Transactions Papers

A Source and Channel-Coding Framework for Vector-Based Data Hiding in Video

Debargha Mukherjee, *Member, IEEE*, Jong Jin Chae, Sanjit K. Mitra, *Fellow, IEEE*, and B. S. Manjunath, *Member, IEEE*

***Abstract*—Digital data hiding is a technology being developed for multimedia services, where significant amounts of secure data is invisibly hidden inside a host data source by the owner, for retrieval only by those authorized. The hidden data should be recoverable even after the host has undergone standard transformations, such as compression. In this paper, we present a source and channel coding framework for data hiding, allowing any tradeoff between the visibility of distortions introduced, the amount of data embedded, and the degree of robustness to noise. The secure data is source coded by vector quantization, and the indices obtained in the process are embedded in the host video using orthogonal transform domain vector perturbations. Transform coefficients of the host are grouped into vectors and perturbed using noise-resilient channel codes derived from multidimensional lattices. The perturbations are constrained by a maximum allowable mean-squared error that can be introduced in the host. Channel-optimized VQ can be used for increased robustness to noise. The generic approach is readily adapted to make retrieval possible for applications where the original host is not available to the retriever. The secure data in our implementations are low spatial and temporal resolution video, and sampled speech, while the host data is QCIF video. The host video with the embedded data is H.263 compressed, before attempting retrieval of the hidden video and speech from the reconstructed video. The quality of the extracted video and speech is shown for varying compression ratios of the host video.**

***Index Terms*—Channel coding, data hiding, lattice VQ, source coding, watermarking, wavelet transform.**

I. Introduction

With the rapid growth in the mass of multimedia data freely available through the Internet, and the associated investment in standardization of hardware and software for open transmission of such data, a mechanism for hidden data transmission over the established infrastructure will provide an economical alternative to expensive dedicated secure channels and specialized terminals. The emerging technology of *data hiding* [1]–[4] therefore presents an overwhelming urge in the world today. Digital watermarking [5]–[12] is a closely related technology for copyright protection that is receiving much attention lately. Here, a small amount of a specific signature information, called the watermark, is invisibly hidden inside a host data source, typically an image or a video sequence, by the owner before distributing freely. The challenge is to enable the owner to retrieve his original signature from the distributed image or video to check authenticity, even after it has undergone significant transformations such as compression. While most of the early work in this area assumes availability of the original host to the retriever, the current trend is toward developing algorithms that allow retrieval even without knowledge of the original host. In *data hiding*, the focus is on hiding larger amounts of data in a host, for a wider range of applications than just copyright protection. Only those authorized with the knowledge of "how to" can retrieve the hidden data, even after standard transformations like compression as required by the transmission system, or media transformations as required by the storage and distribution system, have been applied to the host. Although it is possible for some applications to have the original host data available during retrieval, the real strength of a data-hiding scheme is the ability to make authorized retrieval possible even without the availability of the original host. Data hiding has several defense-type applications, such as inconspicuous transmission of secret information over an insecure but readily available medium such as the Internet. It can also be used for transmitting various kinds of information securely over the existing infrastructure dedicated for transmitting something else, such as transmitting hidden nonstandard format video or hidden speech, using terminals specialized for transmitting H.263 coded video, as in this work. Since a substantial amount has already been invested in the development of the infrastructure for standard-based data transmission, it makes monetary sense to try to use the same infrastructure for transmission of nonstandard data. Another application is in secure transmission of control information along with data in a commercial delivery system. In general, data hiding makes possible invisible mixing of different kinds of secure data along with standardized and

Manuscript received March 26, 1998; revised August 27, 1999. This work was supported by a University of California MICRO grant, with matching funds from Lucent Technologies, Raytheon Missile Systems, Textronix Corporation, and Xerox Corporation, and by the National Science Foundation under Grant IRI9704785. This paper was recommended by Associate Editor H. Gharavi.

D. Mukherjee was with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106 USA. He is now with Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA (e-mail: debargha@hpl.hp.com).

J. J. Chae was with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106 USA. He is now with the Institute for Defense Information Systems, Seoul, Korea (e-mail: chaejj@yahoo.com).

S. K. Mitra and B. S. Manjunath are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106 USA (e-mail: mitra@iplab.ece.ucsb.edu; manj@iplab.ece.ucsb.edu).

Publisher Item Identifier S 1051-8215(00)04894-1.

open forms of data transmission, allowing only those authorized to retrieve the additional hidden information.

Fig. 1 introduces a quantitative framework for the generic data hiding problem by means of a schematic diagram. The original host is modified using the secure data in a deterministic fashion, by the process of *embedding*, to yield a host with embedded data. As a result of embedding, a mean-squared-error MSE_H is introduced between the host with embedded data and the original host. To ensure transparency of embedding, the value of MSE_H should be below a desired level depending on the nature of the application. On distribution, the host with embedded data typically undergoes compression and other standard transformations. The *extraction* process estimates the hidden data from the "noisy" host with embedded data, that is received. It is then required that the mean-squared-error MSE_S between the original secure data and the extracted secure data be as low as possible. Depending on the specific nature of application, the extraction process may or may not assume availability of the original host.

Note that the watermarking problem is a special case of the generic problem described above. In watermarking applications, the allowable MSE_H is very small, and the requirement of robustness against "attacks" is very stringent. As a consequence, the amount of data that can be reliably hidden is very small. In this work, the focus is on the more generic problem where any amount of data may be hidden, but in a manner such that for a given allowable MSE_H the robustness against data transformations, compression, or "attacks," is maximized. Further, since for most useful applications of data hiding, the original host cannot be assumed to be available during extraction, we treat this case separately. In watermarking terminology, the secure data is typically referred to as the *watermark*, while the host with embedded data is referred to as the *watermarked host*. In the rest of the paper we will use interchangeably the set of terms "secure data," "hidden data," and "watermark," and also the set of terms "host with embedded data" and "watermarked host."

We show that the above problem of data hiding readily maps to the source and channel coding problem in digital communications [13]. As such, established concepts from digital communications can be used to solve it. A quantitative treatment allows precise tradeoff between the transparency of embedding (reflected by MSE_H), the amount of data embedded, and the robustness of the data hiding scheme to data transformations (reflected by MSE_S). The secure data is hidden inside the raw host data, which makes it possible to retrieve the hidden information from the host, irrespective of the compression scheme used, or transformations applied during its transmission and distribution. In this work, although we develop the approach under the assumption that the original host is available during retrieval, we later relax the restrictive constraint to obtain a more useful data hiding scheme. In Section II, we discuss in detail our data-hiding approach, where vectors of orthogonal transform coefficients of the host are perturbed using lattice channel codes to represent source coded symbols. The discussion here is completely general, and applies to hiding any kind of secure data in any kind of host. In Section III, we present specifically the methodology used for hiding data inside a video host in the wavelet transform domain. In Section IV, we present two example applications where two different kinds of data are hidden inside 30 frames/s QCIF video. In the first case, the hidden data is low spatial and temporal resolution video, while in the second case, the hidden data is sampled speech. In Section V, we conclude with some notes on future directions.

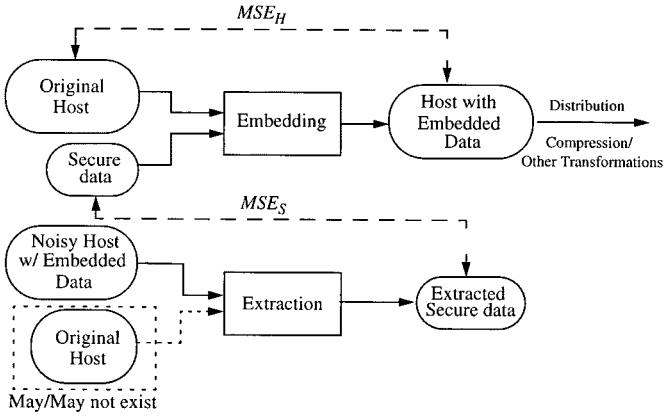

Fig. 1. The data-hiding problem

II. Data Hiding Using Vector Perturbations

The host data is first orthogonally transformed, and the transform coefficients are then perturbed in a definite fashion to represent hidden information. Note that the use of a transform is not essential to this approach because a raw image or a video frame is by itself an expansion on the standard bases. However, orthogonal transformations may yield a subset of coefficients which when perturbed, either result in lower probability of erroneous detection after a particular kind of transformation, or yields less perceptually significant distortions, or strikes a compromise of both.

A. Transparency Constraint

Let us consider a host data sequence X given by $X = (x_1, x_2, \cdots, x_N)$, which is transformed orthogonally to a set of N coefficients $C = (c_1, c_2, \cdots, c_N)$. The transform-domain embedding process perturbs the coefficients into a new set of coefficients given by $\tilde{C} = (\tilde{c}_1, \tilde{c}_2, \cdots, \tilde{c}_N)$. The inverse transformation then yields the watermarked host $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, \cdots \tilde{x}_N)$. Since the transformation is orthogonal, the mean-squared-error introduced in the coefficients is exactly equal to the mean-squared error introduced in the host data; that is

$$MSE_H = \frac{1}{N} \cdot \sum_{i=1}^{N} |x_i - \tilde{x}_i|^2 = \frac{1}{N} \cdot \sum_{i=1}^{N} |c_i - \tilde{c}_i|^2. \quad (1)$$

Now, a *transparency constraint* is imposed on the value of MSE_H. This specifies a maximum value P which upper bounds MSE_H for a given application

$$\frac{1}{N} \cdot \sum_{i=1}^{N} |x_i - \tilde{x}_i|^2 < P \Rightarrow \frac{1}{N} \cdot \sum_{i=1}^{N} |c_i - \tilde{c}_i|^2 < P. \quad (2)$$

The smaller the value of P, the less visible the embedding is, and vice versa.

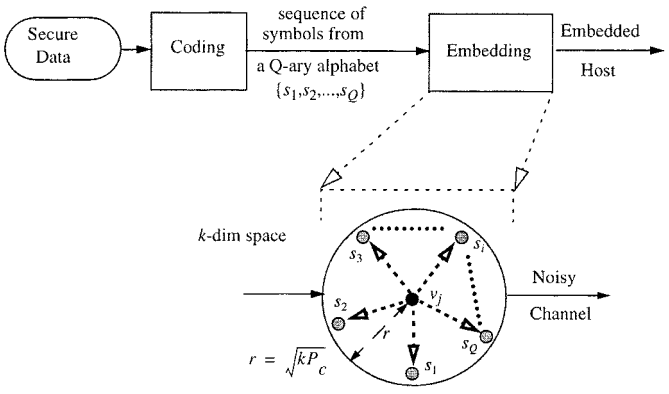

Fig. 2. The Embedding Principle

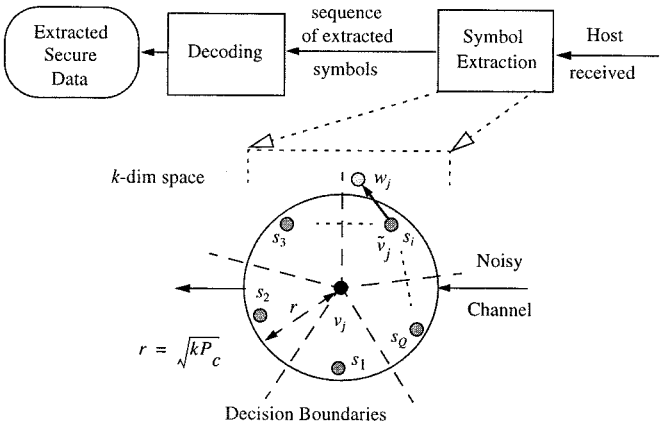

Fig. 3. The Extraction Principle

Since N is typically very large for images and video, it makes sense to simplify the transparency constraint by grouping the coefficients into k-dimensional vectors with $k \ll N$, and satisfying the constraint in each of the vectors individually. Further, it may be convenient to perturb only a limited number M of the N coefficients, say the coefficients in only one particular band of a subband or wavelet decomposition. That is, if the M coefficients to be perturbed are grouped into M/k vectors of dimension k, denoted as v_j, $j = 1, 2, \cdots, M/k$ and the corresponding perturbed vectors are denoted as $\tilde{v}_j$, then for each of the vectors, the following must be satisfied to satisfy the constraint in (2):

$$\frac{1}{k} \cdot \|v_j - \tilde{v}_j\|^2 < P_c = \frac{N}{M} \cdot P, \quad j = 1, 2, \cdots, \frac{M}{k}. \quad (3)$$

P_c represents the maximum allowable per dimension squared perturbation of the vectors, required to satisfy the transparency constraint, and is analogous to a channel power constraint.

B. Embedding Principle

We can now explain the generic embedding principle by means of the diagram in Fig. 2. The secure data is first source-coded, either losslessly or lossily depending on the nature of the data, to generate a sequence of symbols from a Q-ary alphabet $\{s_1, s_2, \cdots, s_Q\}$. The embedding process injects one symbol in each coefficient vector v_j by perturbing it in one of Q possible ways in k-dimensional space to obtain the perturbed vector $\tilde{v}_j$. Note that the possible values of $\tilde{v}_j$ all lie within a shell of radius $\sqrt{kP_c}$ from v_j to satisfy the transparency constraint. The possible perturbations constitute what is in general known as the *channel codebook*, and is of size Q and dimension k. The channel codebook is usually obtained from a noise-resilient channel code by scaling it by a parameter α which determines the transparency constraint. That is, the perturbed vectors $\tilde{v}_j$ are obtained as

$$\tilde{v}_j = v_j = \alpha \cdot C(s_i) \quad (4)$$

where the set of vectors $C(s_i)$, $i = 1, 2, \cdots, Q$ constitute a channel shape codebook of size Q. The perturbed coefficients are inverse transformed back to the host before transmission or distribution.

C. Extraction Principle

The extraction principle is outlined in Fig. 3. Let us say that the jth perturbed vector $\tilde{v}_j$, corresponding to a hidden symbol s_i, has been received as w_j as a result of additive noise n_j due to compression and other transformations

$$w_j = \tilde{v}_j + n_j. \quad (5)$$

The process of extraction is then formulated as a statistical estimation problem that estimates the transmitted symbol from the noisy version received. The solution to the estimation problem yields decision boundaries around the k-dimensional channel codes that depend on the statistical model chosen for the noise. The extraction process uses its knowledge of the original host to decode, from each received vector, the symbol within whose decision boundaries the received perturbation lies. In other words, a nearest-neighbor search with an appropriate distance measure is used. Therefore, if the perturbation corresponding to a symbol s_i is embedded into a vector, and the unperturbed original vector is known during extraction, as long as the received perturbation does not go beyond the decision boundaries around the channel code for symbol s_i in k-dim space, the correct transmitted symbol s_i will still be extracted. The sequence of extracted symbols are then source-decoded to obtain the extracted watermark.

D. Parallels with Data Communication

In this section, we present explicitly the strong parallel that exists between the data hiding problem as described above and that of digital data communication over a power-constrained discrete-input continuous-output channel. Consider a typical passband data communication system where a sequence of symbols from a Q-ary alphabet are to be transmitted over a channel of bandwidth B Hz. Each symbol is transmitted as a particular waveform over a time duration of T seconds. Since the waveform is bandlimited to B Hz and is of duration T seconds, in accordance with the Nyquist sampling theory, it will be completely described by only $k = 2BT$ samples. It follows, therefore, that the signal can be represented as a point in k-dimensional space with respect to any set of k linearly independent basis signals that are each bandlimited to B Hz and over a duration of T seconds. Typically, the basis signal

set chosen is also orthogonal, and therefore, each waveform transmitted over the channel in T seconds can be represented as a point in k-dimensional Euclidean space, with its squared distance from the origin representing the signal energy over time T. Each of the Q-symbols in the alphabet represent one point in a k-dimensional orthogonal signal constellation. The signal points are often referred to as channel codes of the corresponding symbols. Since the transmitter is usually limited in power to transmit no more than energy P_c in time T (power $= P_c/T$), the expected squared distance of the signal point from the origin, is constrained to be less than P_c.

The waveforms as defined above are next transmitted over a noisy channel. While there are various noise models available for various kinds of channels, we will continue this discussion on the basis of an Additive White Gaussian Noise (AWGN) channel of power-spectral-density $N_0/2$. This particular noise model approximates many real channels and also makes analysis simpler. The task of the receiver is then to estimate the symbols transmitted from the noisy waveform that is received. In a correlation receiver, the noisy signal is correlated with all the orthogonal basis signals, to obtain a set of sufficient statistics that also represent a point in the same k-dimensional orthogonal signal space. In a noise-free channel, the received signal point is exactly the same point as the one transmitted. However, as a result of noise, the received vector is different from the one transmitted, and in fact, for the AWGN channel, it can be shown that the components of the noise vector in the signal space are i.i.d. Gaussian random variables with variance $N_0/2$. Using a maximum-likelihood decoder yields symmetric hyperplane decision boundaries, where each waveform is decoded as the symbol to whose signal point representation the received sufficient statistics is closest in Euclidean distance.

We are now ready to see the equivalence between the data hiding problem and the data communication problem. In data hiding, the transparency constraint is similar to the power constraint for channel coding. The host data is orthogonally transformed, and the coefficients are then grouped to form k-dimensional vectors. Each coefficient vector is a point in k-dimensional Euclidean space, which is perturbed using channel codes to represent a hidden symbol. The perturbed coefficients are then inverse transformed to obtain the raw watermarked host. The watermarked host undergoes compression and other transformations in being transmitted to a receiver. Before extracting the hidden data, the receiver orthogonally transforms the received data using the same transform that was used during embedding, grouping the coefficients to form k-dimensional vectors in the same way as before. Let us assume that the element-by-element noise added by compression and other various transformations to the watermarked host is i.i.d. Gaussian with variance σ^2. It follows, therefore, that the noise added to the coefficients in the orthogonal transform domain, is also i.i.d. Gaussian, with precisely the same variance σ^2. Using a maximum likelihood decoder in the transform domain then yields a decoding rule, which for every vector received, chooses the symbol to whose channel code it is closest in Euclidean distance.

E. Channel Coding: Lattice Codes

Before discussing the channel coding issues in data hiding, let us define a rate R for data injection for the perturbed vectors, in bits/dimension as follows:

$$R = \frac{1}{k} \log_2 Q \tag{6}$$

where Q is the size of the alphabet embedded in each k-dimensional host vector.

For a given channel, noise model, and decoding rule, it is possible to write down expressions for the probability of erroneous detection P_e, assuming all source symbols are equally probable. The channel-coding problem is simply stated as that of choosing the channel codes optimally so that for a given power constraint and rate, the probability of error P_e is minimized. For data hiding applications, the value of P_e is a good measure of robustness of a given scheme to transformations of the watermarked host. For a given source coder, the probability of channel-coding error P_e has a direct influence on the value of MSE_S that measures the quality of the extracted data (see Fig. 1). In general, with increase in the dimensionality k, it becomes possible to design more efficient channel codes yielding lower P_e. Further, in accordance with Shanon's channel-coding theorem, for every channel, it is possible to define a channel capacity C as the maximum of the mutual information between the input and the output, such that as long as the rate is $R < C$, virtually error-free transmission can be achieved by choosing a sufficiently large dimension k. As an example, the capacity of a continuous-input continuous-output AWGN channel with noise variance σ^2 and power constraint P_c is given by

$$C = \frac{1}{2} \log_2 \left(1 + \frac{P_c}{\sigma^2}\right) \text{ bits/dimension.} \tag{7}$$

Capacity is the fundamental limit of what rate is achievable for a given channel. From a data hiding perspective, the capacity presents a limit on the quantity of secure data relative to the quantity of host data, that can be embedded inside the host and retrieved error free.

Although capacity is an asymptotic notion that is not really achievable in practice, P_e decreases with increase in dimensionality k and justifies our motivation for using higher dimensional channel codes. This is particularly true when the noise variance is small as compared to the power constraint. In higher dimensions, it is customary to use lattice-based channel codes, where codepoints are chosen as subsets of lattices or lattice-cosets satisfying the power-constraint. Conway and Sloane [14]–[18] have shown that the lattices D_4, E_6, E_8, K_{12}, Λ_{16}, etc. produce very good lattice codes in their respective dimensions, and have also presented tables and graphs [16] with their P_e and nominal coding gain results. Further, lattice channel codes enjoy the advantage of having fast encoding and decoding algorithms. For the case of additive white Gaussian noise with small variance

σ^2, Conway and Sloane computed the value of P_e for a k-dimensional lattice-based channel code with power constraint P_c to be approximately given by

$$P_e \approx \frac{\tau}{2}\text{erfc}\left\{\sqrt{\frac{kS}{2}}\Delta^{1/k}\right\} \tag{8}$$

where $S = P_c/\sigma^2 Q^2$ is the normalized signal-to-noise ratio (SNR) and Δ and τ are the packing density and the kissing number of the lattice, respectively. It follows from (8) that the denser a lattice is in k-dimensional space, the lower its probability of error would be if used for channel coding. Intuitively speaking, in any k-dimensional space, for the same power constraint and rate, points placed on a scaled denser lattice would be further separated from each other than those placed on a scaled thinner lattice. Additionally, even when the dimensionality k increases, it is always possible to find lattices dense enough such that $\Delta^{1/k}$ is bounded below for any k, however large. Thus, if dense enough lattices are chosen, the occurrence of $\sqrt{k}$ in the argument of the erfc$\{\}$ function in (8) ensures that the estimate of P_e decays fairly rapidly with increase in k. Intuitively speaking, with increase in dimensionality of the space, it becomes possible to arrange points on a dense enough lattice with larger per dimension squared distances separating them for a given rate and power constraint.

Most of our data hiding implementations are based on efficient lattice channel codes that yield low values of P_e. Spherical or constant energy codes, derived from the first shell of various multidimensional lattices, are used in many of our implementations. Since all the points in such codes are equidistant from the origin, the MSE_H introduced as a result of embedding is exactly equal to the transparency constraint. In practice, however, for image and video hosts, the effect of rounding the pixels of the watermarked host to integers and limiting them to lie in the range 0–255 may cause minor deviations from the theoretical value.

It is to be noted that more advanced channel coding schemes, such as those based on trellis-coded modulation, can be used to avoid the problems associated with very large dimensional vectors.

F. Source Coding: VQ and COVQ

For most data-hiding applications we envisage, it would be necessary to embed secure data at a rate higher than the channel capacity. Therefore, it makes sense to compress the secure data losslessly or lossily before embedding. If the secure data is compressible, lossy compression schemes can be used for achieving lower rates over the channel. A scheme that works well for correlated sources is vector quantization (VQ) [19]. One advantage of using memoryless VQ, as opposed to other more sophisticated schemes for compression, is that it is inherently very robust to noise because there are no drift effects. The indices obtained by vector quantization form the alphabet that is embedded into the host transform coefficients by vector perturbations derived from noise-resilient channel codes.

Even with compression of the secure data, the rate through the channel may be too high to support error-free communication. For example, the watermarked host may be compressed too severely for a given transparency constraint to allow reliable recovery of the hidden data. In such cases, it is advantageous to combine source and channel coding by using channel-optimized VQ (COVQ) [20], [21] for better noise performance. While standard Euclidean vector quantization obtains the best encoding index $i*$; from a size N_{cb} source codebook with codevectors y_i, $i = 0, 1, \cdots, N_{\text{cb}}-1$ for a given input source vector v, as

$$i* = \arg\min_i(\|v - y_i\|^2), \quad i \in \{0, 1, \cdots, N_{\text{cb}} - 1\} \tag{9}$$

Euclidean channel-optimized vector quantization obtains the best encoding index $i*$ as

$$i* = \arg\min_i \left(\sum_{j=1}^{N_{\text{cb}}} P_{j/i} \cdot \|v - y_j\|^2\right), \quad i \in \{0, 1, \cdots, N_{\text{cb}} - 1\} \tag{10}$$

where the $P_{j/i}$ are the transition probabilities of receiving index j, given index i is transmitted. COVQ strives to minimize the overall end-to-end distortion in a source, transmitted over a given noisy channel. For example, if two channel symbols are close in the channel-signal space, one can often be mistaken for the other, due to noise. However, if the source codes they represent are also close, the distortion introduced by this error will be small. The VQ design algorithm for COVQ [21] is likewise modified to minimize the new distortion metric. Our implementation of hiding video in video uses channel-optimized vector quantization for improved noise resilience.

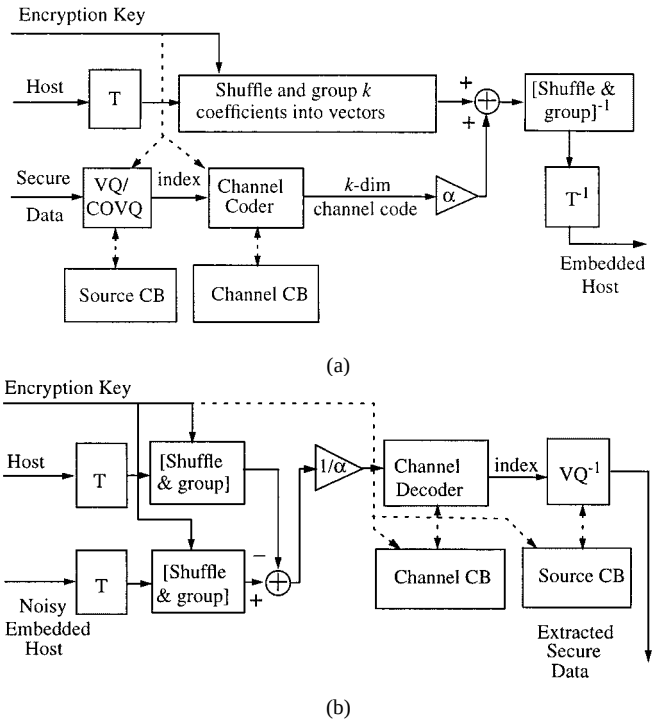

Fig. 4. Schematic diagram of a data-hiding system. (a) Embedding. (b) Extraction.

G. Generic Data Hiding Schematic

Based on the ongoing discussion in this section, we now present in Fig. 4 the schematic diagram of a generic data hiding system for the case when the original host is available during retrieval.

During embedding, the host data is first orthogonally transformed. An encryption key is then used to pseudo-randomly shuffle the coefficients in the transform domain and form k-dimensional vectors. In other words, each vector is formed by taking coefficients from arbitrary positions in the transform domain determined by the encryption key. The selection of the coefficients used for data embedding may also be made adaptively based on certain stable features of the host data. The number of vectors formed depends on the number of symbols to be embedded. The hidden compressible data is next appropriately vector quantized (VQ or COVQ), and the indices obtained in the process are embedded into the k-dimensional host transform vectors by vector perturbations, in accordance with efficient channel codes scaled by a factor α. After embedding, the coefficients are inverse transformed to obtain the watermarked host.

During extraction the original host is assumed to be known. First, the received noisy host and the original host are orthogonally transformed. Given the same encryption key that was used during embedding, the vectors are shuffled and grouped in the same fashion as before to form k-dimensional vectors. The difference between the vectors is scaled, and passed through a channel decoder and then a source decoder to obtain a reconstruction of the hidden data.

The above scheme has three layers of security. The variability in the source and channel codebooks used makes unauthorized retrieval virtually impossible. The knowledge of the algorithm alone is not sufficient to extract the hidden information. The exact source and channel codebooks used for any application instance must be known. The encryption key-based coefficient grouping introduces an additional third layer of security. Even if an attacker knows the exact coefficients used for data embedding, he cannot retrieve it without knowledge of either the source codebook or the channel codebook or the encryption key (that determines the way these coefficients are grouped). Depending on how a specific system is implemented, an attacker may be able to destroy the hidden information; but if implemented appropriately, he cannot do it without significantly degrading the watermarked host.

Another advantage of pseudo-random grouping of coefficients to form vectors is as follows. Typically, the noise introduced as a result of transformations such as compression occur in "bursts." That is, a highly corrupted coefficient is likely to have its neighboring coefficients also heavily corrupted. Therefore, the noise in the components of a vector, if formed by grouping neighboring coefficients, remain too correlated to fit our assumed model of being independent and identically distributed. Pseudo-random shuffling implies that the components of a vector now come from different random locations in the transform domain, and therefore, the noise introduced in the coefficients come closer to being i.i.d. This, in turn, validates the use of the Euclidean distance measure for channel decoding.

The encryption key may also be used to pseudo-randomly change the source and/or channel codebooks, during embedding. For example, the channel vector directions may be changed in the channel codebook, while the indexing in both the source and channel codebooks may be changed. In fact, the family of watermarking techniques using spread spectrum communications, essentially uses pseudo-randomly oriented channel codes in very high dimensions.

H. Hidden Data Extraction without Original Host

We now present a methodology based on the above framework for designing a data hiding system that allows retrieval without knowledge of the original host. This functionality opens up possibilities for numerous other applications besides copyright protection. A previous successful approach to achieving this functionality comprises quantizing the host data elements first and then perturbing them around the quantized values to represent hidden data [1]. The perturbations must be "small" relative to the quantization step-size. While this can be readily combined with the source and channel coding framework described in this paper, and has in fact been used for our implementations, for the sake of completeness, we present an alternative approach based on prediction and estimation in the transform domain, that when used appropriately, can yield results superior to the quantization approach.

During embedding, given the full set of transform coefficients $C = (c_1, c_2, \cdots, c_N)$, a smaller set of M coefficients is chosen as *carriers* for embedding the channel code perturbations. The selection of the carrier coefficients may be made independent of the actual host data (pre-determined, or determined pseudo-randomly by the encryption key), or it may be chosen adaptively based on stable features of the host data that are not likely to change significantly by data transformations. Let us call the set of carrier coefficients $A = (a_1, a_2, \cdots, a_M)$ and the remaining set of coefficients $B = (b_1, b_2, \cdots, b_{N-M})$, where each $a_i \in C$, $b_i \in C$. Next, for each coefficient a_i in set A, a value $\hat{a}_i$ is computed sufficiently close to a_i, which acts as the base value for the channel code perturbations to be introduced subsequently to represent hidden data. Note that $\hat{a}_i$ must be computed in a manner such that it can be recovered reliably at the receiving end. While the quantization method coarsely quantizes a_i to obtain $\hat{a}_i$, in this approach, each a_i is predicted deterministically from the remaining N-M coefficients in set B to obtain $\hat{a}_i$. That is

$$\hat{a}_i = f_{a_i}(B), \quad i = 1, 2, \cdots, M \tag{11}$$

where f_{a_i} is the predictor function predicting coefficient a_i from the coefficients in set B. The predicted coefficients in set A form the base values over which data is embedded. These coefficients are grouped into vectors of appropriate dimensions to form base vectors, which are then perturbed to represent hidden channel codes. The perturbed coefficients replace the original ones in the watermarked host.

When the receiver receives the noisy transform coefficients, it first partitions them into sets A and B in the same manner

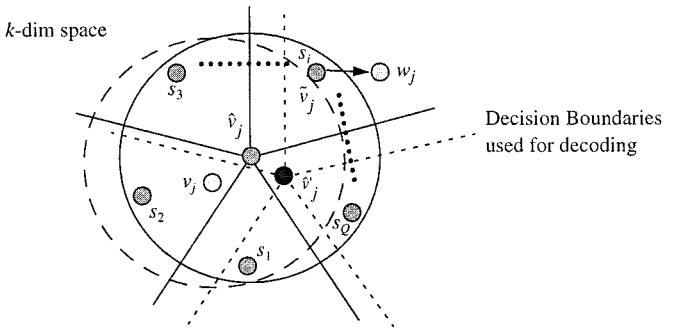

Fig. 5. Illustration of data hiding allowing extraction without knowledge of the original host

as during embedding. Based on the noise model for the coefficients in set B, the receiver then estimates the predicted base values $\hat{a}_i$ of the coefficients in A from the noisy coefficients in B received. Using a minimum mean-squared error estimator amounts to obtaining the estimated base values $\hat{a}'_i$ as

$$\hat{a}'_i = E\left[f_{a_i}(B)/B_{\text{noisy}}\right], \quad i = 1, 2, \cdots, M \qquad (12)$$

where B_{noisy} represents the received noisy coefficients in set B that are used to estimate the value of $f_{a_i}(B)$. As long as this estimation process is accurate, the estimated base values $\hat{a}'_i$ of the coefficients in A would be close to the true base values $\hat{a}_i$ that were actually used for embedding. These values $\hat{a}'_i$ can then be used to decode the hidden channel codes in lieu of the true a_i values.

Fig. 5 shows the methodology diagrammatically in terms of the k-dimensional vectors used for embedding channel codes, after grouping the coefficients in set A. Here, v_j is a vector representing a group of k original set A coefficients, which is predicted as $\hat{v}_j$. As long as the prediction process is accurate, the prediction error $\hat{v}_j - v_j$ will be small. It is this vector $\hat{v}_j$ that is used as the base over which a channel code is embedded in the host. For example, if symbol s_i is to be transmitted using this carrier vector, the vector will be perturbed into $\tilde{v}_j$ as shown in the diagram, where the channel code is given by $\tilde{v}_j - \hat{v}_j$. The vector $\tilde{v}_j$ is the one that actually appears in the watermarked host, so that the overall perturbation of the original vector v_j is given by $\tilde{v}_j - v_j$. In order to decode, the receiver estimates $\hat{v}_j$, the base unperturbed vector, from the noisy coefficients received in set B, to obtain the estimated vector $\hat{v}'_j$. It is this estimate $\hat{v}'_j$ that is assumed to be the base for embedding. Hence, the decision boundaries for decoding are assumed to be centered around $\hat{v}'_j$. As an example in the diagram, if the perturbed vector $\tilde{v}_j$ in the watermarked host representing symbol s_i is received as w_j due to noise, the decoding result will be still correct as long as w_j does not go beyond the decision boundaries for symbol s_i.

Let us assume the variance of the error in prediction (in the prediction-estimation approach) or quantization (in the quantization approach) before embedding to be σ_p^2. Also, assume the expected per dimension squared distance (power) of the channel codes embedded on the predicted vectors to be P_c^*. Since the channel codes representing the hidden information are independent of the prediction error, and assuming both the channel codes and the prediction error to be zero-mean, the expected per-dimension-squared overall perturbation $\tilde{v}_j - v_j$ is given by $P_c^* + \sigma_p^2$. To ensure the transparency constraint, this quantity should be less than P_c as defined in (3). In other words, the available power or transparency constraint is divided into two parts: 1) the part which actually determines the robustness of the code, P_c^* and 2) the part that allows for inaccurate prediction, σ_p^2.

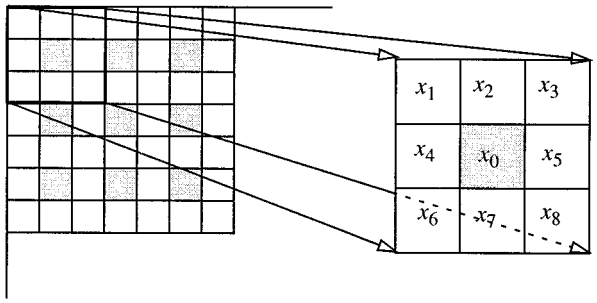

Fig. 6. Example pixel-domain data hiding scheme for image hosts.

Let us next assume the variance of the error in base estimation before decoding to be σ_e^2. Since decoding is based on the estimate $\hat{v}'_j$ rather than on the true base vector $\hat{v}_j$, we can assume the estimation error $(\hat{v}_j - \hat{v}'_j)$ to act as an additional noise component in the received vector w_j. That is

$$w_j = \hat{v}_j + (\tilde{v}_j - \hat{v}_j) + n_j = \hat{v}'_j + (\tilde{v}_j - \hat{v}_j) + (\hat{v}_j - \hat{v}'_j) + n_j. \qquad (13)$$

Assuming the noise in the set B coefficients that determine the estimation error to be independent of the noise in the set A coefficients, the noise variance of the effective channel will be $\sigma^2 + \sigma_e^2$, where σ^2 is the variance of the channel noise appearing in all the coefficients. Thus, in assuming the base vector for embedding to be $\hat{v}'_j$, the receiver effectively decodes symbols from a channel at a higher noise level.

It is now apparent why the prediction-estimation scheme is more advantageous than the quantization scheme, if good predictors and estimators can be obtained. For the quantization scheme, the requirement that the quantized values be sufficiently separated, ensures that σ_p^2 be a large fraction of the power constraint P_c. Therefore, P_c^*, the power of the channel codes that can be used, will be small. On the other hand, in the prediction-estimation approach, if the predictor is good, σ_p^2 will be small, thereby allowing a larger power P_c^* for the channel codes. Of course, decoding is more difficult in the prediction-estimation approach due to a nonzero σ_e^2 (σ_e^2 is zero in the quantization approach, if the quantized base value is always correctly recovered). However, if estimator is good enough, the degradation in robustness will be small. In the long run, for good predictors and estimators, the prediction-estimation approach will yield more robust embedding than the quantization approach.

We now present as an example a pixel-domain data-hiding scheme for image hosts using the above methodology. Fig. 6 shows a portion of an image to act as a host for hidden data. In the diagram, the shaded pixels comprise Set A, while the remaining pixels comprise Set B. The prediction operation is similar to mean-filtering in 3×3 windows around the Set A coefficients, where the set A pixel at the center is not included

in the computation of the mean. Each Set A pixel x_0 is thus replaced by $\hat{x}_0$ given by

$$\hat{x}_0 = \tfrac{1}{8} \sum_{i=1}^{8} x_i. \tag{14}$$

The replaced Set A pixels are then grouped into k-dimensional vectors and perturbed to represent channel codes. At the receiving end, assuming an i.i.d Gaussian model for the noise in the pixels of the received image, the estimate $\hat{x}_0'$ of $\hat{x}_0$ will be given by

$$\hat{x}_0' = \tfrac{1}{8} \sum_{i=1}^{8} x_i(\text{noisy}) \tag{15}$$

which is the same as (14), except that the x_i's are now the noisy pixels received rather than the true ones. If the noise variance is σ^2, the variance of the estimation error will be $\sigma^2/8$. The receiver can then decode the channel codes with $\hat{x}_0'$ as the base value rather than $\hat{x}_0$, and with the noise variance of the effective channel being $\sigma^2 + \sigma^2/8 = 9/8\sigma^2$.

Note that the prediction and estimation functions should be known *a priori* without knowledge of the actual host used for embedding. In practice, predictors and estimators in the transform domain that work well for all instances of hosts of a particular class may be difficult to obtain. Classification based on stable features of the host data can be used for better selection of the set A coefficients, and designing improved predictors specialized for each class. In our implementations, however, we adopt a very simple nonadaptive approach for image or video hosts, that can be regarded as a special case of both the quantization approach and the prediction-estimation approach.

III. Data Hiding in Host Video

While the discussion so far has been considerably generic and applies to hiding most kinds of secure data in most kinds of host, in this section we specialize the scheme to video hosts.

The general principle of data hiding in video is as follows. Each frame of a video sequence is orthogonal wavelet transformed, and the transform coefficients are grouped into vectors based on an encryption key. The signature data is vector quantized, and the indices are embedded into the coefficient vectors in one or more subbands using efficient channel codes. The same hidden data may be repeated in a few successive frames to introduce robustness to low frame-rate compression of video. Note that the frame by frame approach suits well the frame-based compression technology currently in vogue.

We now focus on the issue of choice of subbands for embedding hidden data. In general, hiding data in the lower subbands has several distinct advantages. The nature of current compression algorithms ensures better preservation of the lower frequency data. Furthermore, embedding data in the lower frequencies is not likely to hamper the compression efficiency significantly. Most compression schemes quantize the lower bands finely, and in some way utilize to their advantage the fact that the higher bands have very little energy. Injecting information in the lower bands, therefore, leads to neither easy destruction of the hidden information, nor to any significant change in the coding efficiency. Injecting data in the higher bands, on the other hand, leads to significant deterioration in the compression efficiency by most algorithms. However, a disadvantage is that the distortions introduced by embedding in the lower bands may be perceptually more severe. Weighing the pros and the cons, however, hiding data in the lower subbands is still found to be better if the kind of transformation we are most concerned with is compression.

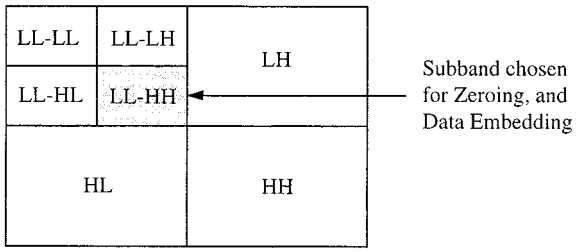

Fig. 7. Subband chosen for zeroing and subsequent data embedding

If, however, extraction is to be made possible without knowledge of the original host using the techniques in Section II-H, hiding data in the lower bands is not found to be appropriate. While on one hand, coefficient prediction (see Section II-H) in the wavelet domain is not an easy problem by any means, on the other hand, because the lower subbands have more energy than the higher subbands, any reasonable predictor or quantizer would yield larger errors in the lower subbands than in the higher bands. As such, the lower subbands do not form a suitable channel for data hiding. Noting that natural images typically have very low energy in the higher bands, we find that for most images, zeroing out some or all of the coefficients in one or more of the higher subbands introduces a very low mean-squared error and affects image detail only inconspicuously in the perceptual sense. Therefore, if the prediction (or quantization) used is simply *zero*, and the hidden data is embedded on the zeroed coefficients, the extraction process only needs to use the zero-vector as its estimated base for decoding the noisy vectors it receives. Both the prediction and the estimation problems become nontrivial in this approach. Furthermore, the estimation error is guaranteed to be zero. In practice, the exact coefficients that are zeroed out and subsequently used for embedding, are either pre-determined, or selected in a pseudo-random manner using the encryption key, or selected image-adaptively based on stable features of the host frame.

Note that the above methodology contradicts the reasons provided earlier on why it may better to embed data in the lower subbands. Therefore, to strike a compromise, only the coefficients in the middle subbands are targeted for data hiding. The scheme used in our implementations is shown in Fig. 7. A two-stage wavelet decomposition of each frame is made, and the hidden data is embedded in the coefficients of the shaded LL-HH subband, after zeroing. Note that embedding data in a single subband is not very appropriate for copyright protection applications because if the subband used for embedding is known, an attacker can easily destroy the hidden information by bandstop filtering. However, in applications involving a broadcast scenario where the same watermarked video is freely

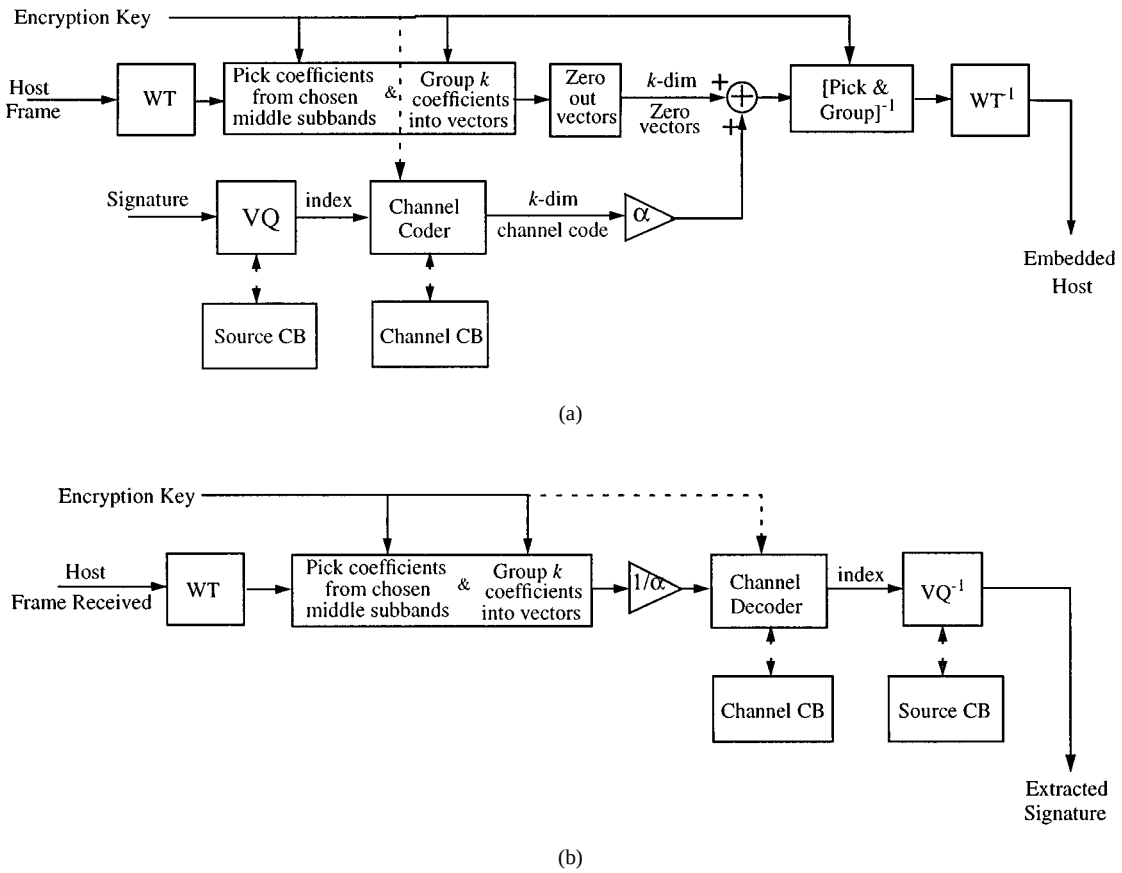

Fig. 8. Embedding and extraction schematic for data hiding in video using zeroed LL-HH subband. (a) Encoder. (b) Decoder.

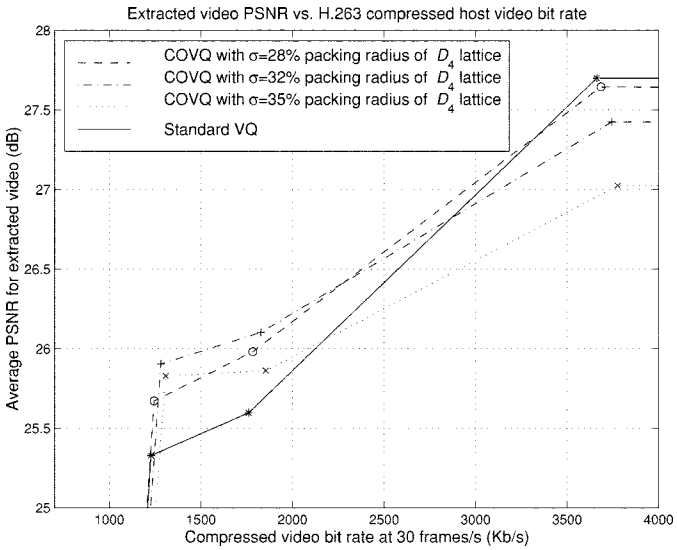

Fig. 9. Average PSNR of extracted quarter QCIF *Hall_Monitorr* sequence versus bit rate for H.263 compressed host *Mother_Daughter* bit stream at 30 frames/s for standard VQ and COVQ, both of size 256. The 256-symbol channel consists of a Voronoi code derived from D_4 lattice.

broadcast or made available to everybody, but only those authorized retrieve the hidden data, protection against watermark destruction is of secondary importance to the protection against unauthorized retrieval. If protection against destructive attacks is desired, the carrier coefficients should be spread over several subbands, including the LL band after quantization, so that an attacker cannot destroy the hidden data without substantially degrading the host video also.

It is appropriate to make a comment on the zeroing-out approach above. While zeroing-out coefficients from one or more subbands before embedding may result in significant distortions or loss of detail for some host videos, in the absence of suitable wavelet domain coefficient predictors, the zeroing-out approach appears to be the only reasonable solution. Also note that this approach is a special case of both the quantization approach and the prediction-estimation approach described in Section II-H.

Fig. 8 shows a schematic diagram for the embedding and extraction mechanism outlined above. The host video is first wavelet transformed frame by frame. An encryption key is used to pseudo-randomly pick coefficients to use as carriers from one or more of the middle subbands chosen for embedding, and also to group them into k-dimensional vectors. The carrier vectors thus formed are each zeroed out. The total number of coefficients picked depends on the amount of hidden data to be embedded in each frame. The hidden compressible data is then appropriately vector quantized, and the indices obtained in the process are embedded into the k-dimensional carrier vectors by assigning them to be the corresponding channel codes scaled by a factor α.

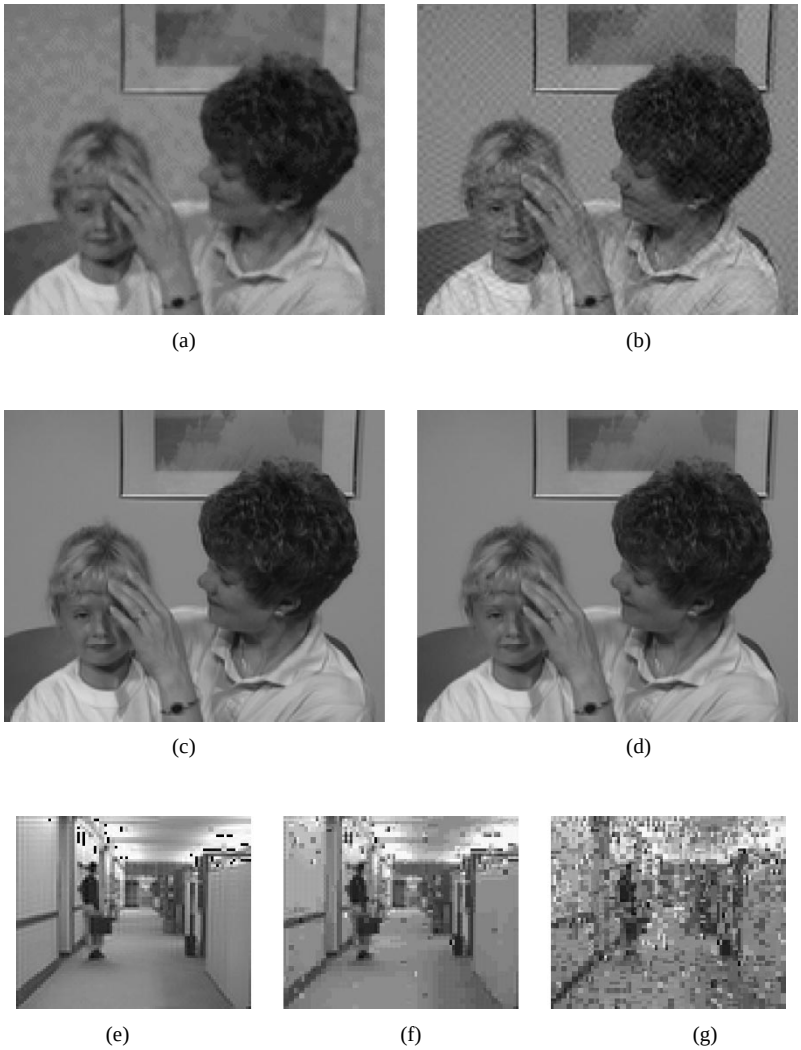

Fig. 10. Visual quality results for hiding quarter QCIF *Hall_Monitor* in QCIF *Mother_Daughter* using COVQ. Frame 20 of the host video and Frame 36 of the hidden video are shown as examples.

IV. Implementation Details and Results

A. General Comments

In this section, we present the implementation details of two example applications: hiding video in video and hiding speech in video [4], both of which allow extraction of hidden data without knowledge of the original host. Though these applications strongly parallel the applications presented by Swanson *et al.* in [1], the current work presents a more universal framework, of which the embedding scheme in [1] is in many ways a special case. While [1] presents a binary source embedding scheme, the current work is more generic in that symbols from a Q-ary source can be embedded in the host data. In terms of channel coding, the scheme in [1] essentially uses a binary channel code in high dimensional space, where the code directions are varied in a pseudo-random manner. One difference however, is that the power of the channel codes in [1] is also varied based on perceptual considerations. While this is good for the sake of invisibility, not all hidden bits are equally protected.

Furthermore, the amount of data embedded in [1] is much smaller than that attempted in this work. In the current work, the use of channel coding principles allows maximizing the amount of data hidden for a given tolerable distortion introduced in the host data, and a desired level of robustness to noise. While the methodology can be effectively used for achieving any tradeoff, the results presented here correspond primarily to the case where the amount of data hidden is large. Although the distortion introduced in the watermarked video is significant, we choose to present these results because similar high data rate embedding results are relatively uncommon in the literature. We also test our scheme by actually compressing the watermarked video by standard algorithms (H.263) and then recovering the hidden data (video or speech) from the reconstructed video. In contrast, [1] presents only the BER results, which is not sufficient to describe the quality of the extracted video when sophisticated source coding schemes like MPEG are used in a noisy environment.

All the concepts outlined in the previous sections are covered in the implementations and results presented below. The wavelet filters used in all cases are the orthogonal Daubechies filters [22] of length 6. Other applications of the methodology in this work will be found in [2], [3], where gray and color images are hidden in larger host images.

B. Hiding Video in Video

In this application, nonstandard Quarter QCIF video at spatial resolution 88 $\times$ 72, and temporal resolution 7.5 frames/s is hidden inside standard 176 $\times$ 144 QCIF video at 30 frames/s. Both the videos are in 4:2:0 format, where the chrominance components Cb and Cr are downsampled by a factor of two in both vertical and horizontal directions. The host and the hidden video are synchronized in time. For the hidden video, four luminance pixels, one Cb pixel, and one Cr pixel, in each 2 $\times$ 2 window, are grouped together to form vectors of dimension 6. Each frame of a quarter QCIF video thus yields $44 \times 36 = 1584$ vectors. Using such data taken from a number of quarter QCIF videos, standard VQ's as well as channel optimized VQ's are designed for different channels. The indices obtained by vector quantization or channel-optimized vector quantization, are embedded into the LL-HH subband obtained by a two-stage orthogonal wavelet decomposition of each frame of the host QCIF video. The watermarked video is piped through a H.263 encoder, and the reconstructed video is used to extract the hidden video segment.

It is assumed that the noise introduced in the pixels of the watermarked host as a result of various transformations is additive, and i.i.d. Gaussian with variance σ^2. It follows, therefore, that the additive noise introduced in the orthogonal wavelet coefficients are also Gaussian distributed with precisely the same variance. Assuming reasonable values for σ^2, the transition probabilities for a given channel are estimated by computer simulations, and are subsequently used in the design of the channel-optimized VQ. The initial codebook for COVQ design [21] is obtained by standard Lloyd's algorithm-based VQ design, followed by an appropriate indexing scheme.

We present the results for two different channel implementations. In the first case, the data is hidden only in the luminance coefficients. The chrominance coefficients are left unperturbed to prevent occurrence of false colors. Note that the LL-HH subband of a luminance QCIF frame contains $44 \times 36 = 1584$ coefficients. Dividing the coefficients from this subband into groups of four wavelet coefficients yields $1584/4 = 396$ vectors of dimension four. The 4-D channel codebook chosen is of size 256, and comprises the Voronoi code derived from the D_4 lattice by centering the lattice at (0, 3/16, 11/32, 17/32). Conway and Sloane [18] have shown such codes to be very efficient, while having fast encoding and decoding algorithms. The source codebook is also of size 256, with each index mapping to a particular channel symbol in the channel code. Note that each frame of the hidden quarter QCIF video has 1584 source vectors, of which only a quarter (396) can be encoded in each QCIF frame. Thus, four host QCIF frames are required to complete the embedding of each hidden frame if the entire LL-HH subband is used as carriers. This constrains the maximum frame rate for the hidden video to be 1/4 the frame rate of the host QCIF video. The frame rate for the hidden video will be still less if the same data is repeated in a few successive host frames to introduce robustness to frame drops during compression.

The quarter QCIF *Hall_Monitor* sequence at 7.5 frames/s was hidden inside the 30 frames/s QCIF *Mother_Daughter* sequence using the above approach. The host video was subsequently compressed using H.263 at 30 frames/s. A standard VQ as well as three different channel-optimized VQ's with varying noise levels are designed for the 256-ary channel described above. The standard deviation σ of the assumed Gaussian noise expressed as a percentage of the packing radius (see [16]) of the D_4 lattice-based channel code are 28%, 32%, and 35%, respectively, for the three COVQ designs. The PSNR for the extracted video against the video bit rate after H.263 compression of the host at 30 frames/s is plotted in Fig. 9 for all the four cases with the same transparency constraint. As expected, at low bit rates of the watermarked host video, the channel-optimized VQ retrieval PSNR results are higher than that of standard VQ, while at higher bit rates the retrieval PSNR results for standard VQ is superior. This is because in the noise-free case, the COVQ codevectors are not optimal because they have been designed for a noisy channel. However, in the presence of noise, the attempt to minimize the overall end-to-end distortion in COVQ bears fruit. Fig. 10 compares for the σ =32% COVQ implementation, the visual quality of the watermarked and compressed frame 20 of the *Mother_Daughter* sequence with the original, and also the extracted Frame 36 of the hidden quarter QCIF *Hall_Monitor* sequence for two different bit rates with the original. The retrieval result in Fig. 10(g) is of barely acceptable quality. The spurious 2 $\times$ 2 impulses correspond to erroneous extraction of the VQ indices. The retrieval quality degrades very fast at bit rates lower than this.

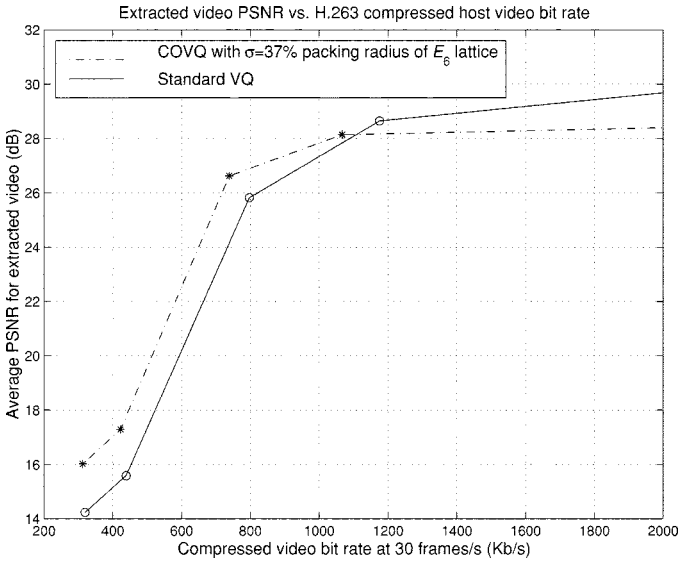

Fig. 11. Average PSNR of extracted quarter QCIF *Coastguard* sequence versus bit rate for H.263 compressed host *Akiyo* bit stream at 30 frames/s for standard VQ and COVQ, both of size 72. The channel code is of size 72, consisting of the first shell of the E_6 lattice.

In the second implementation, data is hidden in both the luminance and chrominance coefficients. The LL-HH subband of a luminance QCIF frame contains 1584 coefficients, while the same subband for the chrominance components contains 396 coefficients. Grouping four luminance wavelet coefficients from the LL-HH subband, and one wavelet coefficient from each of the Cb and Cr LL-HH subbands, yields 396 carrier vectors of dimension 6. The 6-D channel codebook is of size 72, and comprises the first shell of the E_6 lattice. The source codebook is

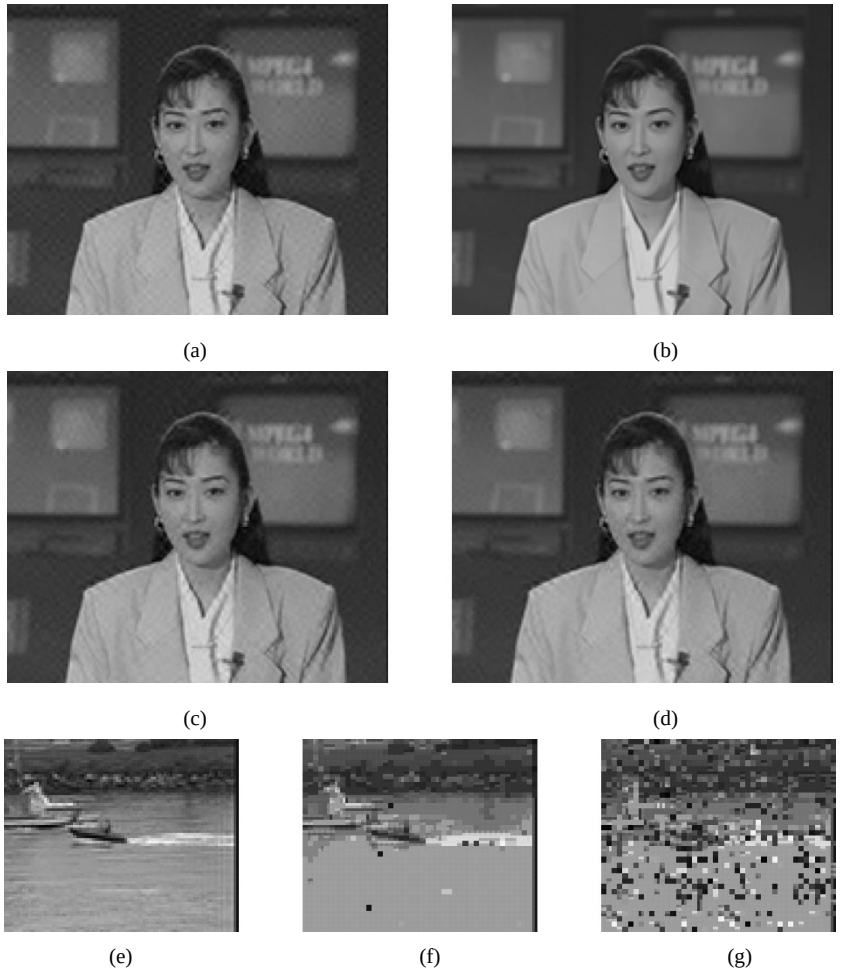

Fig. 12. Visual quality results for hiding quarter QCIF *Coastguard* in QCIF *Akiyo*, using COVQ. Frame 20 of the host video and Frame 36 of the hidden video are shown as examples.

correspondingly of size 72. As before, four QCIF frames are required to host one frame of the source quarter QCIF video.

The quarter QCIF *Coastguard* sequence at 7.5 frames/s was hidden inside the 30 frames/s QCIF *Akiyo* sequence using the above approach. The host video was subsequently compressed using H.263 at 30 frames/s. A standard VQ, as well as a channel-optimized VQ designed for i.i.d. Gaussian noise with standard deviation 37% of the packing radius of the E_6 lattice-based channel code, are designed. The PSNR for the extracted video against the video bit rate after H.263 compression of the host at 30 frames/s, is plotted in Fig. 11 for both standard VQ as well as channel-optimized VQ with the same transparency constraint. As expected, the channel-optimized VQ retrieval PSNR is lower than that obtained by standard VQ at high quality compression of the host (low noise), but overtakes the standard VQ results as the compression becomes more severe (higher noise). Fig. 12 compares for the COVQ implementation, the visual quality of the watermarked and compressed frame 20 of the *Akiyo* sequence with the original, and also the extracted frame 36 of the hidden quarter QCIF *Coastguard* sequence for two different bit rates, with the original. Note that the retrieval results in Fig. 12(f) shows considerable blockiness. This is due to the fact that a small source codebook of size 72 was used. Note further that the retrieval result in Fig. 12(g) is barely acceptable at host bit rates down to 422 kb/s.

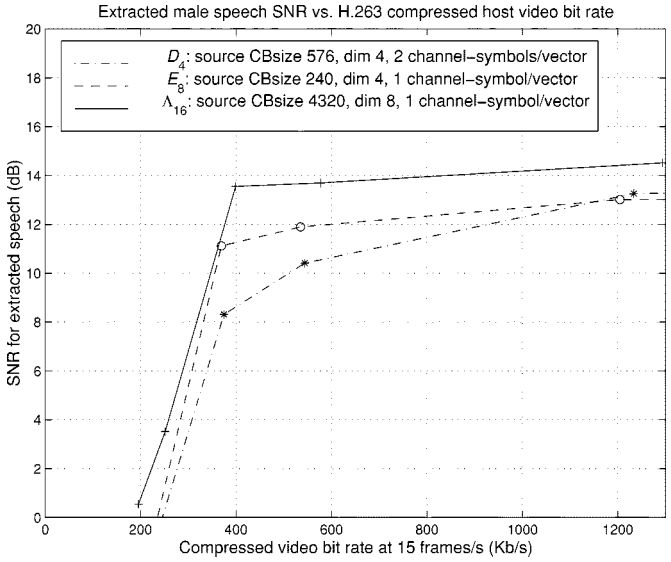

Fig. 13. SNR of extracted hidden male speech versus bit rate for H.263 compressed *News* bit stream at 15 frames/s for D_4, E_8, and L_{16} implementations.

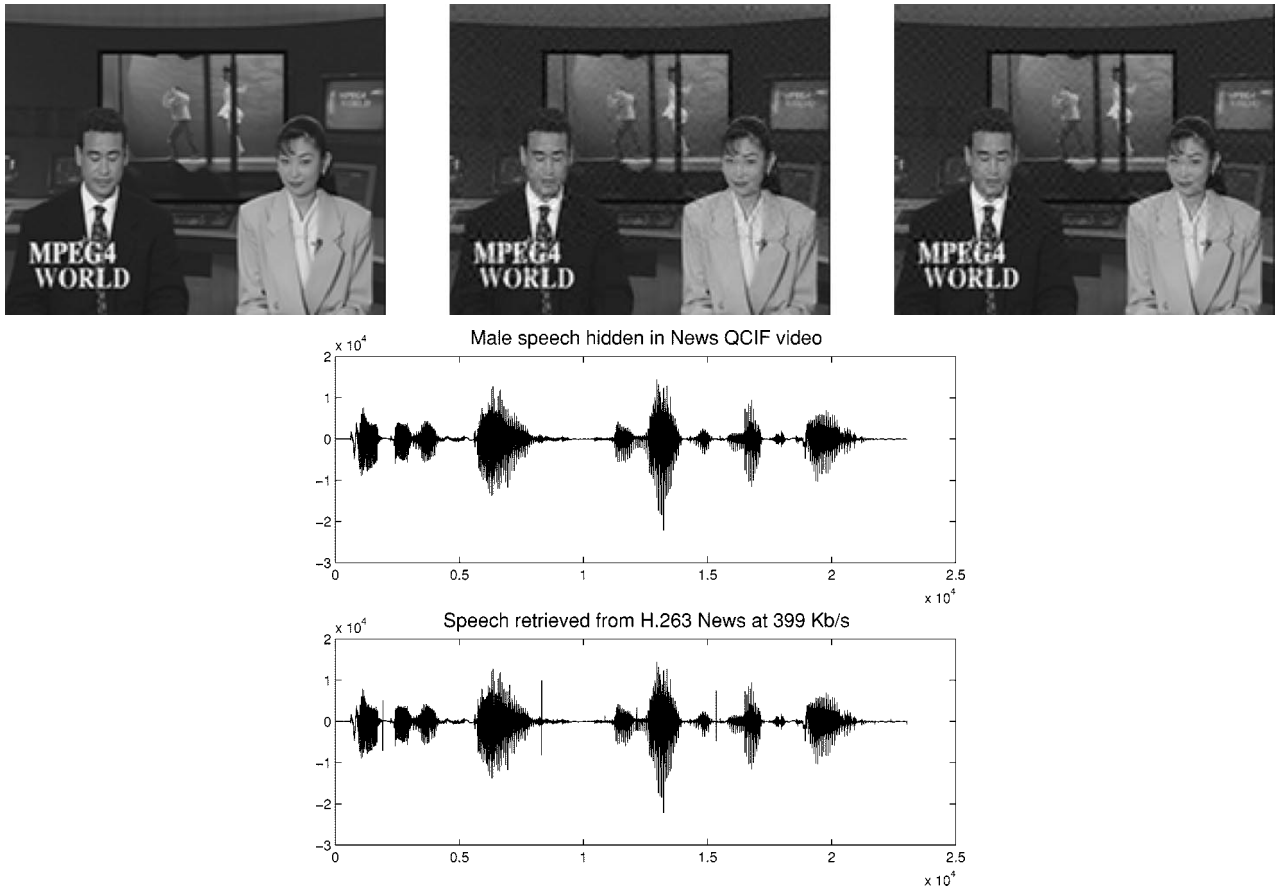

Fig. 14. Visual quality of Frame 40 of the *News* sequence, before and after L_{16} embedding and compression, together with the original and retrieved speech waveforms.

C. Hiding Speech in Video

We implemented a system for hiding 8-kHz sampled speech at 16 bits/sample in a 30 frames/s QCIF video, similar to the one reported in [3]. Successive samples of speech are vector quantized, and the indices are embedded into the LL-HH subband coefficients. Temporal redundancy is incorporated by embedding the same information in several successive frames, so that the embedding becomes robust to lower frame rate compression. The watermarked video is piped through a H.263 encoder as before, and the reconstructed video is used to extract the hidden speech segment.

First, we attempted embedding the secure speech in only the luminance LL-HH subband. We present the details of three different implementations with increasing dimensions of channel codes.

1) The speech is vector quantized with a codebook of size 576 and dimension 4. The index obtained is decomposed into two 24-ary symbols, each of which is embedded into a vector of dimension 4 obtained by grouping four luminance LL-HH coefficients of a two-stage wavelet decomposition. The embedding is done by perturbing the vectors in accordance with a spherical channel code consisting of the first shell of the D_4 lattice (which has 24 points).
2) The speech codebook is of size 240 and dimension 4. The index for each speech vector is used to perturb a group of 8 luminance LL-HH coefficients in accordance with a spherical channel code comprising the 240 points on the first shell of the E_8 lattice.
3) The speech codebook is of size 4320 and dimension 8. The index is embedded into a vector of size 16 obtained by grouping 16 luminance LL-HH coefficients. The channel

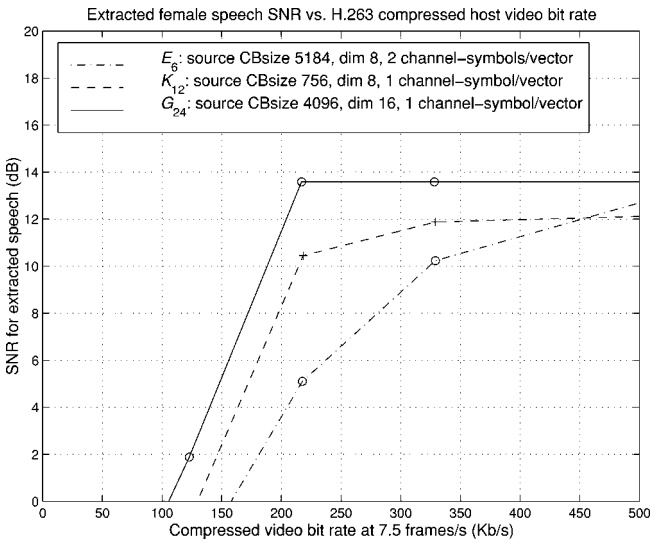

Fig. 15. SNR of extracted hidden female speech versus bit rate for H.263 compressed Grandmother bit stream at 7.5 frames/s for E_6, K_{12}, and G_{24} implementations.

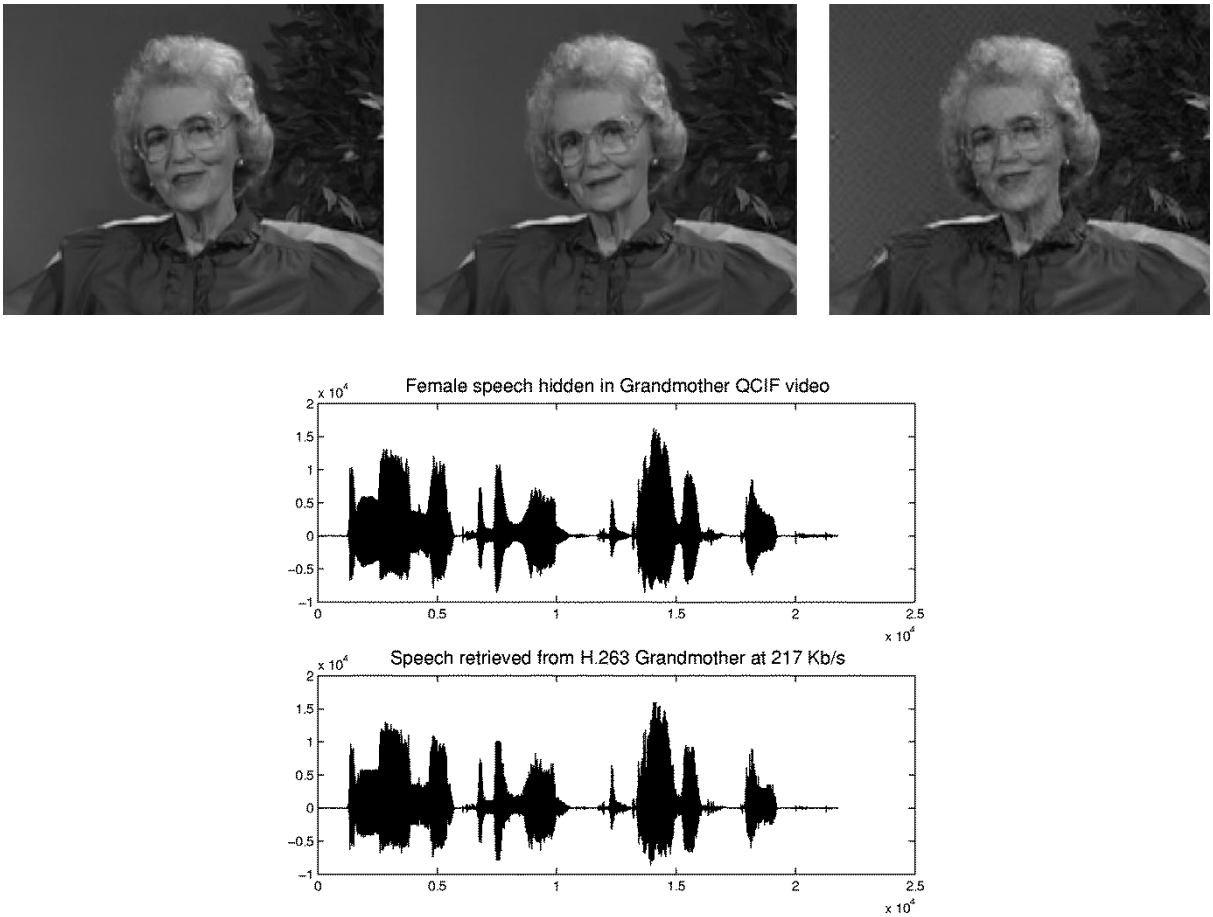

Fig. 16. Visual quality of Frame 40 of the Grandmother sequence before and after G_{24} embedding and compression, together with the original and retrieved speech waveforms.

code comprises the 4320 points on the first shell of the Barnes–Wall Lattice Λ_{16}.

For all the above implementations, the same information is repeated in two successive frames to introduce robustness to low frame-rate compression. The *News* QCIF video is used as the host for hiding a segment of male speech. The SNR for the extracted speech segment against the video bit rate after H.263 compression of the host at 15 frames/s (frameskip $=$ 1) is plotted in Fig. 13. The transparency constraint is the same for all these results. Note that the distortion in the noise-free case in the E_8 implementation is higher than that in the D_4 implementation because a smaller codebook of the same dimension 4 is used. But in presence of noise, a robust E_8-based codebook contributes to superior retrieval PSNR. The 8-D source codebook with 4320 codevectors in the Λ_{16} implementation is superior to both in the noise-free case because it can exploit correlations better, and has a larger number of codevectors. In the presence of noise also, as expected, the Λ_{16}-based channel code yields maximum robustness to noise. Fig. 14 compares the visual quality of the Λ_{16}-embedded and compressed frame 40 of the *News* sequence with the original, and also the extracted speech waveform with the original one hidden at 399 kb/s encoding. Note the spurious spikes in the retrieved speech waveform resulting of erroneous detection.

We next present results for three implementations where both the luminance and the chrominance coefficients are perturbed.

1) The speech codebook is of size 5184 and dimension 8. Each index is decomposed into two 72-ary symbols, which are embedded into two coefficient vectors of dimension 6. Each 6-D coefficient vector is obtained by grouping four luminance LL-HH coefficients and one LL-HH coefficient from each chrominance component. A spherical channel code derived from the first shell of the E_6 lattice (which also has 72 points) is used for each symbol.
2) The speech is vector quantized with a codebook of size 756 and dimension 8. A 12-D coefficient vector is obtained by grouping eight luminance LL-HH coefficients and two LL-HH coefficients from each chrominance component. A spherical channel code consisting of the 756 points on the first shell of the Coxeter–Todd lattice K_{12} is used.
3) The speech is vector quantized with a codebook of size 4096 and dimension 16. A 24-D coefficient vector is obtained by grouping 16 luminance LL-HH coefficients and 4 LL-HH coefficients from each chrominance component. A spherical channel code G_{24}, consisting of 4096 points, is used. It is obtained from the (24, 12) extended Golay code by converting zeroes to ones, and ones to negative ones.

For all the above implementations, the same information is repeated in four successive frames. Fig. 15 presents the retrieval

SNR versus bit rate results for the above methods when a segment of female speech is hidden in the Grandmother QCIF video, which is then coded by H.263 at 7.5 frames/s (frameskip $= 3$). The transparency constraint is the same for all these results. Note that the distortion in the noise-free case in the E_6 implementation is higher than that in the K_{12} implementation because a smaller codebook of the same dimension 8 is used. But, in presence of noise, a robust K_{12}-based codebook contributes to superior retrieval PSNR. The 16-D source codebook with 4096 codevectors in the G_{24} implementation yields a very efficient source VQ in the noise-free case because longer source vectors exploit correlations better. In the presence of noise, as expected, the highest dimensional channel code G_{24} is found to be vastly superior to both. Fig. 16 compares the visual quality of the G_{24}-embedded and compressed frame 40 of the *Grandmother* sequence with the original, and also the extracted speech waveform with the original speech hidden in 217 kb/s video.

V. Conclusions and Future Directions

We have presented a considerably generic framework for data hiding with special emphasis on hiding compressible secure data, such as video and speech, in host video. Our quantitative treatment of the problem is motivated by the identification of its similarity with the source and channel coding problem in digital communications, and allows achieving a desired tradeoff between the visibility of data hiding, amount of secure data hidden, and robustness to host data transformations such as compression. The compressible hidden data is vector quantized, and the indices obtained are then embedded into the host by transform domain vector perturbations using noise-resilient channel codes. Channel-optimized VQ's can be designed for added robustness to noise. An encryption-key-based shuffling and grouping of coefficients, together with uncertainties in source and channel codebooks, make unauthorized retrieval next to impossible, even with the knowledge of the basic algorithm. While the generic approach can be used with success for the case when the original host is available to the retriever, the true potential of data hiding lies in being able to extract the hidden data without using the original host. This makes possible invisible mixing of different kinds of hidden data with standard forms of open data transmission, allowing only those authorized to retrieve the additional hidden information. We showed how the generic scheme can be readily adapted to allow retrieval without knowledge of the original host. We applied the scheme to hiding large amounts of secure video and speech in host QCIF video. The watermarked video is piped through a H.263 coder. The speech and video extracted from the compressed video are found to be intelligible and of acceptable visual quality, respectively, for high enough compression ratios.

Once the equivalence between the data communication problem and the data hiding problem has been established, there are a host of enhancements that could be made to improve on the basic source and channel-coding schemes described in this paper. More sophisticated source-coding schemes rather than simple VQ for the hidden data can be used within this framework. Typically, different parts of a compressed bit stream obtained by a sophisticated compression scheme have different levels of influence on the quality of reconstruction. While a single error in some parts of the symbol stream may have a catastrophic effect on reconstruction, errors in other parts may only be of limited significance. Naturally, the more important parts need to be more heavily protected. Different source and channel codebook combinations with unequal levels of quantization and protection should then be used for different parts of the compressed hidden data. That is, the more critical information symbols need to be embedded at a lower rate for increased robustness to noise, and vice versa. In general, it may not be advisable to use a variable rate compression scheme, which are inherently less resilient to noise than fixed bit rate schemes. From purely a channel-coding perspective, while increase in the dimensionality of a channel increases robustness to noise, it also introduces difficulties in implementation. Established trellis-coded modulation schemes can be used to obtain very high dimensional effective channels, while avoiding the problems associated with too large dimensions.

Finally, we make some honest admissions about the drawbacks of the current scheme. First, to make data hiding more transparent visually, embedding of secure data must be made in perceptually insignificant areas of the host data. In this work, our treatment is based solely on the mean-squared error, which is not always the best measure in the perceptual sense. Investigations on the perceptual aspects must be made within the current framework for greater invisibility. Stable perceptual features of an image, such as activity in a particular region, edginess, or contrast may be used as cues to select the coefficients to use as carriers. Unfortunately, such image-adaptive decision schemes cannot be implemented without sacrificing the amount of secure data hidden. Second, the current work is not robust to transformations such as rotation and clipping. Transforms or features of an image that are invariant to such transformations should be investigated to rectify this drawback.

References

[1] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Data hiding for video-in-video," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, Santa Barbara, CA, Oct. 1997, pp. 676–79.

[2] J. J. Chae, D. Mukherjee, and B. S. Manjunath, "A robust data hiding technique using multidimensional lattices," in *Proc. IEEE Forum Research and Technology Advances in Digital Libraries*, Santa Barbara, CA, Apr. 1998, pp. 319–326.

[3] ——, "Color image embedding using lattice structures," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, Chicago, IL, Oct. 1998, pp. 460–64.

[4] D. Mukherjee, J. J. Chae, and S. K. Mitra, "A source and channel coding approach to data hiding with application to hiding speech in video," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, Chicago, IL, Oct. 1998, pp. 348–52.

[5] I. J. Cox, J. Killian, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Processing*, vol. 6, pp. 1673–87, Dec. 1997.

[6] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Robust data hiding for images," in *Proc. IEEE Digital Signal Processing Workshop (DSP 96)*, Norway, Sept. 1996, pp. 37–40.

[7] J. Ohnishi and K. Matsui, "Embedding a seal into a picture under orthogonal wavelet transform," in *Proc. Int. Conf. Multimedia and Computing Systems*, 1996, pp. 514–512.

[8] S. Craver, N. Memon, B. Yeo, and M. Yeoung, "Can invisible watermarks resolve rightful ownership?," in *Proc. SPIE, Storage and Retrieval for Image and Video Database V*, vol. 3022, 1997, pp. 310–321.

[9] F. Hartung and B. Girod, "Digital watermarking of raw and compressed video," *Syst. Video Commun.*, pp. 205–213, Oct. 1996.

[10] ——, "Watermarking of MPEG-2 encoded video without decoding and re-encoding," *Proc. SPIE*, vol. 3020, pp. 264–274, 1997.
[11] M. D. Swanson, B. Zhu, B. Chau, and A. H. Tewfik, "Object-based transparent video watermarking," in *Proc. IEEE Workshop Multimedia Signal Processing*, 1997, pp. 369–374.
[12] ——, "Multiresolution video watermarking using perceptual models and scene segmentation," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, Santa Barbara, CA, 1997, pp. 558–61.
[13] J. G. Proakis, *Digital Communications*, 3rd ed. New York: McGraw-Hill, 1995.
[14] N. J. A. Sloane, "Tables of sphere packings and spherical codes," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 327–338, May 1981.
[15] J. H. Conway and N. J. A. Sloane, "Voronoi regions of lattices, second moments of polytopes, and quantization," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 211–226, Mar. 1982.
[16] ——, *Sphere Packings, Lattices and Groups*, Second ed. New York: Springer-Verlag, 1993.
[17] ——, "A fast encoding method for lattice codes and quantizers," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 820–824, Nov. 1983.
[18] ——, "Fast quantizing and decoding algorithms for lattice quantizers and codes," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 227–232, Mar. 1982.
[19] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer , 1992.
[20] N. Farvardin, "A study of vector quantization for noisy channels," *IEEE Trans. Inform. Theory*, vol. IT-36, pp. 799–809, July 1990.
[21] N. Farvardin and V. Vaishampayan, "On the performance and complexity of channel-optimized vector quantizers," *IEEE Trans. Inform. Theory*, vol. 37, pp. 155–60, Jan. 1991.
[22] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Commun. Pure and Applied Math.*, vol. 4, pp. 909–96, Nov. 1988.

Debargha Mukherjee (M'98) was born in Calcutta, India, in 1970. He received the B.Tech. degree from the Indian Institute of Technology, Kharagpur, India, in 1993, and the M.S. and Ph.D. degrees from the University of California at Santa Barbara in 1995 and 1999, respectively, all in electrical and computer engineering.

He is currently a Member of Technical Staff in the Imaging Technology Department, Hewlett Packard Laboratories, Palo Alto, CA. During 1993–1994, he was a Software Engineer with Tata Information Systems Limited. In the summer of 1996, he was a summer intern in the Speech Coding Research Group, Texas Instruments, Inc., Dallas, TX. His research interests include image, video and speech compression and communication, signal processing, and information theory.

Dr. Mukherjee received the IEEE Student Paper Award at the IEEE International Conference on Image Processing, Chicago, IL, in 1998.

Jong Jin Chae was born in Seoul, Korea, in 1959. After completing his military service, he received the B.S. and M.S. degrees in electronic engineering from Sogang University, Seoul, Korea, in 1987 and 1990, respectively. He received the Ph.D. degree in electrical and computer engineering from the University of California at Santa Barbara in 1999.

Between 1987 and 1995, he was a Researcher with the Institute for Defense Information Systems (IDIS), Seoul, Korea. Since 1999, he has been a Project Manager at IDIS. His research interests include image processing/analysis, pattern recognition, and digital watermarking/data hiding.

Sanjit K. Mitra (S'59–M'63–SM'69–F'74) received the B.Sc. (Hons.) degree in physics in 1953 from Utkal University, Cuttack, India, the M.Sc. (Tech.) degree in radio physics and electronics from Calcutta University, Calcutta, India, in 1956, the M.S. and Ph.D. degrees in electrical engineering from the University of California at Berkeley in 1960 and 1962, respectively, and an Honorary Doctorate of Technology degree from the Tampere University of Technology, Tampere, Finland.

From 1962 to 1965, he was with Cornell University, Ithaca, NY, as an Assistant Professor of Electrical Engineering. He was with the AT&T Bell Laboratories, Holmdel, NJ, from June 1965 to January 1967. He has been on the faculty of the University of California since 1967, serving as a Professor of Electrical and Computer Engineering since 1977 and Chairman of the Department from July 1979 to June 1982. He has published over 500 papers on signal and image processing, 11 books, and holds five patents.

Dr. Mitra served as the President of the IEEE Circuits and Systems (CAS) Society in 1986 and as a Member-at-Large of the Board of Governors of the IEEE Signal Processing (SP) Society from 1996-99. He is currently a member of the editorial boards of *Multidimensional Systems and Signal Processing*, *Signal Processing*, *Journal of the Franklin Institute*, and *Automatika*. He is the recipient of numerous awards, including the 1973 F.E. Terman Award, the 1985 AT&T Foundation Award of the American Society of Engineering Education, the Education Award of the IEEE CAS Society in 1989, the Distinguished Senior U.S. Scientist Award from the Alexander von Humboldt Foundation of Germany in 1989, the Technical Achievement Award of the IEEE SP Society in 1996, the Mac Van Valkenburg Society Award, and the CAS Golden Jubilee Medal of the IEEE CAS Society in 1999, and the IEEE Millennium Medal in 2000. He is an Academician of the Academy of Finland, a Fellow of the AAAS and SPIE, and a Member of EURASIP and ASEE.

B. S. Manjunath (S'88–M'91) received the B.E. degree in electronics (with distinction) from Bangalore University, Bangalore, India, in 1985, the M.E. degree (with distinction) in systems science and automation from the Indian Institute of Science in 1987, and the Ph.D. degree in electrical engineering from the University of Southern California in 1991.

He joined the Electrical and Computer Engineering Department, University of California at Santa Barbara, in 1991, where he is currently an Associate Professor. His current research interests include multimedia databases, digital libraries, image processing, and computer vision. He also an active participant in the ISO/MPEG-7 standardization.

Dr. Manjunath was a recipient of the National Merit Scholarship during 1978-85 and was awarded the Bangalore University Gold Medal for the best graduating student in electronics engineering in 1985. He is currently an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING and was a Co-Guest Editor of its *Special Issue on Image and Video Processing for Digital* (January 2000.)