

Part V: Collaborative Signal Processing

Akbar Sayeed

V-1

CSP Outline

- Introduction and Overview
 - Illustrative example: single target tracking
 - Different modes of CSP
- Event Detection
- CSP principles for distributed decision making
 - Illustrated in the context of target classification
 - Data fusion versus decision fusion
 - Illustrative examples with real data
- Other CSP applications
 - Target Localization
 - Target Tracking
 - Distributed compression
 - Information driven sensor querying
- Challenges

V-2

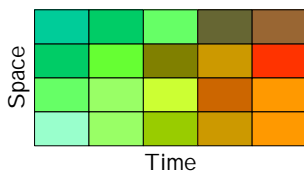
Sensor Networks from a SP Perspective

- Provide a virtual map of the physical world:
 - Monitoring a region in a variety of sensing modalities (acoustic, seismic, thermal, ...)
- Two key components:
 - Networking and routing of information
 - Collaborative signal processing (CSP) for extracting and processing information from the physical world

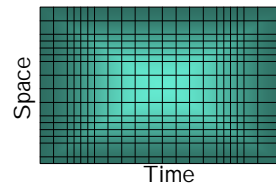
V-3

Space-Time Sampling

- Sensors sample the *spatial* signal field in a particular modality (e.g., acoustic, seismic)
 - Sensor density commensurate with spatial signal variation
- Sampling of time series from each sensor commensurate with signal bandwidth
- Sensor field decomposed into *space-time cells* to enable distributed signal processing (multiple nodes per cell)



Uniform space-time cells

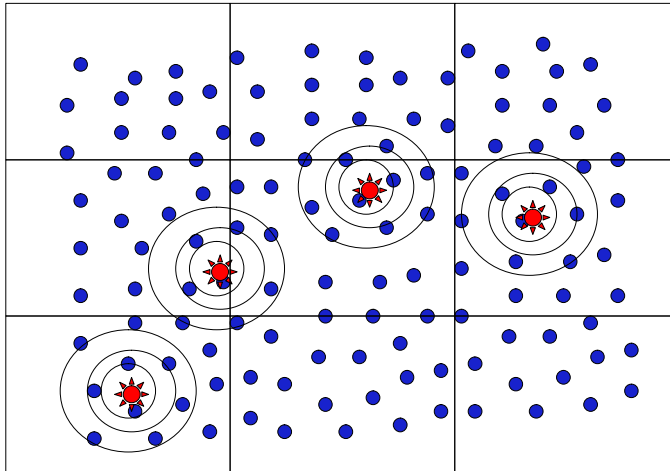


Non-uniform space-time cells

V-4

An Object Moving Through the Network

A moving object corresponds to a spatial peak moving with time
Target tracking corresponds to determining the peak location over time



Manager Nodes

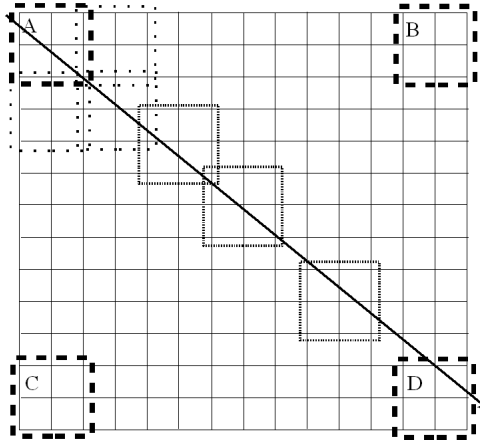
- In each region or cell, a particular node is designated manager node
- The manager node is responsible for:
 - Coordinating the communication between different nodes in the region
 - Receiving data from different collaborating nodes in the region
 - Jointly processing the signals from different collaborating nodes for tasks that require CSP across nodes
 - Communicating requested information back to the querying node

Illustrative Example: Single Target Tracking

Initialization: Cells A,B,C and D are put on detection alert for a specified period

Five-step procedure:

1. A track is initiated when a target is detected in a cell (Cell A – Active cell). Detector outputs of active nodes are sent to the manager node
2. Manager node estimates target location at N successive time instants using outputs of active nodes in Cell A.
3. Target locations are used to predict target location at $M < N$ future time instants
4. Predicted positions are used to create new cells that are put on detection alert
5. Once a new cell detects the target it becomes the active cell



V-7

Component Signal Processing Algorithms

- Single target tracking
 - requires *detection, localization, and location prediction*
- Can track multiple targets if they are sufficiently *separated in space and/or time*
 - separate track for each target
- Can track spatio-temporally overlapping targets with appropriate *classification* algorithms

V-8

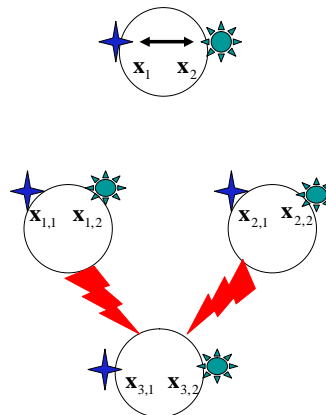
Why CSP?

- In principle, more information about a phenomenon can be gathered from multiple measurements
 - Multiple sensing modalities (acoustic, seismic, etc.)
 - Multiple nodes
- Limited local information gathered by a single node necessitates CSP
 - Inconsistencies between measurements, such as due to malfunctioning nodes, can be resolved
- Variability in signal characteristics and environmental conditions necessitates CSP
 - Complementary information from multiple measurements can improve performance

V-9

Categorization of CSP Algorithms Based on Communication Burden

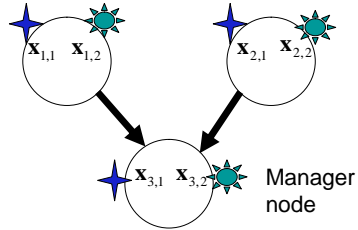
- Intra-node collaboration
 - Multiple sensing modalities
 - E.g., combining acoustic and seismic measurements
 - No communication burden since collaboration is at a particular node
 - Higher computational burden at the node
- Inter-node collaboration
 - Combining a measurements at different nodes
 - Higher communication burden since data is exchanged between nodes
 - Higher computational burden at manager node



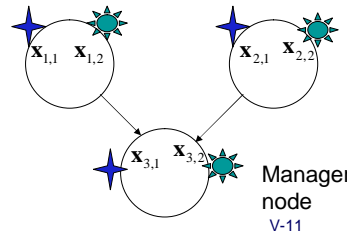
Manager node V-10

Categorization of CSP Algorithms Based on Computational Burden

- Data fusion
 - Time series for different measurements are combined
 - Higher computational burden since higher dimensional data is jointly processed
 - Higher communication burden if different measurements from different nodes



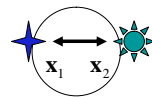
- Decision fusion
 - Decisions (hard or soft) based on different measurements are combined
 - Lower computational burden since lower dimensional data (decisions) is jointly processed
 - Higher communication burden if the component decisions are made at different nodes



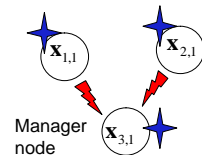
V-11

Various Forms of CSP

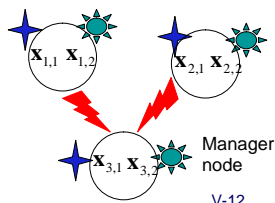
- Single Node, Multiple Modality (SN, MM)
 - Simplest form of CSP: no communication burden
 - Decision fusion
 - Data fusion (higher computational burden)



- Multiple Node, Single Modality (MN, SM)
 - Higher communication burden
 - Decision fusion
 - Data fusion (higher computational burden)



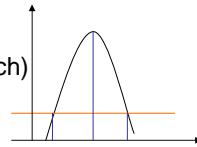
- Multiple Node, Multiple Modality (MN, MM)
 - Highest communication and computational burden
 - Decision fusion across modalities and nodes
 - Data fusion across modalities, decision fusion across nodes
 - Data fusion across modalities and nodes



V-12

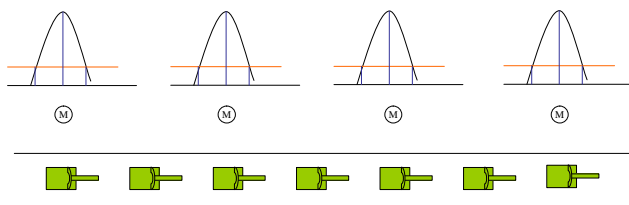
Event Detection

- Simple energy detector
 - Detect a target/event when the output exceeds an adaptive threshold (CFAR)
- Detector output:
 - At any instant is the average energy in a certain window
 - Is sampled at a certain rate based on a priori estimate of target velocity and signal bandwidth
- Output parameters for each event:
 - max value (CPA – closest point of approach)
 - time stamps for: onset, max, offset
 - time series for classification
- Multi-node and multi-modality collaboration



V-13

Detector Output Illustration



V-14

Constant False Alarm Rate (CFAR) Detection

- Energy detector is designed to maintain a CFAR
- Detector threshold is adapted to the statistics of the decision variable under noise hypothesis
- Let $x[n]$ denote a sensor time series
- Energy detector:

$$e[n] = \sum_{k=0}^{W-1} |x[n-k]|^2 \sim \begin{cases} N(\mu_s, \sigma^2) & \text{Target present } (H_1) \\ N(\mu_n, \sigma^2) & \text{Target absent } (H_0) \end{cases}$$

W is the detector window length $\mu \equiv \mu_s - \mu_n > 0$

- Detector decision:

$$e[n] \geq \gamma \quad \text{Target present}$$

$$e[n] < \gamma \quad \text{Target absent}$$

V-15

CFAR Threshold Computation

- Choose the threshold γ so that the False Alarm Rate (FAR) is below a specified value

$$\text{FAR} = P(e[n] \geq \gamma | H_0) \leq p_0$$

- The desired threshold is given by

$$\gamma = \mu_n + \sigma Q^{-1}(p_0)$$

$$Q(x) = \int_x^{\infty} e^{-t^2/2} dt$$

- (μ_n, σ) estimates are continually updated to adapt the threshold

V-16

Single Target Classification: Overview

- Single measurement classifiers
 - MAP/ML Gaussian classifiers
 - NN classifiers (benchmark)
 - Training and Performance Evaluation
 - Confusion matrices
- Multiple measurement classifiers
 - Data fusion (dependent measurements)
 - Decision fusion (independent measurements)
- Different possibilities for CSP-based classification
 - Single node, multiple sensing modalities (SN, MM)
 - Multiple nodes, single sensing modality (MN, SM)
 - Multiple nodes, multiple sensing modalities (MN, MM)

The basic ideas illustrate general CSP principles in distributed decision making

V-17

Single Measurement Classifier

- M possible target classes: $\omega_m \in \Omega = \{m = 1, \dots, M\}$
- \mathbf{x} : N -dim. (complex-valued) event feature vector
 - \mathbf{x} belongs to m -th class with probability $P(\omega_m)$
- C : classifier assigns one of the classes to \mathbf{x}

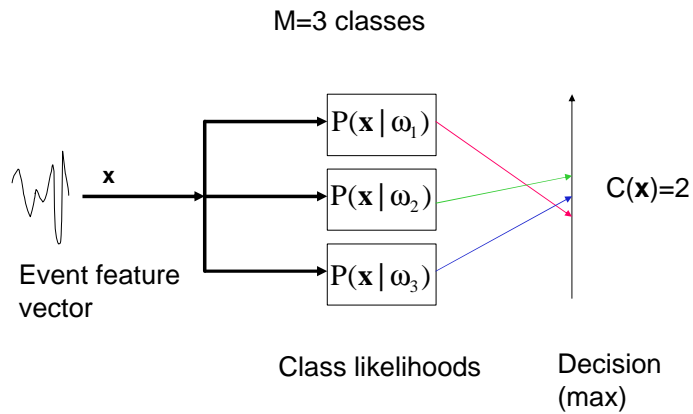
$$\text{MAP: } C(\mathbf{x}) = m \quad \text{if} \quad P(\omega_m | \mathbf{x}) = \max_{j=1, \dots, M} P(\omega_j | \mathbf{x})$$

$$\text{Bayes rule: } C(\mathbf{x}) = \arg \max_{j=1, \dots, M} P(\mathbf{x} | \omega_j) P(\omega_j)$$

$$\text{Equal priors (ML): } C(\mathbf{x}) = \arg \max_{j=1, \dots, M} P(\mathbf{x} | \omega_j)$$

V-18

Single Measurement Classifier – Pictorially



V-19

Effective Event Feature Vectors

- Each event typically yields multiple feature vectors
- An effective feature vector for each event is extracted, such as the mean

$$\mathbf{x} = \frac{1}{N_e} \sum_{n=1}^{N_e} \mathbf{x}_n$$

V-20

Gaussian Classifiers

- Assume that for class j , \mathbf{x} has a complex Gaussian distribution with mean vector $\boldsymbol{\mu}_j = E_j[\mathbf{x}]$ and covariance matrix $\mathbf{S}_j = E_j[(\mathbf{x} - \boldsymbol{\mu}_j)(\mathbf{x} - \boldsymbol{\mu}_j)^H]$
 - $E_j[\bullet]$ denotes ensemble average over class j
 - Superscript H denotes complex conjugate transpose
- Likelihood function for class j

$$P(\mathbf{x} | \omega_j) = \frac{1}{\pi^N |\mathbf{S}_j|} \exp\left[-(\mathbf{x} - \boldsymbol{\mu}_j)^H \mathbf{S}_j^{-1} (\mathbf{x} - \boldsymbol{\mu}_j)\right]$$

$$-\log P(\mathbf{x} | \omega_j) = \log |\mathbf{S}_j| + (\mathbf{x} - \boldsymbol{\mu}_j)^H \mathbf{S}_j^{-1} (\mathbf{x} - \boldsymbol{\mu}_j)$$

V-21

Training and Performance Assessment

- N^{Tr} training events available for each class
- 3-way cross validation – partition data into 3 sets (S_1, S_2, S_3) with equal number of events for each class
- Three sets of experiments:

| | |
|------------|-------|
| Train | Test |
| S_1, S_2 | S_3 |

| | |
|------------|-------|
| Train | Test |
| S_1, S_3 | S_2 |

| | |
|------------|-------|
| Train | Test |
| S_2, S_3 | S_1 |

V-22

Training and Testing

- In each experiment we have:
 - Training phase: estimate mean and covariance for each class from the two training data sets

For $\mathbf{x}_n \in \omega_j \quad j = 1, \dots, M$

$$\hat{\boldsymbol{\mu}}_j = \frac{1}{N_0} \sum_{n=1}^{N_0} \mathbf{x}_n \quad \hat{\mathbf{S}}_j = \frac{1}{N_0} \sum_{n=1}^{N_0} (\mathbf{x}_n - \boldsymbol{\mu}_j)(\mathbf{x}_n - \boldsymbol{\mu}_j)^H$$

- Testing phase: Using $(\hat{\boldsymbol{\mu}}_j, \hat{\mathbf{S}}_j)$ estimated from the two training data sets, test the performance of the classifier on the third testing set

$$C(\mathbf{x}) = \arg \max_{j=1, \dots, M} P(\mathbf{x} | \omega_j)$$

V-23

Confusion Matrix

| | | | | |
|-------------------------------------|----------|----------|-----|----------|
| $C(\mathbf{x}) \backslash \omega_m$ | 1 | 2 | ... | M |
| 1 | n_{11} | n_{12} | | n_{1M} |
| 2 | n_{21} | n_{21} | | n_{2M} |
| ⋮ | | | ⋮ | |
| M | n_{M1} | n_{M2} | | n_{MM} |

$[CM]_{ij} = n_{ij}$ = number of events from ω_i classified as ω_j

V-24

Probability of Detection, Probability of False Alarm, Belief

- Probability of detection for class m

$$PD_m = \frac{n_{mm}}{\sum_{j=1}^M n_{mj}} \quad (\text{m-th row})$$

- Probability of false alarm for class m

$$PFA_m = \frac{\sum_{j=1}^M n_{jm} - n_{mm}}{\sum_{j=1}^M n_{jm}} \quad (\text{m-th column})$$

- Prior belief in the classifier decisions (via training)

$$P(\mathbf{x} \in \omega_m | C(\mathbf{x}) = j) = \frac{n_{mj}}{\sum_{i=1}^M n_{ij}} \quad (\text{j-th column})$$

V-25

Benchmark: Nearest Neighbor (NN) Classifier

- S^{Tr} -- the set of all training event feature vectors \mathbf{x}^{Tr} (containing all classes)
- \mathbf{x} -- test event feature vector to be classified

$$C_{NN}(\mathbf{x}) = \text{class}\left(\arg \min_{\mathbf{x}^{Tr} \in S^{Tr}} \|\mathbf{x} - \mathbf{x}^{Tr}\|\right)$$

That is, find the *training feature vector* that is closest to the *test feature vector*. Assign the label of the closest training feature vector to the test event

V-26

Multiple Measurements

- K measurements (from a detected event)
 - Different nodes or sensing modalities
- \mathbf{x}_k -- event feature vector for k-th measurement
- Classifier C assigns one of the M classes to the K event measurements $\{\mathbf{x}_1, \dots, \mathbf{x}_K\}$

$$C(\mathbf{x}_1, \dots, \mathbf{x}_K) = \arg \max_{j=1, \dots, M} P(\omega_j | \mathbf{x}_1, \dots, \mathbf{x}_K)$$

Equal priors (ML): $C(\mathbf{x}_1, \dots, \mathbf{x}_K) = \arg \max_{j=1, \dots, M} P(\mathbf{x}_1, \dots, \mathbf{x}_K | \omega_j)$

V-27

Data Fusion – Gaussian Classifier

- Assume that different measurements $(\{\mathbf{x}_1, \dots, \mathbf{x}_K\})$ are jointly Gaussian and *correlated*

For $\omega_j, j = 1, \dots, M$

the concatenated event feature vector (KN dim.) $\mathbf{x}^c = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_K \end{bmatrix}$

is Gaussian with mean and covariance:

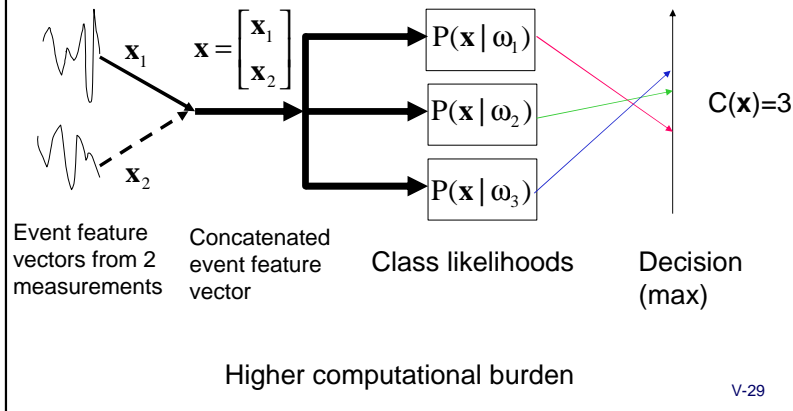
$$\boldsymbol{\mu}_j^c = E_j[\mathbf{x}^c] = \begin{bmatrix} \boldsymbol{\mu}_{j,1} \\ \vdots \\ \boldsymbol{\mu}_{j,K} \end{bmatrix} \quad \mathbf{S}_j^c = E_j[(\mathbf{x}^c - \boldsymbol{\mu}_j^c)(\mathbf{x}^c - \boldsymbol{\mu}_j^c)^H] = \begin{bmatrix} \mathbf{S}_{j,11}, \dots, \mathbf{S}_{j,1K} \\ \vdots \\ \mathbf{S}_{j,K1}, \dots, \mathbf{S}_{j,KK} \end{bmatrix}$$

$(\boldsymbol{\mu}_j^c, \mathbf{S}_j^c)$ characterize the j-th class and can be estimated from training data → cross-validation, CM's, PD, PFA, belief

V-28

Multiple Measurement Classifier – Data Fusion

M=3 classes



Data Fusion – NN Classifier

- Let \mathbf{S}^{Tr} denote the set of all *concatenated* training event feature vectors \mathbf{x}^{cTr} (containing all classes)

$$\mathbf{x}^{\text{cTr}} = \begin{bmatrix} \mathbf{x}_1^{\text{Tr}} \\ \vdots \\ \mathbf{x}_K^{\text{Tr}} \end{bmatrix} \quad (\text{NK dimensional})$$

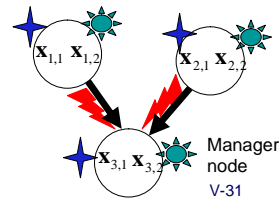
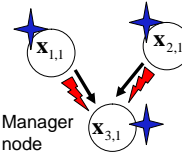
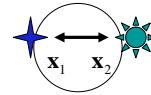
- Let \mathbf{x}^{c} denote the concatenated test event feature vector to be classified

$$C_{\text{NN}}(\mathbf{x}_1, \dots, \mathbf{x}_k) = \text{class} \left(\arg \min_{\mathbf{x}^{\text{cTr}} \in \mathbf{S}^{\text{Tr}}} \|\mathbf{x}^{\text{c}} - \mathbf{x}^{\text{cTr}}\| \right)$$

Forms of Data Fusion in CSP

K modalities, P nodes

- Data fusion of multiple modalities (e.g., acoustic and seismic) at each node (SN, MM)
 - Higher comp. burden (NK dim. data)
 - No additional comm. burden
- Data fusion of a single modality at multiple nodes (MN, SM)
 - Higher computational burden at manager node (PN dim. data)
 - Higher communication burden due to transmission of N dim. data from different nodes to the manager node
- Data fusion of multiple modalities at multiple nodes (MN, MM)
 - Highest computational burden at manager node (NKP dim. data)
 - Highest communication burden due to transmission of KN dim. multi-modality data from different nodes to the manager node



Pros and Cons of Data Fusion

- Pros
 - Maximal exploitation of available information in multiple times series
 - Potentially the best performing classification scheme
- Cons
 - High computational burden
 - High communication burden if data fusion across nodes
 - Need larger amount of data for training
 - Inconsistencies between measurements could cause performance degradation (malfunctioning nodes, e.g.)

Decision Fusion

- Suppose different measurements are statistically *independent*

For each class, $j = 1, \dots, M$

$$P(\mathbf{X} | \omega_j) = P(\mathbf{x}_1, \dots, \mathbf{x}_K | \omega_j) = \prod_{k=1}^K P(\mathbf{x}_k | \omega_j)$$

$$P(\omega_j | \mathbf{X}) \propto P(\mathbf{X} | \omega_j)P(\omega_j) = P(\omega_j) \prod_{k=1}^K P(\mathbf{x}_k | \omega_j)$$

$$\mathbf{S}_j^c = \begin{bmatrix} \mathbf{S}_{j,11}, \dots, \mathbf{S}_{j,1K} \\ \vdots \\ \mathbf{S}_{j,K1}, \dots, \mathbf{S}_{j,KK} \end{bmatrix} = \begin{bmatrix} \mathbf{S}_{j,11}, \mathbf{0}, \dots, \mathbf{0} \\ \vdots \\ \mathbf{0}, \dots, \mathbf{S}_{j,KK} \end{bmatrix} \quad \mathbf{S}_{j,kk'} = \mathbf{0}, k \neq k'$$

- This suggests combining the *decisions* of component classifiers for different measurements

V-33

Hard or Soft Decisions

- Soft decisions for different measurements

$$P(\omega_j | \mathbf{x}_k) \propto P(\mathbf{x}_k | \omega_j)P(\omega_j) \quad \begin{matrix} j = 1, \dots, M \\ k = 1, \dots, K \end{matrix}$$

may be combined to form the final decision

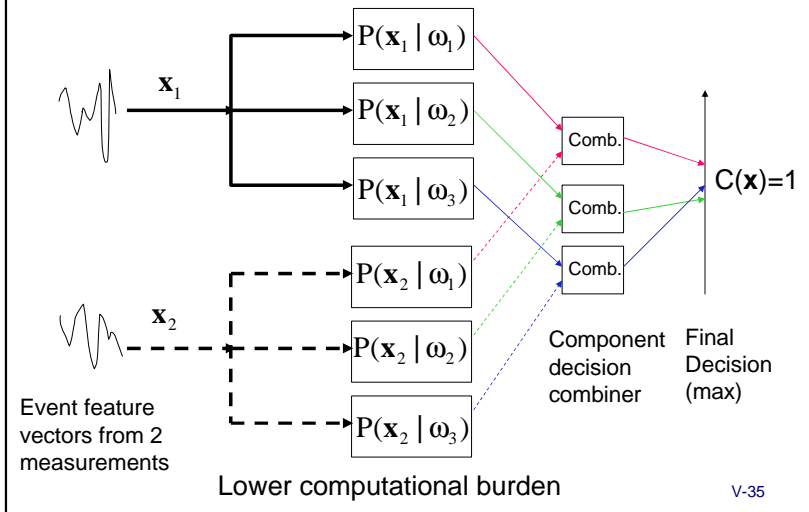
- Hard decisions for different measurements

$$\Delta_{jk} = \begin{cases} 1 & P(\omega_j | \mathbf{x}_k) = \max_{i=1, \dots, M} P(\omega_i | \mathbf{x}_k) \quad j = 1, \dots, M \\ 0 & \text{otherwise} \quad k = 1, \dots, K \end{cases}$$

may be combined to form the final decision

V-34

Multiple Measurement Classifier – Soft Decision Fusion



V-35

Different Ways of Soft Decision Fusion

- All based on the following inequalities

$$\prod_{k=1}^K P(\omega_j | \mathbf{x}_k) \leq \min_{k=1, \dots, K} P(\omega_j | \mathbf{x}_k) \leq \frac{1}{K} \sum_{k=1}^K P(\omega_j | \mathbf{x}_k) \leq \max_{k=1, \dots, K} P(\omega_j | \mathbf{x}_k)$$

- Product rule

$$C(\mathbf{X}) = \arg \max_{j=1, \dots, M} P^{-(K-1)}(\omega_j) \prod_{k=1}^K P(\omega_j | \mathbf{x}_k)$$

Equal priors (ML):

$$C(\mathbf{X}) = \arg \max_{j=1, \dots, M} \prod_{k=1}^K P(\mathbf{x}_k | \omega_j)$$

“On combining classifiers,” Kittler et. al., IEEE Trans. Pattern Anal. Machine Intelligence, March 1998.

V-36

Soft Decision Fusion: Sum Rule

- Product rule suffers from the problem that one small component likelihood would heavily influence the final decision
- Sum Rule

$$C(\mathbf{X}) = \arg \max_{j=1, \dots, M} \left[(1-K)P(\omega_j) + \sum_{k=1}^K P(\omega_j | \mathbf{x}_k) \right]$$

Equal priors (ML):

$$C(\mathbf{X}) = \arg \max_{j=1, \dots, M} \left[\sum_{k=1}^K P(\mathbf{x}_k | \omega_j) \right]$$

V-37

Soft Decision Fusion: Median, Min and Max Rules

- Assume equally likely priors
- Median Rule

$$C(\mathbf{X}) = \arg \max_{j=1, \dots, M} \left[\text{median}_{k=1, \dots, K} P(\mathbf{x}_k | \omega_j) \right]$$

- Min Rule

$$C(\mathbf{X}) = \arg \max_{j=1, \dots, M} \left[\min_{k=1, \dots, K} P(\mathbf{x}_k | \omega_j) \right]$$

- Max Rule

$$C(\mathbf{X}) = \arg \max_{j=1, \dots, M} \left[\max_{k=1, \dots, K} P(\mathbf{x}_k | \omega_j) \right]$$

V-38

Hard Decision Fusion

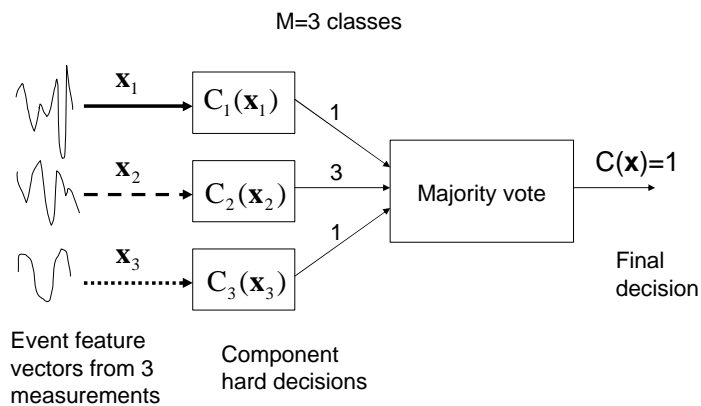
- Majority Vote Rule

$$C(\mathbf{X}) = \arg \max_{j=1, \dots, M} \left[\sum_{k=1}^K \Delta_{jk} \right]$$

$$\Delta_{jk} = \begin{cases} 1 & P(\omega_j | \mathbf{x}_k) = \max_{i=1, \dots, M} P(\omega_i | \mathbf{x}_k) \\ 0 & \text{otherwise} \end{cases} \quad \begin{matrix} j = 1, \dots, M \\ k = 1, \dots, K \end{matrix}$$

V-39

Multiple Measurement Classifier – Hard Decision Fusion



Lower computational burden

V-40

Hard Decision Fusion with Prior Beliefs

- Product Rule Based on Beliefs

Prior belief in class j given hard decisions (m_1, \dots, m_K)

$$\text{bel}(\omega_j) = \eta \prod_{k=1}^K P(\omega_j | C_k(\mathbf{x}_k) = m_k) \quad j=1, \dots, M$$

Can be estimated using the modality confusion matrices

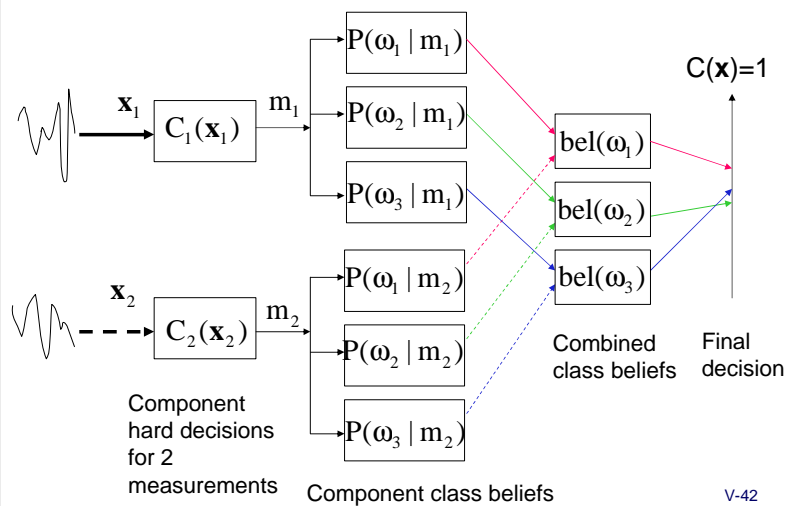
$$P(\omega_j | C_k(\mathbf{x}_k) = m_k) = \frac{n_{jm_k}^{(k)}}{\sum_{i=1}^M n_{im_k}^{(k)}} \quad \begin{array}{l} \text{(m-th column of } CM^{(k)}) \\ j=1, \dots, M \end{array}$$

$$C(\mathbf{X}) = \arg \max_{j=1, \dots, M} \text{bel}(\omega_j)$$

(Can have similar sum, median, min, max rules based on beliefs)

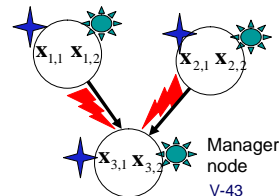
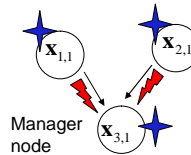
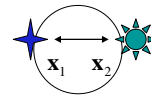
V-41

Multiple Measurement Classifier – Hard Decision Fusion With Prior Beliefs



Forms of Decision Fusion in CSP

- Decision fusion of multiple modalities (e.g. acoustic and seismic) at each node (SN, MM)
 - Relatively low comp. burden (K dim. data)
 - No additional communication burden
- Decision fusion of a single modality at multiple nodes (MN, SM)
 - Relatively low comp. burden at manager node (P dim. data)
 - Relatively low communication burden since decisions are exchanged
- Decision fusion of multiple modalities at multiple nodes (MN, MM)
 - Slightly higher comp. burden at manager node (KP dim. data)
 - Slightly higher comm. burden due to exchange of K dim. multi-modality decisions



Pros and Cons of Decision Fusion

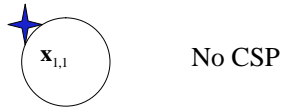
- Pros
 - No loss of information if different measurements are independent
 - Potentially as good as data fusion
 - Significantly lower computational and communication burden relative to data fusion
 - Need less training data to train each component classifier for each modality (joint training needed in data fusion)
- Cons
 - Loss of performance if measurements are correlated
 - Relatively higher communication burden when decision fusion across nodes (compared to intra-node collaboration)
 - Inconsistencies between different modalities could cause performance degradation (less severe than data fusion)

Pictorial Summary: Single Node (SN) CSP

$\mathbf{x}_{p,k}$ Event feature vector for p-th node, k-th modality

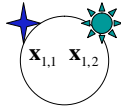
$C_{p,k}$ Component classifier for p-th node, k-th modality

1) Single node, single modality (SN, SM)



2) Single node, multiple modalities (SN, MM)

a) Decision fusion: $C(C_{1,1}(\mathbf{x}_{1,1}), C_{1,2}(\mathbf{x}_{1,2})) \rightarrow m$



b) Data fusion: $C(\mathbf{x}_{1,1}, \mathbf{x}_{1,2}) \rightarrow m$

Higher computational burden

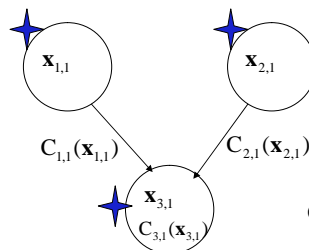
V-45

Pictorial Summary: MN-SM CSP

$\mathbf{x}_{p,k}$ Event feature vector for p-th node, k-th modality

$C_{p,k}$ Component classifier for p-th node, k-th modality

3) Multiple node, single modality (MN, SM)



a) Decision fusion:

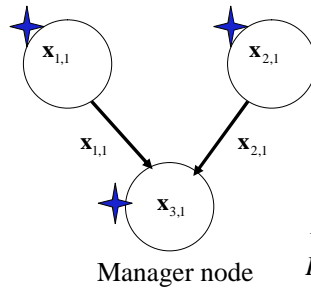
$C(C_{1,1}(\mathbf{x}_{1,1}), C_{2,1}(\mathbf{x}_{2,1}), C_{3,1}(\mathbf{x}_{3,1})) \rightarrow m$

V-46

Pictorial Summary: MN-SM CSP

$\mathbf{x}_{p,k}$ Event feature vector for p-th node, k-th modality
 $C_{p,k}$ Component classifier for p-th node, k-th modality

3) Multiple node, single modality (MN, SM)



b) Data fusion:

$$C(\mathbf{x}_{1,1}, \mathbf{x}_{2,1}, \mathbf{x}_{3,1}) \rightarrow m$$

Higher computational burden

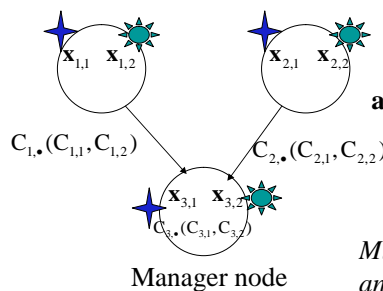
Higher communication burden

V-47

Pictorial Summary: MN-MM CSP

$\mathbf{x}_{p,k}$ Event feature vector for p-th node, k-th modality
 $C_{p,k}$ Component classifier for p-th node, k-th modality

4) Multiple nodes, multiple modalities (MN, MM)



a) Decision fusion in modalities and over nodes:

$$C(C_{1,\bullet}, C_{2,\bullet}, C_{3,\bullet}) \rightarrow m$$

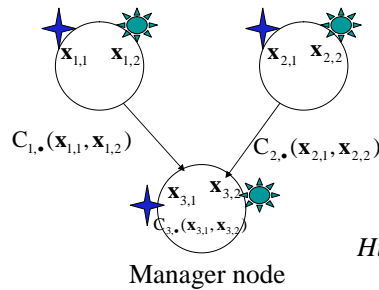
Minimal communication and computational burden

V-48

Pictorial Summary: MN-MM CSP

$\mathbf{x}_{p,k}$ Event feature vector for p-th node, k-th modality
 $C_{p,k}$ Component classifier for p-th node, k-th modality

4) Multiple nodes, multiple modalities (MN, MM)



b) Data fusion across modalities and decision fusion over nodes:

$$C(C_{1,\bullet}, C_{2,\bullet}, C_{3,\bullet}) \rightarrow m$$

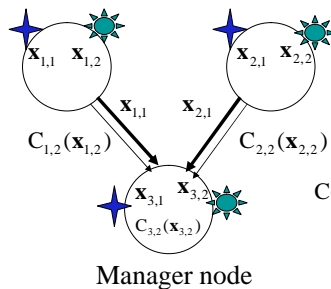
Higher computational burden

V-49

Pictorial Summary: MN-MM CSP

$\mathbf{x}_{p,k}$ Event feature vector for p-th node, k-th modality
 $C_{p,k}$ Component classifier for p-th node, k-th modality

4) Multiple nodes, multiple modalities (MN, MM)



c) Data fusion across nodes in modality 1 and decision fusion in modality 2:

$$C(C_{\bullet,1}(\mathbf{x}_{1,1}, \mathbf{x}_{2,1}, \mathbf{x}_{3,1}), C_{\bullet,2}(C_{1,2}, C_{2,2}, C_{3,2})) \rightarrow m$$

Higher computational and communication burden

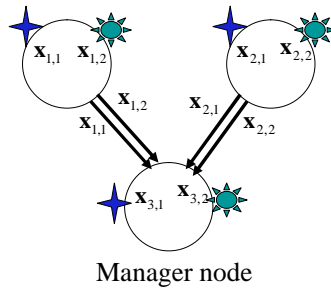
V-50

Pictorial Summary: MN-MM CSP

$\mathbf{x}_{p,k}$ Event feature vector for p-th node, k-th modality

$C_{p,k}$ Component classifier for p-th node, k-th modality

4) Multiple nodes, multiple modalities (MN, MM)



d) Data fusion across nodes in all modalities:

$$C(\mathbf{x}_{1,1}, \mathbf{x}_{1,2}, \mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \mathbf{x}_{3,1}, \mathbf{x}_{3,2}) \rightarrow m$$

Highest computational and communication burden

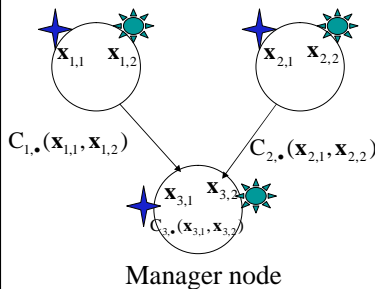
V-51

Pictorial Summary: MN-MM CSP

$\mathbf{x}_{p,k}$ Event feature vector for p-th node, k-th modality

$C_{p,k}$ Component classifier for p-th node, k-th modality

4) Multiple nodes, multiple modalities (MN, MM)



d) Data fusion across nodes in all modalities:

$$C(\mathbf{x}_{1,1}, \mathbf{x}_{1,2}, \mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \mathbf{x}_{3,1}, \mathbf{x}_{3,2}) \rightarrow m$$

Independent nodes \leftrightarrow 4 b)

$$C(C_{1,\bullet}, C_{2,\bullet}, C_{3,\bullet}) \rightarrow m$$

Lower communication and computational burden

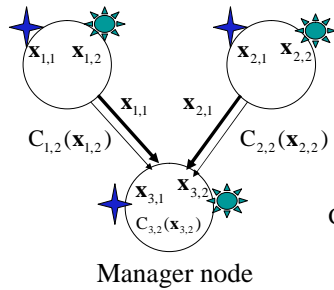
V-52

Pictorial Summary: MN-MM CSP

$\mathbf{x}_{p,k}$ Event feature vector for p-th node, k-th modality

$C_{p,k}$ Component classifier for p-th node, k-th modality

4) Multiple nodes, multiple modalities (MN, MM)



d) Data fusion across nodes in all modalities:

$$C(\mathbf{x}_{1,1}, \mathbf{x}_{1,2}, \mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \mathbf{x}_{3,1}, \mathbf{x}_{3,2}) \rightarrow m$$

Indep. nodes in mod. 2 \leftrightarrow 4 c)

$$C(C_{\bullet,1}(\mathbf{x}_{1,1}, \mathbf{x}_{2,1}, \mathbf{x}_{3,1}), C_{\bullet,2}(C_{1,2}, C_{2,2}, C_{3,2})) \rightarrow m$$

Medium communication and computational burden

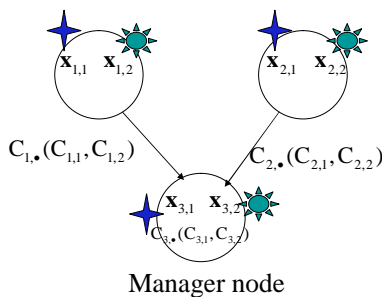
V-53

Pictorial Summary: MN-MM CSP

$\mathbf{x}_{p,k}$ Event feature vector for p-th node, k-th modality

$C_{p,k}$ Component classifier for p-th node, k-th modality

4) Multiple nodes, multiple modalities (MN, MM)



d) Data fusion across nodes in all modalities:

$$C(\mathbf{x}_{1,1}, \mathbf{x}_{1,2}, \mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \mathbf{x}_{3,1}, \mathbf{x}_{3,2}) \rightarrow m$$

Indep. nodes and mods \leftrightarrow 4 a)

$$C(C_{1,\bullet}, C_{2,\bullet}, C_{3,\bullet}) \rightarrow m$$

Lowest communication and computational burden

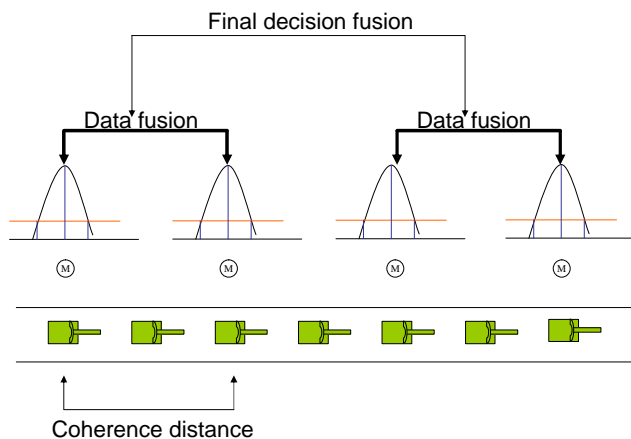
V-54

When to Use Data Fusion Versus Decision Fusion?

- Independent measurements → decision fusion
 - E.g., if different measurements are sufficiently separated in time (and hence space)
- Data fusion across multiple modalities at each node is desirable due to low communication burden
- Decision fusion across nodes is desirable due to lower communication cost
- Inconsistencies between different measurements, such as due to malfunctioning nodes, should also be taken into account
 - Decision fusion may be less sensitive to such inconsistencies

V-55

Sequential Decision Making: Combining Data and Decision Fusion



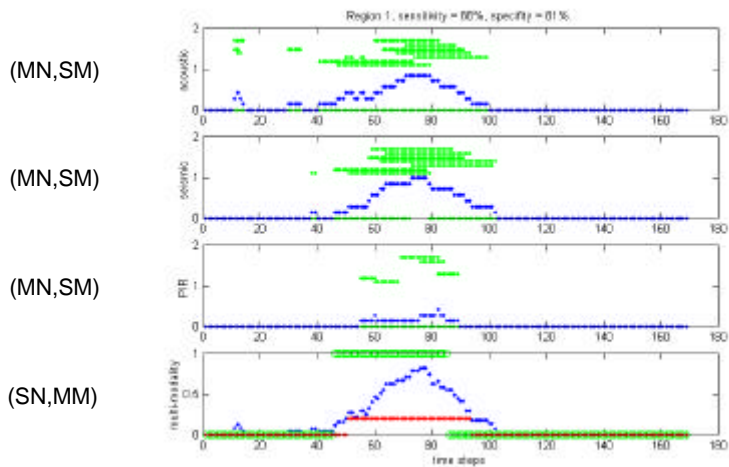
V-56

Illustrative Numerical Results

- SITEX 02 data from DARPA SenseIT project
- Multi-node multiple modality detection
- Classification
 - Gaussian classifiers
 - NN classifier as a benchmark
 - SN-SM
 - SN-MM
 - MN-SM

V-57

Multi-Node Multiple Modality Detection



Green: Node detections Blue: Average of Green Red: Average of Blue v-58

Feature Vectors for Classification

- A classifier operates on some feature vector for each event extracted from the measurement
- What features should be used?
- The answer depends on the characteristics of target signatures
- Often spectral features are useful, particularly when strong harmonic characteristics are present
 - For example, if the target signatures are produced by rotating machinery (engine, gears)
 - FFT of event time series is used as a feature
- Other quantities, such as zero-crossings, moment estimates etc., may also be used

V-59

Spectral Feature Characteristics

- Two modalities: Acoustic and seismic
 - Sampling rate 4960 Hz
- Acoustic signals
 - 512-pt FFT of 512-sample (non-overlapping) segments
 - 9.69 Hz FFT resolution
 - The first 100 positive frequency FFT samples used (969 Hz)
 - 2-pt averaging of the 100 FFT samples yields the final $N=50$ dimensional FFT feature vectors
 - 19.38 Hz resolutions
 - About 50-200 feature vectors in each event depending on the vehicle
 - Event feature vector matrix \mathbf{X} is 50×50 to 50×200
 - 50 dimensional mean event feature vectors \mathbf{x}
- Complex or absolute value FFT features

V-60

Seismic Feature Characteristics

- Seismic signals
 - Sampling rate reduction from 4960 Hz to 512 Hz
 - 512-pt FFT of 512-sample (256-overlap) segments
 - 1 Hz resolution
 - The first 100 positive frequency FFT samples used (100 Hz)
 - 2-pt averaging of the 100 FFT samples yields the final N=50 dimensional FFT feature vectors
 - 2 Hz resolutions
 - About 10-50 feature vectors in each event depending on the vehicle
 - Event feature vector matrix \mathbf{X} is 50x10 to 50x50
 - 50 dimensional mean event feature vectors \mathbf{x}
- Complex or absolute value FFT features

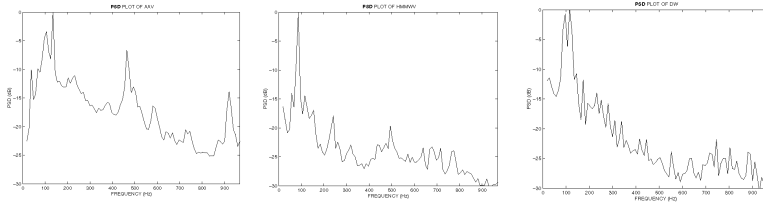
V-61

Class Descriptions

- Tracked vehicle class: AAV (Amphibious Assault Vehicle)
- Wheeled vehicle class: DW (Dragon Wagon) and HMWV (Humvee)
- Locomotion Class and Vehicle Class classification
- Approximately equal number of training and testing events for all classes
- 3-way cross validation for performance assessment

V-62

Representative Acoustic FFT Features



AAV – tracked
(Amphibious
Assault Vehicle)



HMV – wheeled
(Humvee)

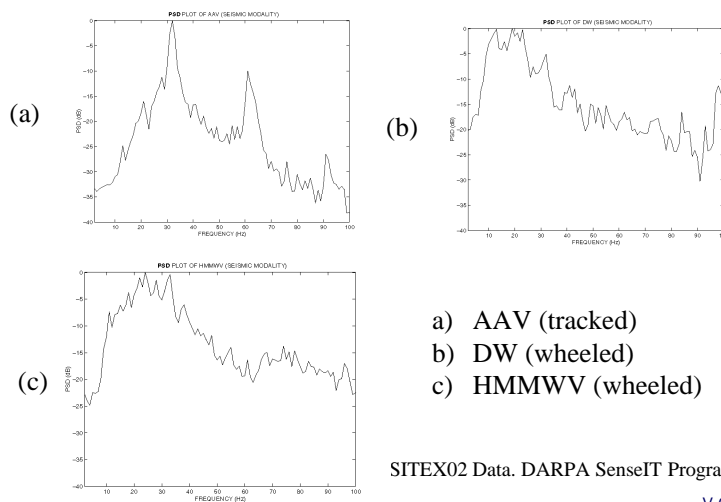


DW – wheeled
(Dragon Wagon)

SITEX02 Data. DARPA SenseIT Program.

V-63

Representative Seismic FFT Features



(a)

(b)

(c)

- a) AAV (tracked)
- b) DW (wheeled)
- c) HMMWV (wheeled)

SITEX02 Data. DARPA SenseIT Program.

V-64

Single Node Single Modality (SN, SM) – Locomotion Class

Absolute-value FFT acoustic features

Gaussian Classifier

NN Classifier (benchmark)

| $\omega_m \backslash C(X)$ | Wheeled | Tracked |
|----------------------------|---------|---------|
| Wheeled | 109 | 11 |
| Tracked | 22 | 98 |

| $\omega_m \backslash C(X)$ | Wheeled | Tracked |
|----------------------------|---------|---------|
| Wheeled | 102 | 18 |
| Tracked | 1 | 119 |

PD = 0.91, 0.82, Ave = 0.86

PD = 0.85, 0.99, Ave = 0.92

PFA = 0.18, 0.09

PFA = 0.01, 0.15

120 events for each class

V-65

Single Node Single Modality (SN, SM) – Vehicle Class

Absolute-value FFT acoustic features

Gaussian Classifier

NN Classifier (benchmark)

| $\omega_m \backslash C(X)$ | AAV | DW | HMV |
|----------------------------|-----|----|-----|
| AAV | 53 | 5 | 2 |
| DW | 12 | 42 | 6 |
| HMV | 15 | 14 | 31 |

| $\omega_m \backslash C(X)$ | AAV | DW | HMV |
|----------------------------|-----|----|-----|
| AAV | 43 | 9 | 8 |
| DW | 0 | 49 | 11 |
| HMV | 1 | 13 | 46 |

PD = 0.88, 0.70, 0.52, Ave = 0.70

PD = 0.72, 0.82, 0.77, Ave = 0.77

PFA = 0.22, 0.16, 0.07

PFA = 0.01, 0.18, 0.16

60 events for each vehicle

V-66

Single Node Multiple Modality (SN, MM) Data Fusion – Locomotion Class

Acoustic and seismic features

Gaussian Classifier

NN Classifier (benchmark)

| $\omega_m \backslash C(X)$ | Wheeled | Tracked |
|----------------------------|---------|---------|
| Wheeled | 117 | 3 |
| Tracked | 25 | 95 |

| $\omega_m \backslash C(X)$ | Wheeled | Tracked |
|----------------------------|---------|---------|
| Wheeled | 106 | 14 |
| Tracked | 4 | 116 |

PD = 0.97, 0.80, Ave = 0.88

PD = 0.88, 0.97, Ave = 0.92

PFA = 0.21, 0.02

PFA = 0.03, 0.12

120 events for each class

V-67

Single Node Multiple Modality (SN, MM) Data Fusion – Vehicle Class

Acoustic and seismic features

Gaussian Classifier

NN Classifier (benchmark)

| $\omega_m \backslash C(X)$ | AAV | DW | HMV |
|----------------------------|-----|----|-----|
| AAV | 59 | 0 | 1 |
| DW | 9 | 46 | 5 |
| HMV | 25 | 12 | 23 |

| $\omega_m \backslash C(X)$ | AAV | DW | HMV |
|----------------------------|-----|----|-----|
| AAV | 43 | 6 | 11 |
| DW | 0 | 47 | 13 |
| HMV | 1 | 22 | 37 |

PD = 0.98, 0.77, 0.38, Ave = 0.71

PD = 0.72, 0.78, 0.62, Ave = 0.71

PFA = 0.28, 0.10, 0.05

PFA = 0.01, 0.23, 0.20

60 events for each vehicle

V-68

Single Node Multiple Modality (SN, MM) Decision Fusion – Locomotion Class

Acoustic and seismic features

Gaussian Classifier

Sum Rule

| $\omega_m \backslash C(X)$ | Wheeled | Tracked |
|----------------------------|---------|---------|
| Wheeled | 110 | 10 |
| Tracked | 32 | 88 |

PD = 0.92, 0.73, Ave = 0.83

PFA = 0.27, 0.08

Product Rule

| $\omega_m \backslash C(X)$ | Wheeled | Tracked |
|----------------------------|---------|---------|
| Wheeled | 116 | 4 |
| Tracked | 45 | 75 |

PD = 0.97, 0.63, Ave = 0.79

PFA = 0.37, 0.03

120 events for each class

V-69

Single Node Multiple Modality (SN, MM) Decision Fusion – Vehicle Class

Acoustic and seismic features

Gaussian Classifier

Sum Rule

| $\omega_m \backslash C(X)$ | AAV | DW | HMV |
|----------------------------|-----|----|-----|
| AAV | 55 | 5 | 0 |
| DW | 8 | 44 | 8 |
| HMV | 20 | 13 | 27 |

PD = 0.92, 0.73, 0.45, Ave = 0.70

PFA = 0.23, 0.15, 0.07

Product Rule

| $\omega_m \backslash C(X)$ | AAV | DW | HMV |
|----------------------------|-----|----|-----|
| AAV | 59 | 0 | 1 |
| DW | 21 | 34 | 5 |
| HMV | 27 | 10 | 23 |

PD = 0.98, 0.57, 0.38, Ave = 0.64

PFA = 0.4, 0.08, 0.05

60 events for each vehicle

V-70

Comparison of Various Forms of CSP – Locomotion Class

Gaussian Classifier

(SN, SM)

| $\omega_m \backslash C(X)$ | Wheeled | Tracked |
|----------------------------|---------|---------|
| Wheeled | 109 | 11 |
| Tracked | 22 | 98 |

PD = 0.91, 0.82,
Ave = 0.86
PFA = 0.18, 0.09

(SN, MM) – Data Fusion (SN, MM) – Dec. Fusion

| $\omega_m \backslash C(X)$ | Wheeled | Tracked |
|----------------------------|---------|---------|
| Wheeled | 117 | 3 |
| Tracked | 25 | 95 |

PD = 0.97, 0.80,
Ave = 0.88
PFA = 0.21, 0.02

| $\omega_m \backslash C(X)$ | Wheeled | Tracked |
|----------------------------|---------|---------|
| Wheeled | 110 | 10 |
| Tracked | 32 | 88 |

PD = 0.92, 0.73,
Ave = 0.83
PFA = 0.27, 0.08

V-71

Comparison of Various Forms of CSP – Vehicle Class

Gaussian Classifier

(SN, SM)

| $\omega_m \backslash C(X)$ | AAV | DW | HMV |
|----------------------------|-----|----|-----|
| AAV | 53 | 5 | 2 |
| DW | 12 | 42 | 6 |
| HMV | 15 | 14 | 31 |

PD = 0.88, 0.70, 0.52,
Ave = 0.70
PFA = 0.22, 0.16, 0.07

(SN, MM) – Data Fusion

| $\omega_m \backslash C(X)$ | AAV | DW | HMV |
|----------------------------|-----|----|-----|
| AAV | 59 | 0 | 1 |
| DW | 9 | 46 | 5 |
| HMV | 25 | 12 | 23 |

PD = 0.98, 0.77, 0.38,
Ave = 0.71
PFA = 0.28, 0.10, 0.05

(SN, MM) – Dec. Fusion

| $\omega_m \backslash C(X)$ | AAV | DW | HMV |
|----------------------------|-----|----|-----|
| AAV | 55 | 5 | 0 |
| DW | 8 | 44 | 8 |
| HMV | 20 | 13 | 27 |

PD = 0.92, 0.73, 0.45,
Ave = 0.70
PFA = 0.23, 0.15, 0.07

Inconsistencies between modalities are present

V-72

Multiple Node Single Modality CSP

- Estimated covariance matrices for different classes at different nodes from SITEX 02 data (Acoustic)
- Synthetically generated noisy measurements at different nodes using the covariance matrices

$$\mathbf{S}_{p,j} = \mathbf{U}_{j,p} \mathbf{\Lambda}_{j,p} \mathbf{U}_{j,p}^H \quad \text{Eigen decomposition}$$

$$\mathbf{x}_{p,j} = \mathbf{U}_{j,p} \mathbf{\Lambda}_{j,p}^{1/2} \mathbf{z} + \mathbf{n}_{j,p} \quad \begin{array}{l} \mathbf{z}: \text{iid unit variance vector} \\ \mathbf{n}: \text{AWGN (20dB SNR)} \end{array}$$

- Compared the performance of (SN,SM) versus (MN,SM) CSP – data fusion and decision fusion

V-73

Comparison of Various Forms of (MN,SM) CSP – Locomotion Class

Gaussian Classifier

(SN, SM)

| $\omega_m \backslash C(\mathbf{X})$ | Wheeled | Tracked |
|-------------------------------------|---------|---------|
| Wheeled | 337 | 163 |
| Tracked | 120 | 380 |

PD = 0.67, 0.76,
Ave = 0.72
PFA = 0.24, 0.32

(MN, SM) – Data Fusion

| $\omega_m \backslash C(\mathbf{X})$ | Wheeled | Tracked |
|-------------------------------------|---------|---------|
| Wheeled | 396 | 104 |
| Tracked | 83 | 417 |

PD = 0.79, 0.83,
Ave = 0.81
PFA = 0.17, 0.21

(MN, SM) – Dec. Fusion
(sum rule)

| $\omega_m \backslash C(\mathbf{X})$ | Wheeled | Tracked |
|-------------------------------------|---------|---------|
| Wheeled | 342 | 158 |
| Tracked | 63 | 437 |

PD = 0.68, 0.87,
Ave = 0.78
PFA = 0.13, 0.32

V-74

Multiple Target Classification

- Key challenge: interference between different target signals
- A solution: orthogonal class subspaces
- Define mixture correlation matrix:

$$\mathbf{C} = E[\mathbf{x}\mathbf{x}^H] = \sum_{i=1}^M P(\omega_i) E[\mathbf{x}_i \mathbf{x}_i^H]$$

- There exists a matrix \mathbf{B} such that $\mathbf{B}^H \mathbf{C} \mathbf{B} = \mathbf{I}$

- Transform feature vectors: $\mathbf{g} = \mathbf{B}^H \mathbf{x}$

- Transformed class correlation matrices

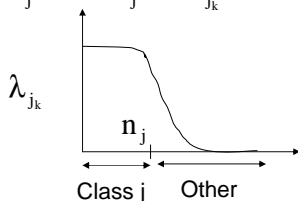
$$\mathbf{C}_j = P(\omega_j) E[\mathbf{g}_j \mathbf{g}_j^H] = \mathbf{A}_j \mathbf{\Lambda}_j \mathbf{A}_j^H \quad (\text{eigen-decomposition})$$

V-75

Orthogonal Projection Classifiers

- Then $\mathbf{C}_j = \mathbf{I} - \tilde{\mathbf{C}}_j = \mathbf{I} - \sum_{\substack{i=1 \\ i \neq j}}^M \mathbf{C}_i$
- \mathbf{C}_j and $\tilde{\mathbf{C}}_j$ share the same eigenfunctions and

$$\mathbf{C}_j = \mathbf{I} - \tilde{\mathbf{C}}_j \Leftrightarrow \lambda_{j_k} = 1 - \tilde{\lambda}_{j_k}$$



- Projection classifier for class j $\mathbf{P}_j = \sum_{k=1}^{n_j} \mathbf{a}_{j_k} \mathbf{a}_{j_k}^H$

Summary – CSP Techniques

- Some form of CSP is in general needed to yield reliable performance in sensor networks
 - Harsh and variable environmental conditions
 - Limited capabilities of sensor nodes
- Key is to combine minimum amount of information that yields desired performance (energy conservation):
 - Minimize computational burden at each node
 - Minimize communication burden across nodes
- Two main forms of CSP: decision fusion and data fusion
 - Decision fusion is desirable whenever feasible
 - Inconsistencies between modalities may compromise the performance gains due to CSP
 - Fusion of complementary modalities/measurements

V-77

Target Localization

- Estimation of target location is needed for tracking
- Some existing techniques for target localization:
 - Beamforming and triangulation using multiple direction-finding sensor arrays within the active region (a sensor array on each node). (*Chen, Yao, and Hudson, IEEE Sig. Proc. Magazine, March 2002.*)
 - Time delay estimation between multiple sensors (on the same node or different nodes). (*Boettcher and Shaw, 4th Int. Conf. Information Fusion, 2001.*)
 - Closest Point of Approach (CPA) based --- crude
 - Energy-based localization: using power decay exponents and energy measurements at different nodes. (*Li, Wong, Hu, and Sayeed, IEEE Sig. Proc. Magazine, March 2002.*)

V-78

Energy Based Localization

- Energy decay assumption
 - Signal emitted from a source decays as a power law as a function of source-sensor distance
- If energy decay exponent is a constant, the distance between source and sensor can be estimated from energy readings at individual sensors

$$y_i(t) = \frac{g_i \cdot s(t)}{\|\mathbf{r}(t) - \mathbf{r}_i\|^\alpha} + \varepsilon_i(t)$$

$y_i(t)$: energy measured at i^{th} sensor node at time t

g_i : sensor gain $s(t)$: source energy at time t

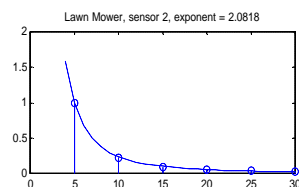
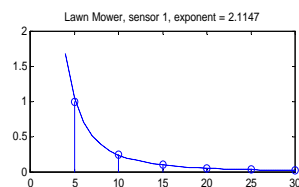
$\mathbf{r}(t)$: source location at time t \mathbf{r}_i : i^{th} sensor location

α : energy decay exponent ε_i : additive noise.

V-79

Energy Decay Profile

For acoustic sensors, within 30 meters from the source, the energy decay exponent was experimentally measured to be approximately = 2



V-80

Direct Parameter Estimation

- Consider the equations

$$y_i(t) = \frac{\mathbf{g}_i \cdot \mathbf{s}(t)}{\|\mathbf{r}(t) - \mathbf{r}_i\|^\alpha} + \varepsilon_i(t)$$

- Given $\{y_i(t), \mathbf{r}_i, \mathbf{g}_i; i = 1, 2, \dots\}$ and α , we want to estimate $\mathbf{r}(t)$ and $\mathbf{s}(t)$

- Assume $\varepsilon_i(t) \sim N(0, \sigma^2)$, the negative log-likelihood function is

$$\ell(\mathbf{r}(t), \mathbf{s}(t)) \propto \sum_i \left| y_i(t) - \frac{\mathbf{g}_i \cdot \mathbf{s}(t)}{\|\mathbf{r}(t) - \mathbf{r}_i\|^\alpha} \right|^2$$

- Minimizing $\ell(\mathbf{r}(t), \mathbf{s}(t))$ with respect to $\mathbf{r}(t)$ and $\mathbf{s}(t)$ leads to the ML estimate of $\mathbf{r}(t)$ and $\mathbf{s}(t)$

- Gradient descent search:

$$\frac{\partial \ell}{\partial \mathbf{s}(t)} = - \sum_i \frac{\varepsilon_i(t) \mathbf{g}_i}{\|\mathbf{r}(t) - \mathbf{r}_i\|^\alpha}$$

$$\begin{aligned} \nabla_{\mathbf{r}(t)} \ell &= -2 \sum_i \varepsilon_i(t) \cdot \frac{\mathbf{s}(t) \mathbf{g}_i (\mathbf{r}(t) - \mathbf{r}_i)}{\|\mathbf{r}(t) - \mathbf{r}_i\|^{\alpha+2}} \\ &= -2 \sum_i \varepsilon_i(t) \cdot (y_i(t) - \varepsilon_i(t)) \cdot \frac{\mathbf{r}(t) - \mathbf{r}_i}{\|\mathbf{r}(t) - \mathbf{r}_i\|^2} \end{aligned}$$

$$\text{where } \varepsilon_i(t) = y_i(t) - \frac{\mathbf{s}(t) \mathbf{g}_i}{\|\mathbf{r}(t) - \mathbf{r}_i\|^\alpha}$$

- May be trapped in a local minimum of the cost function

V-81

Energy Based Localization

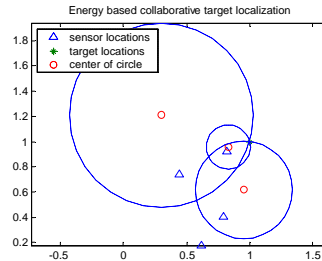
- Eliminate dependence on $\mathbf{s}(t)$ by using ratios $y_i(t)/y_k(t)$

- Ignoring noise, define $\kappa_{ik}^2 = \frac{\|\mathbf{r}(t) - \mathbf{r}_k\|^2}{\|\mathbf{r}(t) - \mathbf{r}_i\|^2} = \left(\frac{y_i(t)/\mathbf{g}_i}{y_k(t)/\mathbf{g}_k} \right)^{-2/\alpha}$

- After simplification we get

$$\left\| \mathbf{r}(t) - \frac{\mathbf{r}_i - \kappa_{ik}^2 \mathbf{r}_k}{1 - \kappa_{ik}^2} \right\|^2 = \frac{\kappa_{ik}^2 \|\mathbf{r}_i - \mathbf{r}_k\|^2}{(1 - \kappa_{ik}^2)^2}$$

$$\text{or } \left\| \mathbf{r}(t) - \mathbf{c}_{ik} \right\|^2 = \rho^2 \quad (*)$$



- Implication:** All the feasible target locations $\mathbf{r}(t)$ that yield $y_i(t)$ and $y_k(t)$ must fall on a circle centered at \mathbf{c}_{ik} with radius ρ , regardless of the source energy $\mathbf{s}(t)$

V-82

Energy Based Localization Algorithm

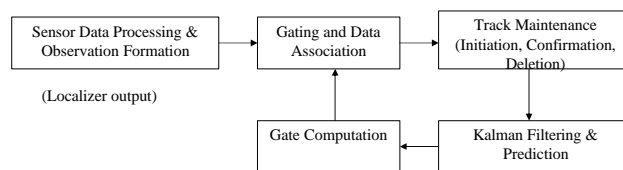
- In theory, at least four sensor nodes are needed to uniquely determine a single target's position
- Given P sensors, P(P-1)/2 energy ratios can be formed
- However, only (P-1) of the ratios (and corresponding circles) are independent
- Factors affecting location estimate accuracy:
 - Errors in energy reading y(t)
 - Errors in sensor locations r_i
 - Errors in decay exponent α
 - Variations in sensor gains g_i
- The (P-1) circles may not intersect at a unique point
- Minimize a nonlinear cost function (local minima possible)

$$J(x, y) = \sum_{i=1}^K \left\| \sqrt{(x - c_{i,x})^2 + (y - c_{i,y})^2} - \rho_i \right\|^2$$

Li, Wong, Hu and Sayeed, IEEE Sig. Proc. Magazine, March 2002

V-83

Basic Elements of a Multiple Target Tracking (MTT) System



Functional Elements of a Recursive Basic Multi-Target Tracking System

Detection and localization algorithms do the necessary sensor data processing to form the observations

Observations are gated with current tracks first
Then more refined data association algorithms are used for final assignments

Observations not assigned to existing tracks initiate new tentative tracks

A tentative track becomes a confirmed track on satisfying some quality test.

A Kalman filter (or other least-squares methods) may be used to predict the track state at future times

The prediction covariance matrix is used to make the gates for next data association recursion

Low quality tracks, as determined by the update history, are deleted

V-84

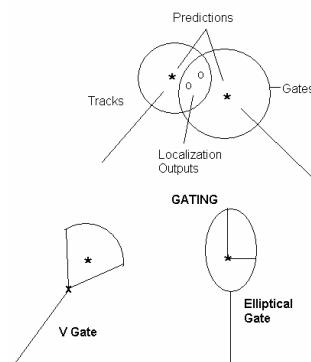
Track Definition

- It is a symbolic representation of a target moving through an area of interest
- Internally, it is represented by a *filter state* which gets updated on each new measurement
- The filter state may include:
 - Past history
 - Track ID, type, quality
 - Present and future (predicted) states, and the covariance matrices in case of Kalman filtering
- Number of tracks a MTT system can process depends on a variety of factors, including the number of nodes used and the accuracy of component algorithms

V-85

Gating and Data Association

- **Gating:** The process of specifying a region for predicting the next localizer output
- Helps in eliminating false localizations
- More sophisticated data association algorithms may be used on the gated localizations
- Different types of gates can be used depending upon data characteristics and physical constraints. E.g., circular, elliptical & V gates.
- The size of the gate is calculated at each recursion from the prediction covariance matrix
- Tracks which share localizations with each other are grouped together into "**clusters**"
- Data association algorithms are run on these clusters.



V-86

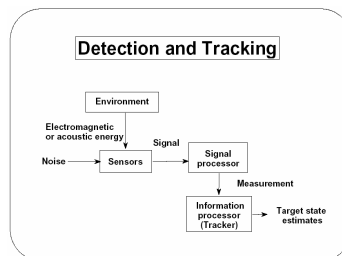
Track Maintenance

- Tracks are maintained for their quality at each iteration
 - Tracks which receive new localizations after gating are updated for predicting track locations (e.g. using a Kalman filter)
 - Tracks which do not receive new localizations are declared miss-detected and hence penalized. Tracks which suffer a certain number of miss-detections are deleted.
 - New tracks, called tentative tracks, are created from unpaired new localizations. Tentative tracks get confirmed after few successive updates.

V-87

Target Dynamics Estimation

- Uses current localization estimates to give more accurate current and future estimates
- State estimation has its basics in least squares estimation
- Kalman Filter (KF) is a good candidate for doing recursive estimation:
 - Process noise can take care of target motion
 - Automatic KF gain adjustment allows appropriate combining of predicted locations and new localizer outputs
 - Estimation accuracy through covariance matrices that are helpful in data association
 - Prediction covariance matrix controls miss-detections
- Can use interacting multiple KFs in parallel to track multiple maneuvering targets



V-88

Kalman Filter

- State of localizer outputs modeled by a linear stochastic difference equation

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{w}(k+1)$$

$$\mathbf{x}(k) = [x \quad v_x \quad y \quad v_y]^T \quad \mathbf{A} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{array}{l} \text{State Transition} \\ \text{Matrix} \end{array}$$

- Constant velocity model – acceleration modeled as white noise $\mathbf{w}(k)$
- Measurement model: $\mathbf{z}(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{v}(k)$ (location estimate)
- Measurement matrix \mathbf{H}

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

- $\mathbf{w}(k)$ and $\mathbf{v}(k)$ represent the process and measurement noise. They are assumed to be iid, independent, and Gaussian

$$p(\mathbf{w}) \sim N(0, \mathbf{Q}) \quad p(\mathbf{v}) \sim N(0, \mathbf{R})$$

V-89

Kalman Filter Iteration

The localizer output $\mathbf{z}(k)$ is used to update the KF

Filtered state: $\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{K}(k)[\mathbf{z}(k) - \mathbf{H}\hat{\mathbf{x}}(k|k-1)]$

Gain update: $\mathbf{K}(k) = \mathbf{P}(k|k-1)\mathbf{H}^T[\mathbf{H}\mathbf{P}(k|k-1)\mathbf{H}^T + \mathbf{R}]^{-1}$

Filt. state cov. update: $\mathbf{P}(k|k) = [\mathbf{I} - \mathbf{K}(k)\mathbf{H}]\mathbf{P}(k|k-1)$

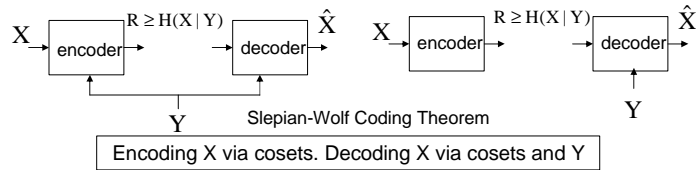
Prediction: $\hat{\mathbf{x}}(k+1|k) = \mathbf{A}\hat{\mathbf{x}}(k|k)$

Pred. Cov. update: $\mathbf{P}(k+1|k) = \mathbf{A}\mathbf{P}(k|k)\mathbf{A}^T + \mathbf{Q}$

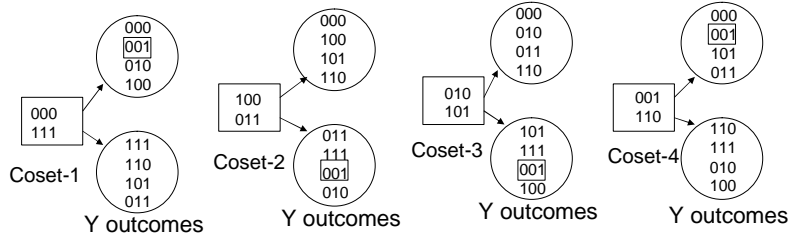
Covariance matrices \mathbf{Q} and \mathbf{R} are initialized by prior estimation from training data

V-90

Distributed Compression



X and Y iid 3-bit words. Correlated \leftrightarrow maximum hamming dist. =1

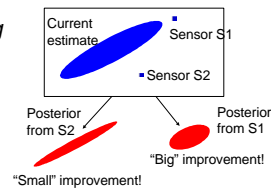


Pradhan, Kusuma and Ramchandran, IEEE Signal Proc. Mag., March 2002. V-91

Information Driven Sensor Querying

Select next sensors to collaborate so as to *maximize information* return while *minimizing latency & bandwidth* consumption

- Routing protocol to automatically direct a sensor query into regions of high information content (information utility measure)
- Tradeoff between maximum information gain and minimum transmission cost



Define dynamic constraints:

$$H(\bar{x}) = \mathbf{a} (\bar{x} - \bar{x}_T)^T \hat{\Sigma}^{-1} (\bar{x} - \bar{x}_T) + (1 - \mathbf{a}) (\bar{x} - \bar{x}_S)^T (\bar{x} - \bar{x}_S)$$

Information utility:
Mahalanobis Distance

Energy utility:
Query path length

\bar{x} Routing node position

\bar{x}_T Target position

\bar{x}_S Querying sensor position

\mathbf{a} "Tradeoff" parameter

Feng Zhao, James Reich, Juan Julia Liu, Jie Liu, Patrick Cheung Palo Alto Research Center.
IEEE Sig. Proc. Mag., March 2002.

V-92

Challenges

- Uncertainty in temporal and spatial measurements critically affects localization, beamforming and bearing estimation:
 - Uncertainty in node locations
 - Uncertainty in timing and synchronization
- Variability in signal characteristics:
 - Doppler shifts due to motion
 - Gear shifts, acceleration in vehicles
- Variability in environmental/sensor conditions:
 - Most algorithms exploit prior statistical information about sources
 - Observed statistical characteristics can vary markedly depending on environmental conditions, such as terrain, foliage, rain, wind etc.
 - Variability in sensor characteristics (e.g., gain calibration)
- A key challenge is to develop CSP algorithms that are robust to such uncertainty/variability in measurements and conditions

V-93

Acknowledgements

- Wisconsin: Yu Hen Hu (detection and localization), Ahtasham Ashraf (Tracking), Ashwin D'Costa (classification)
- PARC: Feng Zhao and Jim Reich (information based sensor querying)
- Michigan: Sandeep Pradhan (distributed compression)
- Penn State –ARL: Richard Brooks (Tracking)

V-94

References

- Kumar, Zhao and Shepherd Guest Eds., IEEE Signal Processing Magazine, March 2002, Special Issue on Collaborative Signal and Information Processing in Microsensor Networks.
- Tracking:
 - Bar-Shalom and Li, *Estimation & Tracking: Principles & Techniques*.
 - Blackman and Popoli, *Design and Analysis of Modern Tracking Systems* Applied Optimal Estimation.
 - Stone, Barlow and Corwin, *Bayesian Multiple Target Tracking*.
 - Bar-Shalom and Li, *Multitarget-Multisensor Tracking : Principles and Techniques*.
 - Brooks and Iyengar, *Multi-Sensor Fusion*.

V-95

References

- Classification:
 - Duda, Hart and Stork, *Pattern Classification*, 2nd Ed, Wiley.
 - Kittler, Hatef, Duin and Matas, *On Combining Classifiers*, IEEE Transactions on Pattern Analysis and Machine Intelligence, March 1998.
 - Foucher, Germain, Boucher, and Benie, *Multisource Classification Using ICM and Demster-Shafer Theory*, IEEE Transactions on Instrumentation and Measurement, April 2002.
 - Kittler, A Method for Determining Class Subspaces, Information Processing Letters, pp. 77-79, June 1977.
- Recent and upcoming special issues in various journals
 - IEEE/ACM Transactions on Networking
 - IEEE JSAC
 - Journal on High Performance Computing

V-96