# Multiple Description Lattice Vector Quantization: Variations and Extensions[*]

Jonathan A. Kelner, Vivek K Goyal and Jelena Kovačević

Mathematics of Communications Research

Bell Labs, Lucent Technologies

kelner@fas.harvard.edu, v.goyal@ieee.org, jelena@bell-labs.com

**Abstract**

Multiple description lattice vector quantization (MDLVQ) is a technique for two-channel multiple description coding. We observe that MDLVQ, in the form introduced by Servetto, Vaishampayan and Sloane in 1999, is inherently optimized for the central decoder; *i.e.*, for a zero probability of a lost description. With a nonzero probability of description loss, performance is improved by modifying the encoding rule (using nearest neighbors with respect to "multiple description distance") and by perturbing the lattice codebook. The perturbation maintains many symmetries and hence does not significantly effect encoding or decoding complexity. An extension to more than two descriptions with attractive decoding properties is outlined.

## 1 Introduction

The contents of this volume and the 1999 edition of these *Proceedings* indicate considerable interest in multiple description (MD) coding. MD coding is a generalization of classical source coding in which multiple bit streams are used to describe a source. Each of these bit streams (descriptions) can be decoded separately; they also can be decoded in any combination. The goal is roughly to have reconstruction quality that improves as the number of received descriptions increases without introducing too much redundancy between the descriptions. Background on MD coding and its applications can be found in other papers in these *Proceedings* and in [1, 2].

For the most part, we consider techniques for two descriptions and use the conventional notation of $R_1$ and $R_2$ for the rates of the two descriptions, $D_1$ and $D_2$, respectively, for the distortions of reconstructions from these descriptions, and $D_0$ for the distortion of the reconstruction computed from both descriptions. Distortions $D_1$ and $D_2$ are called *side distortions* and $D_0$ is called the *central distortion.* Similarly, the corresponding decoders are called side and central decoders. The techniques considered here are *balanced* in that the rates are equal and the expected values of the side distortions are equal.

[*] Jonathan Kelner is with Harvard University, Cambridge MA 02138. This work was performed while he was a summer intern at Bell Labs.

In this work, we concentrate on MD systems based on specially designed quantizers. Furthermore, to achieve the improvement promised by vector quantization while limiting the computational burden, we concentrate on multiple description lattice vector quantization (MDLVQ) instead of less constrained methods (see, *e.g.* [3]). This paper begins, in Section 2, by reviewing the MDLVQ technique proposed last year by Servetto, Vaishampayan, and Sloane (SVS) [4]. The SVS algorithm is presented for squared error distortion, and we maintain that here. Section 3 provides variations on the SVS algorithm. We observe that the SVS encoder minimizes $D_0$ and propose instead a family of encoders that trade off $D_0$ and $D_1$. Since MD techniques are only useful when both descriptions are not always received, the original SVS should basically not be used—though our improvements will not necessarily be large. Small additional improvements can be obtained by altering the decoder to use centroid reconstruction and by perturbing the lattice. Section 4 extends the SVS algorithm to more than two descriptions. The general theory is complicated, but two examples demonstrate encoding for which the decoder map depends only on the number of descriptions received—not on which specific descriptions were received.

## 2 Multiple Description Quantization

The earliest practical MD scheme is Vaishampayan's *multiple description scalar quantization (MDSQ)* [1]. In MDSQ, one quantizes a real number $x \in \mathbb{R}$ using two different scalar quantizers, with one quantizer output sent on each channel. If either channel is received by itself, the original number is known to have been within a given quantization cell of that channel. If both channels are received, the original value is known to have been in the intersection of its quantization cell in one channel and its quantization cell in the other. In this manner, the system provides coarse information to the side decoders and finer information to the central decoder.

One may alternatively view an MDSQ system as a partition of the real line along with an injective mapping between partition cells and ordered pairs of indices. We thus have discrete sets of indices $I_1$ and $I_2$ and a map $\ell : \mathbb{R} \to I_1 \times I_2$. A partition cell is then given by the set $\{x \in \mathbb{R} \mid \ell(x) = (i, j)\}$ for a given $i \in I_1$, $j \in I_2$. From this we get the induced projected mappings $\ell_1 = \pi_1(\ell) : \mathbb{R} \to I_1$ and $\ell_2 = \pi_2(\ell) : \mathbb{R} \to I_2$, which give us the individual scalar quantizers.

However, just as one can construct single description vector quantizers that improve upon the performance of scalar quantizers, one can construct multiple description vector quantizers that outperform their scalar counterparts. Analogously to MDSQ, in *multiple description vector quantization (MDVQ)* we want discrete sets of indices $I_1$ and $I_2$ along with a map $\ell : \mathbb{R}^n \to I_1 \times I_2$ (which induces the projected mappings $\ell_1 = \pi_1(\ell) : \mathbb{R}^n \to I_1$ and $\ell_2 = \pi_2(\ell) : \mathbb{R}^n \to I_2$). The partition cells are given by $\{x \in \mathbb{R}^n \mid \ell(x) = (i, j)\}$ for a given $i \in I_1$, $j \in I_2$. These cells are typically designed to be the Voronoi cells of some collection of points.

### 2.1 Multiple Description Lattice Vector Quantization

Although superior in performance to its scalar counterpart, general vector quantization is computationally expensive. Restricting codebooks to lattices simplifies the necessary calculations for encoding and decoding.

The problem now becomes that of choosing a lattice and designing a way of assigning the indices. Servetto, Vaishampayan, and Sloane (SVS) provided such an algorithm in [4]. Since our algorithms build upon the original algorithm of SVS, it is summarized here for completeness.

We will need the concept of a geometrically similar sublattice:

**Definition 1** *Let $\Lambda$ be a lattice. $\Lambda'$ is a* geometrically similar sublattice *of $\Lambda$ if the points of $\Lambda'$ are a subset of the points of $\Lambda$, and $\Lambda' = cA\Lambda$ for some scalar $c$ and some orthogonal matrix $A$ with determinant 1.*

Thus, a geometrically similar sublattice is a sublattice obtained by scaling and rotating the original lattice.

**The SVS Algorithm** [4] The SVS algorithm finds a triplet $(\Lambda, \Lambda', \ell)$, such that:

1. $\Lambda$ is a lattice;
2. $\Lambda'$ is a geometrically similar sublattice of $\Lambda$; and
3. $\ell : \Lambda \xrightarrow{\text{inj.}} \Lambda' \times \Lambda'$.

The index of the sublattice $N = |\Lambda/\Lambda'|$ controls the redundancy of the system.

We wish to label every point in the lattice with a pair of points on the similar sublattice. We then encode using the Voronoi cells of the lattice points: A point is encoded to $\lambda \in \Lambda$, and then $\pi_1(\ell(\lambda)) \in \Lambda'$ is transmitted over one channel and $\pi_2(\ell(\lambda)) \in \Lambda'$ is transmitted over the other. If one channel is received, one can decode to the sublattice. If both channels are received, one can decode to the lattice itself. This thus provides coarse information if only one channel is transmitted successfully and finer information if both channels are. To design the system, *i.e.*, to find the map $\ell$, SVS outlined the following algorithm (for more details, see [4]):

1. Choose a lattice $\Lambda$, a geometrically similar sublattice $\Lambda'$ of index $N$, and a group $\mathcal{W}$ of rotations of the lattice that map back to the lattice.
2. Define $\equiv$ such that $\lambda_1 \equiv \lambda_2$ if and only if

   $$\text{there exists} \quad \tau \in \mathcal{W} \quad \text{such that} \quad \lambda_1 - \pi_{\Lambda'}(\lambda_1) = \tau(\lambda_2 - \pi_{\Lambda'}(\lambda_2)),$$

   where $\pi_{\Lambda'}$ maps a point to its nearest sublattice neighbor. Points are equivalent under this relation if and only if they are in the same orbit of $\mathcal{W}$ relative to their nearest sublattice neighbors.
3. Define $E \subseteq \Lambda' \times \Lambda'$ by

   $$E = \{(\lambda'_1, \lambda'_2) \in \Lambda' \times \Lambda' \, : \, \|\lambda'_1 - \lambda'_2\| \leq \|c\lambda A\|\},$$

where $\Lambda' = cA\Lambda$ and $\lambda \in \Lambda'$ is a lattice point of maximal norm in the Voronoi cell of $\mathbf{0} \in \Lambda'$. The elements of $E$ will be called edges.

In other words, a valid label $(\lambda_1', \lambda_2')$—an edge—for a point $\lambda$ on the original lattice must consist of sublattice points at a certain, bounded distance from each other. This, of course, makes sense, since we do not want to code a point $p$ with sublattice points too far away from it; it would produce a huge side distortion.

4. Define $\equiv'$ such that $e_1 \equiv' e_2$ with $e_1, e_2 \in E$ if and only if they both serve as minimal vectors in the same similar sublattice of $\Lambda$.

5. Two-color the edges in $E$ so that the colors alternate along any straight line of adjacent edges.

   This step is not strictly necessary (one can randomly assign colors); however, it is in the original algorithm so we include it here. This step breaks the tie in assigning which point would get $(a, b)$ versus $(b, a)$, for example.

6. For each equivalence class of $\equiv$, select an equivalence class of $\equiv'$ to be matched with it. As there will be several ways of choosing this matching, perform a numerical optimization over the different choices to select the one that yields the optimal results.

   This is the most important step; here, one actually finds an optimal index assignment $\ell$. It consists of deciding which orbit of points (out of those in Step 2) gets associated with which class of edges (out of those in Step 4).

7. Using the group and the sublattice, extend the matching of equivalence classes to the entire lattice. Use the coloring from Step 5 to determine the order of the points in the sublattice pairs (*i.e.*, which sublattice point gets transmitted over which channel).

This algorithm facilitates the design of an effective MD vector quantizer. When used to construct a two-dimensional vector quantizer using a hexagonal lattice, the distortion product $d_0 d_1$ improves by 0.3674 dB $\approx$ 8.8% over MDSQ, an improvement per channel of slightly more than the 0.1671 dB by which the hexagonal lattice improves upon the cubic lattice $\mathbb{Z}^2$ [5]. Increasing the dimension of the quantizer further improves the performance, with the eight-dimensional $E_8$ lattice and the 24-dimensional Leech lattice yielding gains of 0.65 dB and 1.03 dB, respectively, over the $\mathbb{Z}^2$ lattice.

The modifications described in the next section provide improvements that are at times larger than these, even while maintaining the essential encoding and decoding complexity of the two-dimensional hexagonal lattice. Thus improving the encoding and decoding rules may be a lower-complexity alternative to increasing the vector dimension. Furthermore, the principles of the next section apply to an MDLVQ of the SVS type for any base lattice and sublattice.

# 3 Altering the Encoding and Decoding

The observation that motivated this work is the following: The encoding rule of the SVS algorithm uses the Voronoi cells of the original (fine) lattice. Since the decoding is to the resolution of the fine lattice only when both descriptions are received, this is inherently an optimization for the central decoder—at the expense of the side decoders. But MD coding is useless unless the side decoders are sometimes used, so it should be possible to improve on the encoding in SVS.

Suppose the criterion of interest is actually a function of the central and side distortions $f(D_0, D_1, D_2)$. Then the encoding should explicitly minimize this quantity. (The SVS criterion is $D_0$.) For concreteness, this section uses the average distortion conditioned on receiving at least one description as the performance criterion. Similar results would be obtained with other performance criteria.

Assume descriptions are lost independently with probability $\rho$. Omitting the case of receiving neither description and normalizing properly gives

$$AMSE = \frac{1-\rho}{1+\rho} D_0 + \frac{\rho}{1+\rho}(D_1 + D_2). \tag{1}$$

$AMSE$ is used to indicate that it is an average of MSE distortions.

## 3.1 Average Performance of MDLVQ

After the choice of a lattice and a sublattice, the original SVS algorithm provides a $(D_0, D_1)$ operating point by optimizing the index assignment. Fig. 4(a) of [4] provides several such points for the two-dimensional hexagonal lattice with sublattice indices ranging from 7 to 127. The source is uniformly distributed over a region much larger than the Voronoi cells of the lattice. These data can be used to compute the optimal index as a function of the parameter $p$, as shown in Fig. 1(a). For $p > 0.01849$, the optimal index is 7.[1] The optimal labeling for index 7 is shown in Fig. 1(b); this example is the basis for the rest of the examples of this section.

## 3.2 Voronoi Cells with Respect to Average MSE

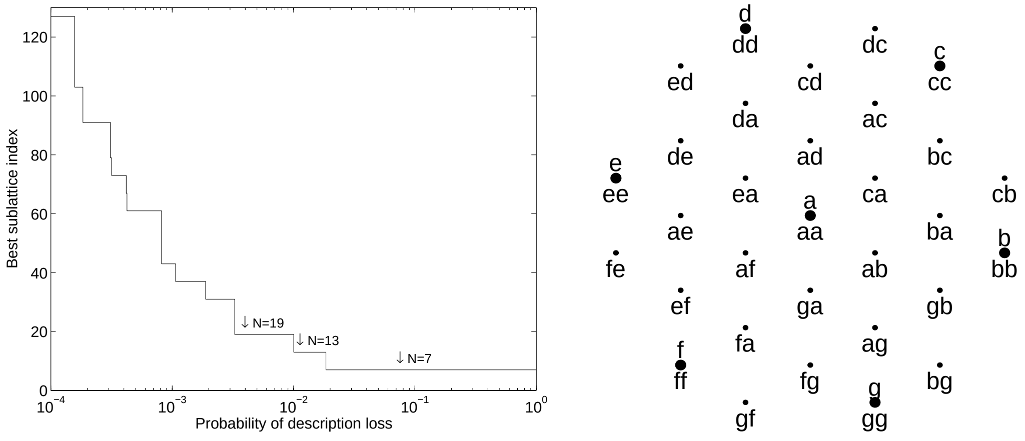
To minimize the average MSE (1), the encoder should use Voronoi cells with respect to a corresponding distance measure. We thus make the following definitions:

**Definition 2** *For $a \in \Lambda$ and $\ell(a) = (x, y)$, where $x, y \in \Lambda'$, let $\pi_1(a) = x$ and $\pi_2(a) = y$.*

**Definition 3** *For $x \in \mathbb{R}^n$ and $a \in \Lambda$, let the* multiple description distance *between $x$ and $a$ at loss parameter $p$ be*

$$\mathfrak{d}_p(x, a) \;=\; \sqrt{\tfrac{1-p}{1+p}\|x-a\|^2 + \tfrac{p}{1+p}(\|x-\pi_1(a)\|^2 + \|x-\pi_2(a)\|^2)}. \tag{2}$$

[1] Only the data from [4] is used, so the index is optimal from among the index set used there. When limited to the original SVS encoding, for sufficiently large $p$ it becomes optimal to simply repeat the data over both channels.



(a) (b)

Fig. 1: (a) The sublattice index that minimizes the $AMSE$ criterion (1) as a function of the parameter $p$ for the two-dimensional hexagonal lattice. Index 7 is optimal for $p > 0.01849$. (b) The optimal index assignment for the index-7 sublattice. Doublet labels are the transmitted indices and singlet labels are the names of the sublattice points.

**Definition 4** *The* Voronoi cell with respect to multiple description distance *of an element $a \in \Lambda$ with loss parameter $p$ is*

$$V_p(a) = \{x \in \mathbb{R}^n : \mathfrak{d}_p(x,a) \leq \mathfrak{d}_p(x,b), \forall b \in \Lambda\}.$$

Encoding using Voronoi cells with respect to multiple description distance gives a family of encoders parameterized by $p$. Although $p$ is inspired by the loss probability $\rho$, they are not necessarily equal. If they are equal, it follows immediately from the definitions that partitioning with Voronoi cells with respect to multiple description distance minimizes $AMSE$.

To test this method and determine the magnitude of the improvements, calculations were made with the MDLVQ shown in Fig. 1(b). The new Voronoi cells for several values of the loss parameter $p$ are shown in Fig. 2. Note how the shapes of the cells change as $p$ increases. When $p$ is zero, the new Voronoi cells are exactly the same as the standard Voronoi cells of $\Lambda$, since a point will be decoded to the point on the fine lattice with probability 1. As $p$ increases, certain "central" cells emerge that are larger than the others. These are the cells of points on the sublattice $\Lambda'$. Encoding to points on $\Lambda'$ is preferred because these points are decoded without error, even at the side decoders. The other cells belong to points of $\Lambda \setminus \Lambda'$. The index assignment $\ell$ maps these other points to ordered pairs of two distinct sublattice points, a closer one and a farther one. As $p$ increases, the large side distortion associated with the farther sublattice point makes encoding to this point unattractive. This effect continues to get more pronounced until the side cells disappear at $p = 1$.

Encoding with the new Voronoi cells gives a set of $(D_0, D_1)$ operating points indexed by $p$. These are shown by the top curve in Fig. 3(a). The leftmost point (circled) is the sole operating point of the SVS algorithm; the modified encoding gives

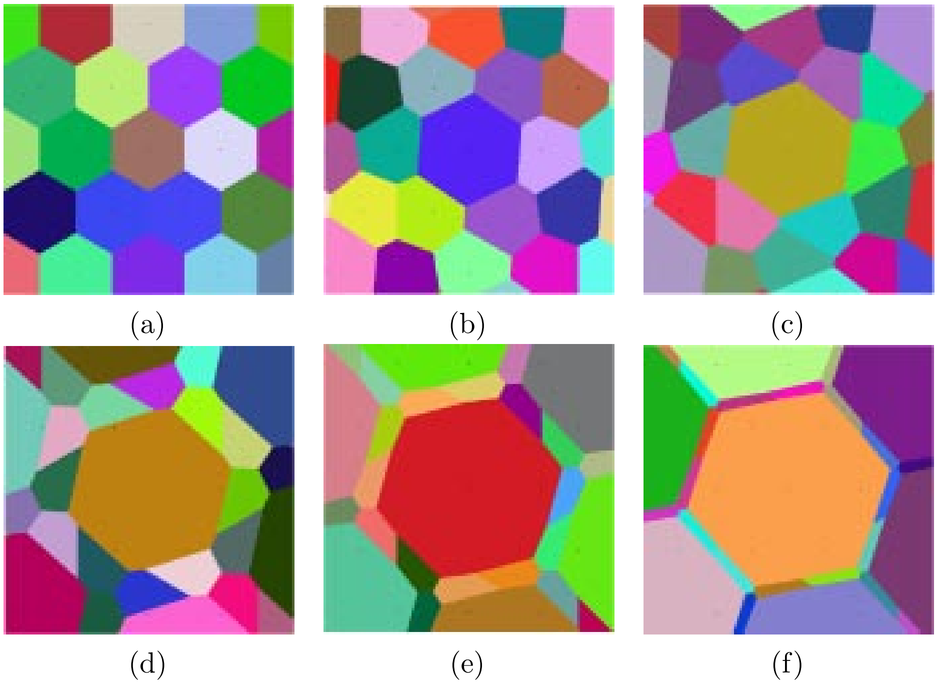


Fig. 2: Shapes of Voronoi cells with respect to multiple description distance for various loss parameters $p$: (a) 0.0, (b) 0.1, (c) 0.2, (d) 0.4, (e) 0.6, (f) 0.8.

a range of operating points. All the reported distortions are normalized such that $D_0$ with the original SVS encoding is 0 dB. The lower curve in Fig. 3(a) shows the improvement obtained by using centroid reconstruction as opposed to reconstructing to the original lattice points.

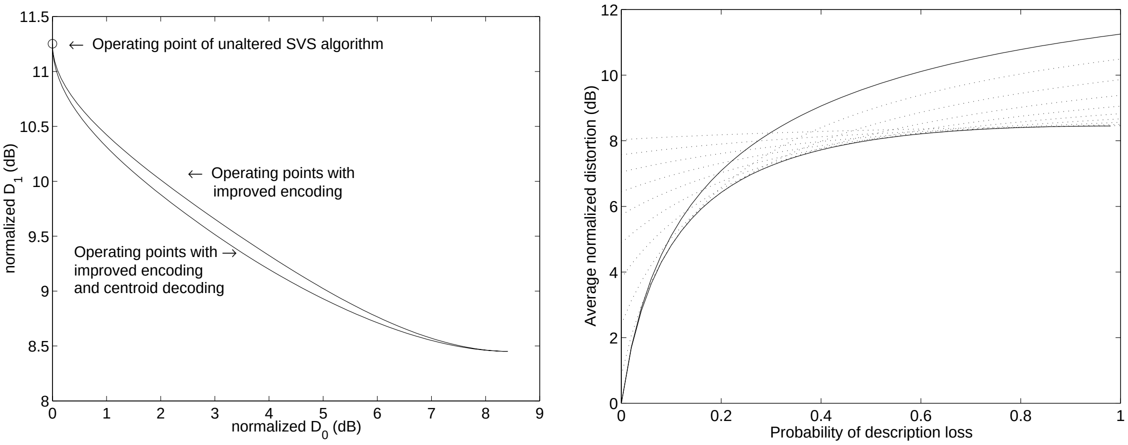
From a $(D_0, D_1)$ operating point, we can compute $AMSE$ as a function of the loss probability. Fig. 3(b) shows a variety of such performance profiles. The top solid curve corresponds to the original SVS algorithm or, equivalently, design parameter $p = 0$. The dotted curves are (from steepest to flattest) for $p = 0.1, 0.2, \ldots, 0.9$. The best performance, shown with the lower solid curve, is obtained when the probability of description loss equals the design parameter $p$. An additional improvement of up to 0.1 dB, peaking at $p \approx 0.28$, is obtained by using centroid reconstruction.

The improvement over the SVS algorithm (in dB) increases approximately linearly with the probability of description loss, leading to large improvements at high probabilities. As this method adds negligibly to the complexity of the system, these improvements come virtually for free.

### 3.3 Perturbing the Lattice

The elongated shapes of the cells associated with $\Lambda \setminus \Lambda'$, along with the fact that these cells do not even contain the corresponding central decoder points at large $p$, suggest that locations of the points can be modified to improve the performance of the system. To this end, we perturb the lattice.

In perturbing the lattice, we wish to retain a highly structured set of points. If



(a)

(b)

Fig. 3: (a) The operating point of the SVS algorithm with the index-7 sublattice of the hexagonal lattice along with the ranges of operating points obtained with our improved encoding and decoding. (The normalization is described in the text.) (b) Average distortion (1) as a function of the probability of description loss. The top solid curve is for loss parameter $p = 0$, *i.e.*, the unaltered SVS algorithm. The dotted curves are for $p = 0.1, 0.2, \ldots, 0.9$. The lower solid curve is the lower convex hull, which is obtained when the parameter $p$ matches the actual probability of description loss.

we fail to do so, the algorithmic complexity of encoding will become prohibitively large. Thus, we leave the sublattice points in their original locations. Next, we use the equivalence relation $\equiv$ defined in the SVS algorithm to partition the remaining points into equivalence classes. (Points are equivalent if they are in the same orbit of $\mathcal{W}$ relative to their nearest sublattice neighbors.) We can now move a single point in each equivalence class and then use the group to extend this perturbation to the rest of the lattice. The specific perturbations that optimize the system can be found through a numerical optimization.

This manner of perturbing the lattice retains the structure of the system and allows the perturbation to be described using a small amount of information. In addition, the asymptotic algorithmic complexity of encoding and decoding is unaffected.

Sample results of numerically optimized perturbations are shown in Fig. 4. Note that the side cells are relatively circular in shape, as desired. The improvement with respect to $AMSE$ is significant, peaking at about 0.18 dB.

## 4 More Descriptions

This section briefly describes extensions of the SVS algorithm to more than two descriptions. The algorithm presented in [4] relies heavily on the the fact that there are exactly two channels, and thus a generalization is not immediate. The algorithm here will use *iterated sublattices*, *i.e.*, an ordered set of lattices such that each lattice is a sublattice of all lattices that precede it. For $M$ descriptions, there are a total

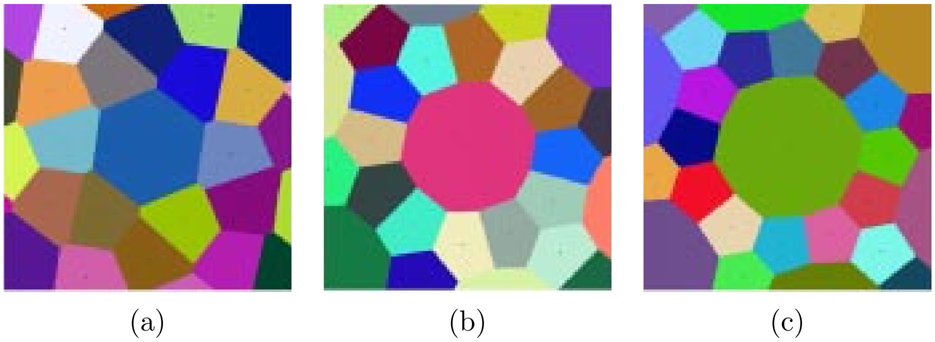


Fig. 4: Shapes of Voronoi cells with respect to multiple description distance after perturbation for various loss parameters $p$: (a) 0.1, (b) 0.2, (c) 0.4.

of $M$ lattices $\Lambda_1 \subset \Lambda_2 \subset \cdots \subset \Lambda_M$. The index assignment problem is to design a mapping $\ell : \Lambda_M \rightarrow \Lambda_1^M$.

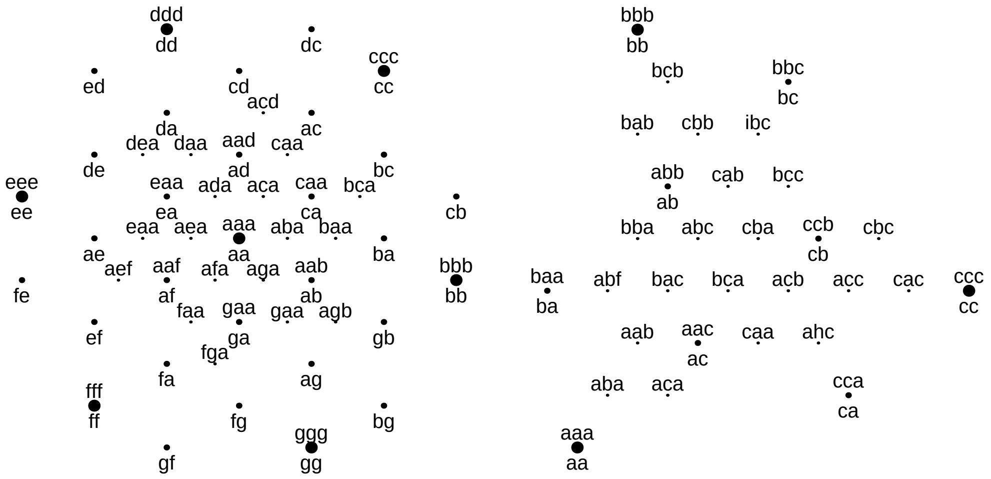
The key innovation in our construction is the requirement that for each number of descriptions received $d$, there is a (single) decoding function that maps the received vector to $\Lambda_d$. This means that we need only $M$ decoders instead of $2^M - 1$ (one for each nonempty subset of descriptions). The general theory is complicated, so we present two examples of three-description labelings to demonstrate the desired properties. These examples, shown in Fig. 5, are again based on the two-dimensional hexagonal lattice.

Consider the example of Fig. 5(a). Suppose the source vector lies close to the point labeled aba (in the Voronoi cell of that point in $\Lambda_3$). The labeling is unique, so if all three descriptions are received, the source will be reconstructed to the resolution of $\Lambda_3$. Deleting one description leaves ba, aa, or ab. (Note that we have kept the ordering of the two received labels.) These are nearby points on $\Lambda_2$, so the distortion is only a little worse than the resolution of $\Lambda_2$. Finally, if one description is received, the reconstruction is the nearest point of $\Lambda_1$ (point a) two-thirds of the time and the second-nearest point of $\Lambda_1$ (point b) one-third of the time. The reader can work out details for other points. The worst-case reconstructions are as follows: from one description, the second closest point of $\Lambda_1$ (including ties); and for two descriptions, the fourth closest point of $\Lambda_2$. The example of Fig. 5(b) is similarly designed to give good performance. The sublattice indices used are higher, so the redundancy is lower.

All of the methods mentioned in the previous section for improving encoding and decoding can be applied to systems with more than two descriptions as well. Encoding and decoding remain simple.

## 5 Conclusions

This paper presents variations on, and extensions of, the MDLVQ algorithm of Servetto, Vaishampayan and Sloane. The variations are focused on providing additional $(D_0, D_1)$ operating points. The merit of these new operating points has been established through the average MSE, a weighted average of central and side distortions. Another way to be convinced of their worth is to note that each point in



(a) (b)

Fig. 5: Examples of index assignments for three-description coding. Triplet labels apply to the finest lattice $\Lambda_3$ and are actually transmitted; doublet labels apply to the middle lattice $\Lambda_2$ and are used for reconstructing from two descriptions; the reconstruction labels for one description are omitted because they are clear from Fig. 1(b). These index assignments allow a single decoder mapping (not three) for one received description and a single decoder mapping (again, not three) for two received descriptions. (a) $|\Lambda_3/\Lambda_2| = 3$, $|\Lambda_2/\Lambda_1| = 7$; (b) $|\Lambda_3/\Lambda_2| = 7$, $|\Lambda_2/\Lambda_1| = 7$.

Fig. 4(a) of [4] can be replaced by a curve as in Fig. 3(a). This certainly decreases (improves) the lower convex hull of $(D_0, D_1)$ points.

The extension of the SVS algorithm provides a technique for more than two descriptions. A trivial generalization would necessitate $2^M - 1$ decoder mappings for $M$ descriptions; the generalization sketched here requires only $M$ decoders.

## References

[1] V. A. Vaishampayan. Design of multiple description scalar quantizers. *IEEE Trans. Inform. Th.*, 39(3):821–834, May 1993.

[2] V. K Goyal. *Beyond Traditional Transform Coding.* PhD thesis, Univ. California, Berkeley, 1998. Distributed as Univ. California, Berkeley, Electron. Res. Lab. Memo. No. UCB/ERL M99/2, Jan. 1999.

[3] M. Fleming and M. Effros. Generalized multiple description vector quantization. In *Proc. IEEE Data Compression Conf.*, pages 3–12, Snowbird, Utah, April 1999.

[4] S. D. Servetto, V. A. Vaishampayan, and N. J. A. Sloane. Multiple description lattice vector quantization. In *Proc. IEEE Data Compression Conf.*, pages 13–22, Snowbird, Utah, April 1999.

[5] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices and Groups.* Springer-Verlag, New York, third edition, 1998.