A Multigrid Approach to the Scalar Quantization Problem

Yair Koren, Irad Yavneh, and Alon Spira

***Abstract*—A multigrid framework for the one–dimensional (scalar) fixed-rate quantization problem is presented. The framework is based on the Lloyd–Max iterative process, which is a central building block in many quantization algorithms. This process iteratively improves a given initial solution and generally converges to a local minimum of the quantization distortion. Contrary to the classical Lloyd–Max process, the convergence of the multigrid algorithm is practically independent of the number of representation levels sought. Using this approach, a local minimum is reached at the cost of just a few Lloyd–Max iterations. The complexity of the proposed method is $O(n)$ operations for the continuous case and $3N + O(n)$ for the discrete case, where n is the number of representation levels sought and N is the size of the discrete probability density function. In addition to its independent attributes, this work is a precursor to the more important vector quantization problem, for which a multiscale framework is also being developed.**

***Index Terms*—Full approximation scheme (FAS), Linde–Buzo–Gray (LBG), Lloyd–Max, multigrid, quantization.**

I. Introduction

Quantization [1], [2] is the process of representing a continuum with only a finite number of representatives or representing an initially rich amount of discrete data with fewer representatives. Rounding off real numbers to the nearest integer is a simple form of quantization.

More formally, given a random process whose probability density function $p : [L, R] \to \mathbb{R}$, or an approximation thereof, is known and a positive integer n, a quantizer $q : [L, R] \to \mathbb{R}$ is defined by n representation levels $\{r_i\}_{i=1}^n$ and n decision regions $\{D_i\}_{i=1}^n$. $\bigcup_{i=1}^n D_i = [L, R]$ and $D_i \bigcap D_j = \phi$ for all $i \neq j$. $q(x)$ is a piecewise-constant approximation of $[L, R]$ where all $x \in D_i$ are represented by r_i. An optimal quantizer minimizes some predefined distortion measure. We confine ourselves to the mean square error (MSE) distortion measure defined as the expectation of the squared quantization error. In this case, the decision regions $\{D_i\}_1^n$ are simply sections in $[L, R]$, henceforth denoted by $\{(d_{i-1}, d_i)\}_{i=1}^n$. The MSE distortion is formally defined by

$$\begin{aligned}\mathcal{D}(q) &= E[|X - q(X)|^2] \\ &= \sum_{i=1}^{n} \int_{d_{i-1}}^{d_i} |x - r_i|^2 \, p(x) \, dx. \end{aligned} \tag{1}$$

In the discrete case, where $p(x)$ is nonzero at only a finite number N of points $\{x_i\}_{i=1}^N$, a similar functional is minimized

$$\begin{aligned}\mathcal{D}(q) &= E[|X - q(X)|^2] \\ &= \sum_{i=1}^{n} \sum_{x_j \in (d_{i-1}, d_i)} |x_j - r_i|^2 p(x_j). \end{aligned} \tag{2}$$

In (2), it is assumed that no x_j coincides with any d_i. If this is not the case, then any such x_j, lying on the boundary between two decision regions, should be represented by either r_i or r_{i+1} and an appropriate term should be added to the distortion (1), (2). In practice, when solving the discrete problem (2), it is usually assumed that $r_i \in \{x_j\}_{j=1}^N$ for all i's. Certain error measures, such as MSE, as well as other monotonic measures, allow polynomial time globally optimal solvers for (2) with discrete r_i's [3]–[5]. These solvers use a dynamic programming scheme that greatly reduces the complexity of an exhaustive search, while maintaining global optimality. The complexity of the fastest algorithm [5] for globally minimizing (2) with discrete r_i's and the MSE error measure is $O(nN)$. This algorithm exploits quantizer monotonicities along with an elegant matrix search algorithm [6]. In the sequel, we present an algorithm that tries to minimize (1) in $O(n)$ and (2) in $3N + O(n)$ operations. Our algorithm ensures only local optimality but permits continuous $p(x)$. Moreover, for large enough n, educated initialization is possible [7], [8], which greatly increases the chances that our method will find the global minimum.

Henceforth, we treat the continuous problem (1) where $p(x)$ is continuous. For simplicity, we permit $p(x)$ to be any nonnegative integrable function, not necessarily a probability density function. The results throughout this correspondence also apply to the discrete case (2). Necessary conditions for a minimum are

$$\begin{aligned} \frac{\partial}{\partial r_i}\mathcal{D} &= 0, \qquad 1 \le i \le n \\ \frac{\partial}{\partial d_i}\mathcal{D} &= 0, \qquad 1 \le i \le n-1. \end{aligned}$$

These conditions yield

$$r_i = \frac{\int_{d_{i-1}}^{d_i} x p(x) dx}{\int_{d_{i-1}}^{d_i} p(x) dx}, \qquad d_i = \frac{r_i + r_{i+1}}{2}. \tag{3}$$

Lloyd [9] and Max [10] came up (separately) with the above equations and proposed "ping-ponging" back and forth between them

$$d_i^k = \frac{r_i^{k-1} + r_{i+1}^{k-1}}{2}, \qquad k = 1, 2, 3, \ldots \tag{4}$$

$$r_i^k = \frac{\int_{d_{i-1}^k}^{d_i^k} x p(x) dx}{\int_{d_{i-1}^k}^{d_i^k} p(x) dx}, \qquad k = 1, 2, 3, \ldots \tag{5}$$

that is, optimizing the d_i's according to the r_i's, followed by optimizing the r_i's according to the new d_i's, and so on. r^0 is an initial guess at the optimal representation levels. This process converges monotonically to a solution that satisfies (3) (see [9]). Fleischer showed that a sufficient condition for a globally minimal distortion is that $\log(p(x))$ be concave [11]. In general, a minimum achieved by the above descent process is not necessarily global.

Since the Lloyd–Max iterative process requires an initial guess at the solution in order to start iterating, many heuristic methods that try to find a good initial approximation have been proposed [7], [8], [12], [13]. After acquiring such an initial approximation, Lloyd–Max iterations may be applied in order to bring it to a local minimum. As an example, the Linde–Buzo–Gray (LBG) [13] algorithm is a grid refinement process that starts by finding a solution (using the generalized Lloyd algorithm) for two r_i's and then splitting each r_i into two new r_i's. This process is repeated recursively until the number of $r_i's$ equals n. The generalized Lloyd algorithm is the above iterative process (4),

Manuscript received June 19, 2003; revised March 13, 2005. This work was supported by the Israel Science Foundation under Grant 48/02, and in part by the Israeli Ministry of Science and Technology.

The authors are with the Department of Computer Science, Technion–Israel Institute of Technology, Technion City, Haifa 32000, Israel (e-mail: yair_k@cs.technion.ac.il; irad@cs.technion.ac.il; salon@cs.technion.ac.il).

Communicated by R. Zamir, Associate Editor for Source Coding.
Digital Object Identifier 10.1109/TIT.2005.851758

(5) with the improvement in distortion used as a stopping criterion, that is, if the improvement made at a certain iteration (relative to the previous one) does not exceed some predefined threshold, the process is stopped.

The multigrid framework [14]–[16] is an efficient iterative method mostly used to numerically solve elliptic partial differential equations. Such equations can be solved iteratively by applying relaxation schemes such as Jacobi or Gauss–Seidel. These schemes are local, due to the local nature of elliptic differential equations. As such, relaxation schemes can eliminate only local (high-frequency) errors. Large-scale (low-frequency) errors cannot be detected locally and are not, therefore, reduced efficiently. The multigrid framework recursively applies local relaxations to decimated versions of the data, exploiting the fact that low frequencies on a fine grid transform to relatively higher frequencies on coarser grids. The idea is that all errors are local on some scale, and they can be reduced efficiently by a local process only on that scale.

The Lloyd–Max process, although not applied to a discretized differential equation, is a local (nonlinear) process, somewhat similar to damped Jacobi relaxation. We show that the multigrid framework can accelerate this process, yielding convergence rates which appear independent of the number of representation levels. These results are similar to those of applying multigrid methods to elliptic differential equations, for which they are considered optimal.

To a large extent, our motivation for studying the scalar problem is to develop insights and tools for dealing with the vector quantization problem that is considered more important and is considerably more difficult. Indeed, based on this research, we have developed algorithms for the vector quantization problem that yield superior results in practice when compared to LBG [17]. In addition, the present method can be used in conjunction with certain vector quantization algorithms that use scalar quantization as a subprocedure, e.g., [18].

Since Lloyd–Max is a gradient descent process, we believe that our method may work with other distortion measures too, as long as they are differentiable.

The correspondence is organized as follows. Section II provides a detailed description of the multigrid full approximation scheme (FAS) algorithm and its application to the Lloyd–Max process. Section III states the computational complexity of the FAS algorithm. Section IV states the convergence rates of the FAS algorithm for constant $p(x)$. Section V compares between the classical Lloyd–Max, LBG, and multigrid methods. Section VI is a conclusion and proposition of future work.

II. The Multigrid (FAS) Algorithm

The key to speeding up the Lloyd–Max iterative process is the multigrid FAS, introduced in 1977 by Brandt [14]. The method is almost exclusively used for solving (usually nonlinear) partial differential equations [15], [16].

Iterative methods for solving boundary-value problems often consist of discretizing the problem on a uniform grid and approximating the derivatives using some finite difference (or volume or element) scheme. Once this is done, an initial solution can be improved by iterative relaxations. The problem is that derivatives are local. That is, at each iteration, the approximate solution at each point is directly affected by its local neighbors alone. In contrast, the solution at any given point generally depends on the solution at all other points in the domain, many of which are distant. Propagation of information from the boundaries inwards, for example, may require many iterations. Multigrid methods overcome this problem.

We first describe the FAS algorithm informally, and later proceed with the formal definition. Instead of exhaustively applying Lloyd–Max iterations (4) and (5), only a small number (ν_1) of such relaxations is performed. The resulting approximate solution is then coarsened by restriction (sampling) to half the resolution. This "descent" process consisting of relaxation/restriction is performed recursively until only a single variable remains, at which point an exact solution is easily calculated. Using interpolation, this coarse solution is used to generate a correction term that is added to the next-level (finer) approximate solution. Next, at the finer resolution, a small number (ν_2) of additional Lloyd–Max relaxations are performed. This "ascent" process consisting of interpolation/relaxation is repeated recursively until returning to the initial finest resolution with a new approximate solution. The whole process is called a "V-cycle." "V" represents the coarsening of the approximate solution until reaching the coarsest resolution (descent), followed by refining (ascent) until returning to the original finest resolution with a new approximate solution.

For a more formal description of the algorithm, we adopt the following conventions. We combine the expressions for both the representation levels and the decision regions (3) into a single expression for the decision regions

$$d_i = \frac{1}{2}\left(\frac{\int_{d_{i-1}}^{d_i} xp(x)dx}{\int_{d_{i-1}}^{d_i} p(x)dx} + \frac{\int_{d_i}^{d_{i+1}} xp(x)dx}{\int_{d_i}^{d_{i+1}} p(x)dx} \right) \tag{6}$$

for all $1 \le i \le n-1$. We define the operator L by

$$L(d)_i \equiv d_i - \frac{1}{2}\left(\frac{\int_{d_{i-1}}^{d_i} xp(x)dx}{\int_{d_{i-1}}^{d_i} p(x)dx} + \frac{\int_{d_i}^{d_{i+1}} xp(x)dx}{\int_{d_i}^{d_{i+1}} p(x)dx} \right) \tag{7}$$

for all $1 \le i \le n-1$. The problem is now defined by

$$L(d) = f \tag{8}$$

where d is the vector of inner decision-region boundaries, $[d_1, d_2, \ldots, d_{n-1}]^T$ and f, the right-hand side is defined as follows. At the finest resolution f is a zero vector of size $n-1$ whereas at the coarser resolutions f is defined in Step 2b of the V-cycle algorithm below. The role of f is to adapt the coarse-level equations to represent the original fine level problem, e.g., [16, Sec. 5.3.7]. Without a modified f, each level approximates a different problem, namely, the quantization problem corresponding to that level. In fact, a modified f on the coarse levels is the key difference between multigrid and LBG. $\tilde{d}$ denotes the current approximation (the tilde will be neglected when no confusion may arise). The residual is defined by

$$r(\tilde{d}) = f - L(\tilde{d}). \tag{9}$$

Upper indices will henceforth denote the number of decision regions, n, which will be omitted when no confusion may arise. For simplicity, we will assume n to be a power of two. $I_{n/2}^n$ will represent the interpolation operator that is applied to the correction term obtained from the coarser, $n/2$-region problem. $I_n^{n/2}$ will represent the restriction operator for the residual $r(\tilde{d}^n)$, and $\tilde{I}_n^{n/2}$ the restriction operator for the approximate solution $\tilde{d}^n$ (from resolution n to resolution $n/2$).

Each V-cycle is considered one iteration. Like the Lloyd–Max iteration, the V-cycle's input is an initial guess and its output is a new approximation. A V-cycle is recursively defined as follows.

$\underline{\tilde{d}^n \leftarrow V^n(\tilde{d}^n, f^n)}$

1) Relax $L(\tilde{d}^n) = f^n$, ν_1 times.
2) If $n = 2$
   a) return $\tilde{d}^n$
   Else
   a) Calculate the residual: $r(\tilde{d}^n) = f^n - L(\tilde{d}^n)$
   b) Set $\tilde{d}^{n/2} = \tilde{I}_n^{n/2}\tilde{d}^n$ and $f^{n/2} = I_n^{n/2} r(\tilde{d}^n) + L(\tilde{d}^{n/2})$.
   c) Recursively call $\tilde{d}^{n/2} \leftarrow V^{n/2}(\tilde{d}^{n/2}, f^{n/2})$.
3) Interpolate and add the correction:

$$\tilde{d}^n \leftarrow \tilde{d}^n + I_{n/2}^n(\tilde{d}^{n/2} - \tilde{I}_n^{n/2}\tilde{d}^n).$$

4) Relax $L(\tilde{d}^n) = f^n$, ν_2 times.

At the coarsest resolution, $n = 2$, where $\tilde{d}^n$ consists of a single coordinate, the ν_1 relaxations can be replaced by an exact calculation of the solution. Henceforth, we denote the V-cycle with ν_1 pre-relaxations and ν_2 post-relaxations V_{ν_1,ν_2}. We generally use $V_{1,1}$ cycles which will be assumed from now on.

To further clarify the role of f, observe that the $n/2$ resolution equation is (Step 2b)

$$L(\tilde{d}^{n/2}) = I_n^{n/2}(f^n - L(\tilde{d}^n)) + L\left(\tilde{I}_n^{n/2}(\tilde{d}^n)\right). \quad (10)$$

Upon convergence, since the residual $f^n - L(d^n)$ is then zero, we obtain

$$L(d^{n/2}) = L\left(\tilde{I}_n^{n/2}(d^n)\right).$$

Hence, as the process converges, the coarse solution $d^{n/2}$ tends to the fine solution d^n restricted to the coarse resolution (e.g., sampled at every other point).

Note that the probability density function $p(x)$ remains at the finest resolution (continuous) throughout the algorithm. The operator L at different resolutions is defined by (7) and differs only due to the different number of coordinates of the approximate solution $\tilde{d}$ at the various resolutions.

We have yet to define the relaxation method and interpolation/restriction operations. We propose the Lloyd–Max process as the relaxation. The Lloyd–Max equations (4) and (5) with only the decision regions as variables are

$$d_i^{\text{new}} = f + \frac{1}{2}\left(\frac{\int_{d_{i-1}^{\text{old}}}^{d_i^{\text{old}}} xp(x)dx}{\int_{d_{i-1}^{\text{old}}}^{d_i^{\text{old}}} p(x)dx} + \frac{\int_{d_i^{\text{old}}}^{d_{i+1}^{\text{old}}} xp(x)dx}{\int_{d_i^{\text{old}}}^{d_{i+1}^{\text{old}}} p(x)dx} \right) \quad (11)$$

with $1 \le i \le n-1$. We actually prefer an over-relaxed version of this method, given by

$$d_i^{\text{over}} = (1-\omega)d_i^{\text{old}} + \omega d_i^{\text{new}} = d_i^{\text{old}} + \omega r\left(d_i^{\text{old}}\right) \quad (12)$$

where d_i^{new} is given by (11) and ω is a constant relaxation parameter. We use $\omega = 4/3$, suggested by our analysis in [19].

The restriction scheme $\tilde{I}_n^{n/2}$ for the current approximation $\tilde{d}^n$ is injection. That is, when transferring the approximate solution to a coarser resolution we simply sample $\tilde{d}^n$ at the even indices

$$\left[\tilde{I}_n^{n/2}\tilde{d}^n\right]_i = \tilde{d}_{2i}^n, \qquad i = 1, \ldots, n/2 - 1. \quad (13)$$

The restriction scheme for the residual $r(\tilde{d}^n)$, $I_n^{n/2}$, is the so-called full weighting operator, scaled by a factor 4, yielding

$$\left[I_n^{n/2} r(\tilde{d}^n)\right]_i = r(\tilde{d}^n)_{2i-1} + 2r(\tilde{d}^n)_{2i} + r(\tilde{d}^n)_{2i+1}, \qquad i = 1, \ldots, n/2 - 1. \quad (14)$$

The factor 4 is a necessary scaling which is explained in [19].

Let $c^{n/2} = \tilde{d}^{n/2} - \tilde{I}_n^{n/2}\tilde{d}^n$ denote the coarse-resolution correction that is interpolated and added to $\tilde{d}^n$ in Step 3 of the algorithm. Using this notation, our interpolation scheme is defined by

$$\left[I_{n/2}^n(c^{n/2})\right]_i = \begin{cases} c_{i/2}^{n/2}, & \text{for even } i \\ \frac{\tilde{d}_{i+1}^n - \tilde{d}_i^n}{\tilde{d}_{i+1}^n - \tilde{d}_{i-1}^n} c_{(i-1)/2}^{n/2} + \frac{\tilde{d}_i^n - \tilde{d}_{i-1}^n}{\tilde{d}_{i+1}^n - \tilde{d}_{i-1}^n} c_{(i+1)/2}^{n/2}, & \text{for odd } i. \end{cases} \quad (15)$$

Here, $\tilde{d}^n$ is the fine-resolution approximation before the coarse-resolution correction is added. We use this nonlinear interpolation to maintain the strict order of $\tilde{d}^n$, i.e., $\tilde{d}_i^n > \tilde{d}_{i-1}^n$, $i = 1, \ldots, n$. This property is not ensured by the usual linear interpolation, for example. The interpolation scheme (15) preserves the order of $\tilde{d}^n$, provided that $\tilde{d}^{n/2}$ is itself properly ordered. However, the ordering might still be lost during the relaxation. While the usual Lloyd–Max process always preserves ordering, our Lloyd–Max relaxations (11), at all but the finest resolution, include a modified nonzero right-hand side f, which may cause disordering. Therefore, after each relaxation, the ordering is tested and, if compromised, the properly ordered $\tilde{d}^n$ from before the relaxation is used.

Even when ordering is preserved, convergence is not assured. In [19], a slight variation of our algorithm, that ensures monotonic convergence, is described.

III. Complexity of FAS

A complete and exact analysis of the computational complexity of the $V_{1,1}$ cycle is documented in [19]. Here we only state the final results for both the discrete (2) and continuous (1) cases.

1) *Discrete Quantization*: The complexity of a single $V_{1,1}$ cycle in the discrete case is $7n$ divisions, $13n$ multiplications, and $40n$ additions/subtractions. The pre-computation requires a one time effort of $2N$ multiplications and N additions.
2) *Continuous Quantization*: The complexity of a single $V_{1,1}$ cycle in the continuous case is $12n$ integral evaluations, $7n$ divisions, $13n$ multiplications, and $26n$ additions.

Integral evaluations are the bulk of the complexity in the continuous case. The Lloyd–Max process requires $2n$ integral evaluations per iteration. Therefore, the complexity of a $V_{1,1}$ cycle in the continuous case is roughly six times the complexity of a Lloyd–Max iteration.

IV. Convergence Analysis

An exact quantitative convergence analysis of the multigrid algorithm for the nonlinear problem is beyond reach. In general, the theory of multigrid solvers for nonlinear problems is quite limited. We therefore analyze the case where $p(x)$ is constant, for which the problem is linear and lends itself to a Fourier analysis. First introduced into multigrid methods in [14], this approach has become the standard tool for accurate prediction of the convergence behavior of multigrid algorithms [16]. Here, we only state the final results. The complete derivation appears in [19]. Much experience with multilevel algorithms, supported

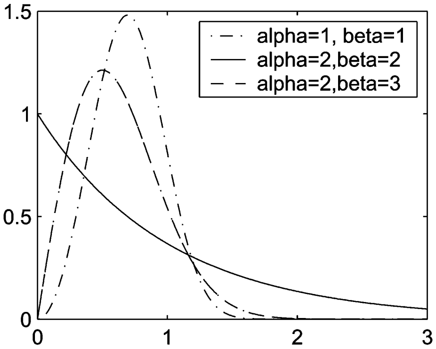

Fig. 1. Weibull density functions $p(x) = \alpha\beta x^{\beta-1}e^{-\alpha x^{\beta}}$, $x > 0, \alpha, \beta > 0$. Solid: $\alpha = 1, \beta = 1$. Dashed: $\alpha = 2, \beta = 2$. Dash-dot: $\alpha = 2, \beta = 3$.

by numerical results below and in [19] indicates that the analysis is relevant also for the nonlinear problem, so long as $p(x)$ undergoes slow relative variations.

When $p(x) \equiv 1$ the problem is linear. That is, both the Lloyd–Max and FAS algorithms have the form

$$d^{\text{new}} = Md^{\text{old}}$$

where M is some matrix and d^{old} and d^{new} are the previous and next approximations to the solution, respectively. In this case, the convergence rate is defined as the spectral radius of the matrix M, i.e., the absolute value of the eigenvalue with largest magnitude.

A. Results of Analysis for the $p(x) \equiv 1$ Case

1) The asymptotic convergence rate of the Lloyd–Max process, is

$$\cos^2\left(\frac{\pi}{2n}\right) = 1 - \left(\frac{\pi}{2n}\right)^2 + O(n^{-4})$$

implying that, in general, $O(n^2)$ iterations are asymptotically required to reduce the error by an order of magnitude.

2) For $\nu_1 = \nu_2 = 1$, which we use in our tests, the convergence rate of the FAS algorithm is $1/4$ for any even n. Thus, the asymptotic convergence factor per multigrid cycle employing a single pre-relaxation and a single post-relaxation Lloyd–Max iteration is expected to be approximately $1/4$ independently of n.

3) The FAS performance can further be improved by introducing a relaxation parameter (12). The optimal relaxation parameter is $\omega = 4/3$, yielding an asymptotic convergence factor of $1/9$. Our numerical tests show that $\omega = 4/3$ remains nearly optimal for nonconstant $p(x)$ as well, and we use this value throughout.

V. Numerical Results

A. Convergence Rates

We first demonstrate the application of multigrid $V_{1,1}$ cycles to the class of Weibull density functions (see Fig. 1)

$$p(x) = \alpha\beta x^{\beta-1}e^{-\alpha x^{\beta}}, \qquad x > 0,\ \ \alpha, \beta > 0 \tag{16}$$

and compare the results with those of the Lloyd–Max and LBG methods [13]. This is precisely the set of problems used in [7]. Various values of α and β yield various commonly encountered density functions. Note that $\beta \geq 1$ yields a single minimum since $\log(p(x))$ is concave [11]. As in [7], we truncate the tails of these infinite densities at the point x_0 for which $\int_{x_0}^{\infty} p(x) < 10^{-4}$. The examples we present are of continuous $p(x)$'s. Application of our method to the discrete versions of these $p(x)$'s yielded similar results, especially when N, the size of the discrete probability density function, was large.

LBG and Lloyd-Max are identical once LBG reaches the highest resolution. We consider the LBG algorithm as a method of acquiring an initial guess at the solution. Therefore, all LBG plots throughout disregard the lower resolutions and display the progress starting from the highest resolution where Lloyd-Max iterations are performed until convergence. Consider the example where $\alpha = 2$, $\beta = 3$, $\Omega = [0, 1.67]$, and assume that 32 representation levels are sought. The initial $\tilde{d}_i$ values for Lloyd-Max and multigrid are placed uniformly over $[0, 1.67]$.

Referring to Fig. 2, the upper left-hand plot displays the converged solution, using any of the three methods, over the distribution function $p(x) = 6x^2e^{-2x^3}$. Solid vertical lines describe the representation levels (r_i's) and dotted vertical lines the boundaries of the decision regions (d_i's). The lower right-hand plot displays the error $|\mathcal{D}^k - \mathcal{D}^{\text{opt}}|$, i.e., the distance between the distortion at iteration k and the optimal distortion. The upper right-hand graph displays the L_2 norm of the residual (9). Note the similar behavior between the residual and the error in distortion. Drawing the distortion graph requires the knowledge of the optimal distortion (that in this case was precomputed). As the optimal distortion is not usually known, the behavior of the residual is often used as an approximation to the behavior of the distortion. Applying the multigrid method causes both the residual and the error to drop by about an order of magnitude at each V-cycle, as predicted by the Fourier analysis for constant $p(x)$ [19]. Applying the Lloyd–Max method, the residual and error drop by a single order of magnitude approximately every 800 iterations. As can be seen, the LBG method does not improve the convergence rate of either the residual or the error over the Lloyd–Max method, it only provides a better initial guess (for this example) with respect to the distortion value. The process was stopped when the residual reached 10^{-12}. While a practical solution is reached long before this point, performing many iterations demonstrates how Lloyd–Max, LBG, and FAS are all processes that converge linearly. The difference, as seen in the right-hand plots of Fig. 2, is that multigrid converges dramatically faster. The lower left-hand graph displays the residual reduction factor against the number of iterations

$$\frac{\|r(\tilde{d}^n)^{k+1}\|_2}{\|r(\tilde{d}^n)^k\|_2} \tag{17}$$

where k is the iteration number and the subscript 2 refers to the L_2 norm.

Indeed, we see that the multigrid method reduces the residual by about an order of magnitude per V-cycle. Both the LBG and the Lloyd–Max residual reduction factors rise quickly and stabilize on about 0.997. The initial lower factor is a result of high-frequency error components that are efficiently reduced by the local Lloyd–Max process [19].

We have stated in Section IV, and proven in [19], that for constant $p(x)$, the number of Lloyd–Max iterations needed to lower the residual by an order of magnitude is proportional to the square of the number of representation levels. Table I shows that the behavior is similar for the nonlinear case $p(x) = 6x^2e^{-2x^3}$. Referring to Table I, the first column indicates the number of representation levels used, n, and the other columns display the average residual reduction rate per iteration for both Lloyd–Max and multigrid.

For Lloyd–Max, Table I demonstrates how doubling the representation levels slows the convergence down by approximately a factor of four, as predicted by the analysis (for constant $p(x)$). Other p's display similar behavior to that described by Table I as long as the relative change in $p(x)$ is locally small.

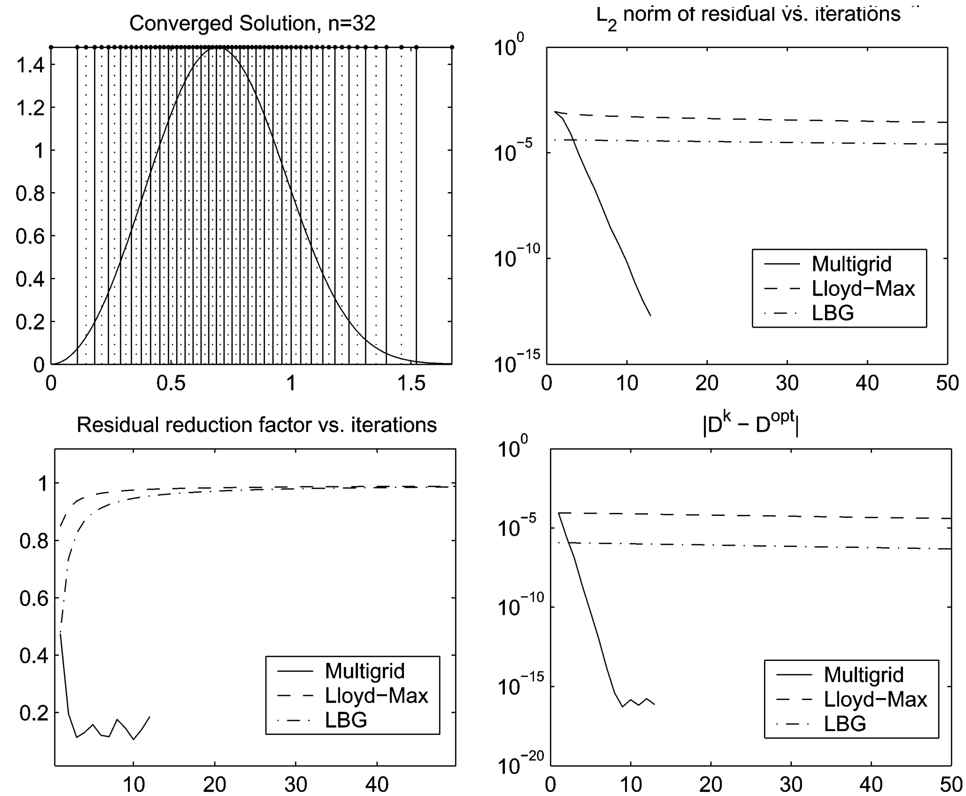

Fig. 2. Weibull density $p(x)$ with $\alpha = 2, \beta = 3$, and $n = 32$. Uniform quantization used as initial guess at the solution.

TABLE I
AVERAGE RESIDUAL REDUCTION RATE PER ITERATION FOR $p(x) = 6x^2e^{-2x^3}$ AND n REPRESENTATION LEVELS

n	Lloyd-Max	multigrid
16	0.9900	0.1725
32	0.9973	0.1782
64	0.9993	0.1847
128	0.9998	0.1962

VI. CONCLUSION AND GENERALIZATIONS

We have presented a multigrid acceleration of the Lloyd–Max iterative process for scalar quantization. The accelerated algorithm is applicable to both the continuous (1) and discrete (2) versions of the problem and normally requires $O(n)$ and $3N + O(n)$ operations, respectively. Multigrid convergence rates seem independent of the number of representation levels, whereas the convergence rates for the Lloyd–Max and LBG methods are quadratically dependent on this factor.

The Lloyd–Max method is applicable to the multidimensional vector quantization (VQ) problem. Based on our research of scalar quantization in conjunction with multigrid methods, we have developed efficient multiscale vector quantization algorithms [17]. Our VQ algorithms consist of grouping the representation levels and decision regions into aggregates. The finer scales are comprised of many aggregates each consisting of a few representation levels, whereas the coarser scales are comprised of fewer aggregates, each consisting of many representation levels. Using criteria based on the theoretical results of Gersho [20], we determine the appropriate number of representation levels per aggregate and redistribute the representation levels accordingly. The redistribution is performed at all scales. Our algorithms are based on insights from conventional multigrid methods such as the one described in this correspondence for scalar quantization. In both the scalar and vector quantization cases, the coarse scales reveal imbalances that were unnoticeable (and therefore irreparable) at the finer scales. This leads to faster solvers and often to better solutions in problems that are difficult for conventional methods.

REFERENCES

[1] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1992.
[2] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 1–63, Oct. 1998.
[3] J. D. Bruce, "Optimal quantization," Sc.D. dissertation, MIT, Cambridge, MA, 1964.
[4] D. K. Sharma, "Design of absolutely optimal quantizers for a wide class of distortion measures," *IEEE Trans. Inf. Theory*, vol. IT-24, no. 6, pp. 693–702, Nov. 1978.
[5] X. Wu and K. Zhang, "Quantizer monotonicities and globally optimal scalar quantizer design," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 1049–1053, May 1993.
[6] A. Aggarwal, M. Klawe, S. Moran, P. Shor, and R. Wilber, "Geometric applications of a matrix-searching algorithm," *Algorithmica*, vol. 2, pp. 195–208, 1987.
[7] X. Wu, "On initialization of Max's algorithm for optimum quantization," *IEEE Trans. Commun.*, vol. 38, no. 10, pp. 1653–1656, Oct. 1990.
[8] G. M. Roe, "Quantizing for minimum distortion," *IEEE Trans. Inf. Theory*, vol. IT-10, no. 4, pp. 384–385, Oct. 1964.
[9] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.
[10] J. Max, "Quantizing for minimal distortion," *IEEE Trans. Inf. Theory*, vol. IT-6, no. 1, pp. 7–12, Mar. 1960.
[11] P. Fleischer, "Sufficient conditions for achieving minimum distortion in a quantizer," in *IEEE Int. Conv. Rec.*, 1964, pp. 104–111.
[12] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Mathematics Statistics and Probability*, vol. 1, Berkeley, CA, 1967, pp. 281–296.
[13] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, no. 1, pp. 84–95, Jan. 1980.

[14] A. Brandt, "Multi-level adaptive solutions to boundary-value problems," *Math. Comput.*, vol. 31, pp. 333–390, 1977.
[15] W. L. Briggs, V. Henson, and S. F. McCormick, *A Multigrid Tutorial*, 2nd ed. Philadelphia, PA: SIAM, 2000.
[16] U. Trottenberg, C. Oosterlee, and A. Schüller, *Multigrid*. San Diego, CA: Academic, 2001.
[17] Y. Koren and I. Yavneh, "Adaptive multiscale redistribution for vector quantization," *SIAM J. Sci. Comput.*, to be published.
[18] R. Balasubramanian, C. Bouman, and J. P. Allebach, "Sequential scalar quantization of vectors: An analysis," *IEEE Trans. Image Process.*, vol. 4, no. 9, pp. 1282–1295, Sep. 1995.
[19] Y. Koren, I. Yavneh, and A. Spira. (2005) A Multigrid Approach to the 1-D Quantization Problem. [Online]. Available: http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-info.cgi?2005/CS/CS-2005-08
[20] A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. Inf. Theory*, vol. IT-25, no. 4, pp. 373–380, Jul. 1979.

On Joint Information Embedding and Lossy Compression

Alina Maor and Neri Merhav, *Fellow, IEEE*

***Abstract*—We consider the problem of optimum joint information embedding and lossy compression with respect to a fidelity criterion. The goal is to find the minimum achievable compression (composite) rate R_c as a function of the embedding rate R_e and the average distortion level Δ allowed, such that the average probability of error in decoding of the embedded message can be made arbitrarily small for sufficiently large block length. We characterize the minimum achievable composite rate for both the public and the private versions of the problem and demonstrate how this minimum can be approached in principle. We also provide an alternative single-letter expression of the maximum achievable embedding rate (embedding capacity) as a function of R_c and Δ, above which there exist no reliable embedding schemes.**

***Index Terms*—Entropy coding, information embedding, information hiding, lossless/lossy compression, rate-distortion theory, watermarking.**

I. Introduction

In the last few years, along with increasing awareness regarding the data protection, there is observed an increased interest in watermarking codes in their various applications. Watermarking is a form of information hiding in a host data set (covertext), usually an image, audio signal, or video, creating a distorted version of the host data (stegotext, composite data). Successful retrieving of the watermark from the examined data indicates ownership, while on the other hand, damaging of the watermark beyond retrieving or its fabrication allows stealing the data or its forgery. There exists a variety of applications for data hiding ranging from classical steganography [1], [2], data authentication, copyright protection, and copy control information [3], [4].

The requirements in watermarking scheme design are quite conflicting: In most applications, the watermark should be *perceptually transparent*, that is, invisible to the naked eye, or, when audio signals are concerned, inaudible to the innocent listener. Watermarks must be *robust* to the distortion of watermarked data, caused by either conventional data processing (e.g., lossy compression, up/down-scaling, filtering, halftoning) or malicious attacks of the parties who wish to invalidate the watermark. As the reconstruction of the watermark is usually performed on a distorted version of stegotext (forgery), various distortion criteria measure the robustness of the embedding, while others ensure its initial transparency.

While most existing practical watermarking applications were designed and tested empirically (see, e.g., [1]–[5]), the information-theoretic research activity in the problem area of watermarking is relatively new, evolving primarily around issues of system modeling, performance criteria, watermarking code design, and theoretical performance bounds. As covertext serves only as a "carrier" of the watermark, it may be considered as side information, as was first proposed in [9]. Thus, from an information-theoretic viewpoint, the watermarking problem is usually regarded as an instance of the problem of channel coding with side information [6]–[8]. The case where the side information is available to the encoder only is named public watermarking, and the case where it is available to the decoder as well is named private watermarking. There exist various models for the watermarking schemes, where two main flows are classified: One can either be interested in complete reconstruction of the watermark or only in identification of its existence [10]. The problem of watermark reconstruction from forgery is often treated as a game between the information hider and the attacker [11]–[14]. In [15], the duality between information embedding problem and the problem of source coding with side information at the decoder is explored, evaluating tradeoffs between the achievable embedding capacity and the allowed covertext distortion.

Another aspect of the watermarking problem is that of joint information embedding and lossy compression, where quantization as well as entropy coding of the stegotext is treated as an integral part of the watermarking scheme. The general problem is given as follows: There is a set of messages to be embedded in the covertext, subject to some distortion constraint. The composite sequence resulting from this embedding is compressed losslessly and the embedded message must be decoded reliably with or without access to the original host data. It should be pointed out that the joint embedding–compression process is lossy, since the perfect covertext reconstruction from the stegotext is not possible, and thus, the stegotext generation *must* take into account both the embedding and the compressibility requirements of the system.

A scheme for successive embedding and compression based on nested lattice coding is proposed in [16] for both Gaussian and independent and identically distributed (i.i.d.) arbitrary memoryless sources. It achieves desired embedding rates and the rate-distortion stegotext compression for some distortion constraint imposed on a system. In [16], the stegotext compression is tuned to the embedding procedure, though, in general, the embedder and the compressor are not cooperative.

Karakos and Papamarcou [17], [18], Karakos [19], and Willems and Kalker [20] study the tradeoffs between the distortion, the embedding rate, and the composite rate for lossless stegotext compression. In [17], the attack-free version of public and private watermarking (fingerprinting) problem is treated, considering the case of a zero-mean Gaussian white noise process: two watermarks are successively hidden in the covertext image, subject to a mean square distortion constraint, resulting in a Gaussian composite sequence, which is then compressed. The achievable rate region is established in terms of the relations between the composite rate, the watermark embedding rates, and the prescribed distortion constraint. In [18], a system similar to [17], which embeds a single watermark in Gaussian covertext and

Manuscript received July 16, 2003; revised March 27, 2005. This correspondence is part of A. Maor's M.Sc. thesis. The material in this correspondence was presented in part at the IEEE International Symposium on Information Theory, Chicago, IL, Jun./Jul. 2004.

The authors are with the Department of Electrical Engineering, Technion–Israel Institute of Technology, Technion City, Haifa 32000, Israel (e-mail: alinam@tx.technion.ac.il; merhav@ee.technion.ac.il).

Communicated by R. W. Yeung, Associate Editor for Shannon Theory.
Digital Object Identifier 10.1109/TIT.2005.851767