Shared Descriptions Fusion Coding for Storage and Selective Retrieval of Correlated Sources*

Sharadh Ramaswamy and Kenneth Rose
ECE Department, University of California, Santa Barbara, CA 93106
E-mail: {rsharadh,rose}@ece.ucsb.edu

Abstract

Motivated by sensor networks, we consider the fusion storage of correlated sources in a database, such that any subset of them may be efficiently retrieved in the future. Only statistical information about future queries is available during encoding and storage. Fusion coding of correlated sources poses new challenges due to the conflicting objectives of exploiting inter-source correlations and enabling efficient *selective* retrieval. Practical signal compression imposes additional constraints on system complexity. We propose a shared-descriptions approach for the design of lossy fusion coding systems, to manage the precise tradeoffs between storage rate, retrieval rate, distortion and system complexity, within one unified framework. An iterative descent algorithm is derived for the design of such fusion coders. The optimized system provides significant gains over traditional quantization techniques that are not directly optimized for fusion coding.

1 Introduction

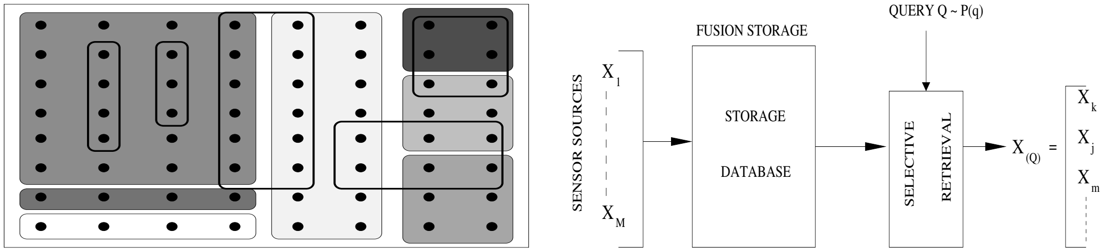
This paper considers the problem of storing correlated sources in a database for future retrieval of any subset of them as queried by users. This problem differs from the well known distributed source coding setting [1] in that all information about the sources is centrally available during encoding for storage in the database. However, only statistical information about future queries is available. Such database design introduces fundamentally new and interesting challenges: On the one hand, inter-source correlations may be exploited via joint coding to reduce the overall storage requirement and to potentially reduce the retrieval time. On the other hand, a future query may select only few of the sources for retrieval, and it would be wasteful to have to retrieve the entire (jointly) compressed data only to reconstruct a small subset.

An example application of the proposed fusion coding of correlated sources is in the arena of sensor networks, which has been the focus of extensive research in recent years.

*This work was supported in part by the NSF under grants IIS-0329267 and CCF-0728986, the University of California MICRO program, Applied Signal Technology, Inc., Cisco Systems, Inc., Dolby Laboratories, Inc. and Qualcomm, Inc.

1068-0314/08 $25.00 © 2008 IEEE
DOI 10.1109/DCC.2008.56

Much of the effort was dedicated to the development of device and communication technologies [2]. But in order to fully realize the potential of most such systems, it is necessary to efficiently store the vast volumes of data generated by the network for future retrieval, as needed for analysis or other uses. As an illustrative example, consider the installation of a dense network of sensors for monitoring purposes. A fusion center stores the signals generated by these sensors, which are expected to be highly correlated, as they cover the same scene. Data from the fusion center are eventually accessed by users, who may be interested in information from only a small subset of the sources at any given time. Figure 1 depicts the setting.



(a) A 2D sensor field: dots represent sensors and boxes represent regions of interest (queries)

(b) Fusion storage and selective retrieval

Figure 1: Fusion coding of correlated sources

In fact, fusion storage of correlated sources has applications even in areas that are remote from traditional signal processing and communications, such as storage and indexing of stock market data streams [3]. The fusion coder problem was first identified in [4], where the authors derived an information-theoretic characterization of an achievable rate region, via reformulation as a multi-terminal source coding problem [5]. More recently, lossy fusion coder design that directly optimizes the distortion-retrieval rate tradeoff for memoryless sources was derived in [6]. The performance gains of such fusion coders increase with storage (rate) capacity. However, the system complexity grows exponentially with the storage rate and, consequently, the design methods of [6] do not scale to a large number of sources. In this paper, we are concerned with the development of systems and design algorithms for fusion coding at high storage rate and for many sources (sensors). We propose a shared descriptions approach that enables constraining the system complexity which can then be traded against distortion , retrieval rate and storage. It also subsumes the fusion coder design of [6] as an extreme special case.

2 Fusion Coding Preliminaries

Let us denote the M correlated sources as the set, $\{X_m, m = 1...M\}$. A subset of sources that need to be retrieved from the database is referred to as a query. Let binary variables $q_i \in \{0, 1\}$ denote whether or not source X_i is called for by the query, i.e., queries are represented by M-tuples of the form $\mathbf{q} = (q_1, ..., q_M) \in \mathcal{Q}$, where $\mathcal{Q} \subseteq \{0,1\}^M$ is the domain-set of queries. A probability distribution is defined over all queries, $P : \mathcal{Q} \rightarrow [0,1]$, where naturally $\sum_{\mathbf{q} \in \mathcal{Q}} P(\mathbf{q}) = 1$. Without loss of generality, we assume that each source is requested with positive probability (i.e., there exists some query with positive probability that calls for this source) and that a query always asks for a non-empty subset of sources, i.e., $P(\mathbf{0}) = 0$. Boldface symbols in lowercase and uppercase represent vectors and random vectors, respectively.

The retrieval time, or the time required to retrieve a subset of sources, is proportional to the number of bits retrieved. Let the retrieval bit rate for query $\mathbf{q}$ be $R_{\mathbf{q}}$. Then, the average retrieval rate is $R_r = \sum\limits_{\mathbf{q} \in \mathcal{Q}} P(\mathbf{q}) R_{\mathbf{q}} = E[R_{\mathbf{Q}}]$.

2.1 Intuition: Lossless Storage and Retrieval

Let us consider two extreme cases, namely, minimum storage rate versus minimum retrieval rate. It follows from Shannon's basic result that given M sources $X_1, \ldots, X_M$, the minimal storage rate is $R_{s,min} = H(X_1, ..., X_M)$. In other words, joint compression is required to minimize the storage rate. When it is time to retrieve information, however, regardless of the query received, the entire (jointly) compressed description needs to be retrieved, imposing a (high) retrieval rate of $R_r = H(X_1, ..., X_M)$.

If we denote the set of sources queried as

$$X_{(\mathbf{q})} = \{X_m, \forall m : q_m = 1\},$$

the minimum conceivable number of bits required to reconstruct the sources queried by $\mathbf{q}$ is $H(X_{(\mathbf{q})})$. Hence, the minimum average retrieval rate is $R_{r,min} = \sum\limits_{\mathbf{q}} P(\mathbf{q})\ H(X_{(\mathbf{q})}) \leq H(X_1, ..., X_M)$. This implies that in order to achieve the best retrieval speed, we need to *separately* compress and store in the database each subset of sources corresponding to a potential query. However, unless M is very small or the set of queries $\mathcal{Q}$ is severely restricted, the storage requirement will quickly exceed practical limitations. In other words, the database will have to individually accommodate a combinatorially large number of queries, with storage rate $R_s = \sum\limits_{\mathbf{q} \in \mathcal{Q}} H(X_{(\mathbf{q})}) >> H(X_1, ..., X_M) = R_{s,min}$. We conclude that the optimal storage technique severely compromises retrieval speed and the optimal retrieval technique is highly wasteful in storage.

2.2 Fusion Coding for Selective Retrieval

In practice, signals are often further compressed by allowing for error or distortion in the reconstruction. A block diagram representing a lossy fusion coder is given in Figure 2. The fusion coder comprises three functional components. The encoder $\mathcal{E}$ compresses data from M sources into R_s bits at every instant. The bit (subset) selector $\mathcal{S}$ is a look-up table that indicates, for a given query $\mathbf{q}$, which of the R_s stored bits to retrieve. The bits from positions $\mathcal{S}(\mathbf{q})$ are retrieved and used by the decoder to reconstruct the relevant sources $\hat{X}_{(\mathbf{q})}$, where $\hat{X} = \mathcal{D}(\mathcal{E}(\mathbf{X}), \mathcal{S}(\mathbf{q}))$. Mathematically,

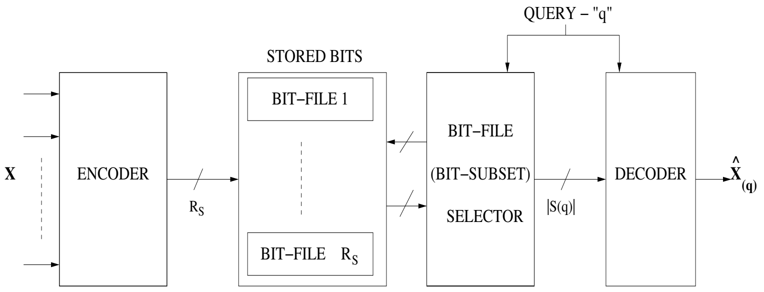


Figure 2: Fusion coder

$$\begin{aligned}\mathcal{E}:\mathcal{R}^M &\rightarrow \mathcal{I}=\{0,1\}^{R_s}\\ \mathcal{S}:\mathcal{Q} &\rightarrow \mathcal{B}=2^{\{1,\dots,R_s\}}\\ \mathcal{D}:\mathcal{I}\times\mathcal{B} &\rightarrow \hat{\mathcal{X}}\end{aligned}$$

where $\mathcal{B}$ is power set (set of all subsets) of the set $\{1,...,R_s\}$, and $\hat{\mathcal{X}}\subset\mathcal{R}^M$ is the corresponding codebook.

For query $\mathbf{q}$, the encoded (stored) bits at positions $\mathcal{S}(\mathbf{q})$ are retrieved, i.e., $R_{\mathbf{q}}=|\mathcal{S}(\mathbf{q})|$ bits (per time instant) are retrieved. The average retrieval rate R_r and distortion D are given by

$$R_r=\sum_{\mathbf{q}}P(\mathbf{q})|\mathcal{S}(\mathbf{q})|=E[|\mathcal{S}(\mathbf{Q})|] \qquad , \qquad D=E[d_{\mathbf{Q}}(\mathbf{X},\mathcal{D}(\mathcal{E}(\mathbf{X}),\mathcal{S}(\mathbf{Q})))],$$

where typically it is assumed that $d_{\mathbf{q}}(\mathbf{x},\mathbf{y})=\sum_m q_m(x_m-y_m)^2$.

2.2.1 Optimal Fusion Coder Design

Given M correlated sources and a storage constraint R_s, an optimal fusion coder is determined by the solution of

$$\min_{\mathcal{E},\mathcal{S},\mathcal{D}} J=\min_{\mathcal{E},\mathcal{S},\mathcal{D}} D(R_s)+\lambda R_r(R_s), \qquad \lambda\geq 0, \tag{1}$$

where λ is a Lagrange multiplier. In practice, expectations are often approximated by averages over available training sets. Necessary conditions for optimality are obtained by setting to zero the partial derivatives of the Lagrangian cost function. We introduce additional notation: given index (or compressed description) $\mathbf{i}\in\mathcal{I}$ and bit subset indicator $e\in\mathcal{B}$ ($e\subseteq\{1,...,R_s\}$), we use $\mathbf{i}_e$ to denote the sub-index obtained by extracting the bits at the positions indicated by e. The necessary conditions for optimality are:

- Optimal Encoder : $\mathcal{E}(\mathbf{x})=\arg\min_{\mathbf{i}\in\mathcal{I}}\sum_{\mathbf{q}}P(\mathbf{q})d_{\mathbf{q}}(\mathbf{x},\mathcal{D}(\mathbf{i},\mathcal{S}(\mathbf{q}))),\forall\mathbf{x}\in\mathcal{X}$

- Optimal Bit Selector : $\mathcal{S}(\mathbf{q})=\arg\min_{e\in\mathcal{B}}\{\frac{1}{|\mathcal{X}|}\sum_{\mathbf{x}}d_{\mathbf{q}}(\mathbf{x},\mathcal{D}(\mathcal{E}(\mathbf{x}),e))+\lambda|e|\},\forall\mathbf{q}\in\mathcal{Q}$

- Optimal Codevectors : $\hat{\mathcal{X}}(\mathbf{i},e)=\frac{1}{|F|}\sum_{\mathbf{x}\in F}\mathbf{x},\forall e\in\mathcal{B},\mathbf{i}\in\mathcal{I}$, where $F=\{\mathbf{x}:\mathcal{E}(\mathbf{x})_e=\mathbf{i}_e\}$.

The design algorithm proposed in [6] consists of iteratively optimizing the encoder, bit-selector and the codebooks.

2.3 Fusion Coder Performance and Scalability

Consider an experimental example of memoryless correlated Gaussian sources of unit variance $X_m, 1 \leq m \leq M$. The correlation between sources X_i and X_j is $\rho_{ij} = \rho^{|i-j|}$, where $-1 \leq \rho \leq 1$. This correlation model is consistent with uniform sampling of a linear sensor field [7]. Queries were assumed to be uniformly distributed over contiguous "neighborhoods" of n sensors (see Figure 3). In our experiments, $M = 50$ sources and any $n = 10$ contiguous sources are queried, which implies that $|\mathcal{Q}| = 41$. We chose $\rho = 0.8$ and generated a database of 40,000 vectors.

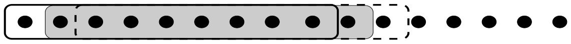


Figure 3: Neighborhood queries on a linear sensor array

The fusion coder was designed at two storage rate constraints, $R_s = 4$ and $R_s = 8$. The competing joint compression technique employed a vector quantizer (VQ), where the compression rate $R_s (= R_r)$ was varied from 1 to 8 bits per vector. Figure 4 provides the performance evaluation (ignore the "shared descriptions" results which will be discussed in the next subsection). It is clear from the figure that the fusion coder provides significant selective retrieval gains over naive joint compression (vector quantization) of sources, and these gains increase with the allowed storage rate. This is because at higher storage rates, there are more degrees of freedom in the design of the bit-subset selector.

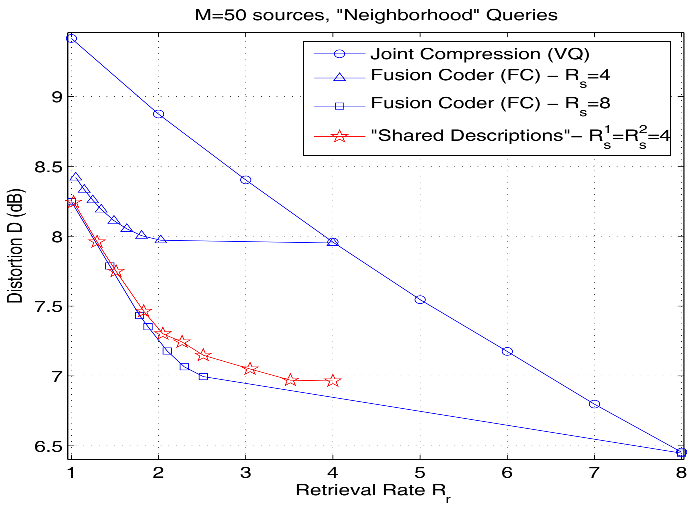


Figure 4: Performance comparison of fusion coding for selective retrieval

It is, however, of considerable practical importance to note that the overall design complexity of the optimal solution is $O(2^{R_s})$. Given storage rate R_s, the encoding operation involves searching for the best index out of 2^{R_s}, i.e. the index that minimizes the query averaged distortion. Note further that this encoding complexity also arises during operation of the fusion coder, not only during offline design. The design of the bit-subset selector searches for the best subset of R_s bits, out of $2^{R_s} - 1$ candidates. The codebook update operation and the codevector storage are also of $O(2^{R_s})$ complexity. Thus, the system complexity (design, operational and codevector storage) grows exponentially with storage rate. This implies that system design and operation scale poorly with the allowed storage rate, which itself grows with the number of sources. This represents a major practical concern.

3 The Shared Descriptions Approach

We reformulate the system and its design such that it enables explicit control of the complexity. A structure needs to be imposed to constrain the complexity in a controlled way so as to optimize tradeoff with performance. For example, in classical vector quantizer design, the split VQ structure might be preferred over full-search VQ because of its lower codevector search complexity [8]. In an analogous fashion, we "split" the storage and spread the complexity over a number of smaller (lower complexity) encoders. Since the complexity is exponential in the storage rate this entails considerable complexity gains. Each encoder now operates independently and we refer to the compressed bits produced by each encoder as a *shared description* for reasons that will shortly become obvious.

We constrain the bit-selection module to select the subset of bits for a query from one of the shared descriptions (see Figure 5). This restriction implies that each description is in fact shared by a group of queries. Since each query is mapped to a particular description and no query retrieves bits from two or more different descriptions, it follows that the different encoders can operate independently of each other. If we employ two encoders (two descriptions) which encode at rates R_s^1 and R_s^2, then the net encoding complexity is $2^{R_s^1} + 2^{R_s^2}$ rather than $2^{R_s^1+R_s^2}$.

The space of queries is now partitioned into groups of queries, one per shared description, though each query in the group may still use a different subset of bits from their shared description. The design complexity for the bit and description selection is $2^{R_s^1} + 2^{R_s^2}$. Similarly, the net codevector update and storage complexity is $2^{R_s^1} + 2^{R_s^2}$. Thus the net system complexity is $O(2^{R_s^1} + 2^{R_s^2}) << O(2^{R_s^1+R_s^2}) = O(2^{R_s})$. The performance of this "split encoder"/"shared description" formulation is presented in Figure 4. There is a small performance loss relative to the unconstrained fusion coder, of about 0.2dB. However, the "shared description" setup considerably reduces system complexity from $O(256)$ to $O(32)$. It also has significant performance advantages over joint compression.

In general, let K be the number of descriptions/encoders. The k^{th} encoder compresses the M-dimensional input vector **x** to R_s^k storage bits at each instant. The total storage would be $R_s = \sum_k R_s^k$. Correspondingly, we introduce notation

$$\mathcal{E}_k : \mathcal{R}^M \rightarrow \mathcal{I}_k = \{0,1\}^{R_s^k}, \forall k = 1, ..., K \tag{2}$$

for the K encoders.

For the k^{th} description, we have the corresponding bit-selector as

$$\mathcal{S}_k : \mathcal{Q} \rightarrow \mathcal{B}_k = 2^{\{1,...,R_s^k\}} \tag{3}$$

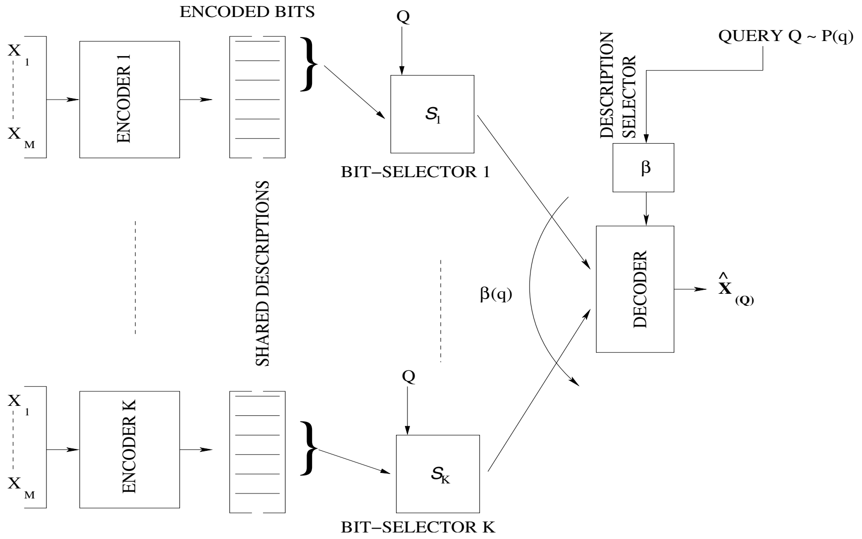


Figure 5: Shared Description Fusion Coder for Memoryless Sources

Next, we use $\beta : \mathcal{Q} \rightarrow \mathcal{K} = \{1, ..., K\}$ to denote the description selector i.e. for query $\mathbf{q}$, bits are retrieved from description $\beta(\mathbf{q})$.

The decoder is now modified to be the map

$$\mathcal{D} : \bigcup_{k=1}^{K} \mathcal{I}_k \times \mathcal{B}_k \times \{k\} \rightarrow \hat{\mathcal{X}} \tag{4}$$

Given $\beta(\mathbf{q}) = k$ for some $\mathbf{q}$, the decoder accesses the bits specified by $S_k(\mathbf{q})$ in description k and estimates $\hat{X} = \mathcal{D}(\mathcal{E}_k(X), S_k(\mathbf{q}), k)$. (Consequently, reconstruction of the relevant sources is $\hat{X}_{(\mathbf{q})}$, where we use the subscript $\mathbf{q}$ to indicate the relevant sources.) If we denote the set of queries that are mapped to the k^{th} description as $A_k = \{q : \beta(q) = k\}$, the average distortion is evaluated as

$$D = \sum_{k=1}^{K} \sum_{\mathbf{q} \in A_k} P(\mathbf{q}) \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x} \in \mathcal{X}} d_{\mathbf{q}}(\mathbf{x}, \mathcal{D}(\mathcal{E}_k(\mathbf{x}), \mathcal{S}_k(\mathbf{q}), k)) \tag{5}$$

where $\mathcal{X}$ is the training set. Likewise, the average retrieval rate is evaluated as

$$R_r = \sum_{k=1}^{K} \sum_{\mathbf{q} \in A_k} P(\mathbf{q}) |\mathcal{S}_k(\mathbf{q})| \tag{6}$$

Let $E_k = \bigcup_{A_k} S_k(\mathbf{q})$. E_k represents the bits within description k that are actually used. Clearly, $R^k_{s,util} = |\bigcup_{\mathbf{q} \in A_k} \mathcal{S}_k(\mathbf{q})| = |E_k|$ and the *true* complexity of the k^{th} encoder is $O(2^{R^k_{s,util}})$. Hence, the total storage and system complexity are evaluated to be

$$R_{s,net} = \sum_{k=1}^{K} R^k_{s,util} \qquad\qquad C_{net} = \sum_{k=1}^{K} 2^{R^k_{s,util}}$$

Given a total storage capacity R_s, allowed average retrieval rate R_{ret}, allowed system complexity C and K shared descriptions, the optimal shared description fusion coder is the solution to

$$\arg\min_{\mathcal{E},\mathcal{D},\mathcal{S}} D \ni R_{s,net} \leq R_s, C_{net} \leq C, R_r \leq R_{ret} \tag{7}$$

Equivalently, we seek solutions of

$$\arg\min_{\mathcal{E},\mathcal{D},\mathcal{S}} J = \arg\min_{\mathcal{E},\mathcal{D},\mathcal{S}} D + \lambda R_r \ni C_{net} \leq C, R_{s,net} \leq R_s \tag{8}$$

where $\lambda \geq 0$ is a Lagrange multiplier. Now, it can be clearly seen that fusion coding [6] is actually a special case of shared descriptions fusion coding (when $C = \infty$, $K = 1$).

3.1 Necessary Conditions for Optimality

The Lagrangian cost J can now be written as

$$J = \sum_{k=1}^{K} [\sum_{\mathbf{q}\in A_k} P(\mathbf{q})\{\frac{1}{|\mathcal{X}|}\sum_{\mathbf{x}\in\mathcal{X}} d_{\mathbf{q}}(\mathbf{x}, \mathcal{D}(\mathcal{E}_k(\mathbf{x}), \mathcal{S}_k(\mathbf{q}), k)) + \lambda|\mathcal{S}_k(\mathbf{q})|\}] \tag{9}$$

Optimal Encoders : Given all the other mappings, it follows from (9) that the optimal encoding index produced by encoder k for input vector $\mathbf{x}$ is

$$\mathcal{E}_k(\mathbf{x}) = \arg\min_{\mathbf{i}\in\mathcal{I}_k} \sum_{\mathbf{q}\in A_k} P(\mathbf{q}) d_{\mathbf{q}}(\mathbf{x}, \mathcal{D}(\mathbf{i}, \mathcal{S}_k(\mathbf{q}), k)), \forall\mathbf{x} \tag{10}$$

Optimal Codevectors : For $k \in \mathcal{K}, \mathbf{i} \in \mathcal{I}_k, e \in \mathcal{B}_k$, we define $F_k = \{\mathbf{x} : (\mathcal{E}_k(\mathbf{x}))_e = (\mathbf{i})_e\}$, and the optimal codevector is

$$\hat{\mathcal{X}}(\mathbf{i}, e, k) = \frac{1}{|F_k|}\sum_{\mathbf{x}\in F_k} \mathbf{x}, \forall\mathbf{i} \in \mathcal{I}_k, \forall\mathbf{e} \in \mathcal{B}_k, \forall k \in \mathcal{K} \tag{11}$$

Optimal Bit-subset Selectors : Let $\tilde{\mathcal{B}}_k = \{e \in \mathcal{B}_k : |e \bigcup E_k| + \sum_{k'\neq k} |E_{k'}| \leq R_s, 2^{|e\bigcup E_k|} + \sum_{k'\neq k} 2^{|E_{k'}|} \leq C\}$. $\tilde{\mathcal{B}}_k$ represents the valid set of bit-selections that do not violate the storage and complexity constraints. Hence, the optimal bit-subset selection, given the encoding indices and the codebook, is the rule

$$\mathcal{S}_k(\mathbf{q}) = \arg\min_{e\in\tilde{\mathcal{B}}_k} \lambda|e| + \frac{1}{|\mathcal{X}|}\sum_{\mathbf{x}} d_{\mathbf{q}}(\mathbf{x}, \mathcal{D}(\mathcal{E}_k(\mathbf{x}), e, k)), \forall\mathbf{q}, k \tag{12}$$

Optimal Description Selector : By reordering the terms of the Lagrangian, the optimal description for a particular query $\mathbf{q}$, given the encoding indices and the codebook, is

$$\beta(\mathbf{q}) = \arg\min_{k\in\mathcal{K}} \min_{e\in\tilde{\mathcal{B}}_k} \lambda|e| + \frac{1}{|\mathcal{X}|}\sum_{\mathbf{x}} d_{\mathbf{q}}(\mathbf{x}, \mathcal{D}(\mathcal{E}_k(\mathbf{x}), e, k)), \forall\mathbf{q} \tag{13}$$

If $\beta(\mathbf{q}) = k$, we update $E_k = E_k \bigcup \mathcal{S}_k(\mathbf{q})$, before optimizing for the next query. We also note that the E_k update is necessary before the bit-selector optimization for the next query.

3.2 Design Algorithm

A natural design algorithm is to iteratively enforce the optimality conditions i.e. optimize each mapping separately, while assuming that the remaining mappings are optimal (and given). There exist only finite number of partitions of a finite training set and there are only finite number of ways to partition bits and queries. As each step of the iteration is monotone non-increasing in the cost, the algorithm must converge to a locally optimal design in a finite number of iterations. However, since the cost is not convex, this simple design approach is dependent on initialization, and multiple runs with different (possibly random) initializations may be necessary to obtain a good solution.

4 Simulation Results

For the same experimental model we considered before, we performed Shared Descriptions Fusion Coding (SDFC) at a storage rate of $R_s = 24$, with $K = 3$ descriptions. The overall complexity constraint imposed was $C = 768$. The SDFC performance was compared with two naive compression techniques that scale well with storage rate - scalar quantization (at 1 bit per source) and split VQ. Since each query consists of 10 sources, scalar quantization is forced to retrieve 10 bits for every query. Split VQ is a standard scheme in speech coding for scaling VQ to high dimensions (equivalent to a large number of sources in our case) which translate into high rates. Here, we perform split VQ by splitting the $M = 50$ sources into (roughly) equal sized groups and compressing each group separately, where the total storage rate $R_s = 24$ is divided equally among all the groups. The number of groups was varied over 24, 12, 8, 6 and 4. For each such grouping of sources, we obtain a point on the retrieval rate-distortion curve.

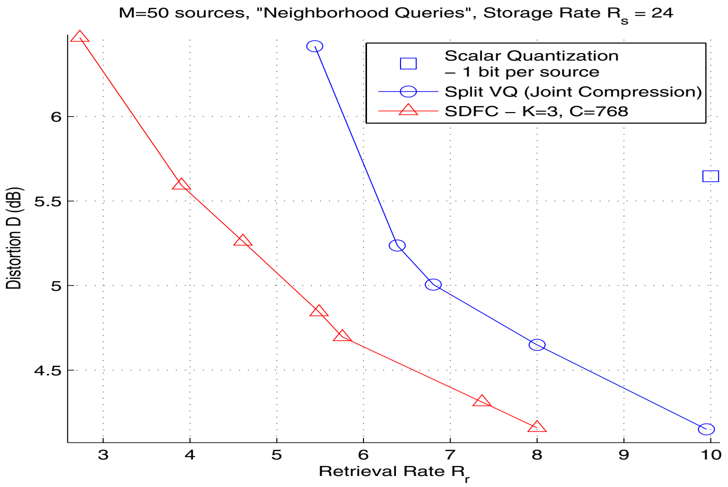


Figure 6: SDFC vs. split VQ (joint compression) vs. Scalar Quantization

From Figure 6, we note significant performance gains of SDFC over both scalar quantization and split VQ. At the same retrieval rate, SDFC offers from 0.5dB to 1.6dB decrease in distortion relative to split VQ. At the same level of distortion, SDFC provides retrieval rate reduction by factors of 1.25X to 2X over split VQ and 2.5X over scalar quantization. We also note that scalar quantization of sources (at 1 bit per source) requires more than twice the storage of SDFC and split VQ.

5 Conclusions

This paper considered the fusion coding of multiple correlated sources given only statistical prior information on queries. Scalability of the unconstrained fusion coder design is compounded by exponential growth of system complexity with storage rate. We proposed a shared-descriptions approach where multiple queries share a description, and where the different descriptions are encoded separately. Moreover, the system makes it possible to manage the precise tradeoffs between distortion, storage rate, retrieval rate and complexity within the same framework. An iterative descent algorithm for the design of shared descriptions fusion coders was derived. The gains offered by shared descriptions fusion coding over other known quantization techniques, at high storage rates, confirm the applicability and scalability of the proposed approach.

6 Acknowledgements

The authors would like to thank Dr. Jayant Nayak for his valuable suggestions on the topic.

References

[1] D. Slepian and J. Wolf, “Noiseless coding of correlated information sources,” *IEEE Trans. on Information Theory*, vol. 19, no. 4, pp. 471–480, Jul 1973.

[2] S.S. Pradhan and K. Ramchandran, “Distributed source coding using syndromes (DISCUS): Design and construction,” in *Data Compression Conference (DCC)*, 1999, pp. 158–167.

[3] X. Liu and H. Ferhatosmanoglu, “Efficient k-NN search on streaming data series,” in *SSTD*, 2003, pp. 83–101.

[4] J. Nayak, S Ramaswamy, and K. Rose, “Correlated source coding for fusion storage and selective retrieval,” in *IEEE International Symposium on Information Theory*, 2005, pp. 92–96.

[5] T. Han and K. Kobayashi, “A unified achievable rate region for a general class of multiterminal source coding systems,” *IEEE Tran. on Information Theory*, vol. 26, no. 3, pp. 277–288, May 1980.

[6] S. Ramaswamy, J. Nayak, and K. Rose, “Code design for fast selective retrieval of fusion stored sensor network/time series data,” in *ICASSP*, 2007, vol. 2, pp. 1005–1008.

[7] R. Cristescu and M. Vetterli, “On the optimal density for real-time data gathering of spatio-temporal processes in sensor networks,” in *IPSN*, 2005, pp. 159–164.

[8] A. Gersho and R. M. Gray, “Vector quantization and signal compression,” 1992, Kluwer Academic Publishers.