Steganography in JPEG2000 Compressed Images

Po-Chyi Su and C.-C. Jay Kuo, Fellow, IEEE

Abstract —*Information hiding in JPEG2000 compressed images is investigated in this research. The challenges of covert communication in this state-of-the-art image codec are analyzed and a steganographic scheme is then proposed to reliably embed high-volume data into the JPEG2000 bit-stream. A special mode of JPEG2000 is employed, and its usage and functions are explained and justified. Experimental results are given to demonstrate the performance of the proposed algorithm.*

Index Terms — Information hiding, steganography, JPEG2000.

I. INTRODUCTION

INFORMATION hiding in digital images, video or audio clips has drawn much attention in recent years [1]-[3]. Some auxiliary information is implicitly combined with a piece of multimedia data, i.e. the host signal, to form a composite signal for certain interesting applications. Digital watermarking is one type of information hiding. The copyright related information about the media data is inserted to enforce intellectual property right protection. Casting a fragile signal into the media file to assist detecting or locating subsequent unauthorized modification may serve as a potential tool for data authentication. The other application is to transmit a large volume of information covertly in a multimedia file via information hiding techniques. The objective is to deliver the data to the intended receiver reliably and secretly. We can view the innocuous host media file as a camouflage to fool possible eavesdroppers or as a secret channel from a communication viewpoint. The case of covert communication can also be termed as *steganography*, which is derived from the Greek words meaning *covered writing*.

In this research, we will focus on steganography in digital images. Two information hiding approaches have been commonly used. One is to cast the information on the imagery data without taking any file format into consideration. The other is to explicitly operate on a specific image format. Most of the previous work in the latter category, such as the methods presented in [4] and [5], was based on JPEG. This is partly because that most of the still images circulated nowadays are compressed with JPEG. The other reason is that, as a block DCT codec, JPEG lends itself to a good candidate for information hiding due to its fixed block structure. Given that images are usually compressed before transmission or storage, information hiding in a compressed data format should be a better choice.

The authors are with the Integrated Media Systems Center and Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, CA 90089. Email: {pochyisu,cckuo}@sipi.usc.edu.

JPEG2000 is an upcoming still image coding standard. This new standard complements JPEG by providing several important features such as resolution/quality progressive image transmission, better resilience to bit-errors, and Region of Interest (ROI) coding, etc. It is believed that JPEG2000 will be used widely and its rich features will benefit many emerging applications. As many images will be compressed by JPEG2000 in the near future, it is worthwhile to investigate how to hide high-volume data in JPEG2000 compressed images efficiently. This is the main objective of our research.

This paper is organized as follows. We first provide a brief review of the basic architecture of JPEG2000 in Section II, which should offer sufficient information to help understand our concerns in designing an information hiding scheme in this image standard. Readers are referred to [6]-[8] for more details. Then, we will point out some challenging issues of information hiding in JPEG2000 and present a steganographic scheme under this compression framework in Section III to achieve reliable covert communication. Experimental results will be shown in Section IV to demonstrate the feasibility of the proposed method. Finally, concluding remarks are given in Section V.

II. REVIEW OF JPEG2000 CODING

The block diagram of JPEG2000 is shown in Fig. 1. In the encoder side, the original image first undergoes the forward image transform, which includes the inter-component transform and the intra-component transform (i.e. the wavelet transform). The resulting wavelet coefficients are then quantized and coded. Scalar or Trellis Coded Quantization is used, which may cause some information loss if the image is lossy compressed. The coding paradigm of JPEG2000 can be viewed as a two-tiered structure as shown in Fig. 1 and will be explained in detail below. Rate control is applied in quantization and coding steps to achieve the targeted bit-rate. The decoder side basically reverses the operations by decoding and dequantizing the bit-stream and applying the inverse image transform to reconstruct the image.

Now, let us take a closer look at the two-tiered coding structure in JPEG2000. We illustrate the concept from the encoder part. In tier-1 coding, the quantization indices for each subband are partitioned into code blocks, which are independently coded using a bit-plane coder. More specifically, the code block is coded one bit-plane at a time starting from the most significant bit-plane to the least significant bit-plane. Each individual bit-plane is coded with three coding passes. The first coding pass is the *significance propagation pass*, which conveys significance and necessary sign information for samples that have not yet been found to be significant and are predicted to become significant. The

Contributed Paper
Manuscript received September 29, 2003

second coding pass is the *magnitude refinement pass*. All bits that became significant in a previous bit-plane are conveyed in this pass by using binary symbols. The final pass is the *cleanup pass*, in which all bits that have not yet been coded during the previous two passes are encoded. The symbols generated from the significance propagation and the magnitude refinement passes can be either raw coded or entropy coded by a context-based adaptive binary arithmetic coder, i.e. MQ coder. The cleanup pass is run-length coded and always processed by the MQ coder. The output of the tier-1 encoding process is therefore a collection of compact representations of coding passes for the code blocks.

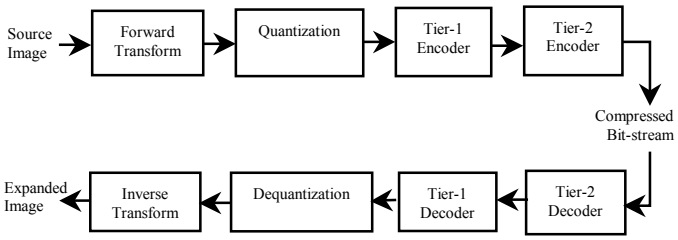

Fig. 1. The block diagram of JPEG2000.

Tier-2 coding operates on the summary information of code blocks, which determines block contributions to the final code stream. The bit-stream of each code-block is truncated in an optimal way so as to minimize distortion subject to the bit-rate constraint. Basically, truncation can only happen at the end of a coding pass. Feasible truncation points have been further identified as those located within the convex hull of the rate-distortion curve. To minimize the distortion with the targeted bit-rate constraint, the exact truncation point is chosen from these feasible ones in each block after the statistics of a collection of code blocks are available. The coding passes are then packaged into *packets* and output to form the final code stream. The ordering of packets in the code stream facilitates progressive transmission of the image by fidelity, resolution or component. Since the rate distortion algorithm of the tier-2 coding is applied after all subband samples have been compressed in tier-1 coding, the rate-control mechanism of JPEG2000 can thus be referred to as Post-Compression Rate-Distortion (PCRD) optimization. Besides, it should be noted that some coding passes may be discarded by this optimized truncation procedure so that the tier-2 coding is another primary source of information loss in the coding path besides quantization.

III. INFORMATION HIDING IN JPEG2000

A. Challenges of Information Hiding

Our objective is to develop an information hiding scheme under the framework of JPEG2000 so that a high volume of data can be secretly transmitted to the intended recipient in a more reliable fashion. First, we have to determine an appropriate position in JPEG2000 coding flow for information hiding. From the structure given in Fig. 1, there are three positions to be considered. We examine their suitability for information hiding below.

(1) *Image Transform*

After the intra-component image transform, the image data are transformed to wavelet coefficients. If we modify the data at this stage, the scheme will be equivalent to many existing wavelet-based watermarking algorithms, which may take other wavelet-based codecs as attacks. For digital watermarking, the payload is usually low, and multiple embedding with the majority detection or the spread-spectrum concept can be applied. The embedded information can thus have sufficient robustness against lossy compression of another codec. However, multiple embedding is not suitable in the data hiding application as the required payload is high and we have to make efficient use of the already limited bandwidth.

(2) *Quantization*

Quantization is an important step in image compression, which reduces certain visual redundancy for efficient coding. As mentioned in Section II, quantization is the primary source of information loss. We can avoid losing the hidden data due to coarser quantization by embedding them in the quantization indices. This solution works for JPEG (as many JPEG-based information hiding schemes operate on the quantization indices), but is not good for JPEG2000. It should be noted that wavelet-based coders usually truncate the compressed bit-stream to fulfill the targeted bit-rate. In JPEG2000, PCRD optimization strategy is adopted so that the truncation mechanism is activated after the whole image has been compressed. If embedding the information at this stage, we cannot predict exactly which quantization index or bit-plane of an index will be included in the final code stream. The embedded information will not be perfectly recovered unless the lossless compression mode is chosen

(3) *Coding*

If the information is embedded in the output of tier-2 coding, i.e. the JPEG2000 packets, it can be guaranteed that all the embedded information will be received without error and in a correct order because we avoid the two major sources of information loss, i.e. quantization and bit-stream truncation. However, we will have difficulty in modifying the packets for information embedding since the bit-streams may have been compactly compressed by the arithmetic coder. Careless modification could result in failure of expanding the compressed image.

B. Progressively Embedding a Hidden Image and Its Drawbacks

There does exist a solution to partially achieve high-volume information hiding in wavelet-based codecs. From the previous discussion, we know that the hidden information could be lost after the subsequent truncation of the compressed bit-stream if it is embedded in the quantization index. However, an intuitive concept indicates that certain indices may have a better chance of survival since they are of more significance. To be more specific, wavelet coefficients in lower frequency subbands are

usually more important than those in higher frequency bands. An extreme example is resolution progressive transmission, in which lower frequency subbands will be sent prior to higher ones. In this case, lower frequency bands should be preserved well at high bit-rates. On the other hand, although some portions of the embedded information may be lost, the recipient can still receive enough information if the significant portions are transmitted successfully. Therefore, if the hidden information is a digital image of a smaller size, we may transmit it by embedding in the quantization indices. This general idea should work in most of the wavelet codecs.

We briefly describe the idea and factors that should be considered. First of all, we decompose the hidden image with the wavelet transform. The number of wavelet decomposing levels and the image size should be related to the host image. For example, if the host image is 512 by 512 and decomposed with 3 levels, we may set the rule that the hidden image is one-fourth the size, i.e. 256 by 256, with the same decomposing levels. We may need to pad the sides of the hidden image when its size is smaller than required. This strategy is to make sure that no image specific information should be necessarily known by the recipient. Besides, it should be noted that coefficients in each level be represented by a fixed number of bits, which is also known by both sides. Then we embed the wavelet subbands of the hidden image according to their importance into the counterparts in the host image. Basically, lower (higher) frequency subbands of the hidden image will be embedded in the lower (higher) frequency subbands of the host image. Embedding is based on modification of lower bit-planes of the host quantization index, i.e. replacing its least significant bits with the bits of the hidden information. The number of the bit-planes used for information embedding will affect both capacity and quality of the resulting image. Embedding follows certain predefined agreements between the information embedder and the recipient. A rule of thumb is that the sign of a coefficient, which is comparatively important than the magnitude, should be embedded more carefully. In addition, bit-planes of lower frequency subbands of the hidden image should also be better embedded than those of the higher frequency subbands. Permutation can be applied to the hidden information before embedding to increase uncertainty. The recipient should be able to permute it back correctly to reconstruct the hidden image. We should not embed any data in the lowest frequency band, i.e. the DC band, of the host image to avoid generating unpleasant artifacts.

Many existing wavelet-based watermarking schemes, such as [9] and [10], may emphasize the function of progressive embedding/detection. However, there are some drawbacks in the idea of progressive hidden image transmission. First of all, we know from the previous discussion that many parameters have to be known in advance by both the sender and the receiver so that the hidden image can be correctly extracted and perceived. Nevertheless, if a lot of information has to be shared beforehand between the sender and the receiver through a certain side channel, the steganographic scheme becomes impractical. Besides, if the hidden information is an image, the eavesdropper may have more chances to detect its existence, given that the characteristics of an image are quite different from that of the host signal. Encryption might not be applicable here since we cannot guarantee that the hidden information be transmitted without errors due to the truncation of JPEG2000 and the fact that an encrypted bit-stream is usually not robust to any error. The most we can do to increase the security level is to permute the position of the wavelet coefficients of the hidden image as mentioned above. It is not clear how this permutation procedure will prevent the eavesdropper from detecting the hidden image. Finally, the requirement of embedding an image as the hidden information significantly limits the usage of this algorithm. The embedded information should be general binary data, in any form such as texts, encrypted bit-streams, raw/compressed images, or else, to achieve practical covert communication. It is apparent that the method presented above may not meet our requirements.

In the following sections, we would like to develop a practical steganographic scheme to convey high-volume general binary data in a more secret and reliable manner.

C. *Information Hiding with Lazy Mode Coding*

As analyzed in Section III-A, in order to transmit general binary data secretly without error, we should embed the hidden information in the JPEG2000 packets. However, we have to avoid modifying the bit-stream that is entropy coded for its correct decoding. In JPEG2000, a so-called *lazy mode* coding option is introduced, in which the arithmetic coding procedure is completely bypassed for most of the significance and the magnitude refinement coding passes. To be more specific, except for the four most significant bit-planes, the significance and magnitude refinement passes in the remaining bit-planes are raw coded. Thanks to this lazy mode coding option, we propose a steganographic scheme, which solves all the above-mentioned problems to achieve reliable covert communication in JPEG2000.

This lazy mode choice for the proposed scheme can be justified below. It has been observed that, at high bit-rates, the symbols produced by the significance and magnitude refinement passes have distributions close to a uniform one so that there is no substantial benefit from arithmetic coding. Bypassing the MQ coder can thus reduce the complexity and improve the execution speed without degrading the coding performance. For information hiding, the images used to host a large volume of data are usually compressed at high bit-rates. Therefore, it is appropriate to embed the information in JPEG2000 compressed bit-stream with the lazy mode enabled. Besides, the hidden information may also be uniformly distributed given that encryption is usually applied. It turns out that the data in these passes are pretty good candidates for information hiding with less chance of being noticed.

Information hiding is achieved by modifying the data in the coding passes. Among the three types of coding passes, only the magnitude refinement passes are chosen in our steganographic scheme. The cleanup passes are definitely

prohibited since they are always entropy/run-length coded and modification of them will cause errors in expanding the bit-stream. The significant passes carry the significance and necessary sign information. Although the significant passes may be raw coded, the modification may cause either sign flipping or decoding errors as well. The magnitude refinement passes carry subsequent bits after the most significant bit for each sample. When the magnitude refinement passes are raw coded, modification of them will not cause problems. Besides, the significant bits or MSB of each sample can act as visual masking so that the change of these subsequent bits can be made less obvious. Furthermore, the amount of raw coded magnitude refinement passes is quite large so carrying a large payload is achievable. By considering these factors, we conclude that only the magnitude refinement pass is suitable for steganographic purposes.

D. Selection of Refinement Passes for Embedding

In order to avoid degrading the composite image severely, we may only use a subset of the raw coded magnitude refinement passes for information embedding. We describe three scenarios of selecting suitable magnitude refinement passes as follows.

(1) *Fixed number of the lowest bit-planes*

The most straightforward method is to examine the bit-planes where these raw coded magnitude refinement passes are located. Given that the total number of meaningful bit-planes in a subband is K, which is signaled explicitly in the code stream, the modification for information hiding is restrained to those bit-planes lower than K-G. The smaller G is assigned, the more bit-planes can be modified so that more hidden information can be carried. Basically, this idea is similar to the common LSB-based information-hiding methodologies, in which only the lowest few bit-planes are modified to avoid introducing visible artifacts. The subtle difference is that, in this steganographic scheme, not all the data in those bit-planes but the data included in the magnitude refinement passes are affected by the embedding process. The advantage of this embedding scenario is its simple implementation since both the information embedding and extraction can be done efficiently in the tier-2 coding. Besides, both of the information embedder and extractor only need to know the parameter G to achieve successful secret communication. The amount of the information that has to be transmitted through other subsidiary channels is thus significantly reduced.

(2) *Bit-planes below the MSB*

A more sophisticated way is to take the MSB of the quantization index into account. In this embedding scenario, the digit of a quantization index is allowed to be modified for information hiding if it is located at the bit-plane that is below the MSB of a sample by at least \$P\$ bit-planes. This idea is somewhat analogous to those watermarking schemes that use the magnitude of a host coefficient to scale the embedded watermark [11]. If a host signal is large in magnitude, we may modify it with a greater scale so that more information or a stronger watermark can be embedded owing to the masking effect. Therefore, the capacity of the steganographic scheme may thus be improved without further affecting the visual quality. In addition, we can intentionally ignore some bit-planes of certain coefficients for embedding due to a more advanced visual masking model or increased security concerns. However, the complexity of the implementation increases accordingly since the information embedding/extraction processes become coefficient-wise, instead of simply viewing the whole bit-planes as a group. In this way, the tier-1 coding has to be involved in the information embedding/extraction processes. Besides, we have to deal with such problems as the varying length of the coding passes and the special patterns reserved for error resilience in a more careful manner.

(3) *Backward embedding*

A better way of selecting suitable magnitude refinement passes for steganography is to consider the importance of these passes to the overall quality of the compressed image. The tier-2 coding achieves rate scalability through multiple quality layers. Each coding pass is either assigned to one of the layers or discarded according to its rate-distortion slope, which is calculated in the tier-1 coding and passed to the tier-2 coding for organizing the code stream. The coding passes with larger rate-distortion slopes are included earlier in the lower layers, while the coding passes with smaller rate-distortion slopes are included later in the higher layers.

Our goal is to hide as much information as possible with the minimal impact on the image quality. Obviously, the embedding process should function in the opposite order of the tier-2 coding by selecting less important coding passes earlier for modifying. Therefore, we propose the idea of backward embedding to take account of the importance of the passes to the overall image quality. After the tier-2 coding determines the passes that will be included into the code stream, an extra procedure embeds the information backward, starting from the last included refinement pass. On one hand, modifying these insignificant passes may be similar to discarding them, which does not severely affect the image, compared to those passes included earlier in the lower layers. On the other hand, the length of these passes could be larger so that we can actually hide more information. The embedding procedure can be carried out until the image quality has been degraded within an acceptable level. A termination pattern may be necessary to signal the end of embedding so that the decoder can learn when to stop extracting the hidden information.

E. Issues on Backward Embedding

By considering the complexity and the performance, we adopt backward embedding in the proposed steganographic scheme. With multiple layers being employed in JPEG2000, backward embedding can easily select those passes that are less important to the compressed image for information hiding without considering the location of the bit-plane and the associated coefficient. The embedding process can thus be done very efficiently since only the tier-2 coding is involved in the embedding process and the coding structure of JPEG2000 can be kept almost the same except that an extra procedure to embed the data in a backward fashion is necessary. It is noteworthy that a major benefit of the proposed JPEG2000

steganographic scheme over the existing JPEG schemes is its controllable rate-distortion trade-off. Here, the rate means the capacity of the hidden information while the distortion is referred to as the additional degradation resulting from the information hiding process. In the existing JPEG embedding schemes, the effect on image quality due to information hiding is usually unpredictable since it is difficult to achieve good rate-control in the JPEG standard. In contrast, our scheme may exploit the characteristics of wavelet-codecs to achieve a better balance between the payload of the hidden information and the resulting image quality.

As the embedding process starts from the last included magnitude refinement pass, the ending point of embedding will decide the distortion of the composite image. A simple scenario for controlling capacity and distortion goes as follows. If the image will be compressed with the bit-rate equal to B bpp, we can guarantee that the composite image will have the quality of the image compressed with C bpp, where $C<B$, by embedding the raw coded magnitude refinement passes until the one that is included in both the bit-stream with B bpp and the bit-stream with C bpp. The idea is easy to implement but the estimation of the quality is conservative since we do not modify all the three coding passes in a bit-plane but magnitude refinement passes, which may only occupy a small portion. We can see this argument from an extreme case that C is equal to 0 while the resulting composite image will still have a reasonably good quality. Therefore, to demonstrate the advantages of the proposed JPEG2000 steganographic scheme over other existing schemes based on the JPEG standard, a more accurate quality measurement is necessary.

If MSE is used as the quality measure, the most accurate way is to calculate the additional distortion in the spatial (or the image) domain. However, the complexity is too high in this approach since we have to expand the compressed bit-stream several times after each embedding of a pass. A more practical way is to evaluate the additional distortion in the wavelet domain along with the generation of the bit-stream. In JPEG2000, the overall distortion in terms of MSE of the compressed image and the original image can be estimated in the wavelet domain directly. For each code block, B_i, the embedded bit-stream is truncated to the rates, $R_i^{n_i}$, with the truncation point, n_i. The contribution from B_i to the distortion improvement in the reconstructed image is denoted by $D_i^{n_i}$. The overall image distortion, D, can be calculated by,

$$D = \sum_i D_i^{n_i} = \sum_i \{\omega_{b_i}^2 \sum_{\mathbf{k}\in B_i} (s_i^{n_i}[\mathbf{k}] - s_i[\mathbf{k}])^2\}, \qquad (1)$$

where $s_i[\mathbf{k}]$ denotes the subband samples in the code block B_i, $s_i^{n_i}[\mathbf{k}]$ denotes the quantized representation of these samples associated with the truncation point n_i, and ω_{b_i} denotes the L2-norm of the wavelet basis function for the subband, b_i, to which the code block belongs. This approximation is valid provided the wavelet transform's basis functions are orthogonal and the quantization errors in each of the samples are uncorrelated. Although neither of the assumptions is held perfectly, the estimation is acceptable in the case of compression. Following the same route, we can calculate the additional distortion introduced by information hiding.

During the embedding process, distortion happens when the bit flips from 0 to 1 or from 1 to 0, i.e. when the original bit and the embedding bit are different. The additional distortion can then be calculated by the sum of the difference between the original quantization index and the index after possible bit flipping and scaled with the quantization step size and the L-2 norm of the wavelet base function. In the implementation, we need not keep the original quantization index or the resulting index to calculate their difference but evaluate the distortion on the fly with each bit-plane processed. For the same index, if a bit in a certain bit-plane changes from 1 to 0, we record it as a negative change while if a bit changes from 0 to 1, we view it as a positive change. The difference between the index values before and after information hiding can be calculated by taking the sum of the positive changes subtracted by the sum of the negative changes. We give a quick example. If the original value is 44 (101100) and the value after embedding is 50 (110010). Each of the positive change and negative change happens twice. The sum of positive changes will be $1\times 2^4 + 1\times 2^1 = 18$ and the sum of negative changes will be $1\times 2^3 + 1\times 2^2 = 12$ so the difference will be 6. This way of calculation is straightforward, and the benefit is its adaptation to the bit-plane coding structure.

However, the exact determination of distortion comes with a few drawbacks, which increase the complexity of the implementation. First of all, the information embedding process is carried out in the tier-2 coding. At this stage, what the tier-2 encoder sees is only a bit-stream. It knows nothing more than the position of the bit-plane or the code-block to which the pass belongs. When flipping a bit in a pass, we may not know exactly which coefficient will be affected. This problem may be solved by passing more information from tier-1 coding to tier-2 coding. Since only the raw coded magnitude refinement passes will be embedded with the hidden information, the tier-1 encoder may have to send extra information to the tier-2 encoder, indicating the correspondence between the bit in the pass and its associated coefficient. For a 64 by 64 block, the extra information may be 64 by 64 bits long for each pass with 1 representing that the pass includes the bit information and 0 representing the null information. The encoder is then able to scan with the same order to identify which coefficient will be affected by a certain bit flipping to evaluate the distortion. Nevertheless, the other problem exists. We have to keep the distortion value for each coefficient in each code block, which increases the memory consumption significantly and contradicts the requirement of efficient memory usage in the tier-2 coding. Next, we provide two rough evaluation methods, which simply operate in the bit-plane level and we will then validate their feasibility by comparing them with the exact distortion approach.

The first method is to calculate the distortion by summing

up MSE of bit flipping in each bit-plane. If a bit flipping happens, we add it to the overall distortion without taking account of the coefficient. The additional distortion ΔD with one bit flipping is expressed as

$$\Delta D = \omega_{b_i}^2 \times (\Delta_{b_i})^2 \times (2^p)^2, \quad (2)$$

where ω_{b_i} is the L-2 norm of the wavelet basis function, Δ_{b_i} is the quantization step size associated with subband b_i and p is the bit-plane position.

It is apparent that the distortion calculated in this way is only a rough estimation since the exact change of the distortion with each coefficient is not recorded and we simply use the sum of the square of the difference in each bit-plane to estimate the square of the sum of the difference in each bit-plane. However, in most of the common cases, we found that many intermediate terms between bit-planes tend to cancel each other. Besides, the overestimate and underestimate compensate each other, as the number of the bit-planes is large. This method will thus provide us a pretty good estimation of the additional distortion introduced by information embedding.

The second estimation method is to utilize the distortion of each pass calculated during the tier-1 encoding. The distortion improvement of each pass is estimated by the difference between the distortion measured before and after including the pass. The distortion is recorded and then evaluated with the rate increase of this pass for rate control. We can view this step as a measurement of importance of the pass since the inclusion of the pass with large distortion improvement makes great impact on the compressed image. We believe that, in information embedding, if the host signal and the hidden signal are both of the uniform distribution, the distortion introduced by embedding or modifying the content of the pass would be very similar to discarding the whole pass. Therefore, we may also use this distortion measurement of a pass as an estimation of the distortion resulting from information embedding. A clear benefit of this method is that the overhead of information hiding is made as small as possible since the embedding process shares the same procedure of distortion measurement in the coding process. Besides, this estimation method allows us to measure the distortion of the host image even before the secret information is embedded. By using this estimation method, we may predict the distortion of many images in advance and choose a more appropriate one for hiding the targeted information. It should be noted that the mid-point reconstruction rule is often employed in the distortion measurement in the tier-1 coding. Information hiding, however, results in bit flipping without mid-point reconstruction as done in the dequantization step. Therefore, we multiply the distortion estimated in the tier-1 coding by 2 as a measurement of the distortion introduced by the modification of this pass.

F. Steganalysis of the Proposed Information-Hiding Scheme

Unlike digital image watermarking, robustness is not the main issue of steganographic applications since we do not expect the attacker will modify the image content by either transcoding or other signal processing procedures, especially in the case that the secret information is hidden in a compressed file for storage or circulation. Capacity, reliability and security are the three major concerns. In our information-hiding scheme with the JPEG2000 standard, we can guarantee that the secret transmission be carried out without errors by embedding the information in raw coded magnitude refinement passes. We can achieve high-volume covert communication by choosing an appropriate amount of passes for information embedding. The remaining issue to be discussed is the security of the proposed scheme. In this section, we try to play the role of an eavesdropper to clarify some security issues to which we should pay attention when designing a steganographic scheme.

The first step the eavesdropper may take is to analyze the bit-stream structure. One drawback of the proposed scheme is that the information-hiding procedure is operated in a special mode of JPEG2000, i.e. the lazy mode. Some people may question that the attacker may suspect the existence of certain hidden information in the JPEG2000 bit-stream if it is compressed with the lazy mode. Eventually, this problem depends on how popular the lazy mode will be. From our viewpoint, the lazy mode coding operation is very useful in high bit-rate image compression. The complexity can be significantly reduced by employing the lazy mode coding because the computationally expensive MQ coding is bypassed while coding efficiency will not be affected much, especially in the high bit-rate coding, which is a very possible case for information hiding. The ROI coding may be the only scenario that the lazy mode is not appropriate to be applied. Therefore, we do not see many reasons for not adopting the lazy mode coding in a broad range of imagery applications.

Next, the eavesdropper may analyze the data in the magnitude refinement passes to see if any unusual distribution appears. As mentioned before, the reason why the MQ coder does not improve coding efficiency in the magnitude refinement passes in the lower few bit-planes is that the distribution of these data is close to a uniform one. Therefore, if we can encrypt or scramble the data in some way so that the hidden information also has a uniform distribution, the chance that the eavesdropper can tell the difference will be small. However, we have to make sure that some special patterns designed for increased error resilience in JPEG2000 should not appear in the modified bit-stream. Aside from the purpose of correct expanding the compressed bit-stream, this cautious strategy can prevent that the appearance of the mark at the wrong position reveals the existence of the hidden information.

The eavesdropper may further expand the compressed bit-stream to see if any abnormal situation happens. In the proposed steganographic scheme, we do not change the length of the magnitude refinement passes but modify the binary content. In general cases, the modified magnitude refinement passes generate the same number of symbols with the original passes. However, some special situations may happen when more symbols or less symbols are generated than expected.

This comes from the fact that extra bits are added by the encoder to avoid generating error resilience patterns as described before. The inaccurate number of symbols will not affect the normal operation of the coding but may give a loophole for the eavesdropper to sense the existence of the hidden information. Besides, the bit-stuffing at the end of the pass to comply with the byte boundary may appear differently if embedding is done in a careless way. We may not embed information into the bits that are used for bit-stuffing if a unified bit-stuffing byte is adopted in most of the JPEG2000 coders. In other words, we should only modify the bits that come from the magnitude refinement passes and avoid the bits used in the simplified implementation or certain markers to ensure better security.

A more advanced eavesdropper may examine the behavior of wavelet coefficients to see if possible hidden information exists. This is actually an interesting topic to investigate if information hiding has different effects on wavelet coefficients from the quantization process. We believe that modifying the JPEG2000 packets may result in some intriguing phenomenon on the inverse wavelet transform. We may study this subject by avoid the quantization step, i.e. by operating the information-hiding process in the lossless compression mode. We leave this part as future research.

IV. Experimental Results

The implementation of our steganographic scheme was based on JASPER [8]. JASPER is a free reference code of JPEG2000 offering the baseline coding with an excellent performance and thus serves as a good framework. In the experiment, we used the four well-known gray-level images, Lena, Boat, Peppers and Baboon, as the host images, all with the same size of 512 by 512, to carry the generalized binary information. We assume that the image will be compressed into the bit-stream with a high bit-rate, i.e. 2 bpp. It should be noted that the length of the bit-stream is not changed by the information-hiding process. In other words, the embedding process will only affect the quality of the image by modifying the bit-stream content, i.e. certain magnitude refinement passes, as described in Section III-C.

First of all, we would like to justify the claim that the lazy mode coding does not give an inferior performance compared with the normal mode in the case of high bit-rate image coding. We compressed Lena, Baboon, Boat and Peppers images from 0.5 bpp to 2 bpp with the normal mode and the lazy mode and then compared the PSNR values of the expanded images in each case. The results are shown in Table I. We see that the difference of PSNR values is very limited. It should be noted that the larger difference, such as Peppers in 2 bpp, comes from the fact that the two compressed bit-streams are different in their lengths although we have tried to make them as close to the target bit-rate as possible. This result may suggest that the lazy mode is applicable in many cases and the compression/decompression speed is thus tremendously improved without much quality degradation.

One of the main advantages of this steganography scheme over other JPEG-based schemes is its controllable distortion during the process of information embedding. As we mentioned before, we can estimate the additional distortion in MSE in the wavelet domain quite precisely. However, in order to simplify the structure without increasing memory consumption, we presented two methods to roughly estimate the distortion introduced by information embedding, i.e. the method one estimating the overall MSE by adding up errors in each bit-plane and the method two utilizing the existing distortion value calculated in the tier-1 coding. We would like to verify their applicability by some experiments. Figure 2 shows the additional MSE measured in Lena, Boat, Peppers and Baboon. The horizontal axis represents the number of magnitude refinement passes that are chosen for information hiding. As more passes are modified for embedding the secret data, the MSE increases accordingly. The dash lines are the actual MSE while the plus marks and cross marks show the estimated MSE values calculated by the method one and method two respectively. We can see that the two methods track the actual distortion pretty well. They both perform better at the beginning of the estimation but worse at the end since the errors accumulate as more passes are processed. Some calibrating steps may need to be taken to achieve a more accurate measurement.

TABLE I
The Comparison of PSNR (dB) using the lazy and the normal modes under various bit-rates (bpp)

Bit-rate	0.50	0.75	1.00	1.25	1.50	1.75	2.00
Lena^n	37.06	38.92	40.31	41.55	42.75	43.94	45.12
Lena^z	37.01	38.87	40.26	41.50	42.69	43.87	45.04
Boat^n	33.15	35.11	36.61	37.98	39.30	40.61	41.91
Boat^z	33.14	35.06	36.55	37.91	39.21	40.52	41.81
Peppers^n	35.60	37.00	38.23	39.44	40.66	41.89	43.11
Peppers^z	35.55	36.94	38.17	39.36	40.55	41.76	42.96
Baboon^n	25.47	27.41	29.06	30.58	32.02	33.41	34.73
Baboon^z	25.47	27.41	29.06	30.59	32.03	33.41	34.73

Image^n: image compressed with the normal mode.
Image^z: image compressed with the lazy mode.

In steganographic applications, we are interested in the relationship between the capacity and the resulting composite image. We embedded the four images with the same binary data and examined the MSE increase of the image due to the embedding process associated with the payload of the hidden information. We can see from Fig. 3 that the capacity varies in the four images even though they are compressed with the same ratio. This phenomenon should not be too surprising since the compression affects images in different ways. In our scheme, the capacity is eventually determined by the number of magnitude refinement passes that are raw coded so the distribution of the data across the three passes will affect the payload. Fig. 3 also shows that, with more information being embedded, the MSE value grows as expected. However, they do not relate to each other linearly. We take Lena as an example. Embedding the first 3000 bytes of the binary data only results in about 1 additional MSE of each pixel in average

but embedding the next 1000 bytes quickly increases the MSE to 2. The last 100 bytes even cause the MSE to change by more than 9. Therefore, the embedding process should evaluate this curve to decide how much information is appropriate to be embedded. It should be noted that the MSE increase may be roughly estimated in conjunction with the embedding process so that we can stop embedding at the point that a minimal acceptable quality is reached.

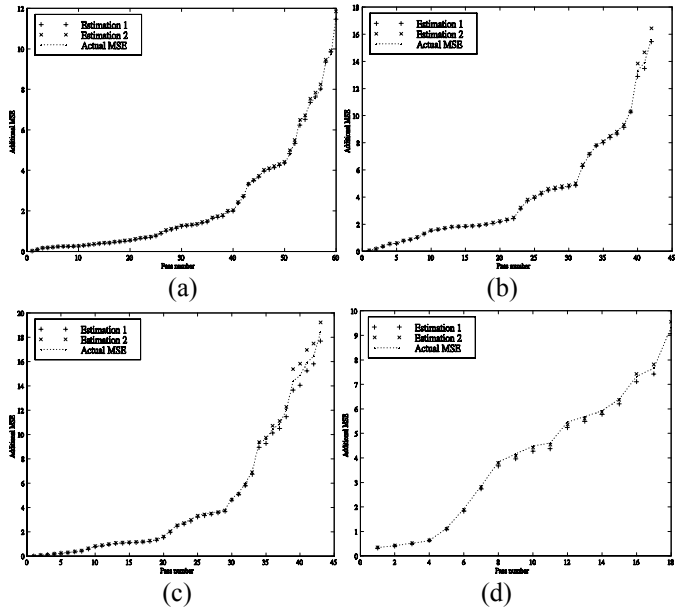

Fig. 2. Additional MSE estimation of (a) Lena, (b) Boat, (c) Peppers and (d) Baboon.

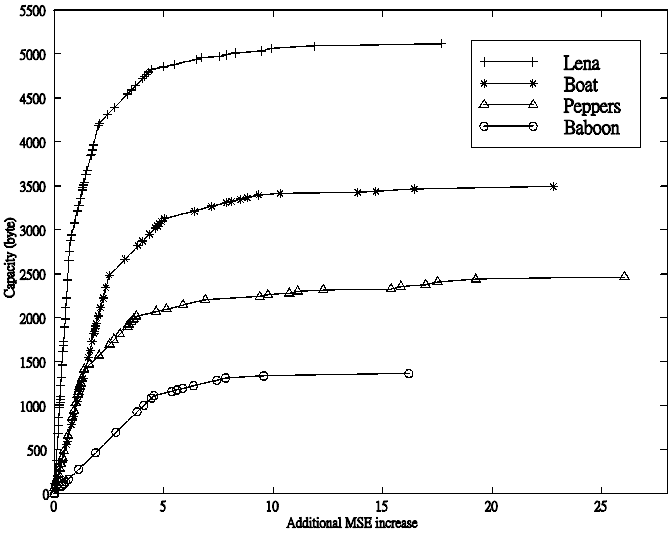

Fig. 3. Capacity vs. additional MSE of the composite image.

Since the payload is quite large in our experiment, we may consider using an image with a smaller size as the intended binary data for information embedding. We compressed the image, F-16 (128 by 128), into a JPEG2000 bit-stream and embedded it into the four host images. The relationship between the PSNR of the composite image and that of the hidden image along with embedding is shown in Fig. 4. The result demonstrates the benefits of backward embedding, in which the less important refinement passes are used to carry the more important information of the hidden image owing to the layered structure of JPEG2000. Progressive transmission of the hidden image can thus be achieved. At the beginning of the embedding, the composite image degrades little while the PSNR of the hidden image boosts quickly. At the latter part of the embedding, the large sacrifice of the composite image only helps to improve the finer detail of the hidden image so the increase of the PSNR value is limited. Under this scenario, the embedding process should proceed until both the composite and the hidden images have an acceptable quality.

V. CONCLUSION

A steganographic scheme was proposed to hide a large volume of data into JPEG2000 compressed images for covert communication. Several design issues were examined to help achieve reliable information hiding in this state-of-the-art image coding standard. Experimental results showed the practicability of the proposed algorithms and the decent performance. Progressive transmission of the hidden information was also demonstrated. As future extension, we would like to apply a more rigorous steganalysis to the proposed scheme to further ensure its security.

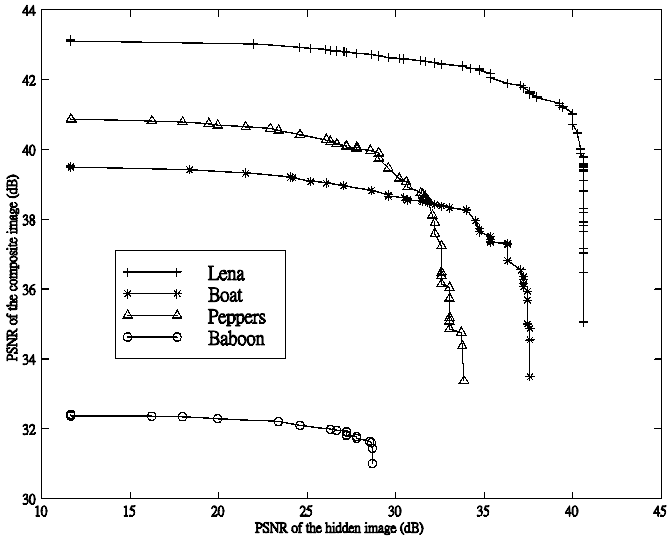

Fig. 4. PSNR of the composite image vs. PSNR of the hidden image

REFERENCES

[1] W. Bender, D. Gruhl, N. Morimoto and A. Lu, "Techniques for data hiding," *IBM System Journal*, vol. 35, no. 3, pp. 313-336, 1996.

[2] M. D. Swanson, M. Kobayashi and A. H. Tewfik, "Multimedia data embedding and watermarking technologies," *Proc. IEEE*, vol. 86, pp. 1064-1087, 1998.

[3] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proc IEEE*, vol. 87, no. 2, pp. 1079-1107, 1999.

[4] M. Holliman, N. Memon, B.-L. Yeo and M. Yeung, "Adaptive public watermarking of DCT-based compressed images," in *Proc. SPIE Photonics West, Security and Watermarking of Multimedia Contents*, San Jose, CA, Jan. 1998.

[5] C. I. Podilchuk and W. Zeng, "Watermarking of the JPEG bit-stream," in *Proc.International Conference Image Science, Systems and Technology,* Las Vegas, NV, June 1997.

[6] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. on Image Processing*, vol. 9, no. 7 pp. 1158-1170, July 2000.
[7] C. Christopoulos, A. Skodras and T. Ebrahimi, "The JPEG2000 still image coding system: An overview," *IEEE Trans. on Consumer Electronics,* vol. 46, no. 4, pp. 1103-1127, Nov. 2000.
[8] M. D. Adams, "The JPEG-2000 still image compression standard," ISO/IEC JTC 1/SC 29/WG 1, Tech. Rep., Sep. 2001.
[9] X. G. Xia, C. G. Boncelet and G.R. Arce, "A multiresolution watermark for digital images," in *Proc. IEEE International Conference on Image Processing*, vol. 1, Santa Barbara, CA, July 1997.
[10] M.D. Swanson, B. Zhu and A. H. Tewfik, "Multiresolution scene-based video watermarking using perceptual models," *IEEE Journal Selected Areas in Communications*, vol. 16, pp. 540-550, May 1998.
[11] I. J. Cox, F. T. Leighton and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. on Image Processing*, vol. 6, no. 12, 1997.

Po-Chyi Su was born in Taipei, Taiwan in 1973. He received the B.S. degree from the National Taiwan University, Taipei, Taiwan, in 1995 and the M.S. and Ph.D. degrees from the University of Southern California, Los Angeles, in 1998 and 2003, respectively, all in Electrical Engineering. His research interests include information hiding, digital image processing and image/video compression.

C.-C. Jay Kuo received the B.S. degree from the National Taiwan University, Taipei, in 1980 and the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1985 and 1987, respectively, all in Electrical Engineering. Dr. Kuo was Computational and Applied Mathematics (CAM) Research Assistant Professor in the Department of Mathematics at the University of California, Los Angeles, from October 1987 to December 1988. Since January 1989, he has been with the Department of Electrical Engineering-Systems and the Signal and Image Processing Institute at the University of Southern California, where he currently has a joint appointment as Professor of Electrical Engineering and Mathematics. His research interests are in the areas of digital signal and image processing, audio and video coding, wavelet theory and applications, multimedia technologies and large-scale scientific computing. He has authored around 500 technical publications in international conferences and journals, and guided more than 50 students to their Ph.D. degrees. Dr. Kuo is a member of SIAM, ACM, a Fellow of IEEE and SPIE. He is the Editor-in-Chief for the Journal of Visual Communication and Image Representation. Dr. Kuo received the National Science Foundation Young Investigator Award (NYI) and Presidential Faculty Fellow (PFF) Award in 1992 and 1993, respectively.