PREDICTIVE FUSION CODING OF SPATIO-TEMPORALLY CORRELATED SOURCES

Sharad Ramaswamy and Kenneth Rose

ECE Department, University of California, Santa Barbara, CA 93106, USA.

{rsharadh,rose}@ece.ucsb.edu

ABSTRACT

This paper considers the problem of predictive fusion coding for storage of multiple spatio-temporally correlated sources so as to enable efficient selective retrieval of data from subsets of sources as designated by future queries. Only statistical information about future queries is available during encoding. While temporal correlations can be exploited by coding over large blocks, the growth in encoding complexity renders this approach impractical and hence the interest in a low complexity predictive coding approach. However, the design of optimal predictive fusion coding systems is considerably complicated by the presence of the prediction loop, and the potentially exponential growth of the query sets. We propose a complexity-constrained predictive fusion coder and derive an iterative algorithm for its design, which is based on the "Asymptotic Closed Loop" framework and hence, circumvents convergence and stability issues of traditional predictive quantizer design. The proposed predictive fusion coder optimizes the distortion - retrieval rate tradeoff, given a fixed storage capacity, and provides significant gains over storage schemes that perform only joint compression or memoryless fusion coding of all sources.

***Index Terms*—** Multisensor systems, Database query processing, Linear predictive coding, Vector quantization, Data compression

1. INTRODUCTION

We are motivated by the problem of data storage for sensor networks. As an illustrative example, suppose we consider a dense network of sensors installed for surveillance/monitoring. The signals generated by these sensors are expected to be highly correlated, since they are covering the same scene. This data is sent to a fusion center to be stored for possible future analysis, possibly using efficient distributed source codes (see [1], [2]). For the sake of clarity of explanation, we assume that all sources are stored in a single fusion center.

We would like to emphasize here that while we are motivated by sensor networks, the problem generalizes to any collection of correlated sources. When the data from the fusion center is eventually accessed by a user, it is very likely that the information from only a small subset, and not all, sources will be requested at any given time. An interesting tradeoff emerges between conflicting objectives: On the one hand, the inter-source spatial and temporal correlation may be exploited via joint (predictive) coding to minimize the overall storage requirement and to minimize the retrieval bit rate (and hence time) required for retrieving highly correlated data. On the other hand, the specific nature of the query may result in selecting only very few of the sources to be reconstructed, and it would be wasteful to have to retrieve the entire compressed data only to reconstruct a small subset. Thus, the central issue in fusion coding for storage is, *given a fixed storage capacity and a query distribution, how does one minimize the retrieval rate*. In practice, where lossy compression would be inevitable, one seeks to minimize the retrieval rate given an additional constraint on the allowable distortion. Figure 1 is representative of the situation.

The work is supported in part by the NSF under grant II-0329267, the University of California MICRO program, Mindspeed Technologies Inc., Applied Signal Technology, Inc., Dolby Laboratories Inc., and Qualcomm Inc.

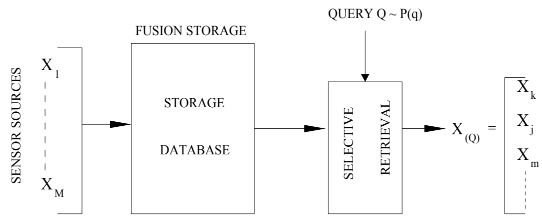

Fig. 1. Fusion Storage vs. Selective Retrieval

The problem was first identified in [3], where the authors also provided information-theoretic (asymptotic) analysis to determine an achievable rate region for lossless storage and reconstruction. More recently, fusion coders that exploit inter-source correlation, adapt to the query distribution and directly optimize the distortion retrieval rate trade-off for i.i.d (memoryless) sources, were proposed in [4]. However, streaming data such as video/sensor streams exhibit significant temporal correlations and efficient storage and retrieval would have to exploit these correlations as well. But, as we shall show in subsequent sections, the design of optimal predictive fusion coders is compounded both by the presence of the prediction loop and the need to accommodate a (possibly *exponentially*) large query set. We propose a complexity constrained approach, that still yields huge gains over naive joint compression (vector quantization (VQ) or predictive VQ) of all sources and memoryless fusion coding.

2. FUSION CODING PRELIMINARIES

Let us denote the M correlated sources as the set, $\{X_m, m = 1...M\}$. We define a query as the subset of sources that need to be retrieved. Employing binary variables $q_i \in \{0, 1\}$ to denote whether source X_i is requested or not, we represent queries by M-tuples of the form

$$\mathbf{q} = (q_1, ..., q_M) \in \mathcal{Q} \tag{1}$$

where $\mathcal{Q} \subseteq \{0, 1\}^M$ represents the domain-set of queries. We next introduce notation for the query distribution, or the probability mass function (pmf),

$$P : \mathcal{Q} \rightarrow [0, 1] \tag{2}$$

It is to be noted that there are conceivably 2^M possible queries and $\sum_{\mathbf{q} \in \mathcal{Q}} P(\mathbf{q}) = 1$. Without loss of generality, we assume that each source is requested with positive probability (i.e., there exists some query with positive probability whose requested subset includes the source) and that a query always asks for a non-empty subset of sources, i.e., $P(\mathbf{0}) = 0$. It is to be noted that in our notation, bold-face letters in lowercase and upper case represent vectors and random vectors, respectively.

Given a database of constant size, the *retrieval time* or the time required to retrieve a subset of sources is proportional to the number of bits retrieved *per sample*, which we term the *retrieval rate*. Let the number of bits per sample answer retrieved for query **q** be $R_{\mathbf{q}}$. Then, the average retrieval rate is

$$R_r = \sum_{\mathbf{q} \in \mathcal{Q}} P(\mathbf{q}) R_{\mathbf{q}} = E_{\mathbf{Q}}[R_{\mathbf{Q}}] \tag{3}$$

2.1. Lossless Storage and Retrieval

From Shannon's basic result, it follows that the minimum number of bits required to store M sources $X_1, \ldots, X_M$, i.e. the minimal *storage rate* is $R_{s,min} = H(X_1, ..., X_M)$ and hence joint compression is optimal in minimizing storage rate. However, for any query, *the entire compressed description needs to be retrieved* imposing a (high) retrieval rate $R_r = H(X_1, ..., X_M)$.

If we denote the set of sources queried as,

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

the minimum number of bits required to reconstruct the sources requested in query **q** is $H(X_{(\mathbf{q})})$ and hence, the minimum average retrieval rate is $R_{r,min} = \sum_{\mathbf{q}} P(\mathbf{q})\ H(X_{(\mathbf{q})}) \leq H(X_1, ..., X_M)$ (for any query distribution). This implies that in order to have the fastest retrieval speed, we need to *compress and store each subset* of sources that may be requested, *separately*.

However, unless M is very small or the set of queries $\mathcal{Q}$ is severely restricted, the storage requirement would be impractically high as it would have to individually accommodate a very large (possibly an *exponential*) number of queries, i.e. $R_s = \sum_{\mathbf{q} \in \mathcal{Q}} H(X_{(\mathbf{q})}) >> H(X_1, ..., X_M) = R_{s,min}$. Clearly the optimum storage technique is wasteful in retrieval speed and the optimal retrieval technique is wasteful in storage.

2.2. Practical Fusion Coding for Memoryless Sources

In practice, compression of signals is impossible without allowing for error or distortion in the reconstruction. A block diagram, representative of a practical fusion coder, is given in Figure 2. Fusion coders (without memory) are composed of an ***encoder*** $\mathcal{E}$, that compresses data from M sources into R_s bits at every instant, a query specific ***bit-subset selector*** $\mathcal{S}$, which is a look-up table to decide which set of bits to retrieve for a given query and the ***decoder*** $\mathcal{D}$, which reconstructs the sources. Mathematically,

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

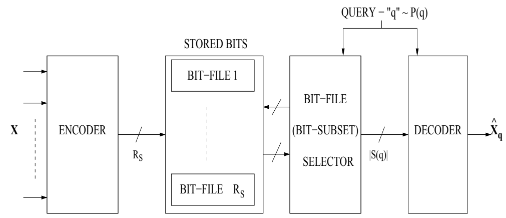

Fig. 2. Fusion Coder for Memoryless Sources

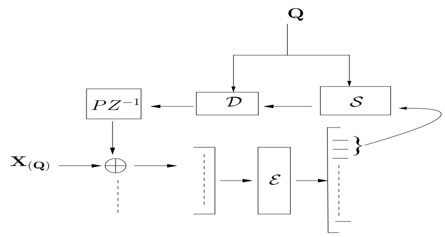

Fig. 3. Optimal Predictive Fusion Coding: Encoder

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. It is to be noted that $\mathcal{S}(\mathbf{q}) \subseteq \{1, \ldots, R_s\}$, $\forall \mathbf{q}$ and $\mathcal{D}(\mathcal{E}(\mathbf{x}), \mathcal{S}(\mathbf{q}))$ denotes the reconstruction of input **x** for query **q**, using the bits at locations indicated by $\mathcal{S}(\mathbf{q})$.

The distortion D is averaged across both the source and query distributions i.e.

$$D = E_{\mathbf{X},\mathbf{Q}}[d_{\mathbf{Q}}(\mathbf{X}, \mathcal{D}(\mathcal{E}(\mathbf{X}), \mathcal{S}(\mathbf{Q})))] \tag{4}$$

where typically it is assumed that $d_{\mathbf{q}}(\mathbf{x}, \mathbf{y}) = \sum_m q_m(x_m - y_m)^2$. For query **q**, the encoded (stored) bits at locations indicated by $\mathcal{S}(\mathbf{q})$ are retrieved i.e. $R_{\mathbf{q}} = |\mathcal{S}(\mathbf{q})|$ bits per sample are retrieved, where $|A|$ denotes the cardinality of set A. Then, the average retrieval rate is

$$R_r = \sum_{\mathbf{q}} P(\mathbf{q}) |\mathcal{S}(\mathbf{q})| = E_{\mathbf{Q}}[|\mathcal{S}(\mathbf{Q})|] \tag{5}$$

Given M correlated sources and a storage constraint R_s optimal fusion coders are solutions of

$$\min_{\mathcal{E},\mathcal{S},\mathcal{D}} J = D(R_s) + \lambda R_r(R_s), \lambda \geq 0 \tag{6}$$

where λ is a Lagrange multiplier. In practice, since the database designer has access to only training sets, and not actual distributions, expectations are replaced by averages over the training sets. The design algorithm proposed in [4] consists of iteratively optimizing the encoding, bit-selection and the codebooks, till convergence.

3. OPTIMAL PREDICTIVE FUSION CODING

While time-correlations could conceivably exploited by fusion coding over larger blocks, this would be a high complexity approach. Linear predictive coding, on the other hand, is attractive as a low-complexity alternative that can yet exploit temporal correlations. We first note that since all the sources X_m are available at the encoder, they can be equivalently replaced by query-specific super sources i.e. vector sources of the form $\mathbf{X}_{(\mathbf{q})} = [..., X_m, ...]^T, \forall m \ni q_m = 1$.

Now in optimal (linear) predictive fusion coding, at every instant, each such super source $\mathbf{X}_{(\mathbf{q})}$ is predicted and all the prediction residuals are fusion coded to exploit spatial correlations efficiently (see Figure 3). However, this imposes on the encoder the need to accommodate the entire (possibly exponentially large) query set. This would require $|\mathcal{Q}|$ prediction loops and the encoder would need to handle input vectors whose size $\sum_{\mathbf{q}\in\mathcal{Q}} |\mathbf{q}|$ grows with the query set $\mathcal{Q}$ and hence, could be of impractically high complexity.

4. CONSTRAINED PREDICTIVE FUSION CODING

Hence, we propose predictive fusion coding with constraints i.e. where queries and sources share predictors. More specifically, we shall consider the situation where all queries *share a single predictor*. We assume one-step prediction and that the prediction matrix P is estimated from open-loop statistics. Figure 4 represents our predictive fusion coding encoder. We continue with the notations developed in section 2.2 and introduce two new mappings.

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

At each instant, the predictor returns an estimate of all sources, based on a subset $\mathcal{S}_P$ of the immediately past encoding bits. The prediction error is fed as input to the encoder $\mathcal{E}$, where the error residuals are compressed to the best possible index (set of encoding bits) $\mathcal{E}(\mathbf{e})$, that are then stored in the database. The compressed error residual $\hat{\mathbf{e}} = \mathcal{D}_P(\mathcal{E}(\mathbf{e}), \mathcal{S}_P)$ is fed as input to the prediction filter.

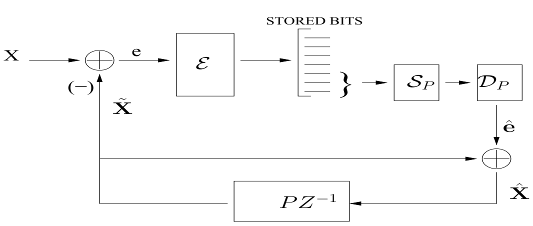

Fig. 4. Constrained Predictive Fusion Coding: Encoder

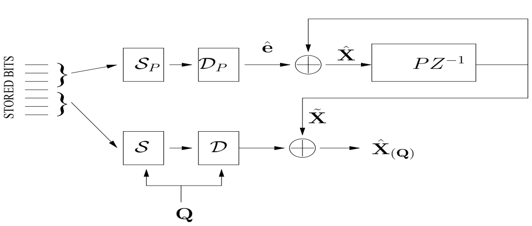

Fig. 5. Constrained Predictive Fusion Coding: Decoder

During query processing, the process is reversed. For all queries, the stored bits from the locations given by $\mathcal{S}_P$ are extracted and the reconstructed residual $\hat{\mathbf{e}}$ is fed into the prediction filter. For each query $\mathbf{q}$, additional bits are extracted (if necessary) and the output of the prediction filter $\tilde{\mathbf{X}}$ is augmented by $\hat{\mathbf{e}}_\mathbf{Q} = \mathcal{D}(\mathcal{E}(\mathbf{x}), \mathcal{S}(\mathbf{q}))$, to reconstruct the queried-sources $\hat{\mathbf{X}}_{(\mathbf{Q})}$. At this point, we note that the retrieval rate per sample for query $\mathbf{q}$ is $R_\mathbf{q} = |\mathcal{S}(\mathbf{q}) \cup \mathcal{S}_P|$ bits.

5. DESIGN BY ASYMPTOTIC CLOSED LOOP

The design of predictive coding systems is complicated by the feedback loop of the prediction filter [5]. As a result, quantizer design in open loop, while stable, is sub-optimal, while closed-loop design is inherently unstable (except at high rates), as the training set of vectors changes within every iteration. The asymptotic closed loop design principle [6], however, designs the system *always in open loop*, using the same training set within each iteration to design all components and hence, is stable. However, from one iteration to the next, the training set of error residuals is gradually modified till *asymptotically* they match the true residuals generated by the prediction loop. In other words, asymptotically the prediction loop is *closed*. We extend the asymptotic closed loop (ACL) design principle towards the design of predictive fusion coding systems in the following manner.

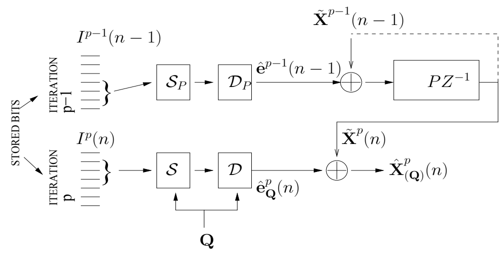

Fig. 6. Decoder: Design by Asymptotic Closed Loop

In subsequent discussion, the superscript p denotes the iteration number, while n denotes time. If $I \in \{0,1\}^{R_s}$ is a typical encoding index and $f \subset \{1,...,R_s\}$, we use the notation I_f to denote the sub-index formed by the bits in I at the locations in f. The error residuals at iteration p, $\{\mathbf{e}^p(n)\}$ are encoded to the indexes (stored vectors of bits) $\{I^p(n)\}$. $\{\mathbf{e}^p(n)\}$ form the training set at iteration p and are computed as

$$\begin{aligned} \mathbf{e}^p(n) &= \mathbf{x}(n) - \tilde{\mathbf{x}}^p(n) = \mathbf{x}(n) - P\hat{\mathbf{x}}^{p-1}(n-1) \\ \Rightarrow \mathbf{e}^p(n) &= \mathbf{x}(n) - P[\hat{\mathbf{e}}^{p-1}(n-1) + \tilde{\mathbf{x}}^{p-1}(n-1)] \end{aligned}$$

The ACL predictive fusion decoder is shown in Figure 6. Note that $\hat{\mathbf{x}}^p_\mathbf{q}(n) = \tilde{\mathbf{x}}^p(n) + \hat{\mathbf{e}}^p_\mathbf{q}(n)$, where $\hat{\mathbf{e}}^p(n) = \mathcal{D}_P(I^p(n), \mathcal{S}_P)$ and $\hat{\mathbf{e}}^p_\mathbf{q}(n) = \mathcal{D}(I^p(n), \mathcal{S}(\mathbf{q}))$. During iteration $p-1$, we seek to minimize the cost at iteration p. Asymptotically this does not matter, but this subterfuge is useful in obtaining effective update rules for $\mathcal{S}_P$ and $\mathcal{D}_P$, since both $\hat{\mathbf{e}}^{p-1}(n-1)$ and $\hat{\mathbf{e}}^p_\mathbf{q}(n)$ affect $\hat{\mathbf{x}}^p_{(\mathbf{q})}$ at time n. The design algorithm that minimizes $J = D + \lambda R_r$ cost (for a given Lagrange multiplier λ) is presented in the following sub-section.

5.1. ACL algorithm for Predictive Fusion Coder Design

1. Initialize (e.g. randomly) all query-codebooks and predictor-codebooks, bit-selectors $\mathcal{S}_P$, $\mathcal{S}$, design prediction matrix P, set $p = 0$
2. Increment p
3. Compute the new training set of residuals $\mathbf{e}^p(n)$
4. Update **encoding indexes** : $\forall n$
   $I^p(n) = \mathcal{E}(\mathbf{e}^p(n)) = \arg\min_{I\in\mathcal{I}} E_\mathbf{Q}[d_\mathbf{Q}(\mathbf{e}^p(n), \mathcal{D}(I, \mathcal{S}(\mathbf{Q})))]$
5. Update **query bit-subset selection** : $\forall \mathbf{q} \in \mathcal{Q}$
   $\mathcal{S}(\mathbf{q}) = \arg\min_{f\in\mathcal{B}} \lambda|f \cup S_P| + \frac{1}{N}\sum_n d_\mathbf{q}(\mathbf{e}^p(n), \mathcal{D}(I^p(n), f))$
6. Update **query-codebooks** : $\forall I \in \mathcal{I}, f \in \mathcal{B}$
   $\hat{\mathcal{X}}(I, f) = \frac{1}{|F|} \sum_{n:n\in F} \mathbf{e}^p(n)$, where $F = \{n : I^p_f(n) = I_f\}$

7. Update **predictor bit-subset selection** :
$\mathcal{S}_P = \arg\min_{f\in\mathcal{B}} E_{\mathbf{Q}}[\lambda|\mathcal{S}(\mathbf{q})\cup f| + \sum_n \frac{1}{N} d_{\mathbf{Q}}(\tilde{\mathbf{e}}(n), \hat{\mathbf{e}}^p_{\mathbf{Q}}(n))]$,
where $\tilde{\mathbf{e}}(n) = \mathbf{x}(n) - P[\mathcal{D}_P(I^{p-1}(n-1)), f) + \tilde{\mathbf{x}}^{p-1}(n-1)]$ and $\hat{\mathbf{e}}^p_{\mathbf{Q}}(n) = \mathcal{D}(I^p(n)), \mathcal{S}(\mathbf{Q}))$

8. Update **predictor codebooks** : $\forall I \in \mathcal{I}, f \in \mathcal{B}$
$\hat{\mathcal{X}}_P(I, f) = \arg\min_{\phi} E_{\mathbf{Q}}[\sum_{n\in G} \frac{1}{N} d_{\mathbf{Q}}(\tilde{\mathbf{e}}_\phi(n), \hat{\mathbf{e}}^p_{\mathbf{Q}}(n))]$,
where $G = \{n : I_f^{p-1}(n-1)) = I_f\}$ and
$\tilde{\mathbf{e}}_\phi(n) = \mathbf{x}(n) - P[\phi + \tilde{\mathbf{x}}^{p-1}(n-1)]$

9. Close the loop and evaluate the Lagrangian $J^p = D^p + \lambda R_r^p$. If $\frac{|J^{p-1}-J^p|}{J^{p-1}} < \epsilon$, STOP. Else go to step 2.

6. SIMULATION RESULTS

We used the first order Gauss-Markov source model for our simulations i.e. $X_m(n) = \beta_m X_m(n-1) + W_m(n)$, where $\{W_m(n)\}_1^M$ are i.i.d, zero-mean, unit variance jointly Gaussian random variables with the pairwise correlation coefficient $\rho_{jk} = E[W_j(n), W_k(n)] = \rho^{|j-k|}$. In all our simulations, $\beta_m = 0.8, \forall m$ and $\rho = 0.95$. This model would be representative of a linear sensor array. Our query distribution was a uniform distribution over contiguous "neighborhoods" of n sensors (see Figure 7). In our experiments, $M = 100$ sources and any $n = 10$ contiguous sources are queried, which implies that $|\mathcal{Q}| = 91$ and $\sum_{\mathbf{q}} |\mathbf{q}| = 910 >> M = 100$. In all our experiments, the maximum allowed storage capacity was $R_s = 6$ bits per sample and a training set of length 3000 samples was used.

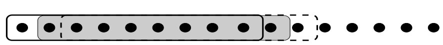

Fig. 7. Neighborhood Queries on a Linear Sensor Array

We compared the final closed-loop performance of the predictive fusion coder, with three competing techniques - (memoryless) vector quantizer (VQ) and predictive VQ (which compress all sources jointly i.e. they are joint compression techniques) and (memoryless) fusion coder (FC). The joint compression (here, VQ) techniques are compelled to retrieve all the compressed data i.e. $R_r = R_s$ and can reduce retrieval rate (time) only by reducing the compression (storage) rate.

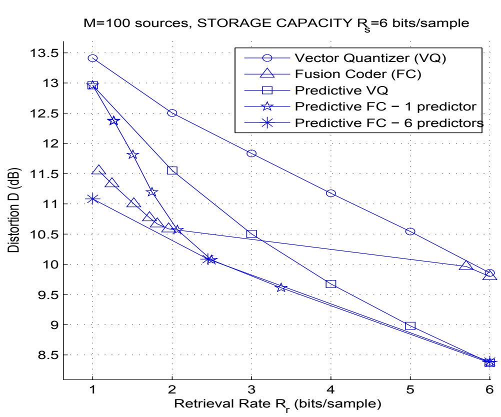

Fig. 8. Joint Compression (VQ) versus Fusion Coding

We note significant gains of the predictive fusion coding over the three competing methods. We obtain distortion gains about 2 dB over VQ, 1.5 DB over the FC and 1 dB over predictive VQ at $R_r = 2$ bits per sample.The bigger gains over VQ and FC are possible because time-correlations were exploited by the predictive FC. A nearly 1.5X reduction in retrieval rate was achieved over the predictive vector quantizer, given the same distortion of $D = 10.5$ dB. In the very low rate region, we note that the FC has a slightly better performance than predictive FC. Since we constrained the predictive FC to have only prediction loop, in the low rate region $|\mathcal{S}_P \cup \mathcal{S}(\mathbf{q})| = |\mathcal{S}_P| = 1, \forall \mathbf{q}$. As we allow more prediction loops, there would be greater freedom in designing the query bit-selector $\mathcal{S}(\mathbf{q})$, and the performance gets better. We show a performance plot with 6 prediction loops, but the description of the algorithm (for multiple predictors) is beyond the scope of this paper.

7. CONCLUSIONS

We have proposed a complexity constrained approach toward predictive fusion coding or predictive coding for storage of many correlated sources in a database. The design of optimal predictive fusion coders is compounded by the need to accommodate the entire query set and the prediction loop. In our approach, queries and sources share a single predictor, thereby reducing complexity of encoding, and the stability issues of design are circumvented by an "Asymptotic Closed Loop" algorithm. By exploiting spatio-temporal correlations efficiently, our predictive fusion coder provides significant gains over competing techniques i.e. joint compression (VQ) techniques and memoryless fusion coding. However, the cost surface is riddled with local optima and while the design complexity is $O(2^{R_s})$. In future work, we shall study intelligent initialization heuristics along with multiple predictors and complexity constrained designs.

8. ACKNOWLEDGEMENTS

The authors would like to thank Ankur Saxena for his valuable inputs regarding the asymptotic closed loop design principle.

9. REFERENCES

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

[2] A. Saxena and K. Rose, "Distributed predictive coding for spatio-temporally correlated sources," in *IEEE International Symposium on Information Theory (ISIT)*, 2007, pp. 1506–1510.

[3] 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.

[4] S. Ramaswamy, J. Nayak, and K. Rose, "Code design for fast selective retrieval of fusion stored sensor network/time series data," in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2007, vol. 2, pp. 1005–1008.

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

[6] H. Khalil, K. Rose, and S. L. Regunathan, "The asymptotic closed-loop approach to predictive vector quantizer design with application in video coding," *IEEE Trans. on Image Processing*, vol. 10, no. 1, pp. 15–23, Jan 2001.