DISCUS: Distributed Compression for Sensor Networks

K. Ramchandran, K. Pister, M. Jordan, J. Malik, V. Anantharam,
L. Doherty, J. Hill, J. Kusuma, W. Leven, A. Ng, S. Pradhan, R. Szewczyk, J. Yeh

http://basics.eecs.berkeley.edu/sensorwebs

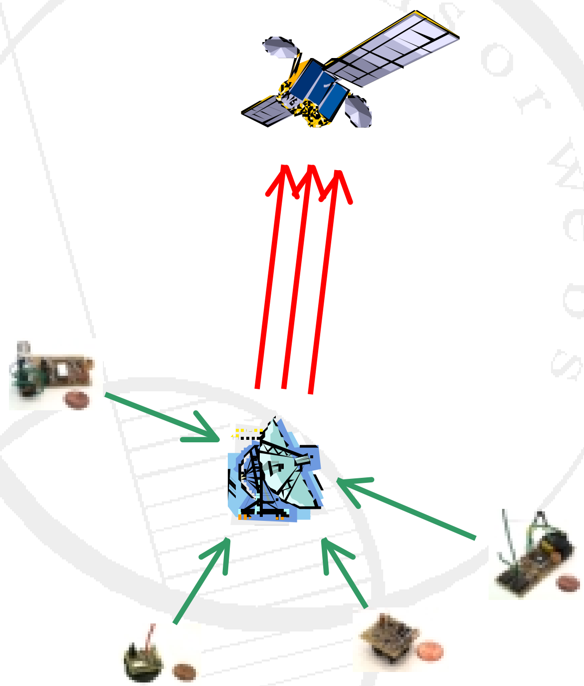
BASiCS Group

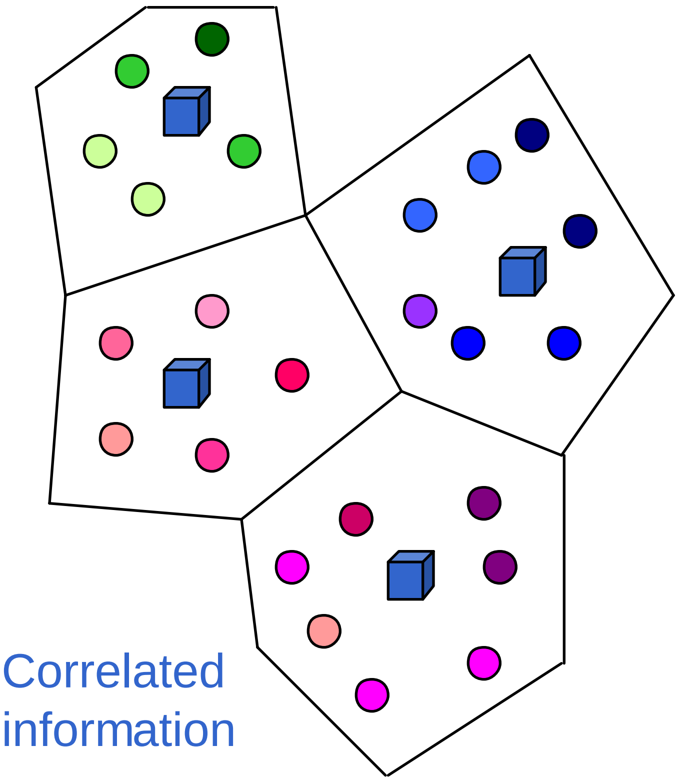
University of California at Berkeley

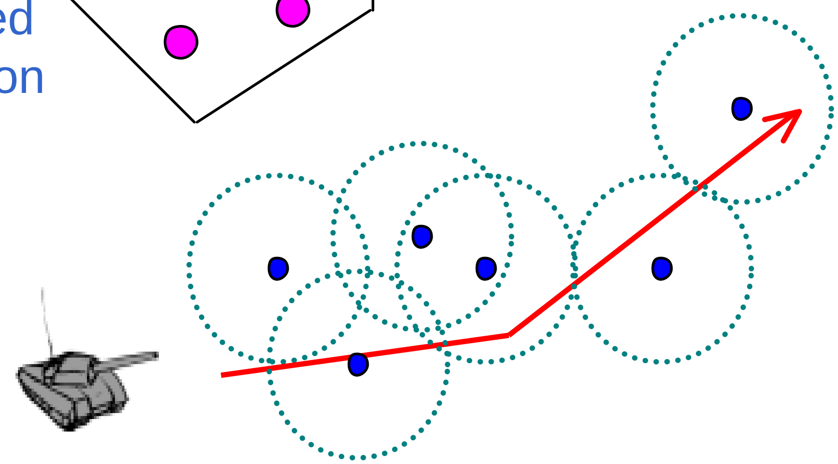
Motivations

- SmartDust and MEMS: quantum leap in device and computing device technology
- Excellent platform for efficient, smart devices and collaborative, distributed theory and algorithms
- Bring unified theoretical approach to address specific scenarios
- Sensor networks -> correlated observations: want to compress in efficient and distributed manner

Mission Statement



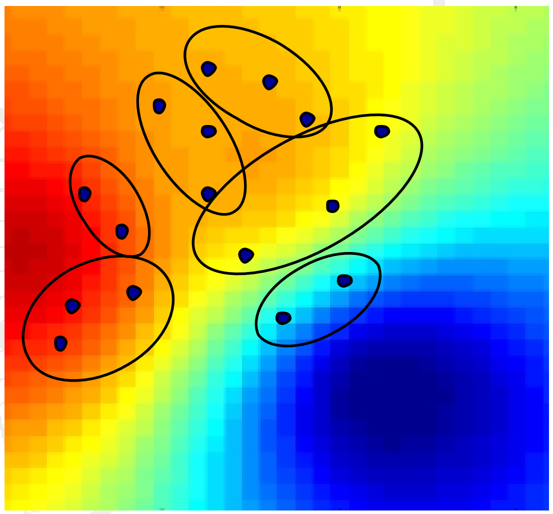




- Experimental testbed to leverage advances in SmartDust MEMS technology and demonstrate developed theories and algorithms -- the BCSN (Berkeley Campus Sensor Network).

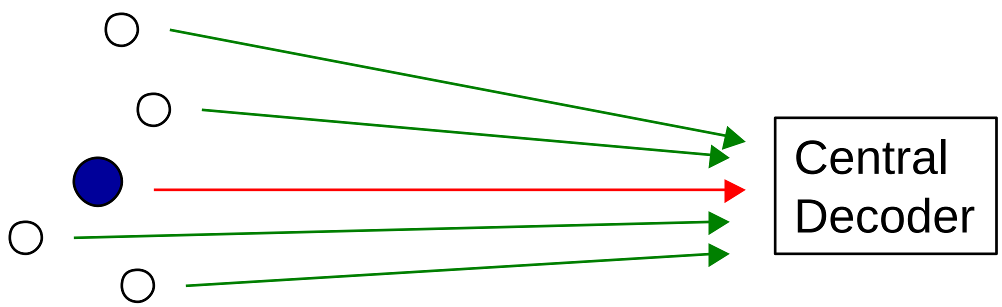
Scenario:
correlated observations

- Varying temperature field



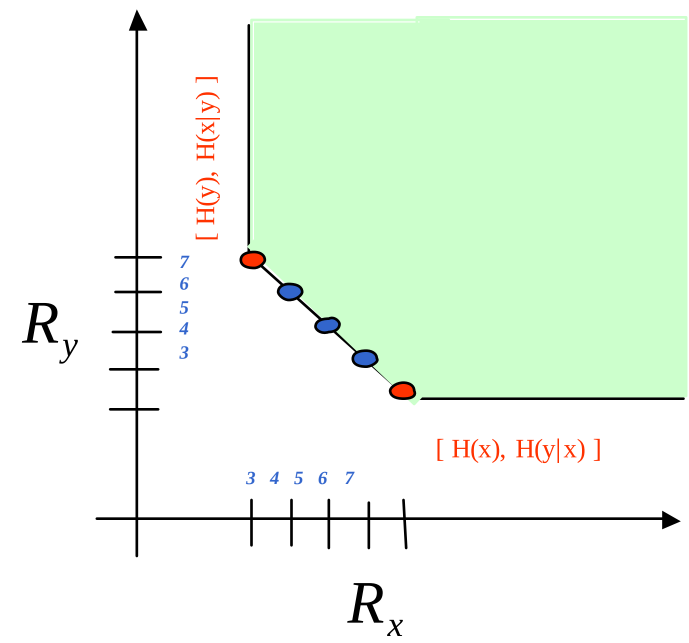
Information Theory: can gain performance boost thanks to correlation

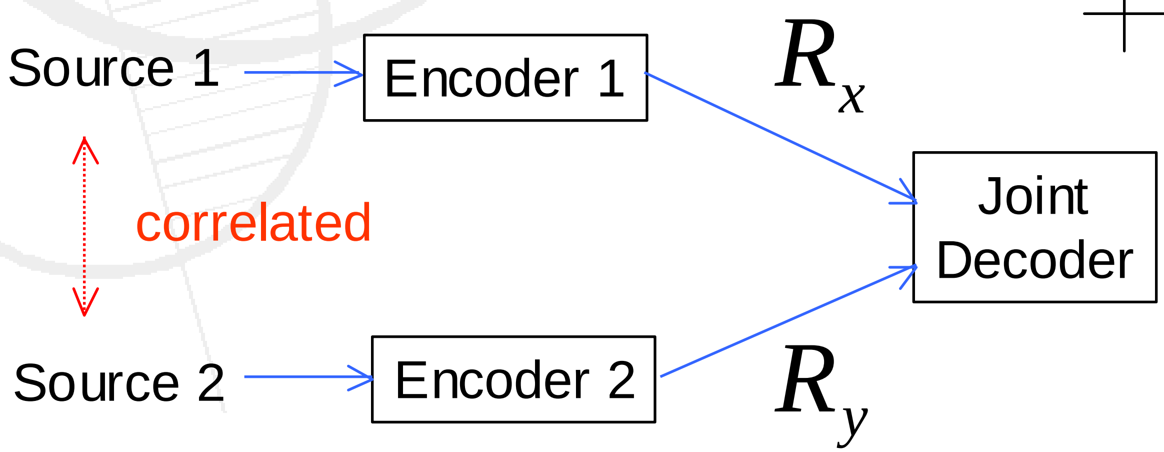
DISCUS: constructive way of harnessing correlation for blind joint compression close to optimum



At The End of The Day

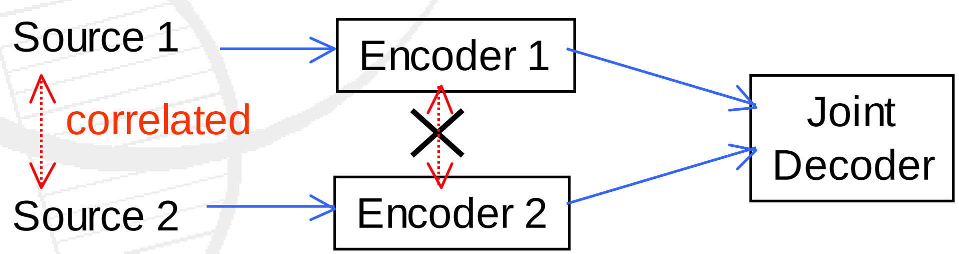
- Want to compress correlated observations
- Don't want communication between motes

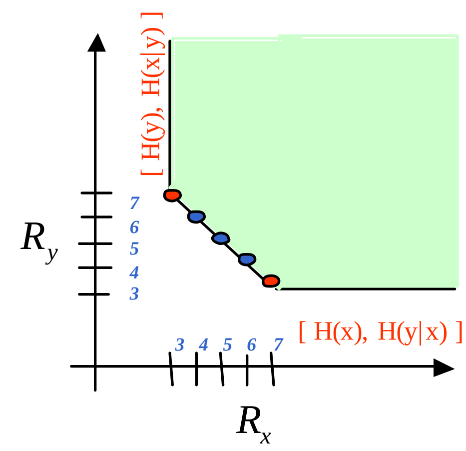




Distributed Compression

- Slepian-Wolf, Wyner-Ziv (ca. 70's): information-theoretically can jointly compress correlated observations, even with no communication between motes.

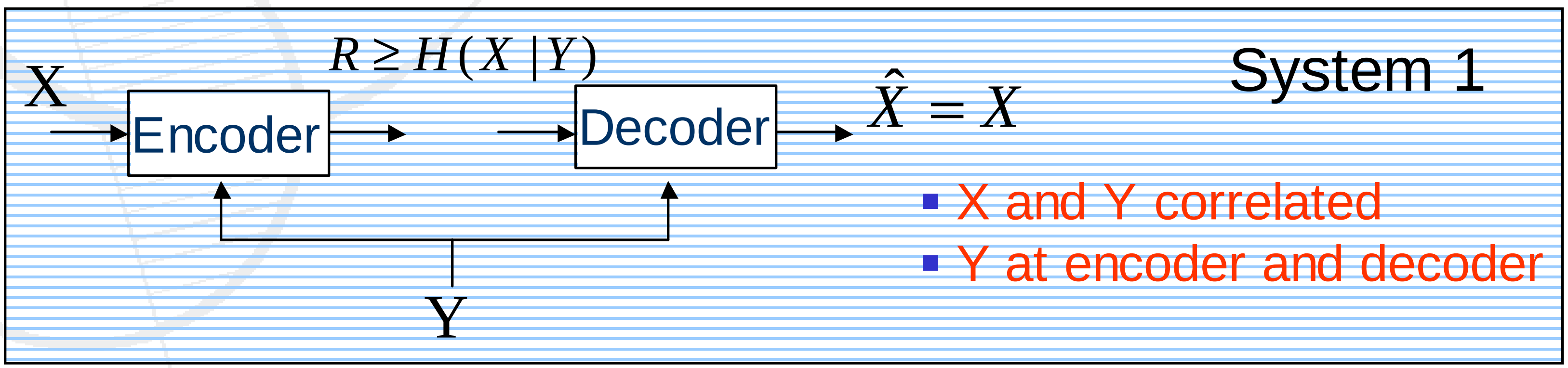




- DISCUS: constructive framework, using well-studied error-correcting codes from coding theory.

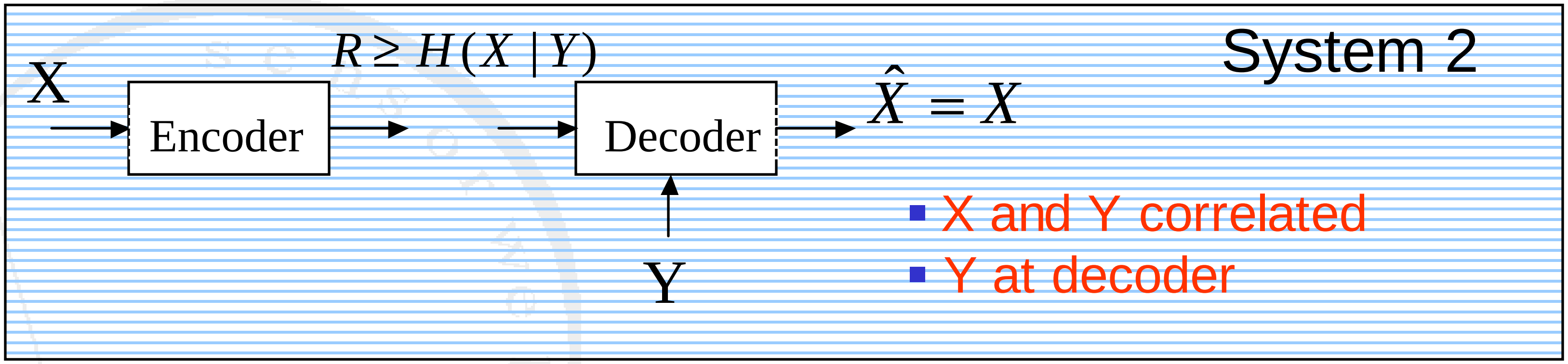
Source Coding with Side Information – discrete alphabets

- X and Y => length-3 binary data (equally likely),
- Correlation: Hamming distance between X and Y is at most 1.
  Example: When X=[0 1 0],
  Y => [0 1 0], [0 1 1], [0 0 0], [1 1 0].



$$X+Y=\begin{cases} 0\ 0\ 0 \\ 0\ 0\ 1 \\ 0\ 1\ 0 \\ 1\ 0\ 0 \end{cases}$$

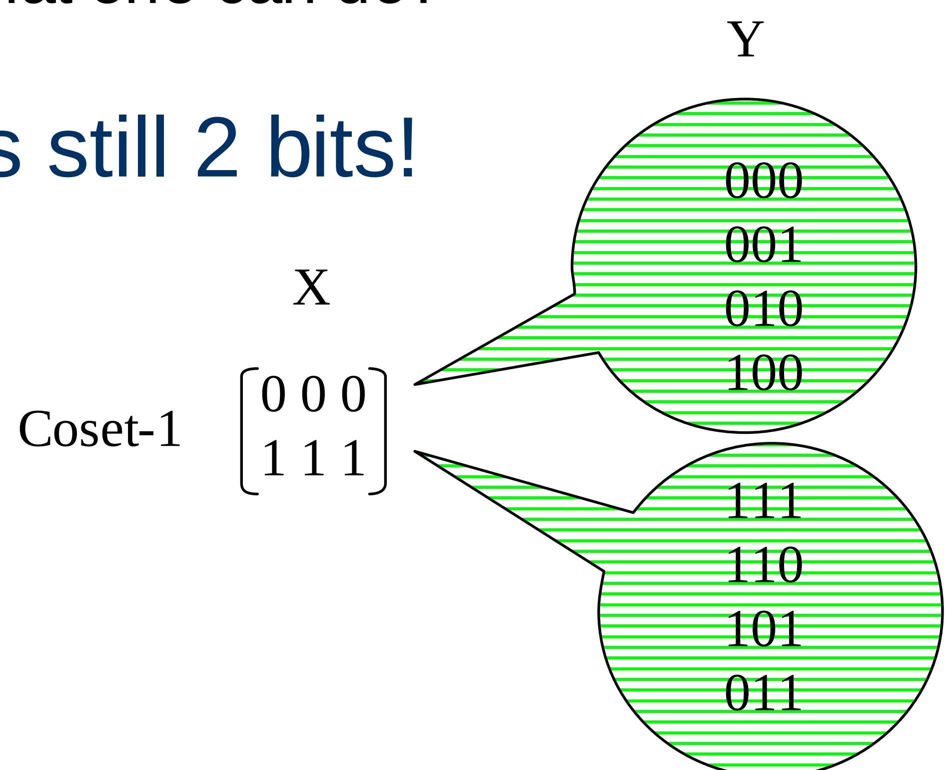
Need 2 bits to index this.



- What is the best that one can do?

- The answer is still 2 bits!

How?



Coset-1 $\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$ Coset-2 $\begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$

Coset-3 $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$ Coset-4 $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$

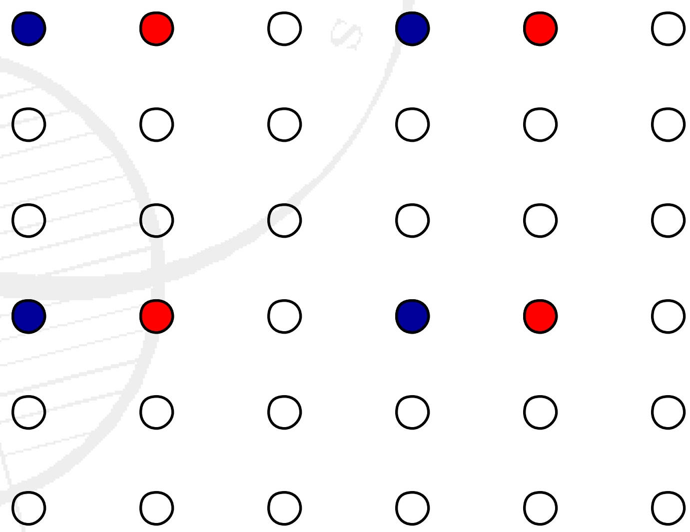
- Encoder -> index of the coset containing X.
- Decoder ->X in given coset.

Note:

- Coset-1 -> repetition code.
- Each coset -> unique "syndrome"
- DIstributed Source Coding Using Syndromes

Discrete case – handwaving argument

- Consider abstraction using lattices
- Correlation structure translates to distance between X and Y

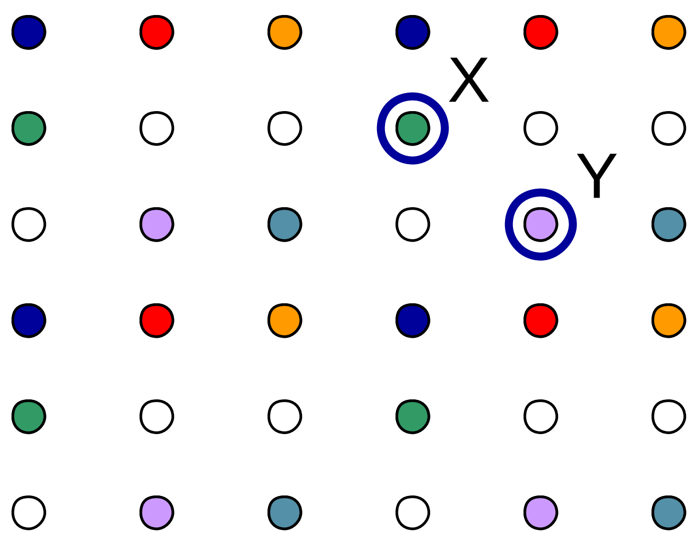


- Suppose X and Y have distance at most 1
- How to compress X if we know that Y is close to X?

- Use cosets!
- Partition lattice space into cosets and send coset index!

A little accounting

- How much have we compressed?
- Without compression: $\log_2 36$ bits
- With compression: $\log_2 9$ bits



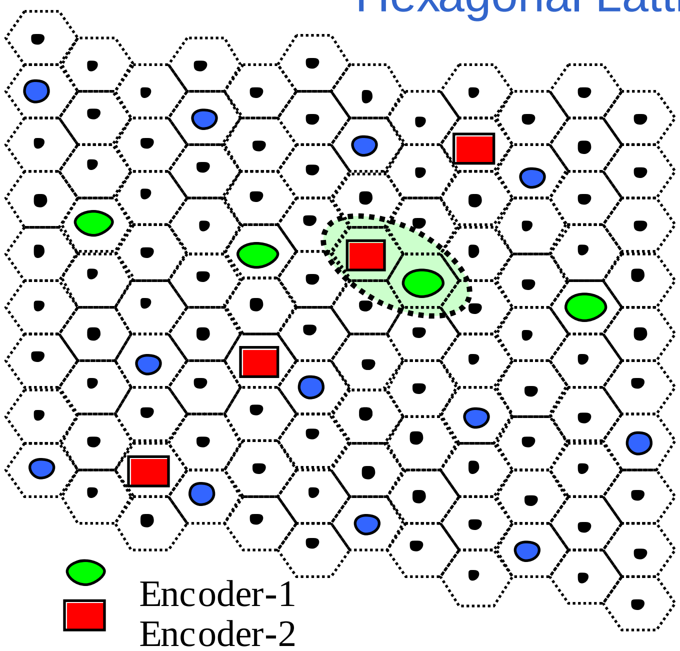
And now the math cleanup!

- Can use well-studied channel codes to build codes on euclidean space!
- Basic idea: want to select points as far apart as possible (i.e. cosets) – done using channel codes
- Send ambiguous information, side information will disambiguate information!
- Can use Hamming, TCM, RS, etc. codes

Generalized coset codes: a/symmetric applications

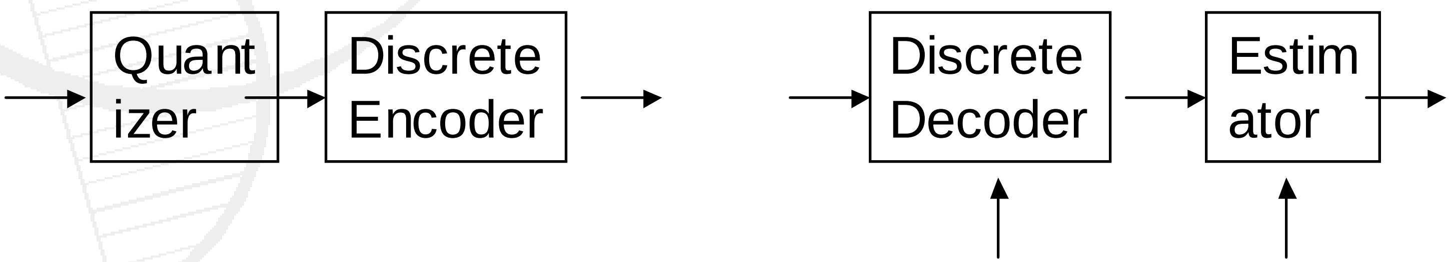
- How to compress to arbitrary (but information-theoretically sound) rates?
- Use lattice idea again: 2 encoders send ambiguous information, but only one true answer that satisfy correlation constraint

Hexagonal Lattice



Continuous case – quantization and estimation

- Modular boxes:



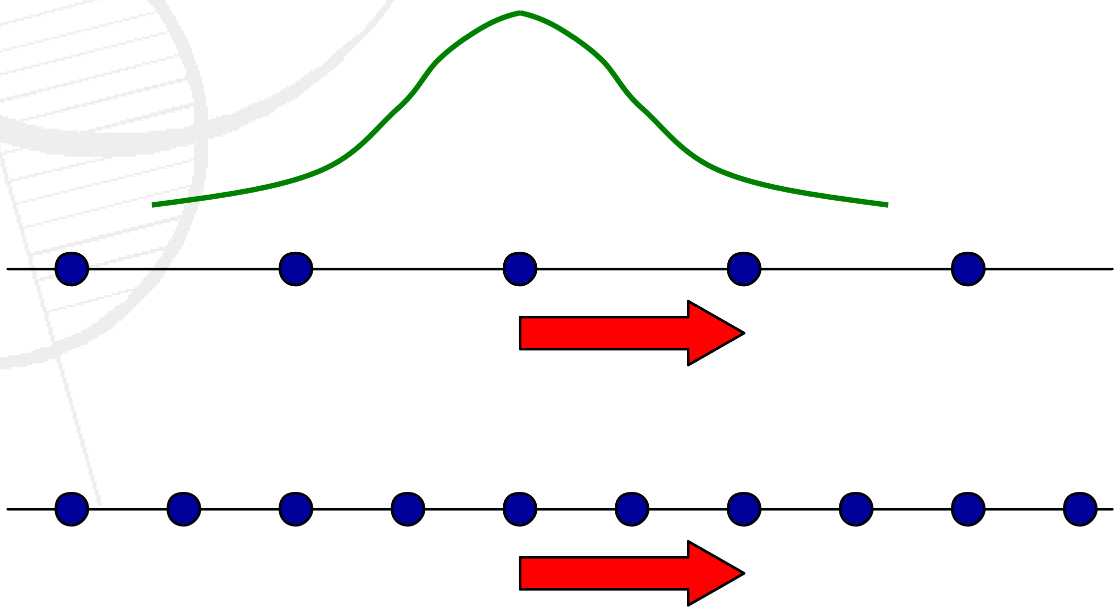
Gaussian case example

- Estimation with side information:
  - First quantize sensor information
  - Use discrete DISCUS encoders / decoders
  - Estimate given decoder output and side information

$$\hat{X} = E\left[X \mid Y, X \in \Gamma_i\right]$$

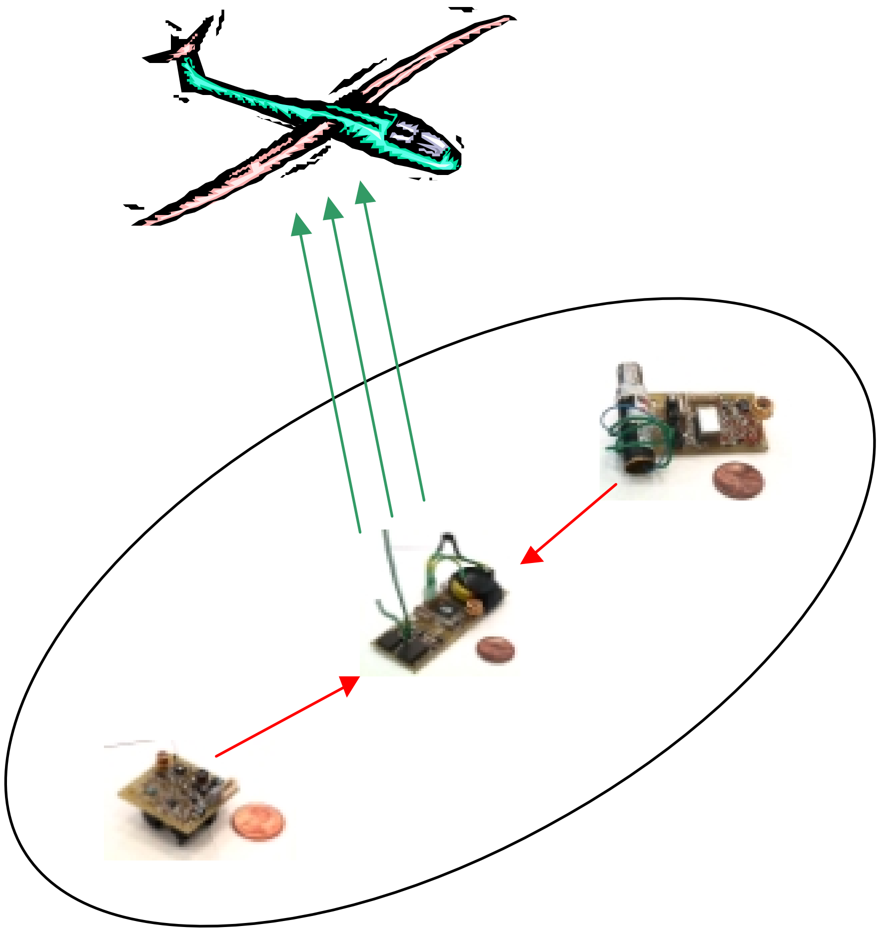
No Such Thing as Free Lunch

- Big issue: finer quantization means larger error distances – need stronger code to enable finer quantization!!!



Deployment Scenario

- Observe readings: learn correlation
- Use clustering algorithm
- Assign codebooks to different motes
- Can rotate “centroid” mote in each group
- Centroid mote report to central decoder



Dynamic Tracking Scenario

- Wish to dynamically update clustering
- Good news: no need for child nodes to be aware!!! Codebook assignment not an issue!
- Only central decoder needs to be aware of clustering
- Can also rotate centroid node within each cluster: can detect correlation changes

Conclusions

- Can use efficient error-correcting codes to enable distributed compression
- Power/bandwidth/quality tradeoff through quantization and codebook selection
- Very little work for encoders!
- Tremendous gains in a sensor network