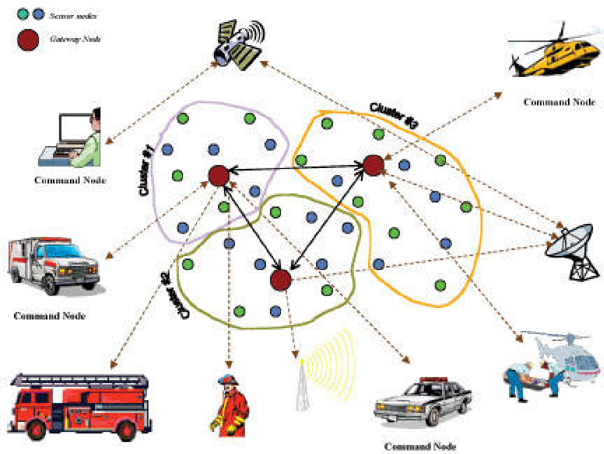
Code Design for Fast Selective Retrieval of Fusion Stored Sensor Network/Time-Series Data

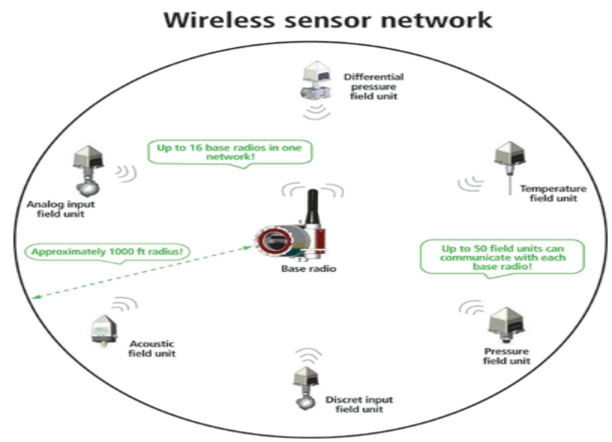
Sharadh Ramaswamy, Jayanth Nayak and Kenneth Rose

Monitoring with Sensor N/Ws

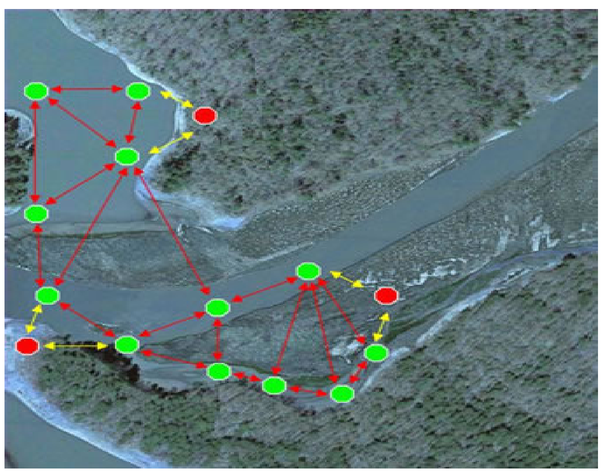
http://faculty.cua.edu/elsharkawy/images/cluster.jpg



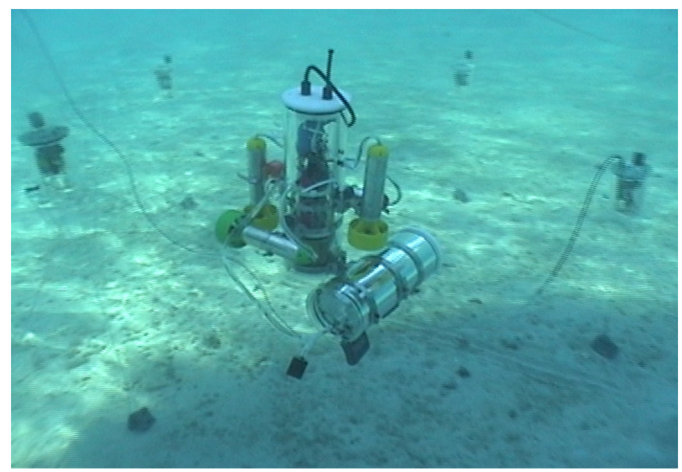
http://www.isa.org/Images/InTech/2005/July/2005070553-3.gif



http://gis.clemson.edu/cpost/images/sensor_network_small.jpg



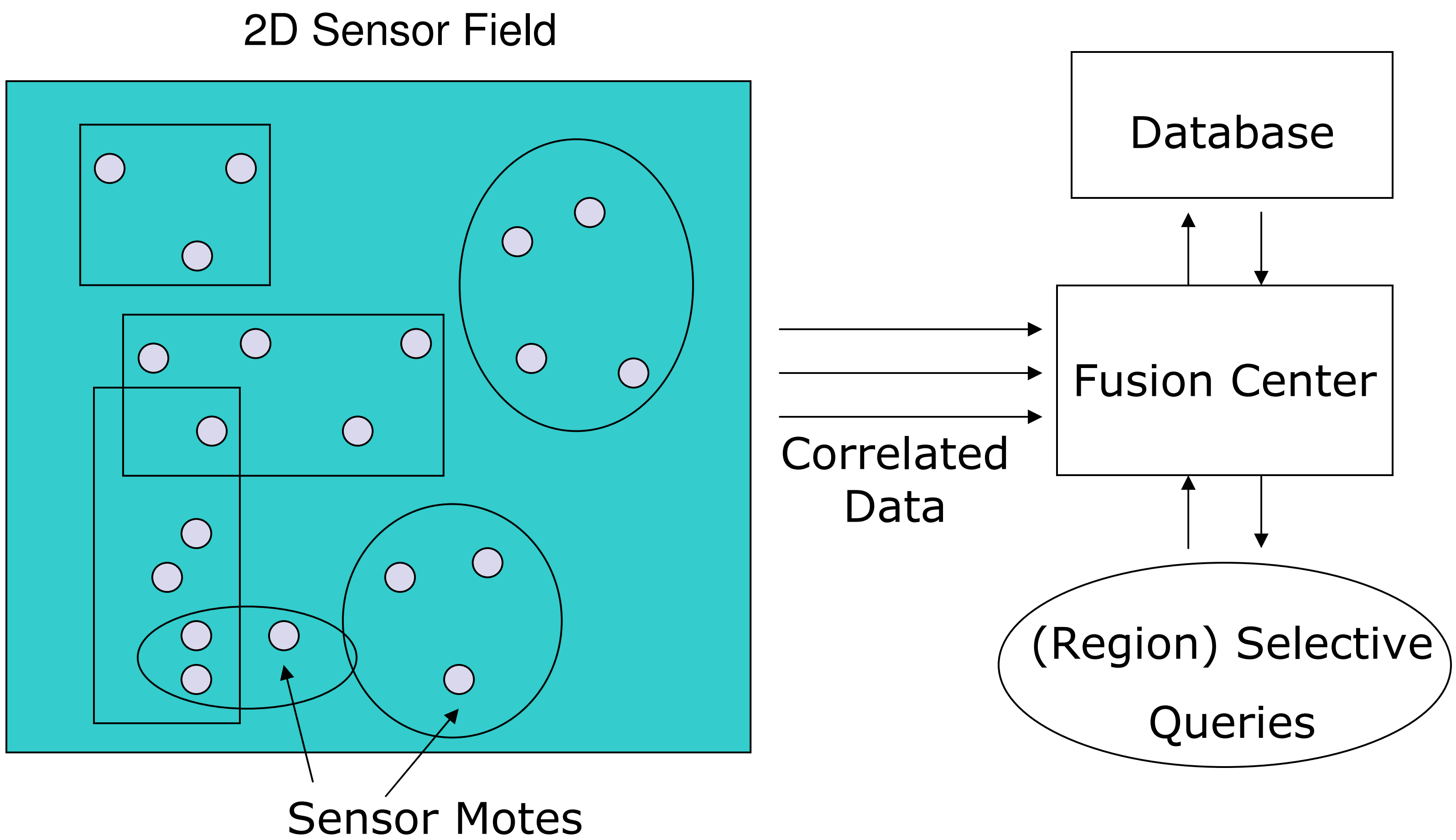
http://groups.csail.mit.edu/drl/wiki/images/d/db/amour_with_sensors.jpg



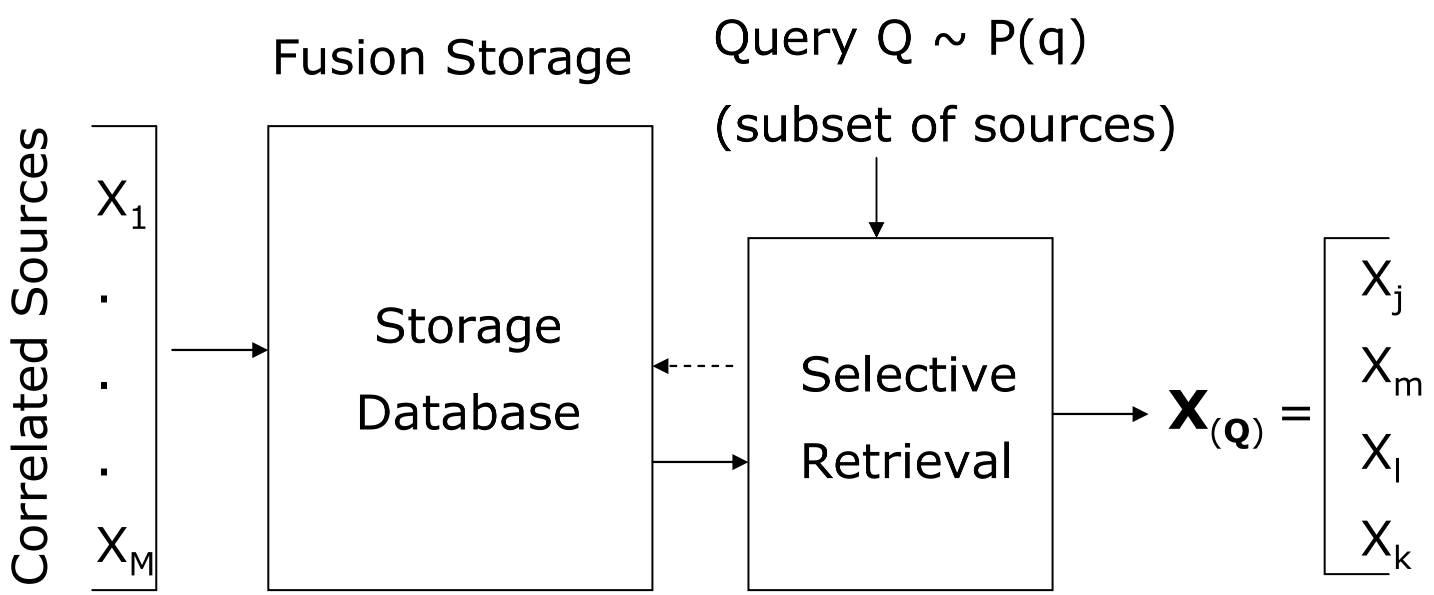
Motivation

- Dense sensor N/Ws in many apps. (monitoring/tracking/surveillance)
- Correlated measurements are collected and stored
- Distributed compression for efficient data collection
- Optimal storage strategy?

Typical User/Query Model



Fusion Storage vs. Selective Retrieval



Storage Rate R_s vs. Retrieval Rate R_r

Fusion Storage vs. Selective Retrieval

- Fundamentally a *storage* problem
- Challenge: exploit correlations to
  - min. storage rate
  - min. retrieval rate
- Conflicting objectives ?!
- Trade-off characterized in prior work (Nayak et. al. ISIT'05)

Naive Storage Strategy I

- **Jointly compress all sources**
  - optimal in exploiting inter-source corr.
  - minimizes storage rate R_s

    Lossless Coding $\Rightarrow R_{s,min}=H(X_1,\ldots,X_M)$

  - retrieves all stored data even for a small subset
  - high retrieval rate/time!! $R_r=R_s$

Naive Storage Strategy II

- **Compress and store every subset of sources separately**
  - retrieves only minimum info. reqd.
  - optimal in retrieval rate/time

$$\text{Lossless} \Rightarrow R_{r,min} = \Sigma_{\mathbf{q}}\, P(\mathbf{q})H(\mathbf{X}_{(\mathbf{q})})$$

  - reqd. storage grows with size of query set
  - (*combinatorially*) high storage rate!!

$$\text{Lossless} \Rightarrow R_s = \Sigma_{\mathbf{q}}\, H(\mathbf{X}_{(\mathbf{q})}) >> H(X_1,\ldots,X_M)$$

Storage with Distortion

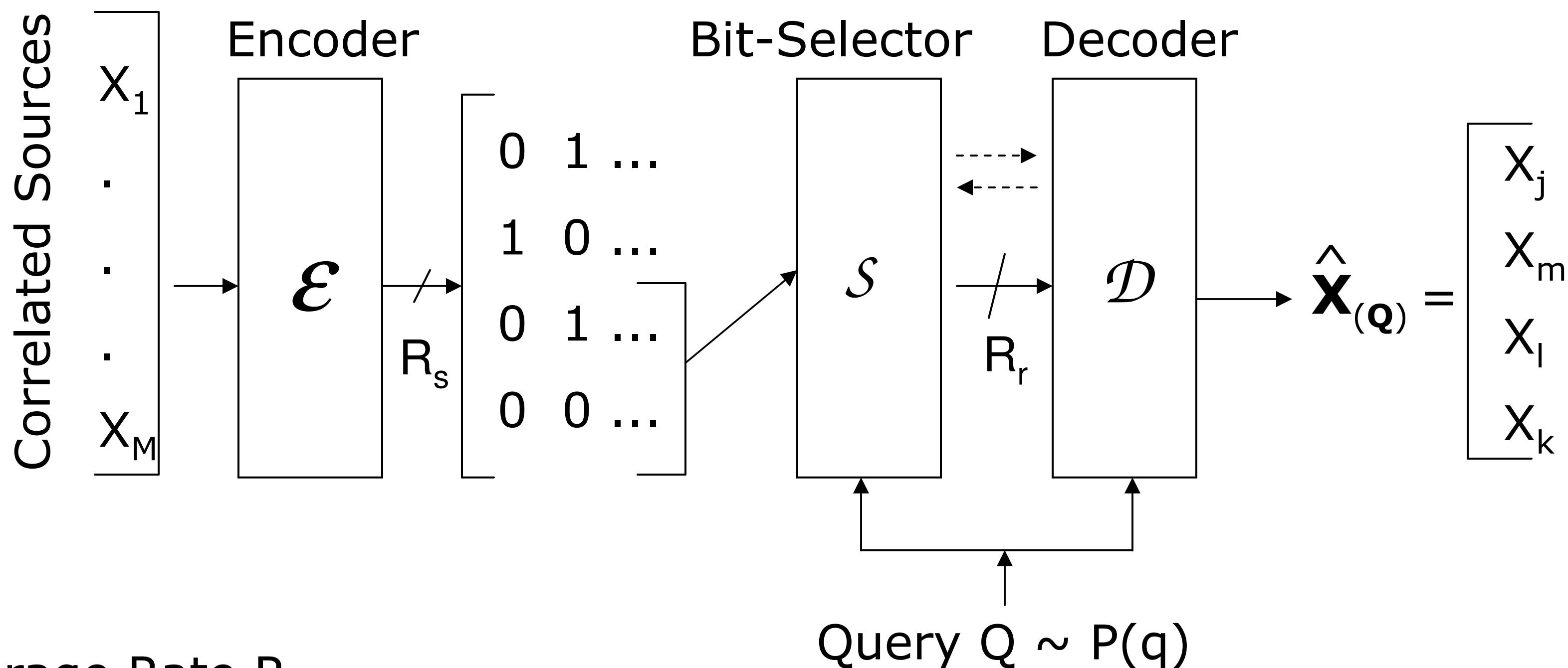
- Perfect reconstruction not always necessary
- Trade-off -

  Storage Rate vs. Retrieval Rate vs. Distortion

- In practice, storage devices are of fixed capacity
- Practical design by fixing maximum allowable R_s

- Query-dependent “bit-selection” (and relevant codebooks) for selective retrieval…

Proposed Fusion Coding Approach



Storage Rate R_s
Distortion D
Retrieval Rate R_r

Mathematically…

Encoder: $\mathcal{E} : \mathcal{R}^M \rightarrow \mathcal{I} = \{0,1\}^{R_s}$

Decoder: $\mathcal{D} : \mathcal{I} \times \mathcal{B} \rightarrow \hat{\mathcal{X}}$

Bit-Selector: $\mathcal{S} : \mathcal{Q} \rightarrow \mathcal{B} = 2^{\{1,\ldots,R_s\}}$

$$D = \sum_{\mathbf{q}\in\mathcal{Q}} P(\mathbf{q}) \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x}\in\mathcal{X}} d_{\mathbf{q}}(\mathbf{x}, \hat{\mathbf{x}})$$

$$R_r = \sum_{\mathbf{q}} P(\mathbf{q}) R_{\mathbf{q}} = \sum_{\mathbf{q}} P(\mathbf{q}) |\mathcal{S}(\mathbf{q})|$$

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

Optimality and Design

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

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

Optimal Decoder: $\mathcal{D}(\mathbf{i}, \mathbf{e}) = \frac{1}{|F|} \sum_{\mathbf{x}\in F} \mathbf{x}, \forall \mathbf{e}, \mathbf{i}$

Design by Gradient Descent

where $F = \{\mathbf{x} : (\mathcal{E}(\mathbf{x}))_{\mathbf{e}} = (\mathbf{i})_{\mathbf{e}}\}.$

Experimental Set-up

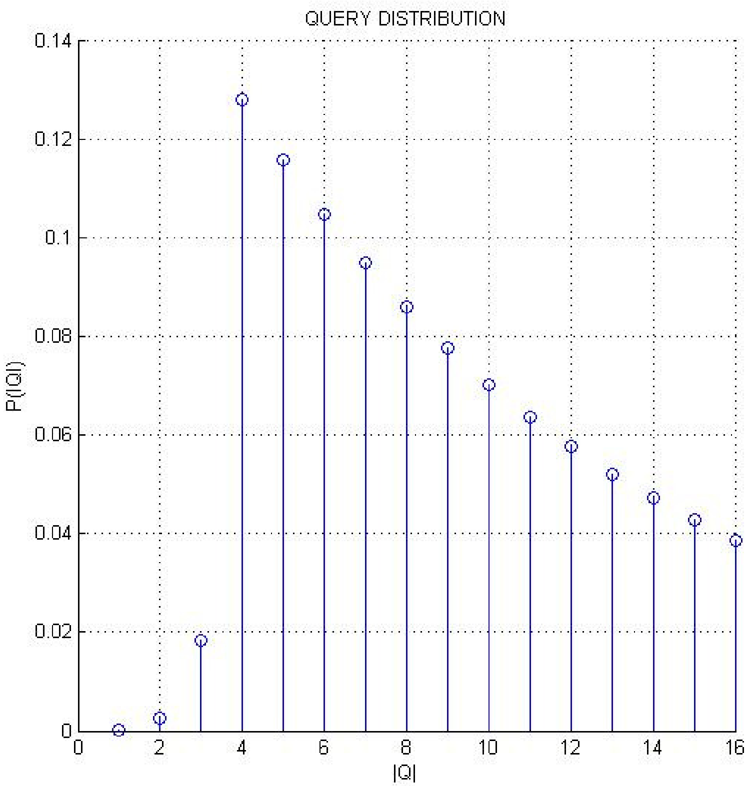
- Sensor data modeled as corr. Gaussian

$$C = \sigma^2 \begin{pmatrix} 1 & \rho & \dots & \rho^{M-2} & \rho^{M-1} \\ \rho & 1 & \rho & \dots & \rho^{M-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \rho^{M-2} & \dots & \rho & 1 & \rho \\ \rho^{M-1} & \rho^{M-2} & \dots & \rho & 1 \end{pmatrix}$$

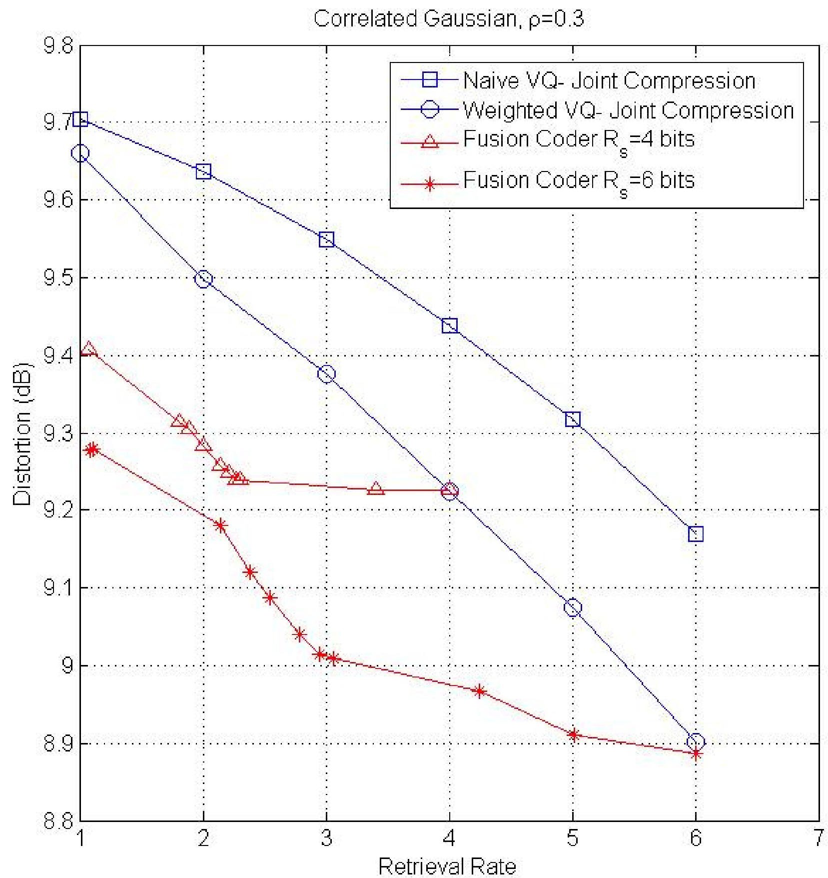
- Stock market data (UCR Data-mining archive) – 93 stocks

Exponential Query Distribution

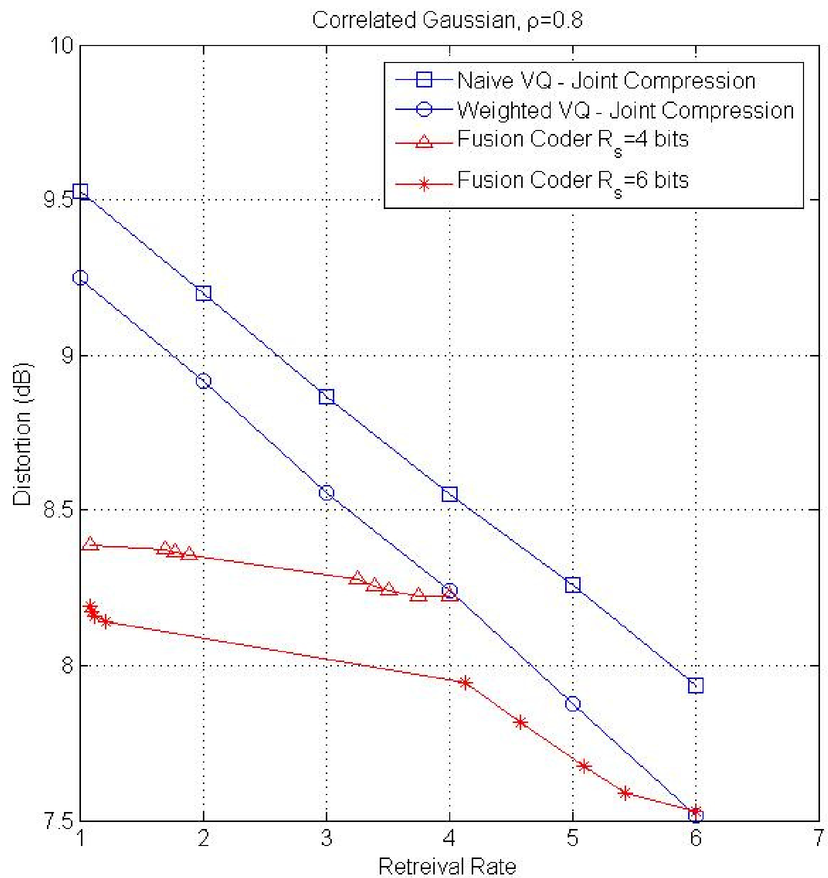
- Query distribution modeled as exponential in query size
- Queries of same size equally likely
- Distribution approximated by a training set of 335 queries



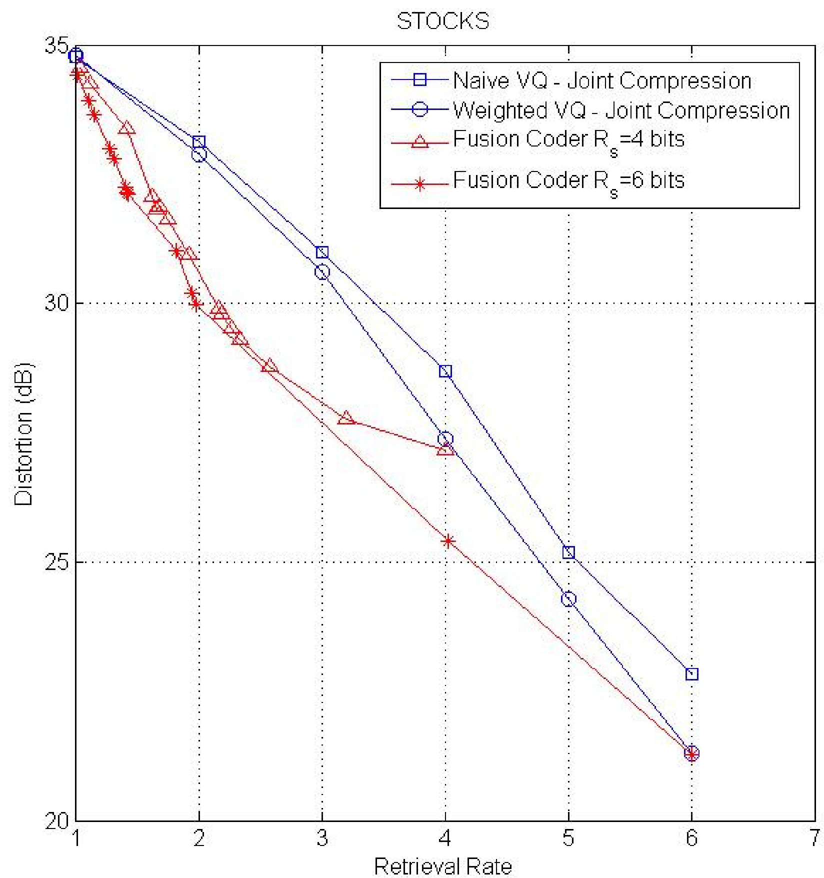
Correlated Gaussian ρ=0.3



Correlated Gaussian ρ=0.8



Stock Market Data



Conclusions

- Fusion storage vs. selective retrieval
  - optimization of conflicting objectives
- Fusion Coder proposed for practical design
- Fusion Coder provides large gains over joint compression (VQ) schemes