Text Data-Hiding for Digital and Printed Documents: Theoretical and Practical Considerations

R. Villán, S. Voloshynovskiy, O. Koval, J. Vila,
E. Topak, F. Deguillaume, Y. Rytsar, and T. Pun

Computer Vision and Multimedia Laboratory - University of Geneva
24, rue du Général-Dufour - 1211 Geneva 4, Switzerland

Keywords: Text Data-Hiding, Gel'fand-Pinsker Problem, Color Quantization, Halftone Quantization.

ABSTRACT

In this paper, we propose a new theoretical framework for the data-hiding problem of digital and printed text documents. We explain how this problem can be seen as an instance of the well-known Gel'fand-Pinsker problem. The main idea for this interpretation is to consider a text character as a data structure consisting of multiple quantifiable features such as shape, position, orientation, size, color, etc. We also introduce *color quantization*, a new semi-fragile text data-hiding method that is fully automatable, has high information embedding rate, and can be applied to both digital and printed text documents. The main idea of this method is to quantize the color or luminance intensity of each character in such a manner that the human visual system is not able to distinguish between the original and quantized characters, but it can be easily performed by a specialized reader machine. We also describe halftone quantization, a related method that applies mainly to printed text documents. Since these methods may not be completely robust to printing and scanning, an outer coding layer is proposed to solve this issue. Finally, we describe a practical implementation of the color quantization method and present experimental results for comparison with other existing methods.

1. INTRODUCTION

In the past decade, a number of data-hiding schemes have been proposed in literature, however, the majority of them deals only with digital image, audio or video documents. Nonetheless, text documents, either in printed or electronic form, are still the most common and almost unavoidable form of information communication among humans. Text documents are omnipresent everyday in the form of newspapers, books, web pages, contracts, advertisements, checks, identification documents, etc.

One possible explanation of this situation is that text documents have a relatively small number of features that can be exploited in order to hide (or embed) information in comparison to image, audio or video documents. Indeed, a text document can be seen as a form of a highly structured image, which is precisely the kind of images to which the human visual system (HVS) is more sensitive. For the same reason, the data embedding rate in text documents is comparatively much smaller than those in image, audio or video documents.

Four major groups of methods for data-hiding in text documents have appeared in literature: *syntactic methods*,[1,2] where the diction or structure of sentences is transformed without significantly altering their meaning; *semantic methods*,[1,2] where words are replaced by their synonyms and/or sentences are transformed via suppression or inclusion of noun phrase coreferences; *open space methods*,[1,3] where either the inter-line space, the inter-word space or the inter-character space is modulated; and *character feature methods*,[3–5] where features such as shape, size or position are manipulated.

However, syntactic and semantic methods are not suitable for all types of documents (e.g. contracts, identity documents, literary texts) and need, in general, human supervision. Some open space methods such as inter-line space modulation and inter-word space modulation can be automated, are robust against printing and scanning, but have low information embedding rates. On the other hand, inter-character space modulation and existing character feature methods have higher information embedding rates, but are less or not robust at all against

For further information contact S. Voloshynovskiy. E-mail: svolos@cui.unige.ch (http://sip.unige.ch)

printing and scanning. Since automation is very important for practical applications, we will not consider in this paper syntactic or semantic methods.

Is the problem of text data-hiding too difficult to solve? The answer to this question depends on the application requirements. For example, for copyright protection applications where robust data-hiding is required, the answer might be YES, since the attacker can always use optical character recognition (OCR) to completely remove the hidden data. On the other hand, we advocate that the answer is NO for applications where either semi-fragile or fragile data-hiding is required (e.g. identification, authentication and tamper proofing applications[6]).

The goal of this paper is to propose a new theoretical framework for the text data-hiding problem and introduce a new semi-fragile text data-hiding method, *color quantization*, that is fully automatable, has high information embedding rate, is resistant to printing and scanning, and can be applied simultaneously to both digital and printed text documents.

This paper is organized as follows. The interpretation of the text data-hiding problem as a Gel'fand-Pinsker problem is given in Section 2. Concrete examples of Gel'fand-Pinsker text data-hiding are explained in Section 3. Experimental results about a practical implementation of the color quantization method are given in Section 4. Finally, Section 5 concludes this paper.

Notations. We use capital letters, e.g. X, to denote scalar random variables, bold capital letters, e.g. $\mathbf{X}$, to denote vector random variables, and corresponding small letters, e.g. x and $\mathbf{x}$, to denote their realizations. The superscript N is used to designate length–N vectors, e.g. $\mathbf{x} = x^N = (x_1, x_2, \ldots, x_N)$, with $n-$th element x_n. We use $X \sim p_X(\cdot)$ to indicate that the random variable X is distributed according to $p_X(\cdot)$. When no confusion is possible we write $p(x)$ instead of $p_X(x)$. The mathematical expectation of a random variable $X \sim p_X(\cdot)$ is denoted by $\mathrm{E}_{p_X}[X]$ or simply by $\mathrm{E}[X]$. $\mathrm{Var}[X]$ or σ_X^2 denote the variance of X. Calligraphic letters, e.g. $\mathcal{X}$, denote sets and $|\mathcal{X}|$ denotes the cardinality of $\mathcal{X}$.

2. TEXT DATA-HIDING AS A GEL'FAND-PINSKER PROBLEM

The Gel'fand-Pinsker problem of digital communications with noncausal side information available at the encoder is depicted in Figure 1. An encoder sends a message $m \in \mathcal{M} = \{1, 2, \ldots, 2^{NR}\}$, $|\mathcal{M}| = 2^{NR}$, to a decoder by

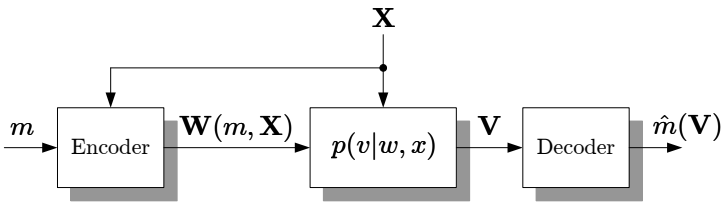


Figure 1. The Gel'fand-Pinsker problem.

mapping it into a length–N sequence $\mathbf{W}(m, \mathbf{X})$. The channel $p_{V|W,X}(v|w,x)$ has interference in the form of a sequence $\mathbf{X}$ that is output from a discrete memoryless source according to $p_{\mathbf{X}}(\cdot)$, i.e.:

$$p_{\mathbf{X}}(\mathbf{x}) = \prod_{n=1}^{N} p_X(x_n).$$

This channel is assumed to be memoryless, i.e.:

$$p_{\mathbf{V}|\mathbf{W},\mathbf{X}}(\mathbf{v}|\mathbf{w},\mathbf{x}) = \prod_{n=1}^{N} p_{V|W,X}(v_n|w_n,x_n).$$

Moreover, the entire sequence $\mathbf{X}$ is available in a noncausal fashion at the encoder but not at the decoder. The goal is to design the encoder and decoder to maximize the communication rate R while ensuring that

$\Pr\{\hat{m}(\mathbf{V}) \neq m\}$ can be made an arbitrarily small positive number. The capacity C is the supremum of such rates R.

Gel'fand and Pinsker[7] showed that the capacity C of this channel is:

$$C = \max_{p(u,w|x)} [\, \mathrm{I}(U;V) - \mathrm{I}(U;X) \,],$$

where U is an auxiliary random variable with conditional distribution $p_{U|X}(u|x)$ such that $V \leftrightarrow (W,X) \leftrightarrow U$ form a Markov chain.

The codebook construction in the proof of the achievability part of this theorem uses a random binning technique and the concept of strong joint typicality*. The main idea is to trade-off the number of codewords needed at the encoder in each message bin in order to cancel the interference $\mathbf{X}$ and the number of uniquely distinguishable codewords at the decoder.

Codebook construction: Introduce an auxiliary random variable U with alphabet $\mathcal{U}$ via $p_{U|X}(u|x)$. Generate $|\mathcal{J}||\mathcal{M}|$ codewords $\mathbf{u}(m,j)$, where $m \in \mathcal{M}$, $j \in \mathcal{J} = \{1,2,\ldots,2^{NR'}\}$, $|\mathcal{J}| = 2^{NR'}$, independently at random according to the marginal distribution $p_U(u)$. The number NR' can be interpreted as the number of bits used to represent the interference $\mathbf{x}$. The codebook is organized as shown in Figure 2.

$$\begin{pmatrix} & \begin{pmatrix} \mathbf{u}(1,1) \\ \mathbf{u}(1,2) \\ \vdots \\ \mathbf{u}(1,|\mathcal{J}|) \end{pmatrix} \\ 1 & \\ & \\ \vdots & \vdots \\ & \\ & \begin{pmatrix} \mathbf{u}(|\mathcal{M}|,1) \\ \mathbf{u}(|\mathcal{M}|,2) \\ \vdots \\ \mathbf{u}(|\mathcal{M}|,|\mathcal{J}|) \end{pmatrix} \\ |\mathcal{M}| & \end{pmatrix}$$

Figure 2. Gel'fand-Pinsker codebook construction.

Encoder (see Figure 3(a)): Given the message m and the interference $\mathbf{x}$, the encoder seeks a codeword $\mathbf{u}(m,j)$ such that $(\mathbf{u}(m,j),\mathbf{x}) \in A_\epsilon^{*(N)}(U,X)$, i.e. the encoder seeks a strongly jointly typical pair $(\mathbf{u}(m,j),\mathbf{x})$ in the set of strongly jointly typical sequences $A_\epsilon^{*(N)}(U,X)$. Therefore, the message m defines the bin and the interference $\mathbf{x}$ is exploited to select a particular codeword $\mathbf{u}(m,j)$ from this bin (see Figure 2). If such a codeword is found, the encoder produces $\mathbf{w}$ according to a deterministic mapping $\mathbf{w} = f^N(\mathbf{u}(m,j),\mathbf{x})$.

Decoder (see Figure 3(b)): Given the channel output $\mathbf{v}$, the decoder seeks a codeword $\mathbf{u}(m,j)$ such that $(\mathbf{u}(m,j),\mathbf{v}) \in A_\epsilon^{*(N)}(U,V)$ in the set of all $|\mathcal{J}||\mathcal{M}|$ codewords. If the decoder finds a unique strongly jointly typical pair, then it declares that the sent message was $\hat{m} = m$. Otherwise, an error is declared.

We explain now how the text data-hiding problem can be considered as a particular instance of the Gel'fand-Pinsker problem† (see Figure 4). The text, where some message m is to be hidden, is represented by $\mathbf{X}$ and called cover text. Each component X_n, $n = 1,2,\ldots,N$, of $\mathbf{X}$ represents one character from this text. Here, we

*The concept of strong typicality is nicely introduced in the book by Cover and Thomas[8]

†Usually, in the context of data-hiding, a secret key K is shared between both encoder and decoder. The secret key K is used for security purposes. For the sake of completeness Figure 4 explicitly shows the secret key K. However, since in this paper we do not perform a security analysis, we will not refer to it.

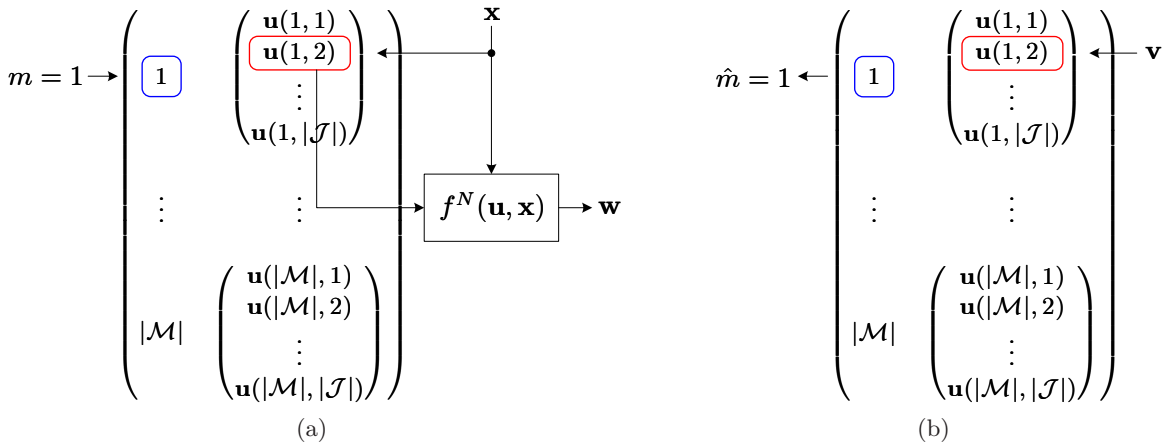


Figure 3. (a) Gel'fand-Pinsker encoder; (b) Gel'fand-Pinsker decoder.

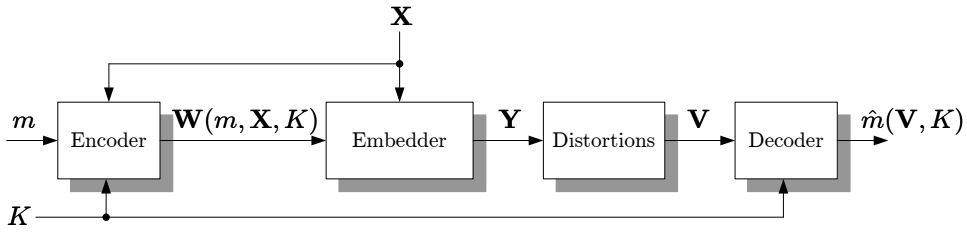


Figure 4. Text data-hiding as an instance of the Gelfand-Pinsker problem.

define a character as an element from a given language alphabet (for instance, the latin alphabet {A, B, ..., Z}). This alphabet can also contain punctuation characters as well as other special characters[‡]. To be more precise, we conceive each character X_n as a data structure consisting of multiple *quantifiable* component fields (features): *shape* (geometric definition), *position*, *orientation*, *size*, *color*, etc. Given a message m and the cover text $\mathbf{X}$, the encoder looks for a jointly strongly typical pair $(\mathbf{U}, \mathbf{X}) \in A_\epsilon^{*(N)}(U, X)$. The watermark $\mathbf{W}$ is found via a deterministic mapping $\mathbf{W} = f^N(\mathbf{U}, \mathbf{X})$. The influence of the channel $p(v|w, x)$ is divided in two stages. In the first stage, $\mathbf{W}$ and $\mathbf{X}$ are combined via a deterministic mapping, defined as the embedder, to produce the stego text $\mathbf{Y} = \psi^N(\mathbf{W}, \mathbf{X})$. In general, $\mathbf{Y}$ needs to satisfy a number of requirements, which can be different according to the selected application. In the second stage, $\mathbf{Y}$ may suffer from some intentional or unintentional distortions. We denote by $\mathbf{V}$ the resulting distorted version of the stego text. Finally, $\mathbf{V}$ is fed to the decoder, which tries to obtain an estimate $\hat{m}$ of message m.

2.1. Costa's Problem

Costa considered the Gel'fand-Pinsker problem in the particular case of zero-mean additive white Gaussian interference and zero-mean additive white Gaussian noise (AWGN):

$$V = W + X + Z, \quad X \sim \mathcal{N}(0, \sigma_X^2), \quad Z \sim \mathcal{N}(0, \sigma_Z^2).$$

Given the watermark power constraint $\mathrm{E}\left[\|\mathbf{W}\|^2\right] \leq N\sigma_W^2$, Costa[9] demonstrated that if U is defined via $U = W + \alpha X$, where $W \sim \mathcal{N}(0, \sigma_W^2)$ is independent from X and $\alpha = \sigma_W^2/(\sigma_W^2 + \sigma_Z^2)$, then the capacity of this channel coincides with the capacity of the AWGN channel $V = W + Z$, i.e.:

$$C = C_{\mathrm{AWGN}} = \frac{1}{2}\log_2\left(1 + \frac{\sigma_W^2}{\sigma_Z^2}\right).$$

[‡]In fact, we show in Sections 3 and 4 that the particular language alphabet is not relevant for the practical implementation of a text data-hiding scheme based on the Gel'fand-Pinsker framework.

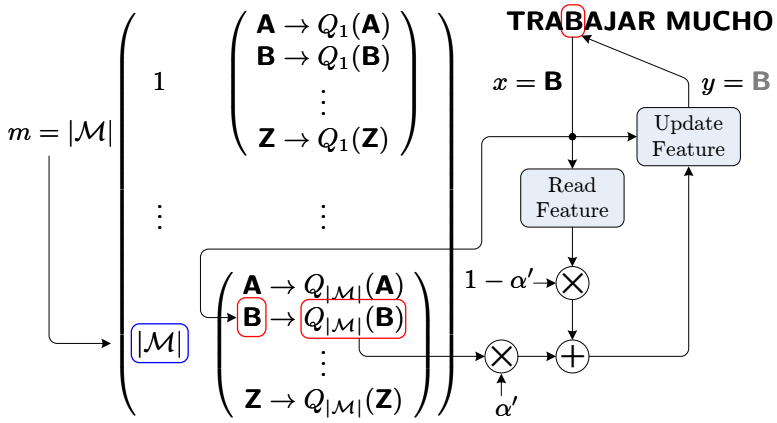


Figure 5. SCS codebook for text data-hiding ($N = 1$).

2.2. Practical Implementation of Costa's Problem

It should be noticed that Costa's result still makes use of *random* codebooks with an exponential number of codewords in order to achieve capacity. To reduce the complexity of practical implementations of such coding schemes, the use of *structured* codebooks instead of random ones has been proposed.[10, 11] These codebooks are designed based on high-rate quantizers providing the independence between W and X. For example, in the so-called Scalar Costa Scheme[11] (SCS) the auxiliary random variable U is approximated by:

$$U = W + \alpha' X = \alpha' Q_m(X),$$

where $Q_m(\cdot)$ is a scalar quantizer for the message m and α' is a compensation parameter. This produces a uniformly distributed watermark $W = U - \alpha' X = \alpha' Q_m(X) - \alpha' X$. The resulting stego text is obtained as:

$$Y = W + X = \alpha' Q_m(X) + (1 - \alpha')X. \tag{1}$$

To illustrate the Gel'fand-Pinsker text data-hiding interpretation, we consider a practical implementation based on the SCS. For this, we select one character feature, e.g. color, and use it as the cover character X in (1). We show in Figure 5 the resulting SCS codebook and an illustration of how to use it for text data-hiding.

Naturally, one can select simultaneously more than one character feature, e.g. size and color. In this case, the quantizer $Q_m(\cdot)$ in (1) becomes a vector quantizer $\mathbf{Q}_m(\cdot)$ acting on the selected character features. This scheme has the advantage of having a higher data embedding rate.

Moreover, the generalization of SCS to the vector case, where N-dimensional vector quantizers are used instead of scalar ones, is also perfectly suited for the text data-hiding problem. For this setup, we have:

$$\mathbf{U} = \mathbf{W} + \alpha' \mathbf{X} = \alpha' \mathbf{Q}_m(\mathbf{X}),$$

and:

$$\mathbf{Y} = \mathbf{W} + \mathbf{X} = \alpha' \mathbf{Q}_m(\mathbf{X}) + (1 - \alpha')\mathbf{X}. \tag{2}$$

In our interpretation, this accounts for taking groups of characters in order to build a suitable codebook. For example, for $N = 8$, such a codebook would contain the entry $\mathbf{Q}_m(\mathbf{TRABAJAR})$, corresponding to message m and the group of characters $\mathbf{x} = \mathbf{TRABAJAR}$. Once again, one may consider using one or multiple features per character.

Notice finally that all open space methods modifying either inter-character, inter-word or inter-line space are all particular cases of the more general scheme described by (2), where the exploited character feature is *position*. Moreover, all previously proposed character feature methods can be also considered as particular cases of the scheme described by (1).

3. PRACTICAL IMPLEMENTATION OF GEL'FAND-PINSKER TEXT DATA-HIDING

In identification, authentication and tamper proofing applications, the alteration of the hidden data allows to indicate that the text document has been modified. Thus, either fragile or semi-fragile text data-hiding is needed. A fragile method, intolerant to any modification, would only address digital documents, whereas a semi-fragile one, robust against some unintentional attacks (e.g. print-and-scan), would address both digital and printed documents.[12]

In our view, the main requirements for a semi-fragile text data-hiding method should be the following:

R1. It should work for text documents both in electronic and printed forms.

R2. It should be independent of the document file format provided that this format supports a reasonable level of text description. Modern file formats satisfying this condition are: Microsoft Office Word (DOC), Rich Text Format (RTF), PostScript (PS), Portable Document Format (PDF), LaTeX, Hypertext Markup Language (HTML), and many others.

R3. Marked digital text documents should be converted from one format to another retaining the hidden information in a transparent manner for the end user.

R4. Marked text documents should be perceptually undistinguishable from the original versions.

R5. It should have a high information embedding rate. Even a single page of text should be capable to hold some basic information. For example, the author's name, time and date of creation, comments, etc.

R6. It should be easy to automate. Automation and unsupervised processing are very important features to make the solution attractive for practical applications.

As an illustration of the theory developed in Section 2, we explain two semi-fragile text data-hiding methods. The first method, *color quantization*, is new and can be used for both digital and printed text documents. The second method, *halftone quantization*, mainly addresses printed text documents.

3.1. Color Quantization

In this method,[13] the stego text is obtained via (1), where $\alpha' = 1$, $Q_m(\cdot)$ is a scalar quantizer and the character feature X to quantize is color (see also Figure 5). The main idea of this method is to quantize the color of each character in such a manner that the HVS is not able to distinguish between the original and quantized characters, but it is still possible to do it by a specialized reader, e.g. a high dynamic range scanner in the case of printed documents.

An example illustrating this method is shown in Fig 6. Therein, dark characters encode a 0, whereas light ones encode a 1. Thus, a binary sequence can be sequentially embedded into the cover text. Notice that the embedding rate is comparatively higher than the rate of inter-line or inter-word space modulation methods. Furthermore, according to the desired level of robustness against digital-to-analog-to-digital (D-A-D) conversion, e.g. print-and-scan, one can choose which characters will be used to embed the data and which ones will be ignored. Indeed, small characters, like periods and commas, may not be good information carriers for printed text documents. This problem is less likely to occur in a digital-only environment.

(a) (b)

Figure 6. Color quantization: (a) original text; (b) marked text (exaggerated).

Obviously, this method satisfies requirements R1, R2, and R3. Requirement R4 is also satisfied because we know from the HVS characteristics that slight luminance variations will not be noticed by the human eye.

Moreover, luminance variations over bright (and dark) backgrounds are less visible than luminance variations over gray backgrounds.[14] Luckily, most text documents are written using dark color characters over a bright background. By using a modern word processor, one can easily verify that in a digital-only environment this method can embed up to 4 bits per character (using gray levels from 0 to 15) while still satisfying requirement R4. If the digital stego text document is to be printed-and-scanned, we expect an embedding rate of 1-2 bits per character. Concerning the automated processing of printed-and-scanned stego text documents, one can use one of the numerous existing document segmentation algorithms depending on the targeted application and on the a priori knowledge of the document layout.[15] The correct segmentation of individual characters is essential in order to estimate their color; however, the recognition of the characters themselves (OCR) is not necessary. Therefore, our method also satisfies requirements R5 and R6.

From the point of view of practical applications, this method is efficient for identification, authentication and tamper proofing because of the following reasons:

- As indicated above, robustness against D-A-D conversion can be accomplished by selecting the set of characters to be used to embed the data. Further improvement can be obtained by grouping small sequences of characters and/or by using error control codes.
- Repetitive embedding can be used to recover the hidden data from incomplete text documents and to perform both synchronization and channel compensation prior to decoding.

The reader is referred to Deguillaume et al.[12,13] for further information on the last two points.

Two-level quantizer. The easiest method to embed information is by using a two-level quantizer. In this approach, we fix a reference color representing bit 0. A good choice is to use the document's original color (most of the time black). We choose a lighter shade in order to represent bit 1. We show in Figure 7 a concrete example of this method, where we used $Q_0(x) = 0$ (black) and $Q_1(x) = 46$ to mark the characters. The experimental performance evaluation of this method, for both digital and printed text documents, is given in Section 5.

Four major groups of methods for data-hiding in digital text documents have appeared in literature: syntactic methods, where the diction or structure of sentences is transformed without significantly altering their meaning; semantic methods, where words are replaced by their synonyms and/or sentences are transformed via suppression or inclusion of noun phrase coreferences; open space methods, where either inter-line space, inter-word space or inter-character space is modulated; and character feature methods, where features such as shape, size or position are manipulated.

(a)

Four major groups of methods for data-hiding in digital text documents have appeared in literature: syntactic methods, where the diction or structure of sentences is transformed without significantly altering their meaning; semantic methods, where words are replaced by their synonyms and/or sentences are transformed via suppression or inclusion of noun phrase coreferences; open space methods, where either inter-line space, inter-word space or inter-character space is modulated; and character feature methods, where features such as shape, size or position are manipulated.

(b)

Figure 7. Two-level color quantization: (a) original text; (b) marked text using $Q_0(x) = 0$ and $Q_1(x) = 46$.

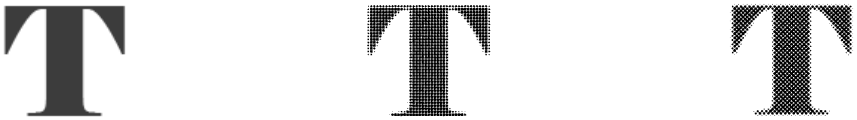
Multilevel quantizer. The above method can be easily extended to multiple levels. Instead of using two color levels, one may use four or eight levels, i.e. a multilevel quantizer. The data embedding rate could in this case be multiplied by a factor of two or three, according to the number of embedded bits per character. Naturally, for practical applications, this method relies on the quality of the printing and scanning machines. With the advent of very high quality printers, e.g. very high resolution ink jet printers, and very high dynamic range scanners, this extension should not be disregarded.

3.2. Halftone Quantization

This method, which is similar to the one proposed by Matsui and Tanaka,[16] relies on *halftoning*, a widely used printing technology that enables continuous tone images to be printed with one color ink (grayscale) or a few color inks (color). Here, we restrict our discussion to black & white printers.

In order to simulate a given gray shade a halftone printer uses a halftone screen. Our method exploits the fact that there exist several possible choices for the halftone screen leading to the same gray shade. Therefore, one can use this property in order to hide data on each text character by using a different halftone screen according to the message m that we wish to embed. Typical halftone screen characteristics that can be exploited for this purpose are: screen angle and screen dot shape (elliptical, round, square).

Again, the stego text is obtained via (1) for $\alpha' = 1$. However, the quantizer $Q_m(\cdot)$ must be seen as a vector quantizer acting on the gray values of the pixels making up each character. We show in Figure 8 an example of this method where a screen angle of 0° is used to encode bit 0 whereas a screen angle of 45° is used to encode bit 1. The major strength of this method is that all characters in the stego text will have the same grade shade. On the other hand, unless combined with the color quantization method, this method is intended mainly for printed documents. For example, if $|\mathcal{M}| = 2$, one may use the set of grade shades $\{Q_0(x) = 45, Q_1(x) = 46\}$ to embed binary data into the digital version of the text document, and a halftone pattern screen together with two screen angles to embed binary data into the printed version of the text document.



(a) (b) (c)

Figure 8. Halftone quantization: (a) original character; (b) marked character for $m = 0$; (c) marked character for $m = 1$.

3.3. Error Control Coding for Print-and-Scan Channels

The quantization-based methods explained above, which are by themselves channel codes, may not be completely robust to printing and scanning, i.e the decoder's output may contain some errors. In fact, it can be verified by experimentation that there is a trade-off between the invisibility of the watermark and the decoding accuracy. In order to reduce the error rate to an acceptable level, an outer layer of coding can be used. The correct design of such an outer layer takes into account the channel formed by the quantization encoder, the print-and-scan channel, and the quantization decoder. However, the operation of the overall decoding machinery may require some modifications in order to get full benefit of soft-decision decoding techniques. For example, the quantization decoder may be modified so that it outputs soft estimates rather than hard estimates. In the case of the color quantization method, effective coding techniques for print-and-scan channels have already been studied in the context of two-dimensional bar codes.[17] The main idea is to consider each character as a 2D symbol and use a suitable model for the print-and-scan channel. In particular, if a multilevel quantizer is used, then a multilevel encoder together with a multistage decoder can be designed to reliably communicate information through any print-and-scan channel. The interested reader is refereed to the above publication for more details on the implementation of this approach.

4. EXPERIMENTAL RESULTS

In this part, we describe a practical implementation of the color quantization method described in Section 3.1. As explained above, this method can be used for both digital and printed text documents.

The implementation of this method in a digital-only environment is straightforward. In our experiments, we implemented a prototype for Microsoft Office Word documents capable of embedding and extracting any arbitrary message. Assuming perfect synchronization when reading the marked characters, our prototype was able to extract the embedded messages without any errors. Thus, for this case, the use of error control codes

for the reliable extraction of an embedded message is not needed. We also verified that the conversion from the DOC format to the PDF or PS formats retains the color information of each character. Our implementation was also able to successfully extract the embedded message from the PDF or PS versions of a DOC document.

Now, we describe an extended implementation of the color quantization method for text documents that are subject to D-A-D conversion. This implementation considers only a two-level quantizer, but it can be readily extended to consider multilevel quantizers. In Table 1, we list the exploited equipment for performing the experiments. The default printer parameters (resolution, screen frequency, halftone algorithm) were used

Table 1. Equipment Used for Experimentation

Model	Type
HP Color LaserJet 4600	laser printer
Epson Perfection 3170 Photo	CCD scanner
Epson Perfection 4990 Photo	CCD scanner
Canon LiDE 50	CCD scanner

for printing the digital text documents. For scanning the printed text documents, we used a resolution of $r_s = 600$ ppi, grayscale mode, 8 bits of bit-depth, full dynamic range (from 0 to 255 according to the bit-depth setting), γ-correction set to 1, and an unsharp mask filter of high level according to each scanner's driver interface.

For the sake of simplicity, we first selected a set of random black digital texts written using the Latin alphabet (A,B, ..., Z, a, b, ..., z), common punctuation and special symbols (comma, period, colon, semicolon, -, ?, !, “, ”, ‘, ’, (,), <, >, @, |), numbers (0, 1, ..., 9), and arithmetic operators (+, −, ∗, /, =). Our practical system can nonetheless work with other languages using other alphabets. We used Arial font characters of size 10 pt (1 pt $\approx$ 1/72 in). In order to be robust against printing and scanning, some characters were deliberately not used for embedding information. These characters are the following: comma, period, colon, semicolon, “, ”, ‘, ’, -. Secondly, an equal number of arbitrary messages were embedded into the digital texts using the two-level color quantization method. The marked digital texts were subsequently printed and scanned with the equipment listed in Table 1. Finally, the scanned digital texts were processed in order to extract the embedded messages.

The extraction process can be divided in three parts: segmentation of characters, demodulation of the character feature (in this case, color), and quantization-based decoding.

For the segmentation of characters, we assumed good compensation (document deskewing) of a possible document misalignment while scanning. The implemented segmentation algorithm is based on off-the-shelf methods for character segmentation.[3, 18] We briefly explain the involved steps (see also Figure 9):

i. Apply a γ-correction factor of 2.6,

ii. Identify the text line boundaries by computing the luminance vertical profile (i.e. by projecting the pixel luminance values onto the Y-axis),

iii. Provisionally eliminate the halftone patterns via a local median filter,

iv. Identify the text character boundaries on each line using the SUSAN edge detector,[18]

v. Identify the areas where the demodulation of features will be performed.

Due to the used printing technology (halftoning), one has two options for the demodulation of the character feature: computation of the character's average luminance or analysis of the character's halftone pattern (in this case, quantify whether a halftone pattern is present or not). We tested both approaches and found that the latter is more robust against printing and scanning.

Finally, the two-level quantization decoder was implemented by experimental optimization of a threshold.

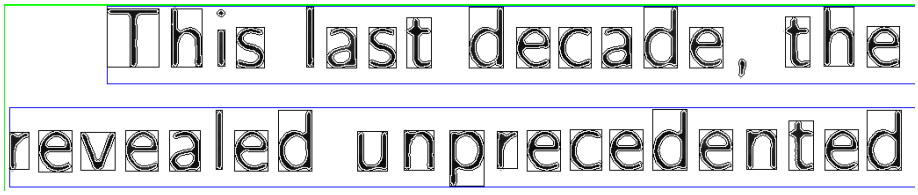


Figure 9. Segmentation of characters.

The obtained results were similar for all the exploited scanners. For this reason, we only show in Table 2 the results for the Epson Perfection 3170 Photo scanner. These results were obtained using halftone pattern demodulation for texts of $J = 4104$ characters. In this table, $Q_0(x)$ and $Q_1(x)$ represent the luminance values employed to mark the characters.

Table 2. Performance of the two-level color quantization method ($J = 4104$)

$Q_0(x)$	$Q_1(x)$	Error count	Error rate
0	41	1342	32.7%
0	46	824	20.1%
0	51	315	7.7%
0	56	120	2.9%
0	61	62	1.5%
0	66	23	0.6%

5. CONCLUSIONS

In this paper, we proposed a new theoretical framework for the problem of data-hiding in text documents. We explained how this problem can be seen as an instance of the well-known Gel'fand-Pinsker problem. In particular, we considered Costa's setup and the family of quantization-based methods in order to show how they can be applied in text data-hiding applications. The main idea was to consider a text character as a data structure consisting of multiple quantifiable features such as shape, position, orientation, size, color, etc. We showed that previous text data-hiding techniques, namely open space methods and character feature methods, are particular cases of a general quantization-based text data-hiding technique. Finally, we presented color quantization as a new method for semi-fragile data-hiding in digital and printed text documents. The experimental work confirmed that this method has high perceptual invisibility, high information embedding rate, and is fully automatable. It was also emphasized that this method is suitable for document identification, authentication, and tamper proofing applications. Our paper[6] presents the results concerning the application of this method to the authentication problem.

ACKNOWLEDGMENTS

This paper was partially supported by the Swiss National Science Foundation (SNF) professorship grant no. PP002-68653/1, the Interactive Multimodal Information Management (IM2) project, and the European Commission through the IST program under contract IST-2002-507932 ECRYPT and the sixth framework program under the number FP6-507609 SIMILAR. The information in this document reflects only the author's views, is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

REFERENCES

1. W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Systems Journal* **35**(Nos 3&4), pp. 313–336, 1996.
2. M. Topkara, C. Taskiran, and E. J. Delp, "Natural Language Watermarking," in *Proceedings of SPIE-IS&T Electronic Imaging 2005, Security, Steganography, and Watermarking of Multimedia Contents VII*, (San Jose, USA), January 17–21 2005.
3. J. T. Brassil, S. Low, and N. F. Maxemchuk, "Copyright protection for electronic distribution of text documents," *Proceedings of the IEEE (USA)* **87**(7), pp. 1181–1196, 1999.
4. A. K. Bhattacharjya and H. Ancin, "Data Embedding in Text for a Copier System," in *Proceedings of the ICIP*, **2**, pp. 245–249, 1999.
5. Q. Mei, E. K. Wong, and N. Memon, "Data hiding in binary text documents," in *Proceedings of SPIE, Security and Watermarking of Multimedia Contents III*, **4314**, pp. 369–375, August 2001.
6. S. Voloshynovskiy, O. Koval, R. Villán, E. Topak, J. E. Vila-Forcén, F. Deguillaume, Y. Rytsar, and T. Pun, "Information-Theoretic Analysis of Electronic and Printed Document Authentication," in *Proceedings of SPIE-IS&T Electronic Imaging 2006, Security, Steganography, and Watermarking of Multimedia Contents VIII*, (San Jose, USA), January 15–19 2006.
7. S. Gel'fand and M. Pinsker, "Coding for channel with random parameters," *Problems of Control and Information Theory* **9**(1), pp. 19–31, 1980.
8. T. Cover and J. Thomas, *Elements of Information Theory*, Wiley and Sons, New York, 1991.
9. M. Costa, "Writing on dirty paper," *IEEE Trans. on Information Theory* **29**(3), pp. 439–441, 1983.
10. B. Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. on Information Theory* **47**, pp. 1423–1443, 2001.
11. J. Eggers, J. Su, and B. Girod, "A blind watermarking scheme based on structured codebooks," in *Secure Images and Image Authentication, IEE Colloquium*, pp. 4/1–4/6, (London, UK), April 2000.
12. F. Deguillaume, Y. Rytsar, S. Voloshynovskiy, and T. Pun, "Data-hiding based text document security and automatic processing," in *IEEE International Conference on Multimedia & Expo (ICME) 2005*, (Amsterdam, The Netherlands), July 6–8 2005.
13. F. Deguillaume, S. Voloshynovskiy, and T. Pun, "Character and vector graphics watermark for structured electronic documents security." US Patent Application 10/949,318 (pending), September 27 2004.
14. M. Barni and F. Bartolini, *Watermarking Systems Engineering*, Marcel Dekker, Inc., New York, 2004.
15. A. Nagy, "Twenty Years of Document Image Analysis in PAMI," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, pp. 38–62, January 2000.
16. K. Matsui and K. Tanaka, "Video-steganography: How to secretly embed a signature in a picture," in *Proceedings of IMA Intellectual Property Project*, **1**(1), 1994.
17. R. Villán, S. Voloshynovskiy, O. Koval, and T. Pun, "Multilevel 2D Bar Codes: Towards High Capacity Storage Modules for Multimedia Security and Management," in *Proceedings of SPIE-IS&T Electronic Imaging 2005, Security, Steganography, and Watermarking of Multimedia Contents VII*, **5681**, pp. 453–464, (San Jose, USA), January 16–20 2005.
18. S. M. Smith and J. M. Brady, "SUSAN—A New Approach to Low Level Image Processing," *Int. J. Comput. Vision* **23**(1), pp. 45–78, 1997.