Adaptive Multiscale Redistribution for Vector Quantization

Yair Koren and Irad Yavneh
Department of Computer Science
Technion—Israel Institute of Technology
Haifa 32000, Israel
January 2004

***Abstract*— Vector quantization is a classical problem which appears in many fields. Unfortunately, the quantization problem is generally non-convex, and therefore affords many local minima. The main problem is finding an initial approximation, which is close to a "good" local minimum. Once such an approximation is found, several methods exist for reaching the local minimum near it. In recent years, much improvement has been made with respect to reducing computational costs of quantization algorithms, while the task of finding better initial approximations received somewhat less attention. We present a novel multiscale scheme for the quantization problem. The scheme is based on the so called point density function and focuses on achieving better local minima than other contemporary methods such as LBG. When quantizing signals with sparse and patchy histograms, as may occur in color images for example, the improvement in distortion relative to LBG may be arbitrarily large.**

***Index Terms*— Quantization, Lloyd-Max, LBG, multigrid, FAS, point density.**

I. Introduction

Quantization [9], [11] is the process of representing continuum with only a finite number of representatives or representing an initially rich amount of discrete data with a lesser amount of representatives. Rounding off real numbers to the nearest integer is a simple form of scalar quantization. Representing color images with a lesser amount of colors is an example of vector quantization.

In general, many digital signal processing problems require some form of quantization. In particular, as the world is moving to digital media, processing voice, digital images and digital video involves a lot of quantization. Some problems which would benefit from better quantization algorithms are analogue to digital converters, sampling in general and compression.

A. Formal definition

Stating the problem more formally, given a random process whose probability density function $p : \Omega \to \mathbb{R}$, or an approximation thereof, is known and a positive integer n, a quantizer $q : \Omega \to \mathbb{R}$ is defined by n representation levels $R = \{\vec{r}_i\}_{i=1}^n$ and n decision regions $\{D_i\}_{i=1}^n$, where $\cup_i D_i = \Omega$. That is, $q(\Omega)$ is a piecewise constant approximation of Ω where all $\vec{x} \in D_i$ are represented by $\vec{r}_i$. An optimal quantizer achieves minimal distortion, which is defined as the expectation of the squared quantization error

$$\mathcal{D}(q) = E[\|X - q(X)\|^2] = \sum_{i=1}^{n} \int_{D_i} \|\vec{x} - \vec{r}_i\|^2 \, p(\vec{x}) \, d\vec{x}, \tag{1}$$

where $\|\cdot\|$ is a norm. Henceforth, we assume $\|\vec{x}\|^2 = \sum x_i^2$, i.e., the L_2 norm. In the discrete case, where $p(\vec{x})$ is non-zero at only a finite number of points $\{\vec{x}_i\}_{i=1}^N$, a similar functional is minimized

$$\mathcal{D}(q) = E[\|X - q(X)\|^2] = \sum_{i=1}^{n} \sum_{\vec{x}_j \in D_i} \|\vec{x}_j - \vec{r}_i\|^2 p(\vec{x}_j). \tag{2}$$

Henceforth, we permit $p(\vec{x})$ to be any non-negative function (not necessarily a distribution density function). In general, minimizing (2), is NP hard [7].

B. Previous Work

In recent years, many methods have been presented which reduce the complexity of finding a solution, e.g., by solving multiple 1-D problems [1], by recursively splitting the domain in two [18], [12], [23] or by accelerating known algorithms [25], [20]. These methods are usually greedy and/or heuristic and there is a complexity/distortion tradeoff. That is, lower complexity algorithms generally arrive at solutions with higher distortion. Wu [24] generalizes [18], [12], [23] and attempts to arrive at a good local minimum by making an observation about a constant number of consecutive splits and using dynamic programming.

C. The Lloyd-Max Iterative Process

Necessary conditions for a minimum of (1) are

$$\frac{\partial}{\partial r_i}\mathcal{D}(q) = 0, \qquad 1 < i < n$$
$$\frac{\partial}{\partial D_i}\mathcal{D}(q) = 0, \qquad 1 < i < n.$$

These conditions yield

$$\vec{r}_i = \frac{\int_{D_i} \vec{x} p(\vec{x}) d\vec{x}}{\int_{D_i} p(\vec{x}) d\vec{x}}, \quad D_i = V(\vec{r}_i), \tag{3}$$

where $V(\vec{r}_i)$ is the Voronoi cell corresponding to $\vec{r}_i$. Lloyd [15] and Max [17] came up (separately) with (3) and proposed "ping-ponging" back and forth between its equations,

$$D_i^\ell = V((\vec{r}_i)^\ell), \qquad \ell = 0,1,2,... \tag{4}$$

$$(\vec{r}_i)^{\ell+1} = \frac{\int\limits_{D_i^\ell} \vec{x}p(\vec{x})d\vec{x}}{\int\limits_{D_i^\ell} p(\vec{x})d\vec{x}}, \qquad \ell = 0,1,2,... \tag{5}$$

that is, optimizing the D_i's according to the $\vec{r}_i$'s (computing the Voronoi regions of the $\vec{r}_i$'s), followed by optimizing the $\vec{r}_i$'s according to the new D_i's (computing the centers of mass of the D_i's), and so on. $\vec{r}^0$ is an initial guess at the solution. This process converges monotonically to a solution which satisfies (3) (see [15]).

When applying Lloyd-Max iterations (4, 5) in the discrete case (2) one need not actually compute the Voronoi diagram which takes $\theta(n \log n + n^{\lceil k/2 \rceil})$ for dimension k. For high dimensions it is both sufficient and more efficient to determine for each $\vec{x} \in \Omega$, which representation level $\vec{r}_i$ is closest to $\vec{x}$. The naive method of inspecting all n possibilities for each $\vec{x} \in \Omega$ and choosing the minimal takes $O(kNn)$ operations where $N = |\{\vec{x} : p(\vec{x}) \neq 0\}|$. Using a k-d tree [6] for finding best matches, the expected computation time of a Lloyd-Max iteration is $O(kN \log n)$.

For scalar quantization ($k = 1$), Fleischer showed that a sufficient condition for a globally minimal distortion is that $\log(p(x))$ be concave [5]. For vector quantization, we are unaware of a similar result.

Since the Lloyd-Max iterative process requires an initial guess at the solution in order to start iterating, many heuristic methods, such as k-means [16] and LBG [14], have been proposed which try to find a good initial approximation. The LBG [14] algorithm is a "grid refinement" type process which starts by finding a solution (using the generalized Lloyd algorithm) for two $\vec{r}_i$'s and then splitting each $\vec{r}_i$ into two new $\vec{r}_i$'s. This process is repeated recursively until the number of $\vec{r}_i$'s equals n. The Generalized Lloyd algorithm is the iterative process (4, 5) with the reduction in distortion used as stopping criterion. That is, the process is stopped at the iteration in which the ratio between the current distortion and the previous iteration's distortion exceeds some threshold. In the sequel, we shall compare our method with LBG.

A multigrid [4], [22] approach for scalar (1-D) quantization was presented in [13]. Our approach was basically a multigrid full approximation scheme algorithm [3] with the Lloyd-Max iteration (4, 5) used as relaxation. The benefits are 3-fold. First, much faster convergence is achieved. The convergence rates of our method seem independent of the initial guess or number of representation levels sought. In contrast, both Lloyd-Max and LBG converge at a rate which is quadratically dependent on the number of representation levels. Second, the speed up alleviates the need for convergence/stopping criteria such as the one used for LBG. This is because reaching machine accuracy requires only several iterations when using our method. Third, our method sometimes arrives at better minima than other methods.

The paper is organized as follows. Section II provides a detailed description of our proposed adaptive multiscale redistribution (AMR) algorithms for vector quantization. Section III includes results of our vector quantization algorithms and a comparison with LBG. Section IV is a conclusion and further discussion.

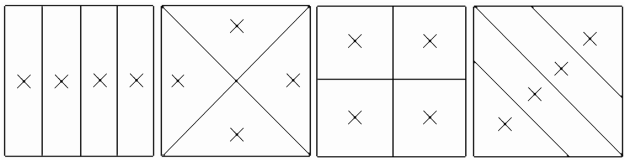

Fig. 1. Multiple local minima for $p(x,y) \equiv 1$. The x's are the representation levels which are the centers of mass of the decision regions (Voronoi cells)

II. Adaptive Multiscale Redistribution Algorithms

The quantization problem is generally non-convex and therefore affords multiple local minima [10]. Figure (1) depicts several local minima for the simple $p(x,y) \equiv 1$ case with 4 representation levels. For scalar quantization, we have shown how to accelerate the convergence rate from initial guess to local minimum [13]. We consider the acceleration of convergence less important than converging to a solution of lower distortion. In this section we develop methods for both scalar and vector quantization which aim to overcome this acute multiple local minima problem.

A. High Resolution Approximations

The point density function [15] of a vector quantizer q is a nonnegative, usually smooth function $\lambda(\vec{x})$ that, when integrated over a region, determines the approximate fraction of representation levels in that region. That is,

$$\frac{|\{r_i \ : \ r_i \in S\}|}{n} \approx \int\limits_S \lambda(\vec{x})d\vec{x}. \tag{6}$$

As with any density function which describes a discrete set of points, the point density function is not defined uniquely for a quantizer q. Instead, it is intended to roughly estimate the location of the representation levels, just as the probability density function roughly estimates the location of the original data points [11]. As an example, given a quantizer q, a piecewise constant point density function for q may be defined as

$$\lambda_q(\vec{x}) = (nVol(D_i))^{-1} = \frac{1}{n \int\limits_{D_i} d\vec{x}}, \qquad (7)$$

where D_i is the decision region containing $\vec{x}$ and n is the number of representation levels.

Many theoretical results exist regarding optimal point density functions and their use for quantizer analysis [2], [15], [26], [19], [8], [21]. The quantization methods and algorithms described in this section exploit these results in order to obtain lower distortion efficiently.

When the number of representation levels is large we may assume that $p(\vec{x}) \approx p(\vec{r}_i)$ for all $x \in D_i$, i.e., we may assume that the distribution is approximately uniform inside each decision region. In both 1 and 2 dimensions it is known [21], that the optimal Voronoi diagram for a uniform distribution function $p(\vec{x})$ is approximately a lattice tessellation of the domain. That is, all Voronoi cells (decision regions) are congruent, with the possible exception of cells touching the boundary of the domain. For 1 and 2 dimensions, the tessellation-generating polytopes are intervals and hexagons respectively. Gersho [8] made the now widely accepted conjecture that for any dimension k, when the number of representation levels is large, most cells (excluding the ones touching boundaries) are congruent to some k-dimensional polytope.

Based on his conjecture, Gersho [8] showed that for large n, the optimal quantizer point density function obeys

$$\int\limits_{S} \lambda(\vec{x}) d\vec{x} \approx \frac{\int\limits_{S} p^{\frac{k}{k+2}}(\vec{x}) d\vec{x}}{\int\limits_{\Omega} p^{\frac{k}{k+2}}(\vec{x})\, d\vec{x}}, \qquad (8)$$

for any measurable set S in Ω. Gersho also showed [8] the following property for an optimal quantizer in the large n limit.

Lemma 1: For an optimal quantizer q, each decision region makes an equal contribution to the distortion (1), (2).

The property indicated by lemma 1 was observed for $k = 1$ by Panter and Dite [19] and for $k = 2$ by Fejes Toth [21].

We now exploit the theoretical results described above within a novel Adaptive Multiscale Redistribution (AMR) algorithm. We begin by defining three methods for building a quantizer. Our ultimate quantization algorithm involves concatenating these three methods.

The first method is an extremely quick way of attaining a good initial set of n representation levels. This method heavily relies on (8) and achieves a distortion which is nearly as low as our final result. The complexity of obtaining this initial guess is proportional to $\log(n)$ Lloyd-Max relaxations where n is the number of representation levels sought. In contrast, LBG [14] requires approximately $O(n)$ Lloyd-Max relaxations to come up with an initial approximation, i.e., to reach the finest grid where there are n representation levels.

The second method is used to improve an initial guess. The improvement is based on lemma 1, and is performed by moving representation levels from low distortion areas to areas with higher distortion. The longer the method is applied, the better the minimum obtained.

The last method is simply applying Lloyd-Max iterations (4), (5) to the best approximation found to date, thus reaching a local minimum of the quantization distortion. We propose a quantization algorithm which is made up of these 3 steps. First, find an initial guess. Second, improve the initial guess. Third, relax until a local minimum has been reached.

B. The General Approach

In both acquiring an initial guess and improving it, we use the following adaptive multiscale redistribution (AMR) approach. A hierarchy of scales is defined. Scale m is a partition of the set of representation levels R into m aggregates,

$$R^m = \{R_j^m\}_{j=1}^m, \; R = \bigcup_j R_j^m. \qquad (9)$$

At any scale m, each of the m aggregates are approximately equal in size. At the finest scale, each aggregate is made up of a single representation level, that is, $m = n$. When passing from scale m to scale $m/2$ (the next coarser scale), the m aggregates are grouped in pairs to form approximately $m/2$ new aggregates. This set of $m/2$ new aggregates is scale $m/2$. The coarsest scale consists of two aggregates.

At each scale, we determine the hungriest aggregate, i.e., the aggregate which would benefit the most from an additional representation level. In a similar manner, we determine the most satisfied aggregate, i.e., the aggregate which would suffer the least from giving up a representation level. Then we transfer a representation level from the most satisfied to the hungriest aggregate.

C. Acquiring an Initial Guess

For the purpose of acquiring an initial guess we exploit our knowledge regarding the optimal point density function (8). Let

$$D(R_j^m) = \bigcup_{r_i \in R_j^m} D_i, \tag{10}$$

i.e., $D(R_j^m)$ is the union of the decision regions of all the representation levels in R_j^m. For each aggregate R_j^m of the m aggregates in R^m, (8) tells us how many representation levels an optimal quantizer allocates to the region $D(R_j^m)$. Subtracting the actual number of representation levels in $D(R_j^m)$ from this amount, we arrive at the "hunger", H_j^m, of aggregate R_j^m. More formally,

$$H_j^m = \frac{\int\limits_{D(R_j^m)} p^{\frac{k}{k+2}}(\vec{x}) d\vec{x}}{\int\limits_{\Omega} p^{\frac{k}{k+2}}(\vec{x})\, d\vec{x}} - \left|R_j^m\right|. \tag{11}$$

We call aggregates whose hunger is above zero "hungry aggregates" and aggregates whose hunger is below zero "satisfied aggregates".

Our algorithm for obtaining an initial guess is as follows. We initialize R^n by randomly choosing n representation levels uniformly in Ω. We use this uniform initial approximation for two reasons. First, it is extremely cheap to compute. Second and more importantly, the $Obtain-Guess$ algorithm transfers representation levels to places in Ω which already contain other representation levels. Therefore, a uniform initial approximation gives an equal chance for all areas in Ω to receive representation levels. Next we call the following recursive function with R^n as input (note that $n = |R^n|$).

$Obtain - Guess(p(\vec{x}), R^m)$

1) Perform a single Lloyd-Max relaxation.
2) Calculate $H^m = \{H_i^m\}_{i=1}^m$.
3) Set $H_{max}^m \leftarrow \max(H^m)$, $H_{min}^m \leftarrow \min(H^m)$.
4) *While* $H_{max}^m - H_{min}^m > 1$, *do*:
   a) $\vec{r}_j \leftarrow Select - Source(R_{min}^m)$ (*).
   b) $\vec{t} \leftarrow Select - Destination(R_{max}^m)$ (**).
   c) Transfer $\vec{r}_j$ from R_{min}^m to R_{max}^m at location $\vec{t}$.
   d) $H_{min}^m \leftarrow H_{min}^m + 1$, $H_{max}^m \leftarrow H_{max}^m - 1$.
5) $R^{m/2} \leftarrow Match(R^m)$ (***).
6) *If* $R^{m/2} = R^m$
   a) *Return* R^m.
7) *Else*,
   a) Recursively call $Obtain - Guess(p(\vec{x}), R^{m/2})$.

The single Lloyd-Max relaxation at each scale (step 1 of $Obtain - Guess$) is performed to improve the solution. In addition, the calculation of H_i^m requires knowing the decision regions and a Lloyd-Max relaxation is a means of finding the optimal decision regions for a given set of representation levels.

Note that $Obtain - Guess$ is an adaptive multiscale redistribution of the representation levels. The goal of this redistribution is that the quantizer's point density function obey (8) as best as possible. Since checking (8) for all possible sets S is intractable, we only do so for various unions of decision regions. The following lemma shows how each transition of representation level at step 4c of $Obtain - Guess$ makes the point density function of the quantizer more coherent with (8).

Lemma 2: When $H_{max}^m - H_{min}^m > 1$ (step 4 of $Obtain - Guess$) then the transition of a representation level (step 4c of $Obtain - Guess$) reduces $\sum_i |H_i^m|$.

Proof: We shall prove that $|H_{max}^m - 1| + |H_{min}^m + 1| < |H_{max}^m| + |H_{min}^m|$. Assume that $0 < H_{max}^m < 1$ and $-1 < H_{min}^m < 0$. When this is not true, the result is trivial. Note that $H_{max}^m > 0$ and $H_{min}^m < 0$ always holds. $|H_{max}^m - 1| = 1 - H_{max}^m < -H_{min}^m = |H_{min}^m|$. The first equality is implied by $H_{max}^m < 1$ and the inequality by $H_{max}^m - H_{min}^m > 1$. In a similar manner, $|H_{min}^m + 1| = 1 + H_{min}^m < H_{max}^m = |H_{max}^m|$ where the first equality is implied by $H_{min}^m > -1$ and the inequality by $H_{max}^m - H_{min}^m > 1$. ∎

The selection (*) of a representation level in the most satisfied aggregate is performed as follows.

$\vec{r}_i \leftarrow Select - Source(R_j^m)$

1) *If* $\left|R_j^m\right| > 1$
   a) *If* $R_j^m = R_s^{2m}$ for some $1 \le s \le 2m$,
      i) Recursively call $Select - Source(R_s^{2m})$
   b) *Else*,
      i) $R_j^m = R_s^{2m} \cup R_t^{2m}$ for some $1 \le s, t \le 2m$. Let R_ℓ^{2m} be the aggregate which achieves $\min\{H_s^{2m}, H_t^{2m}\}$.
      ii) Recursively call $Select - Source(R_\ell^{2m})$
2) *Else*,
   a) *Return* the single $\vec{r}_i \in R_j^m$

Less formally, we wish to remove the representation level in R_j^m whose decision region is the most satisfied at all scales. We traverse the scales from the present scale to finer scales. At each scale m we deal with a single aggregate R_j^m. R_j^m is usually made up of two $2m$-scale aggregates (the actual matching is performed by $Match$). We choose the most satisfied aggregate among the two and continue recursively. When we reach an aggregate containing a single representation level, we choose it as the representation level to move.

The selection (**) of a destination in the hungriest aggregate is performed as follows.

$\vec{t} \leftarrow Select - Destination(R_j^m)$

1) $If \quad |R_j^m| > 2$
   a) $If \quad R_j^m = R_s^{2m}$ for some $1 \le s \le 2m$,
      i) Recursively call $Select - Destination(R_s^{2m})$.
   b) $Else$,
      i) $R_j^m = R_s^{2m} \cup R_t^{2m}$ for some $1 \le s, t \le 2m$. Let R_ℓ^{2m} be the aggregate which achieves $\max\{H_s^{2m}, H_t^{2m}\}$.
      ii) Recursively call $Select - Source(R_\ell^{2m})$
2) $Else$,
   a) $Return$ the mean of the set R_j^m

Less formally, we wish to insert a representation level at a location which is least satisfied at all scales. In a manner similar to $Select - Source$, we traverse the scales from the present scale to finer scales. This time, at each scale m we choose the hungriest aggregate among the two which make up R_j^m and continue recursively. When we reach an aggregate containing just two representation levels, we choose the destination as the midpoint between the two levels.

The coarser scales are required to take care of the situations where many small aggregates are somewhat hungry and many other small aggregates are somewhat satisfied. In this case, the grouping of several somewhat hungry aggregates should form a larger aggregate which is hungry enough to sustain an additional representation level. In a similar fashion, the grouping of several somewhat satisfied aggregates should form a larger aggregate which is satisfied enough to allow a representation level to be removed.

In light of the above, the matching (***) between pairs of aggregates, which is designed to produce larger aggregates, should only match aggregates which are on the same side of the hunger/satisfaction fence. In addition we prefer to match aggregates which are close to each other. This is because adding a new representation level to an aggregate made up of two distant regions only reduces the hunger of the region which accepts the representation level. If indeed this is beneficial, it should have been done at a finer scale, where each of the two regions was a separate aggregate. We define the distance between two aggregates as the distance between the mean of the representation levels in each aggregate, i.e.,

$$\|R_i^m, R_j^m\| = \left\| \frac{\sum_{\vec{r}_i \in R_i^m}(\vec{r}_i)}{|R_i^m|} - \frac{\sum_{\vec{r}_j \in R_j^m}(\vec{r}_j)}{|R_j^m|} \right\|_2 . \quad (12)$$

More formally, the matching algorithm is defined as follows.

$R^{m/2} \leftarrow Match(R^m)$

1) Initialize $R^{m/2} = \phi$.
2) Sort the aggregates in R^m according to $|H_i^m|$.
3) Traverse the sorted list, for each aggregate R_j^m, do:
   a) Denote by Δ the set of unmatched aggregates, R_k^m, which satisfy $\mathrm{sign}(H_k^m) = \mathrm{sign}(H_j^m)$.
   b) If $\Delta = \phi$
      i) Add R_j^m to $R^{m/2}$ and mark R_j^m as matched.
   c) Else,
      i) Find $R_i^m = \operatorname*{argmin}_{R_k^m \in \Delta} \|R_k^m, R_j^m\|$.
      ii) Add $R_j^m \cup R_i^m$ to $R^{m/2}$ and mark both R_j^m and R_i^m as matched.
4) $Return \quad R^{m/2}$

Using k-d trees [6] to compute $\operatorname{argmin} \|R_k^m, R_j^m\|$ in step 3(c)i of $Match$, the expected complexity of the matching algorithm is $O(km \log m)$ where m is the number of aggregates. Note that this is asymptotically less than the $O(kN \log n)$ required for a Lloyd-Max relaxation.

We apply $Obtain - Guess$ several times, until no representation levels are moved at any scale. Each call to $Obtain - Guess$ applies $O(\log(n))$ Lloyd-Max relaxations which is the bulk of its time complexity. In practice we have applied this method to many examples. We've found that about 2 - 3 calls to $Obtain - Guess$ generate a "steady" initial guess, i.e., one that is not affected by another call to $Obtain - Guess$. LBG on the other hand, requires approximately $O(n)$ Lloyd-Max relaxations in order to acquire an initial guess, that is, to reach the finest scale.

From our tests, the distortion of the initial guess thus obtained is on average only 5% higher than the final distortion achieved by concatenating our 3 methods. For $p(\vec{x})$ that are sparse and patchy (e.g. color images), the ratio between this distortion and that achieved by LBG may be arbitrarily small as shall be demonstrated in section III.

D. Improving an Initial Guess

$\underline{Improve-Guess(R^m, T, \nu)}$

1) Perform a single Lloyd-Max relaxation.
2) Perform the following as long as the distortion of R^m, $\mathcal{D}(R^m)$, decreases:
   a) Calculate $H^m = \{H_i^m\}_{i=1}^m$.
   b) $If \quad \max(H^m) - \min(H^m) > T$,
      i) Move 1 representation level from the hungriest aggregate to the most satisfied aggregate, using $Select-Source$ (*) and $Select-Destination$ (**). Denote by $\tilde{R}^m$ the set of aggregates after the transition.
      ii) $\tilde{R}^m \leftarrow Beat-Target-Distortion(\tilde{R}^m, \mathcal{D}(R^m), \nu)$.
      iii) If the distortion has not improved,
         A) discard the new $\tilde{R}^m$ and keep R^m.
      iv) $Else$,
         A) $R^m = \tilde{R}^m$.
3) Form a set of new aggregates: $R^{m/2} \leftarrow Match(R^m)$.
4) $If\ R^{m/2} = R^m$
   a) $Return\ R^m$.
5) $Else$,
   a) Recursively call $Improve-Guess(R^{m/2}, \nu)$.

Improving a guess requires more leniency than acquiring an initial guess. The harsh demand that $\max(H^m) - \min(H^m) > 1$ (step 4 of $Obtain-Guess$) in order to perform a transition is relaxed by $\max(H^m) - \min(H^m) > T$ for some threshold T. In addition we re-define an aggregate's hunger. This time the definition is based on lemma 1.

$$H_j^m = \frac{average\ distortion\ at\ aggregate\ j}{average\ distortion\ in\ \Omega} - 1, \quad (13)$$

where "$average\ distortion$" is the average distortion per representation level.

According to lemma 1, an optimal quantizer obeys $H_j^m = 0$, for all $1 \leq j \leq m \leq n$. This definition still permits us to call aggregates whose hunger is above zero - "hungry aggregates" and aggregates whose hunger is below zero - "satisfied aggregates".

Our improvement method uses the above distortion based hunger definition (13). From our tests, the distortion based hunger definition (13) leads to better results. On the other hand, a meaningful threshold for the initial guess phase (step 4 in $Obtain-Guess$) such as the one which lemma 2 provides for (11) is more difficult to come by for (13). This is why we use the point density based hunger definition (11) with threshold $T = 1$ when acquiring an initial guess.

$Beat-Target-Distortion$ applies Lloyd-Max relaxations to $\tilde{R}^m$, attempting to achieve a distortion lower than $\mathcal{D}(R^m)$. After each relaxation the function tries to assess if ν additional relaxations are likely to reduce the distortion enough. This assessment is made by assuming that each additional Lloyd-Max relaxation will reduce the distortion by the actual reduction achieved via the previous Lloyd-Max relaxation.

The linear extrapolation we use to forecast the distortion reduction has the following property. It is optimistic in the early stages, at which time a more accurate approximation may halt the process prematurely, while in the later stages, as the distortion reduction rate reduces, it becomes more accurate. If the target distortion is beaten, then $Beat-Target-Distortion$ applies ν additional Lloyd-Max relaxations and the result is returned. The additional ν relaxations are performed to allow a solution to realize its potential. More formally,

$\underline{Beat-Target-Distortion(R^m, \mathcal{D}, \nu)}$

1) Do: {
   a) Denote the current distortion $\mathcal{D}^{old} = \mathcal{D}(R^m)$.
   b) Perform a Lloyd-Max relaxation on R^m.
   c) Denote the new distortion by $\mathcal{D}^{new} = \mathcal{D}(R^m)$.
   d) if $\mathcal{D}^{new} < \mathcal{D}$
      i) Apply ν additional Lloyd-Max relaxations to R^m and return the result.

   } $While \quad \mathcal{D}^{new} - \nu|\mathcal{D}^{new} - \mathcal{D}^{old}| < \mathcal{D}$.
2) $Return \quad R^m$.

When a representation level is added or removed (step 2(b)i of $Improve-Guess$), a Lloyd-Max iteration propagates this change only to the neighbors of this representation level (4), (5). Assuming that the representation levels are spread uniformly in the domain Ω, it would take approximately $n^{1/k}$ Lloyd-Max iterations to propagate this change throughout all representation levels. For this reason we recommend using $\nu \geq \lceil n^{1/k} \rceil$. We chose $\nu = \lceil n^{1/k} \rceil$ in our tests. For the threshold T, we used $T = 1/2$.

Coarse scale changes may upset the balance achieved at the finer scales. Therefore, it may be beneficial to call $Improve-Guess$ several times. Before calling $Improve-Guess$ an additional time, ν may be raised and T lowered. In our examples we have always called $Improve-Guess$ until no transitions of representation levels were performed at any scale.

The AMR process can be stopped at any time and from our tests, no matter when the process is stopped, the results are better than those achieved by LBG after the same amount of time.

III. Vector Quantization Results

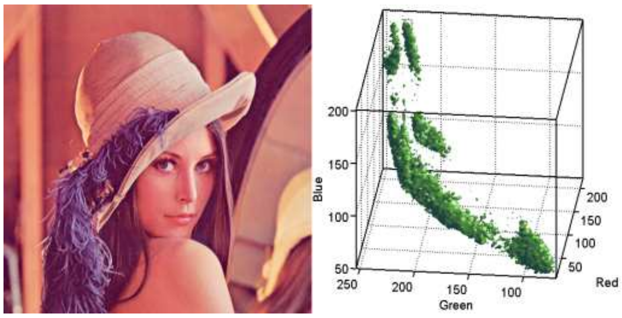

Fig. 2.
Left: Lena, 512x512 RGB image. Right: Lena's histogram $p(\vec{x})$.

Many digital signal processing applications involve some form of quantization. As an example, color images may be compressed by representing them with only a fraction of the initial amount of colors. The histograms of RGB (color) images are usually sparse and patchy, as Figure 2 illustrates. We demonstrate our algorithms on discrete 2-D data using the discrete distortion definition (2). Let

$$p_{ij} = \begin{cases} 10 & i,j \in \{0,1,2,...,100\} \\ 3/4 & i,j \in \{200,201,202,...,255\} \\ 0 & otherwise \end{cases} \quad (14)$$

p_{ij} is comprised of 2 patches (see Figure 14), one larger and of higher value than the other. From p_{ij}'s structure alone we can deduce that most representation levels should be allocated to the larger patch. Note that the fact that the patches are square is completely unimportant, we defined p_{ij} this way solely for simplicity.

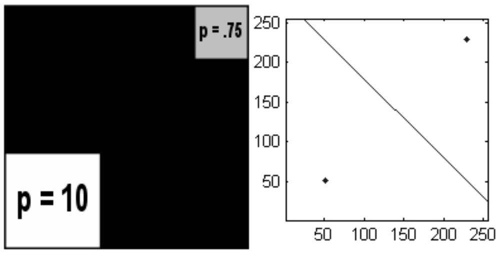

Fig. 3. Left: p_{ij}. Right: Globally optimal solution for $n = 2$, produced by both LBG and AMR.

We applied both LBG and AMR to this problem. LBG was used with threshold 10^{-4}, that is, Lloyd-Max iterations were performed at each scale until the reduction in distortion ($\mathcal{D}^{new}/\mathcal{D}^{old}$) exceeded $1-10^{-4}$. Once the highest resolution was reached, Lloyd-Max relaxations were performed until the residual dropped to machine zero, that is, until no further improvement could be made.

For $n = 2$ both LBG and our method produce the same solution (see Figure 3), namely, one representation level for each patch (see Figure 3). Figure 4 displays the solutions reached by AMR and LBG for $n = 4, 8, 16, 32$. Figure 4 implies that LBG may not be able to remedy errors made at coarse scales. At the stage where $n = 2$ LBG decided to allocate one representation level to each patch. From that moment on, no representation levels could move from patch to patch since for any representation level, the center of mass of its decision region is always located within its patch.

n	LBG	AMR	AMR:LBG Ratio
2	$167.8 \cdot 10^6$	$167.8 \cdot 10^6$	100%
4	$104.9 \cdot 10^6$	$674 \cdot 10^5$	64.26%
8	$419.6 \cdot 10^5$	$265.4 \cdot 10^5$	63.25%
16	$218 \cdot 10^5$	$124.8 \cdot 10^5$	57.25%
32	$104.7 \cdot 10^5$	$625.4 \cdot 10^4$	59.73%

TABLE I
RESULTS (DISTORTION VALUE) OF OUR ADAPTIVE MULTISCALE REDISTRIBUTION (AMR) METHOD VS. LBG

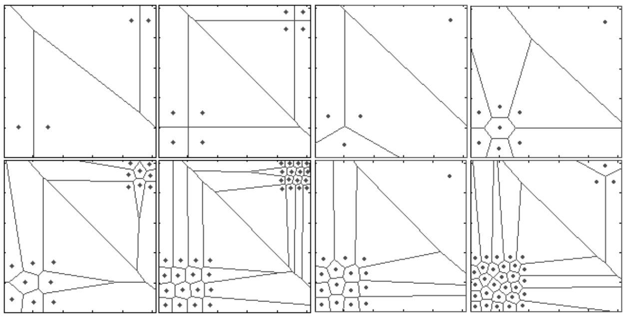

Fig. 4. Quantization results of AMR and LBG on p_{ij} with $n = 4, 8, 16, 32$. 4 left hand plots: LBG. 4 right-hand plots: AMR.

Table I displays the distortion values of the solutions attained by AMR and LBG for various n's. The last column displays the ratio between the AMR distortion and the LBG distortion.

From Figure 4 it is obvious that raising the LBG distortion improvement threshold for any $n > 2$ would not be beneficial. Raising the threshold would yield a solution with lower valued distortion at each of the coarse scales. LBG implicitly assumes a certain connection between the solutions at the various scales. Figure

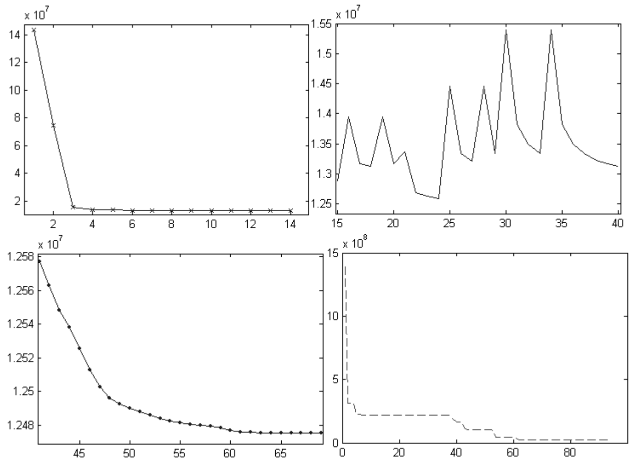

Fig. 5. Distortion vs. iterations for AMR and LBG methods applied to p_{ij}, $n = 16$. Note the different scale in each plot. Top left: AMR acquiring an initial guess. Top right: AMR improving the initial guess. Bottom left: AMR applying Lloyd-Max until local minimum is reached. Bottom right: LBG.

4 demonstrates, for example when moving from $n = 2$ to $n = 4$, that this is not necessarily the case.

In contrast to LBG, where the coarse scales lead us astray, AMR uses the coarse scales effectively to improve the fine scale solution. This is done by correcting errors at the coarse scales which were unrecognizable at the fine scale.

Figure 5 displays the distortion vs. Lloyd-Max iterations for both LBG and AMR for $n = 16$. At its first stage, depicted in the upper left-hand plot, AMR begins with a uniform guess and redistributes the representation levels between the patches using the hunger function (11). Note the substantial drop in distortion achieved by the first 3 iterations. LBG falls short of this result even when run to completion. At its second phase, depicted in the upper right-hand plot, the algorithm tries to improve the solution by using (13). The purpose of this improvement stage is to traverse other feasible solutions which may have been overlooked in the first phase. Indeed an improvement is made at iteration 24. Finally, the best solution to date, in this case, the one found at iteration 24, is brought to a local minimum by application of Lloyd-Max iterations (4, 5). That is, Lloyd-Max iterations are performed until no further improvement is possible. This process is depicted in the lower left-hand plot. The lower right-hand plot displays the LBG Distortion vs. iteration number throughout the algorithm.

The time complexity, measured in Lloyd-Max iterations, is lower for our method than for LBG in all cases shown. Our multiscale improvement method, $Improve - Guess$, was performed until the first time that no transitions of representation levels were made at any scale.

Table I demonstrates how the optimal solution (which may be better than the AMR solution) may be nearly twice as good as the LBG solution. We now demonstrate how the LBG solution may be arbitrarily bad.

Consider as an example, continuous or discrete, a patchy $p(\vec{x})$. Let $p(\vec{x})$ be nonzero only at ℓ disjoint patches $\{\Omega_i\}_{i=1}^{\ell}$. One patch covers a considerable part of the domain Ω and the other $\ell - 1$ patches cover an arbitrarily small part of Ω. Assume that $p(\vec{x})$ is constant within each patch and that $\int_{\Omega_i} p(\vec{x})$ is equal for all $1 \leq i \leq \ell$. That is, all patches have the same mass. Figure 6 is an example of such a $p(\vec{x})$ with 6 patches.

At the scale $n = \ell$, LBG's solution would include approximately one representation level per patch. At finer scales, each representation level is doubled and no representation level may move from one patch to another just as with the previous p_{ij} (14) (see Figure 4). For large n the majority of the representation levels should be allocated to the large patch S due to the quadratic penalty for points which are far from their representative (1). In contrast, LBG is forced to keep an approximately equal number of representation levels per patch. The larger the number of patches ℓ and the smaller the area of Ω which each patch covers (except for the big one), the worse the LBG solution becomes relative to the optimal solution. As an example, consider the $p(\vec{x})$ displayed in figure 6. AMR (which may not achieve the optimal solution) achieved a distortion which is approximately a third of the distortion achieved by LBG for $n = 8, 16$ and 32.

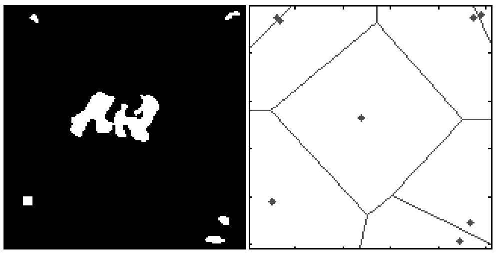

Fig. 6. Left: $p(\vec{x})$ with 6 patches of different sizes and shapes but same mass. Right: LBG applied to $p(\vec{x})$, $n = 8$.

IV. Conclusions and Further Discussion

With respect to the multiple local minima problem, we have demonstrated an adaptive multiscale redistribution (AMR) strategy based on the point density function and distortion of an optimal quantizer. This strategy is non-local and may determine from a local glance that the solution is not globally optimal, an attribute not shared by the Lloyd-Max method. Our strategy includes an extremely fast method for obtaining an initial guess, as well as a method for improving a guess and bringing it to a local minimum. From our tests the strategy may beat LBG by an arbitrary factor, and it may do so in a fraction of the computation time required by LBG.

A more general distortion definition than (1), where the quantization error is raised to some arbitrary power instead of two is possible, i.e.,

$$\mathcal{D}(q) = E[\|X - q\left(X\right)\|^t] = \sum_{i=1}^{n} \int_{D_i} \|\vec{x} - \vec{r}_i\|^t \, p\left(\vec{x}\right) d\vec{x}. \tag{15}$$

Gersho's optimal point density results [8] generalize to this case and therefore AMR methods are worthwhile here too.

In a previous paper [13] we have presented a multigrid acceleration of the 1-D Lloyd-Max iterative process. That is, given an initial guess, a local minimum near it may be reached at a complexity equivalent to just a few Lloyd-Max iterations. Using the AMR methods for the scalar case may be very powerful. This is because, at the guess improvement stage, we may replace every application of Lloyd-Max iterations (step 2(b)i of $Improve - Guess$) with our monotonically convergent multigrid solver [13]. Doing so implies that we need not guess if a transition of a representation level is worthwhile. Instead we can use our multigrid method to arrive at a local minimum and compare its distortion with the best distortion to date.

An interesting question is whether or not an acceleration scheme similar to that of the scalar quantization problem exists for vector quantization. We believe that such a multigrid method does not exist since the problem doesn't behave "elliptically". A small change in the solution at some point may cause a large change in the solution at some other distant point. For this reason, the Lloyd-Max relaxations do not smooth the distortion/residual which is a basic demand for multigrid schemes.

We believe that the ideas proposed in this paper may be applied to other interesting representation problems. For example, representing a k-dimensional manifold $S(\vec{x})$ by n points. From these n points, given some interpolation scheme, one can compute an approximating manifold $S'(\vec{x})$. The objective is to minimize

$$\int \|S(\vec{x}) - S'(\vec{x})\|^2 d\vec{x}. \tag{16}$$

With suitable adaptations, our proposed methods may also be able to deal with other image processing problems such as halftoning, where the average color value of the halftoned image should equal the average color value of the original image at all scales.

References

[1] R. Balasubramanian, C. Bouman, and J. P. Allebach, "Sequential scalar quantization of vectors: An analysis," *IEEE Transactions on Image Processing*, vol. 4, no. 9, pp. 1282–1295, September 1995.

[2] W. R. Bennett, "Spectra of quantized signals," *Bell Systems Technical Journal*, vol. 27, pp. 446–472, July 1948.

[3] A. Brandt, "Multi-level adaptive solutions to boundary-value problems," *Mathematics of Computation*, vol. 31, pp. 333–390, 1977.

[4] W. L. Briggs, V. Henson, and S. F. McCormick, *A Multigrid Tutorial*, 2nd ed. SIAM, 2000.

[5] P. Fleischer, "Sufficient conditions for achieving minimum distortion in a quantizer," *IEEE International Convention Record*, pp. 104–111, 1964.

[6] J. H. Freidman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Math. Softw.*, vol. 3, no. 3, pp. 209–226, 1977.

[7] M. R. Garey, D. S. Johnson, and H. S. Witsenhausen, "The complexity of the generalized Lloyd-Max problem," *IEEE Transactions on Information Theory*, vol. 28, pp. 255–256, March 1982.

[8] A. Gersho, "Asymptotically optimal block quantization," *IEEE Transactions on Information Theory*, vol. IT-25, no. 4, pp. 373–380, July 1979.

[9] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Kluwer Academic Publishers, 1992.

[10] R. M. Gray and E. D. Karnin, "Multiple local optima in vector quantizers," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 256–261, March 1982.

[11] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 1–63, October 1998.

[12] P. S. Heckbert, "Color image quantization for frame buffer display," *ACM Computer Graphics (ACM SIGGRAPH '82 Proceedings)*, vol. 16, no. 3, pp. 297–307, 1982.

[13] Y. Koren, I. Yavneh, and A. Spira, "A multigrid approach to the 1-D quantization problem," 2003, submitted for publication.

[14] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. COM-28, pp. 84–95, January 1980.

[15] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, March 1982, unpublished Bell Laboratories Technical Note 1957.

[16] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Fifth Berkeley Symposium on Mathematics Statistics and Probability*, vol. 1, 1967, pp. 281–296.

[17] J. Max, "Quantizing for minimal distortion," *IEEE Transactions on Information Theory*, vol. 6, no. 1, pp. 7–12, March 1960.

[18] M. T. Orchard and C. Bouman, "Color quantization of images," *IEEE Transactions on Signal Processing*, vol. 39, no. 12, pp. 2677–2690, December 1991.

[19] P. F. Panter and W. Dite, "Quantizing distortion in pulse-count modulation with nonuniform spacing of levels," in *Proc. IRE*, vol. 39, January 1951, pp. 44–48.

[20] L. Torres and J. Huguet, "An improvement on codebook search for vector quantization," *IEEE Transactions on Communications*, vol. 42, no. 2-4, pp. 208–210, Part 1 Febuary-April 1994.

[21] L. F. Toth, "Sur la representation d'une population infinie par un nombre fini d'elements," *Acta Math. Acad. Sci. Hung.*, vol. IT-25, no. 4, pp. 76–81, 1959.
[22] U. Trottenberg, C. Oosterlee, and A. Schüller, *Multigrid*. Academic Press, 2001.
[23] S. J. Wan and S. K. M. Wong, "An algorithm for multidimensional data clustering," *ACM Transactions on Mathematical Software*, vol. 14, no. 2, pp. 153–162, 1988.
[24] X. Wu, "Color quantization by dynamic programming and principal analysis," *ACM Transactions on Graphics*, vol. 11, no. 4, pp. 348–372, October 1992.
[25] X. Wu and L. Guan, "Acceleration of the lbg algorithm," *IEEE Transactions on Communications*, vol. 42, no. 2-4, pp. 1518–1523, Part 3 Febuary-April 1994.
[26] P. L. Zador, "Development and evaluation of procedures for quantizing multivariate distributions," Ph.D. dissertation, Stanford University, 1963.