KASSPER
Real-Time Embedded Signal Processor Testbed

Glenn Schrader

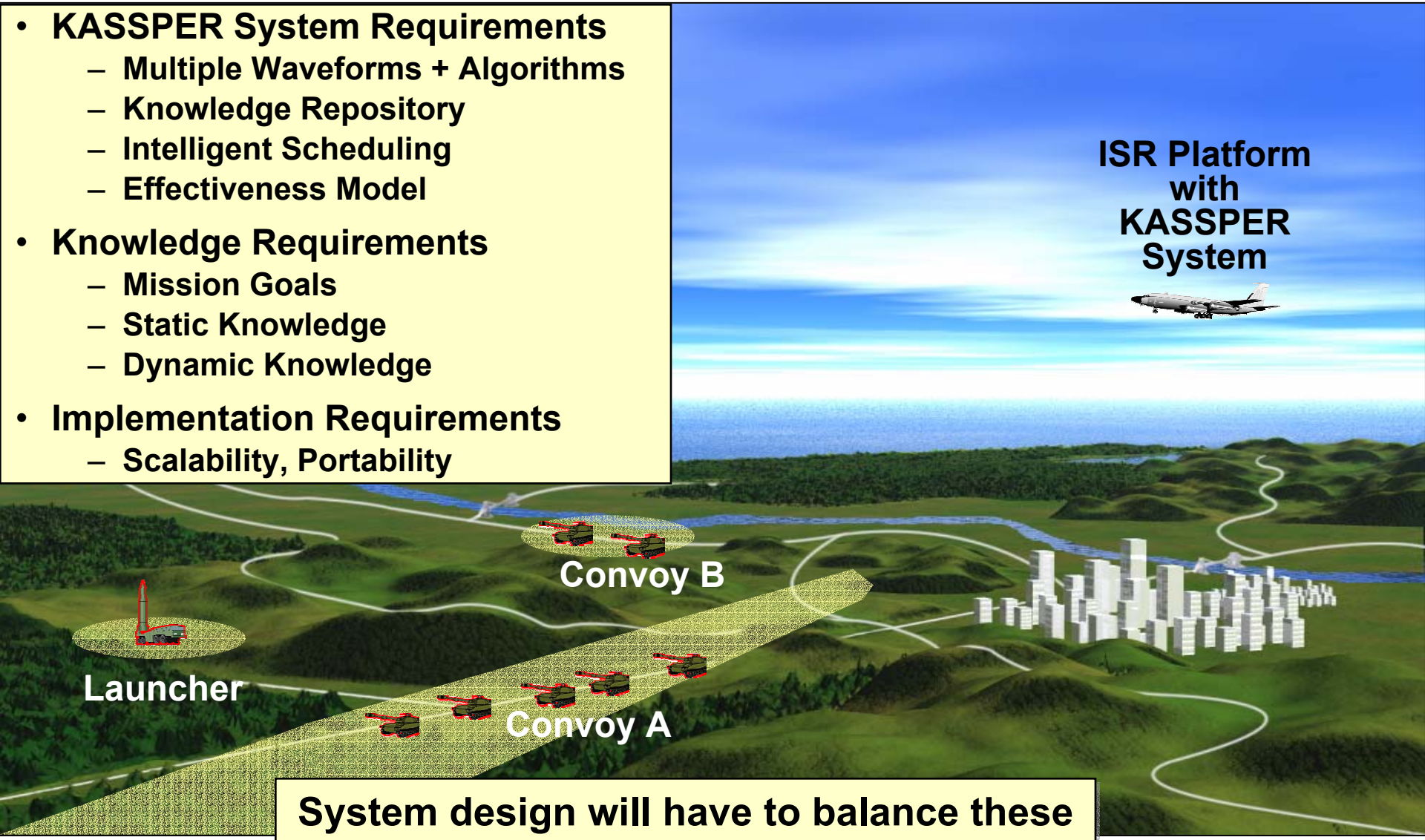
Massachusetts Institute of Technology
Lincoln Laboratory

5-7 April 2004

This work is sponsored by the Defense Advanced Research Projects Agency, under Air Force Contract F19628-00-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the author and are not necessarily endorsed by the United States Government.

Representative KASSPER Problem

- **KASSPER System Requirements**
  - Multiple Waveforms + Algorithms
  - Knowledge Repository
  - Intelligent Scheduling
  - Effectiveness Model
- **Knowledge Requirements**
  - Mission Goals
  - Static Knowledge
  - Dynamic Knowledge
- **Implementation Requirements**
  - Scalability, Portability

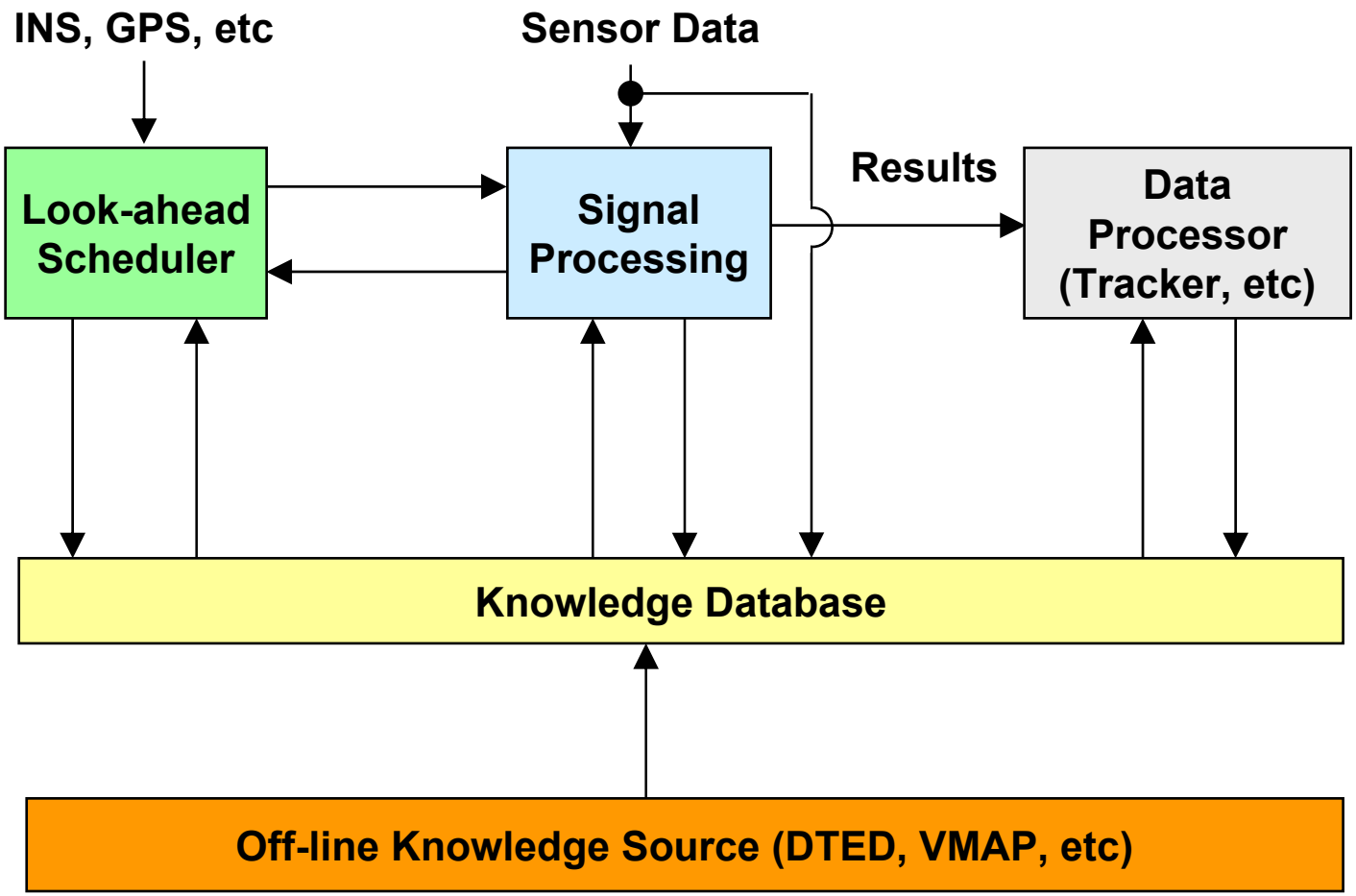


System design will have to balance these (possibly competing) requirements

Outline

- System Architecture
- Knowledge Database
- Look-ahead Scheduling
- Knowledge Data Case Study
- Summary

High Level KASSPER Architecture



Signal Processing Architecture

Data Input

Sensor Data

Command

INS, GPS, etc

SAR Algorithms

Parametric Tracker
Courtesy of Jon Jones (AFRL)

GMTI Algorithms

Look-ahead Scheduler

Data Processor (Tracker, etc)

Data Output

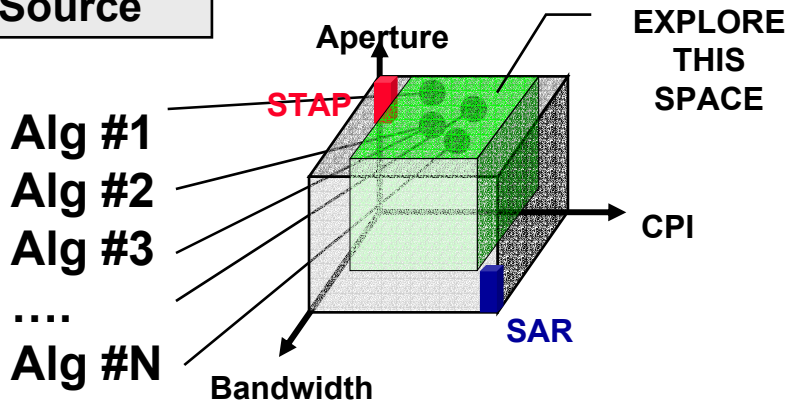
CPU Load

Knowledge Database

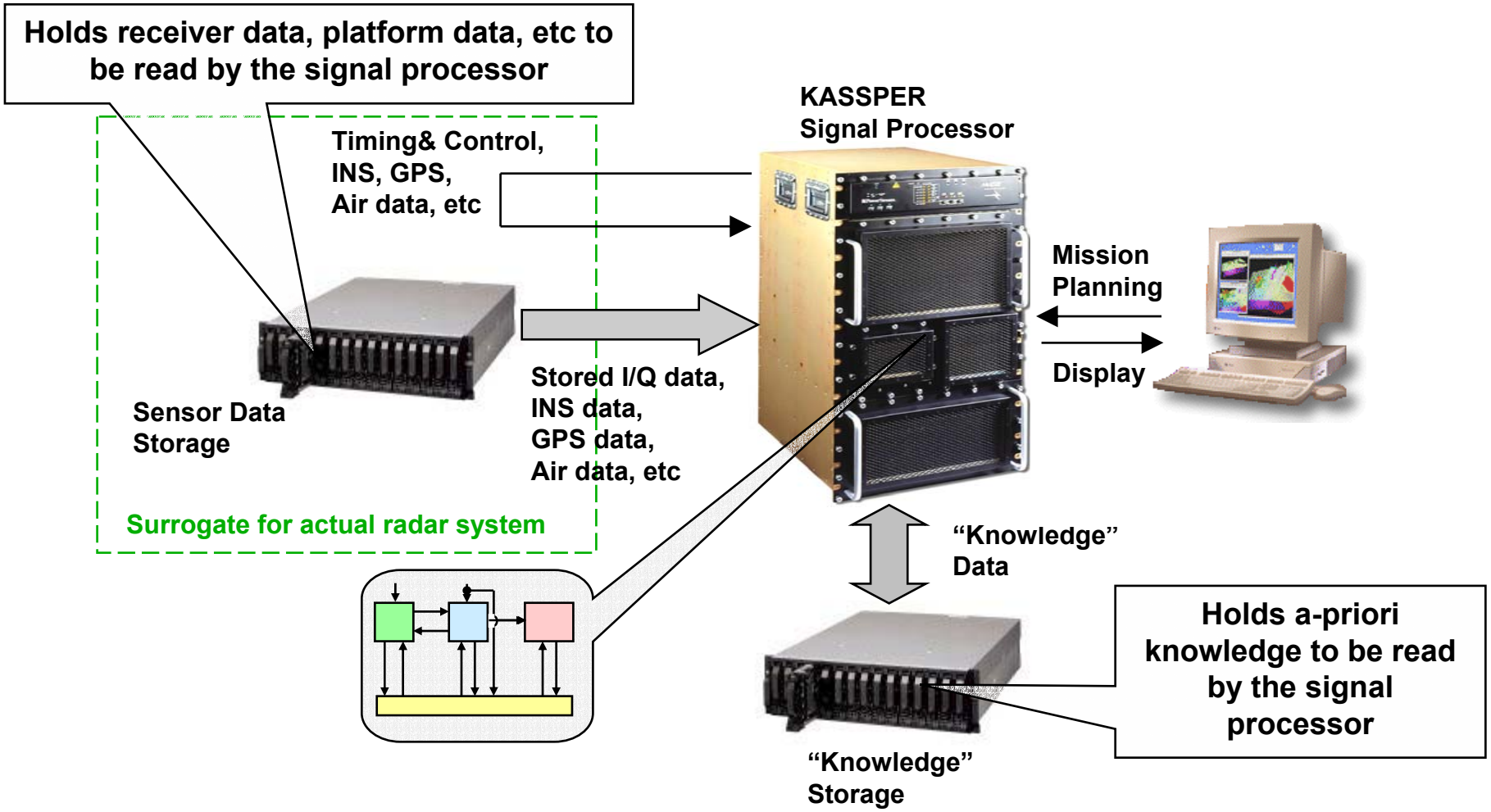
Pre-mission Data Load

Off-line Knowledge Source

Modular signal processing architecture supports exploration of KASSPER algorithms



Initial Architecture Evaluation Testbed

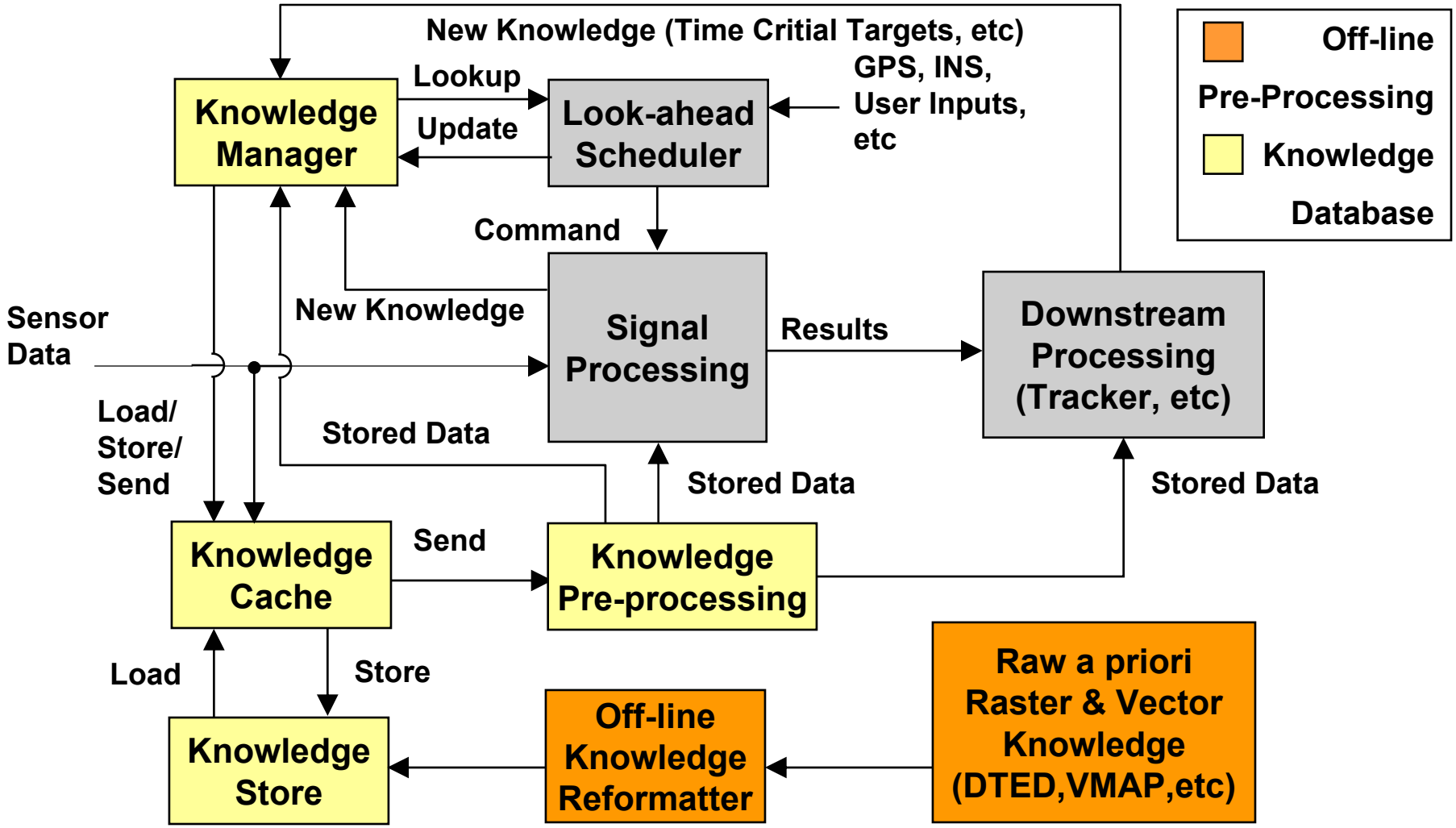


Scaled for 9ch 20mhz GMTI system

Outline

- System Architecture
- Knowledge Database
- Look-ahead Scheduling
- Knowledge Data Case Study
- Summary

Knowledge Database Architecture



The Knowledge Database must:

- Support various knowledge types
- Converted raw knowledge into radar-centric data
- Hide storage details from the signal processing

Vector Data

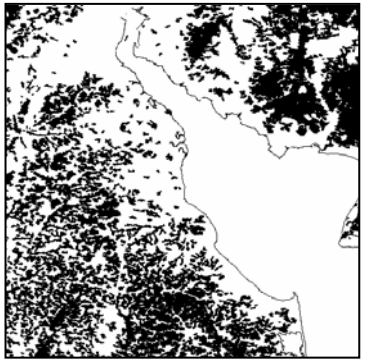
Each feature represented by a set of point locations:

- Points - longitude and latitude (i.e. towers, etc)
- Lines - list of points, first and last points are not connected (i.e. roads, rail, etc)
- Areas - list of points, first and last point are connected (i.e. forest, urban, etc)

Standard Vector Formats

- Vector Smart Map (VMAP)
  - Supplied by NIMA
  - Contains data "planes" such as political boundaries, vegetation, etc.
  - Replaces Digital Feature Analysis Data (DFAD)
- ArcInfo
  - Commercial format developed by Environmental System Research Institute, inc
- Sample VMAP data
  - 110KM x 110KM tiles

Political (line)



Trees (area)

Roads (line)

- Data rates are incompatible with magnetic media, caching is unavoidable
- Cache memory size and bandwidth may be a problem at high resolutions

Raster Data

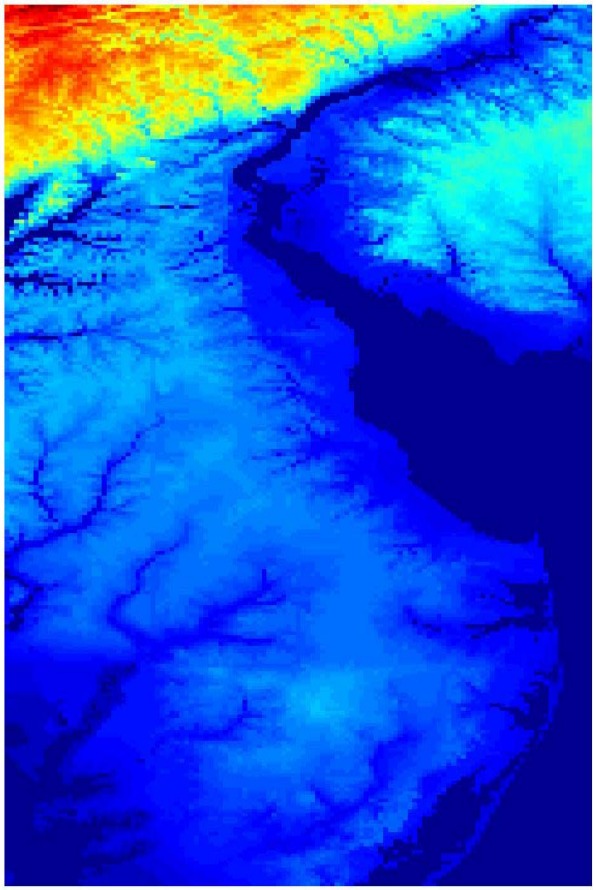
Rectangular arrays of evenly spaced data points

Current Raster Formats

- Digital Terrain Elevation Data (DTED)
  - Supplied by NIMA
  - Resolutions down to 30 meters
- Digital Elevation Map (DEM)
  - Resolutions to 2 arc-sec (≈ 60 meters)

Future Raster Formats

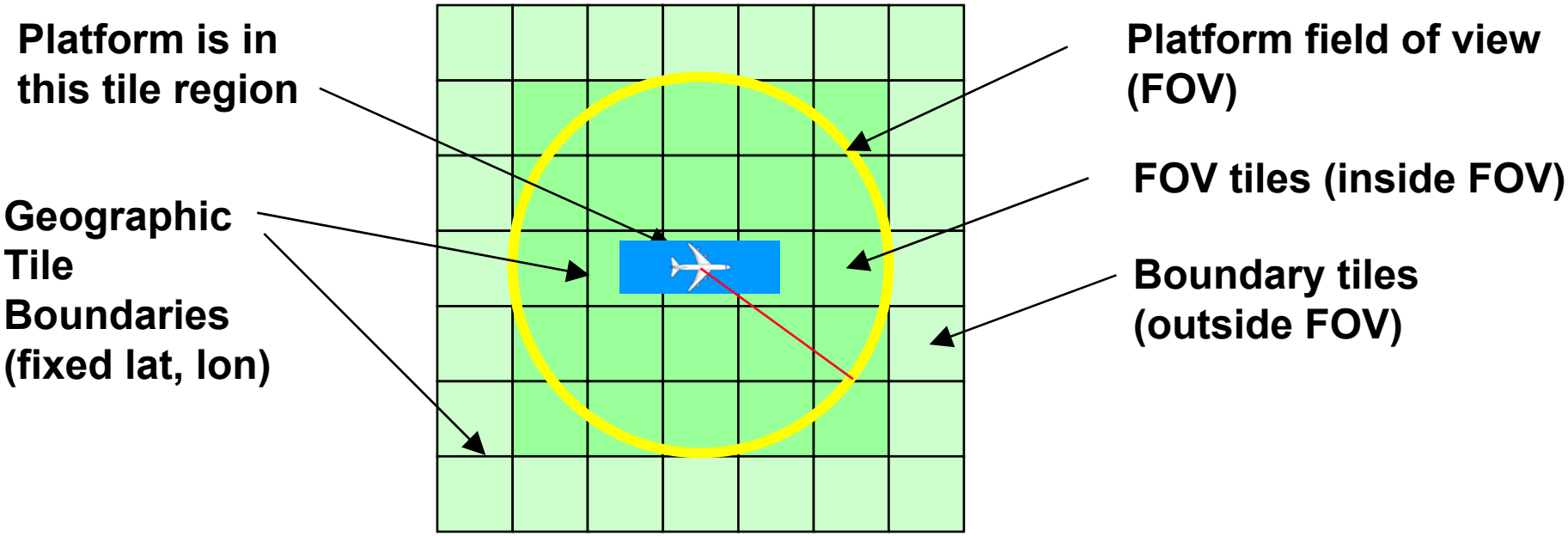
- High Resolution Terrain Information (HRTI)
  - NIMA HRTI still in draft stage
  - Resolutions down to 1 meter



110KM x 220KM (DTED Level 0)

- Data storage volume is challenging, but feasible with current technology
- Data rates are incompatible with magnetic media, caching is unavoidable
- Cache memory size and bandwidth WILL be a problem at high resolutions

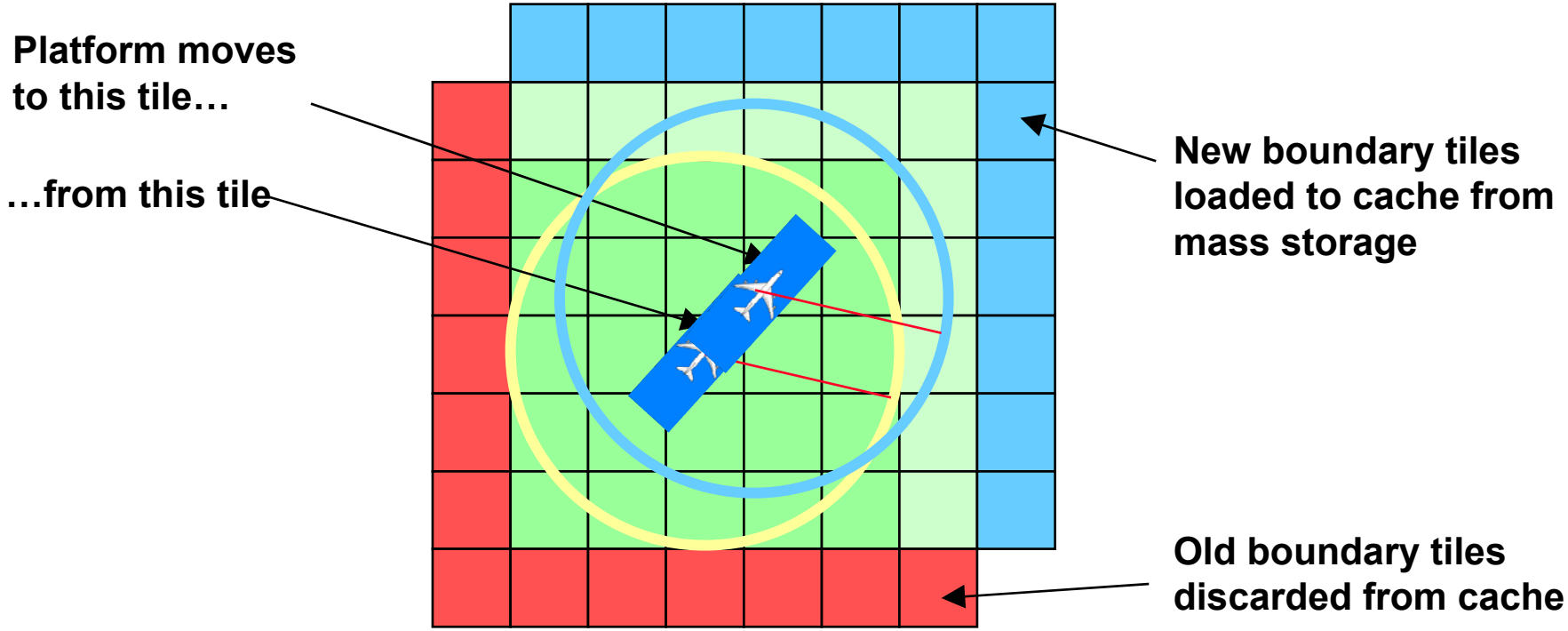
Geographic Knowledge Processing



- Tile size is based on the per tile memory, processing, and communication constraints
- Processing is done on both FOV tiles and boundary tiles
- Each geographic tile will be processed independently
- Results combined across tiles as last step.

Geographic Knowledge must be segmented, processed in parallel, and the results combined

Geographic Knowledge Cache Updating



- Load new boundary tiles as tiles enter the radar's FOV
- Discard boundary tiles as tiles leave the radar's FOV
- Boundary tiles loaded or discarded based on direction of motion
- With 100KM^2 tiles and 510kt, new tiles needed every 380sec

Loading or discarding tiles which are outside the FOV de-couples use of geographic knowledge from the cache

DTED & VMAP Accuracy

Data Source	Post Spacing	Absolute Accuracy		Relative Accuracy	
		Horizontal	Vertical	Horizontal	Vertical
DTED Lvl 0	1000M	Unspecified			
DTED Lvl 1	100M	± 50M	± 30M	± 30M	± 20M
DTED Lvl 2	30M	± 50M	± 30M	± 30M	± 20M
HRTI Lvl 3	12M	± 10M	± 10M	± 3M	± 4M
HRTI Lvl 4	6M	± 10M	± 5M	± 2M	± 1M
HRTI Lvl 5	1M	± 5M	± 5M	± 0.5M	± 0.5M

Data Source	Map Scale	Feature Data Quality			
		Class 1	Class 2	Class 3	Class 4
VMAP Lvl 0	1 - 1000000	Unspecified			
VMAP Lvl 1	1 - 250000	± 125M	± 250M	± 500M	± >500M
VMAP Lvl 2	1 - 50000	± 25 M	± 50M	± 100M	± >100M
UVMAP	1 - 10000	± 12.5M	± 25M	± 50M	± >50M

DTED Data Volume and Rates (worst case analysis)

Data volume and rates for a single 100KM^2 tile assuming that the tile data needs to be processed for each doppler @ PRF=1Khz

Note - Area of Iraq ≈ 439K KM^2 ≈ 44 tiles
JSTARS FOV ≈ 250KM ≈ 25 tiles

Data Source	Post Spacing	Tile Data Size	CPUs needed/ tile
DTED Lvl 0	1000M	0.03 MB	<<1
DTED Lvl 1	100M	3 MB	<<1
DTED Lvl 2	30M	23 MB	<1
HRTI Lvl 3	12M	278 MB	2
HRTI Lvl 4	6M	1.1 GB	9
HRTI Lvl 5	1M	40 GB	313

- Data storage volume is challenging, but feasible with current technology

Vector Data Volume and Rates

Data volume and rates for a single 100KM^2 tile assuming that the tile data needs to be processed for each doppler @ PRF=1Khz

Note - Area of Iraq ≈ 439K KM^2 ≈ 44 tiles
JSTARS FOV ≈ 250KM ≈ 25 tiles

Data Source	Scale	Tile Data Size	CPUs needed/ tile
VMAP Lvl 0	1 - 1000000	≈ 0.1 MB	<<1
VMAP Lvl 1	1 - 250000	≈ 4 MB	<<1
VMAP Lvl 2	1 - 50000	? 20 MB	? <1
UVMAP	1 - 10000	? 100 MB	? ≈1

- Data storage volume is feasible with current technology

Knowledge Cache Data Rates

- Assume 100KM^2 tiles
- Assume radar range = 250KM
- Cache size = 7 tiles x 7 tiles
- JSTARS max patrol speed = 510kt
- 510kt=263M/sec -> 380sec/100KM
- Worst case = load 14 tiles every (sqrt(2) * 380) sec

Data Source	Tile Data Size	Mass Stg Data Rate
DTED Lvl 0	0.03 MB	0.8 KB/s
DTED Lvl 1	3 MB	78 KB/s
DTED Lvl 2	23 MB	0.6 MB/s
HRTI Lvl 3	278 MB	8 MB/s
HRTI Lvl 4	1.1 GB	29 MB/s
HRTI Lvl 5	40 GB	1042 MB/s

Data Source	Tile Data Size	Mass Stg Data Rate
VMAP Lvl 0	≈ 0.1 MB	3 KB/s
VMAP Lvl 1	≈ 4 MB	67 KB/s
VMAP Lvl 2	? 20 MB	? 600 KB/s
UVMAP	? 100 MB	? 3 MB/s

Cache load rate from mass storage is challenging, but feasible with current technology

Outline

- System Architecture
- Knowledge Database
- Look-ahead Scheduling
- Knowledge Data Case Study
- Summary

The Scheduling Problem

Surveillance Duties

- General area surveillance
  - Everything in the platform's FOV
- A priori interesting locations
  - Points, Lines, or Areas of interest (tunnel openings, roads, DMZs, etc.)
- Evolving interesting locations
  - Real-time intelligence
  - Target tracks
  - Time-critical targets

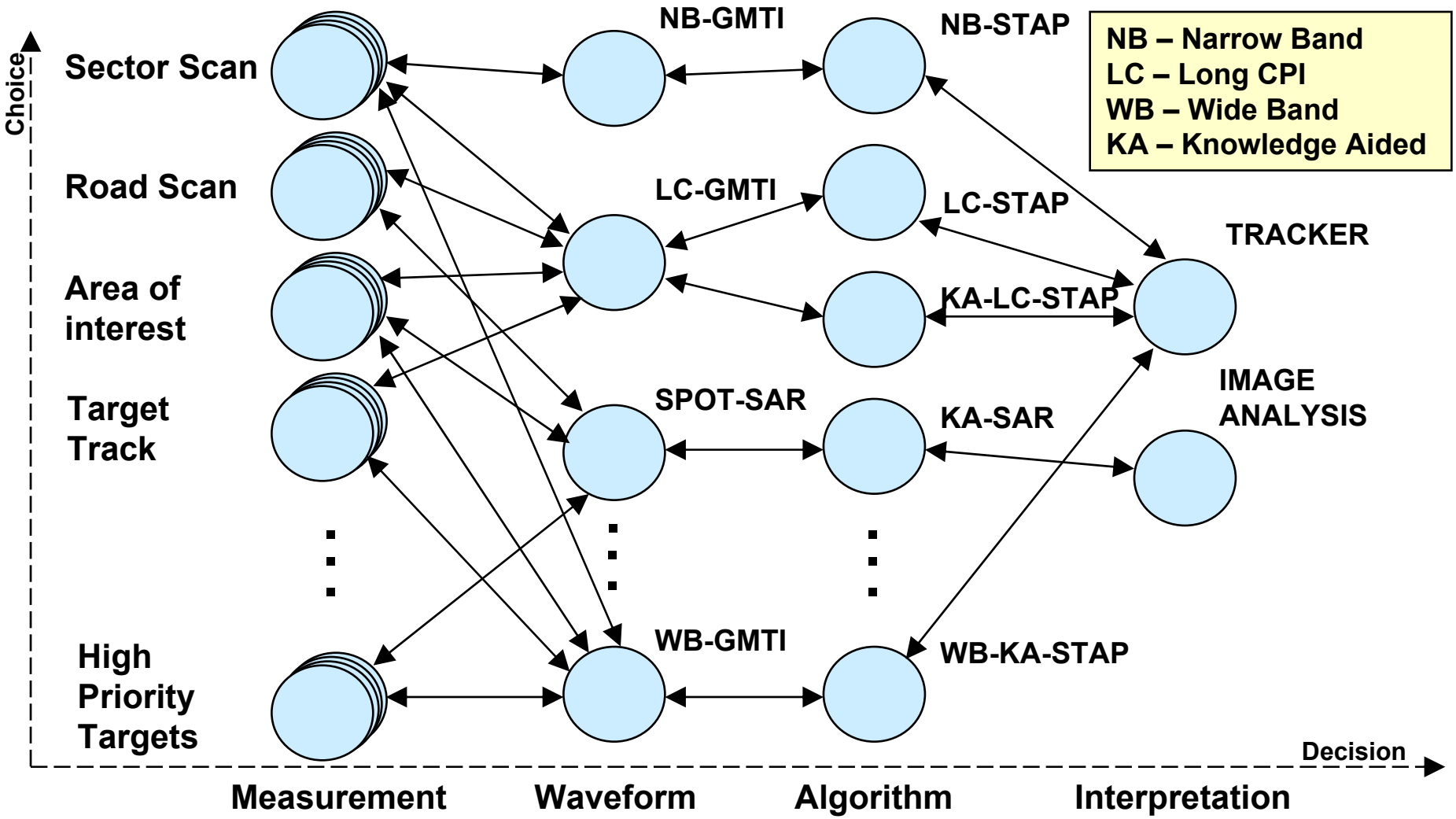
The scheduling problem has a number of facets

- Waveform / Algorithm selection
  - The "best" waveform / algorithm should be used
  - Revisit rate must be maintained
- Sensor and processor timeline
  - Time is a limited resource

Static vs. Dynamic Scheduling

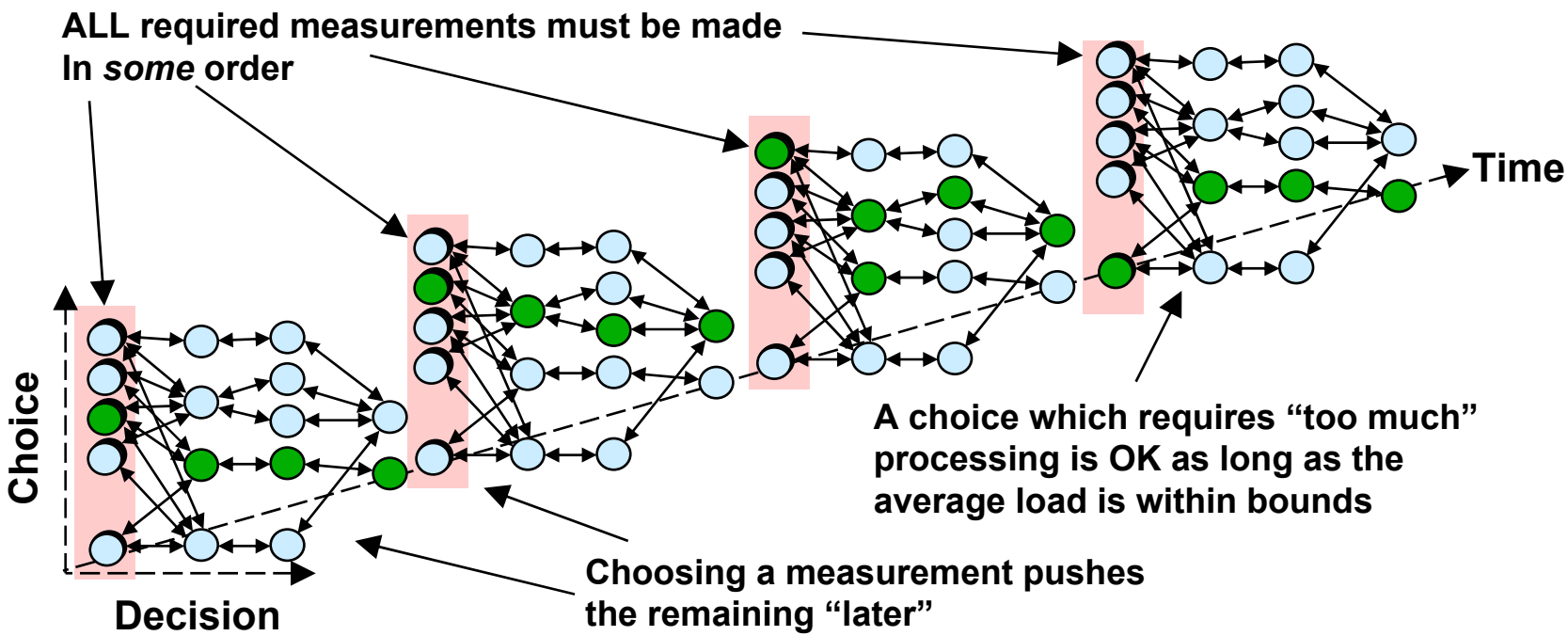
- Static scheduling
  - Does not adapt to its environment
- Dynamic scheduling
  - Adapt based on dynamic knowledge

CPI Scheduling Decision Graph



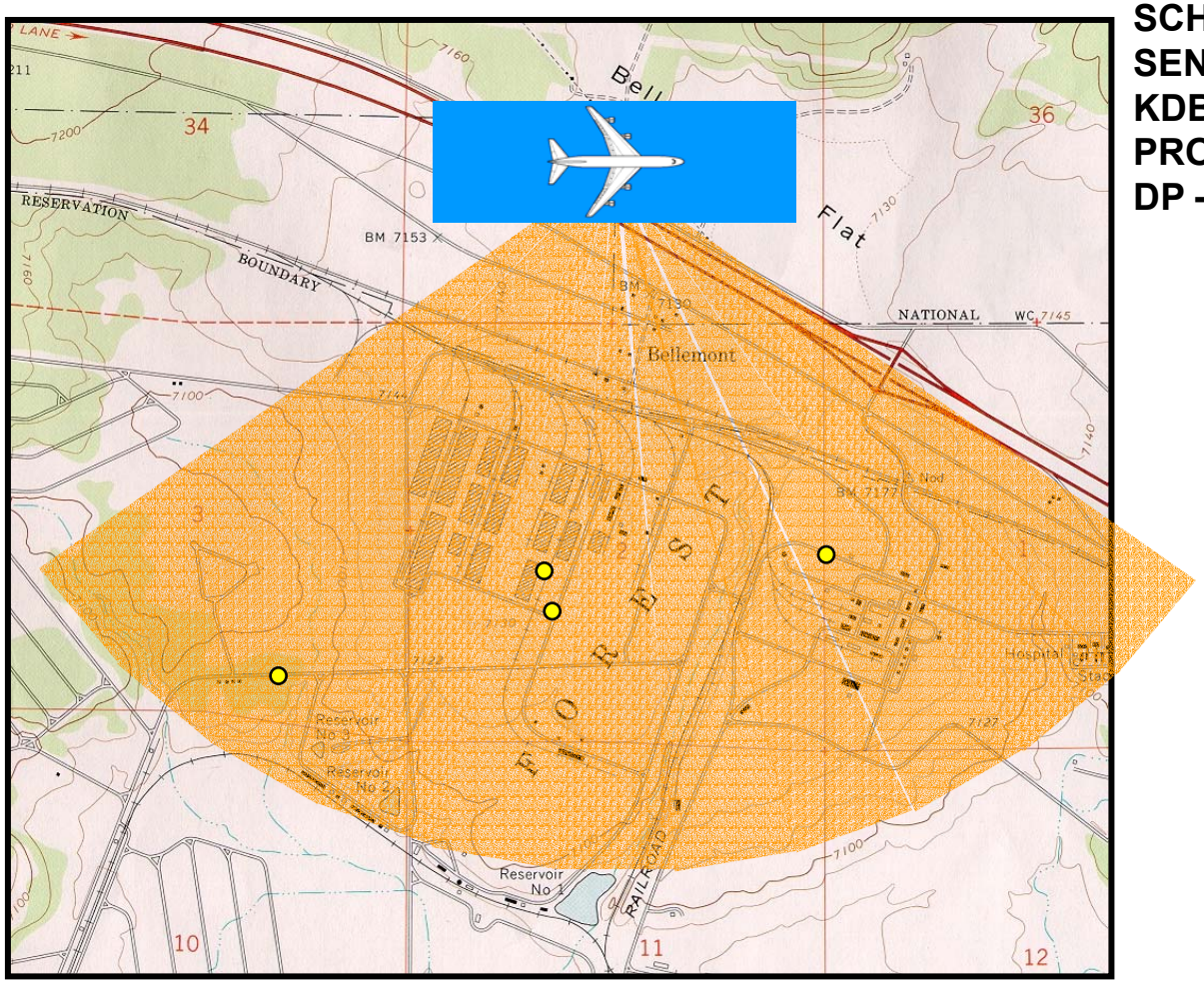
"Scheduling" a CPI by picking from each column to form a "decision path"

Multiple CPI Scheduling

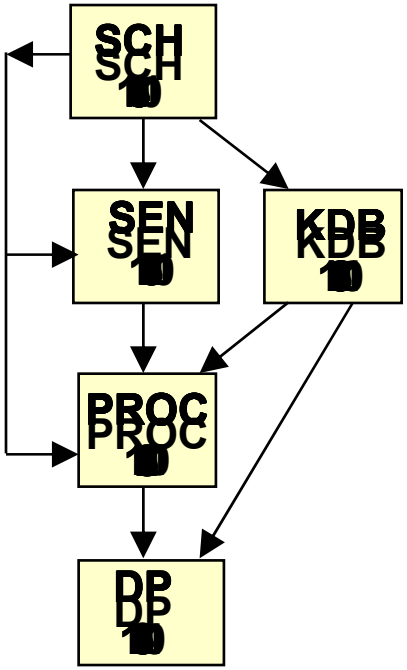


CPIs scheduled across time affect each other

Simple Area Scan

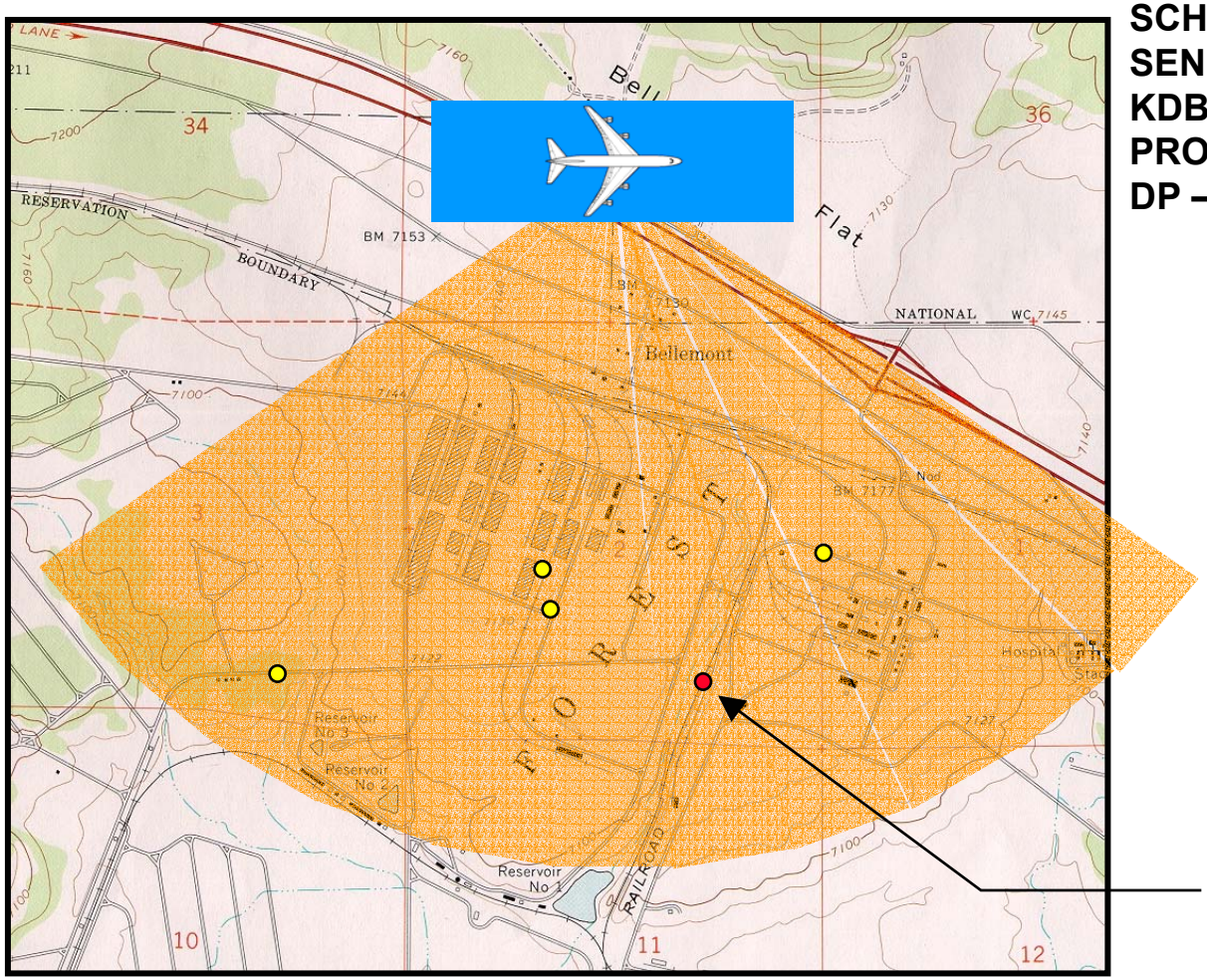


SCH - Scheduler
SEN - Sensor
KDB - Knowledge Database
PROC – Signal Processor
DP – Data Processing

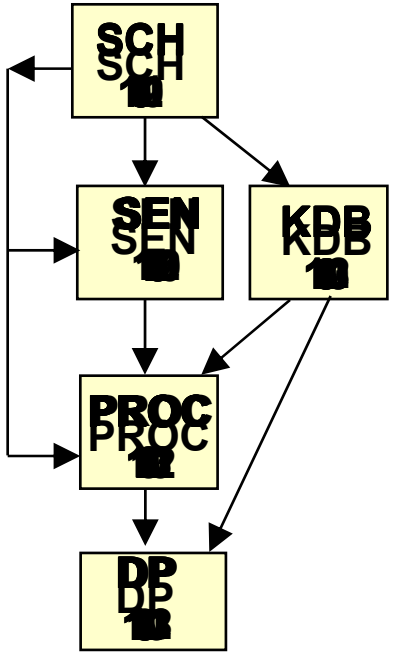


Statically scheduled azimuth sweep

Surveillance Scan/w Time Critical Target



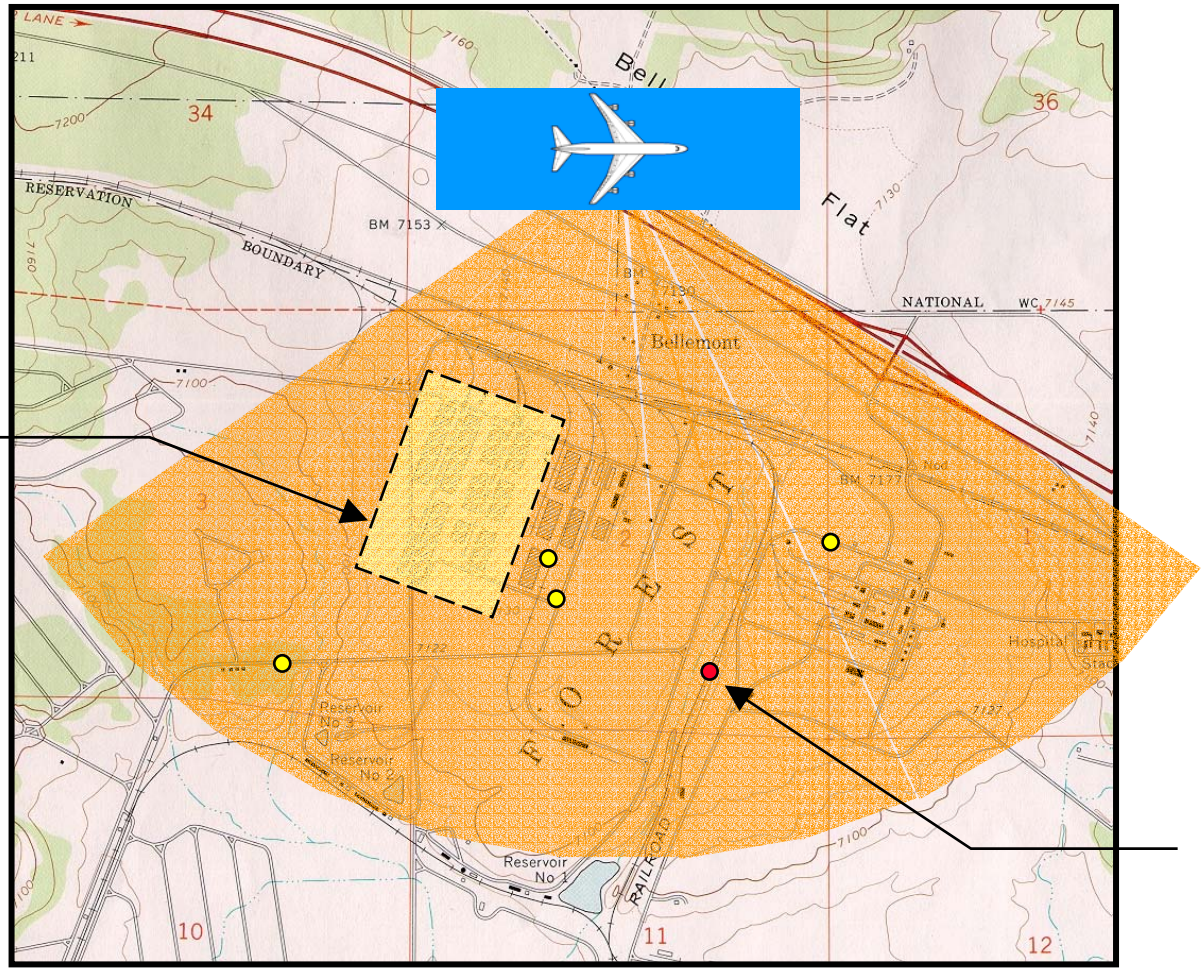
SCH - Scheduler
SEN - Sensor
KDB - Knowledge Database
PROC – Signal Processor
DP – Data Processing



Time Critical Target (TCT)

Dynamically scheduled TCT/w background azimuth sweep

Surveillance Scan/w Time Critical Target & Area of Interest



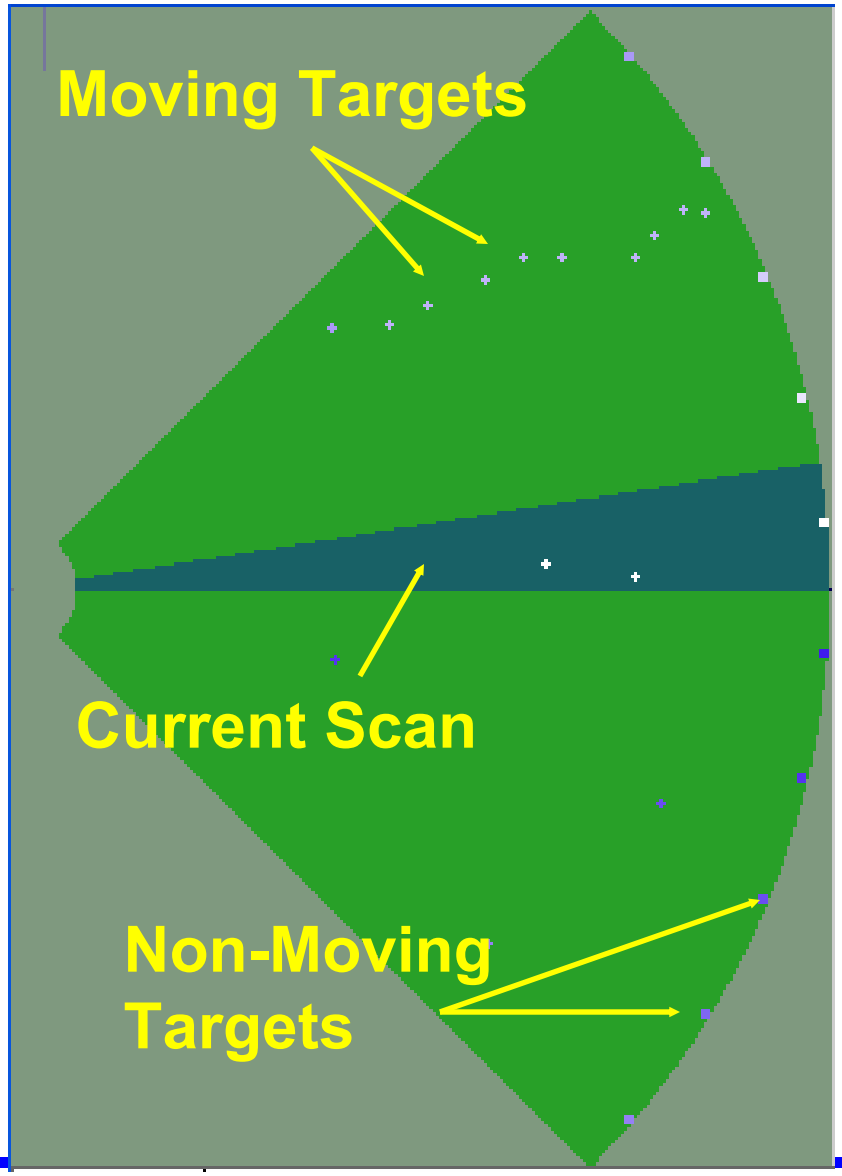
Dynamically scheduled TCT and AOI/w background azimuth sweep

Scheduler Development Approach

- Actual mission data (i.e. Tuxedo, etc.) is pre-scheduled
  - No scheduling flexibility
  - Not suitable for scheduler development
- Scheduler Testbed
  - Scheduling issues can be explored using simulated targets
  - Targets have a geographic position, velocity, revisit rate, priority
  - Radar data cubes are not necessary since scheduling is performed prior to the collection of radar data based on tracker and other knowledge.

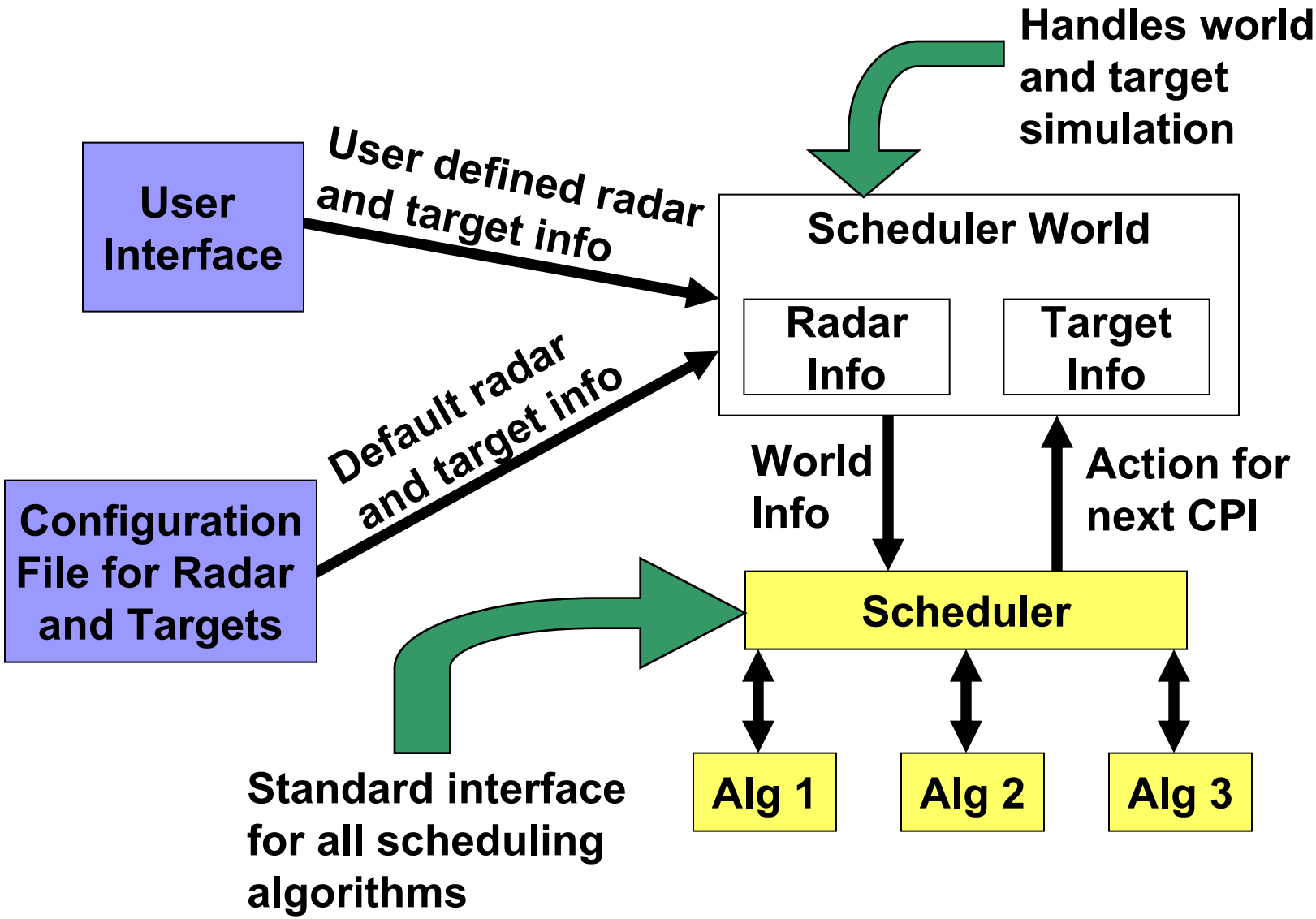
Simulated data will be used to allow infrastructure level testing using more realistic scenarios

Scheduler Test-bed Overview



- Determine which sector to visit to track targets and maintain a general area surveillance.
- Targets have several attributes influencing when they are scheduled.
  - Location
  - Priority
  - Revisit rate
  - Velocity and direction
- Potential Scheduling Algorithms
  - Simple greedy heuristics
  - Decision Directed Learning
  - Others…
- Issues
  - Real-time Performance
  - Perfect 'optimal' solution is not necessary

Scheduler Test-bed Architecture



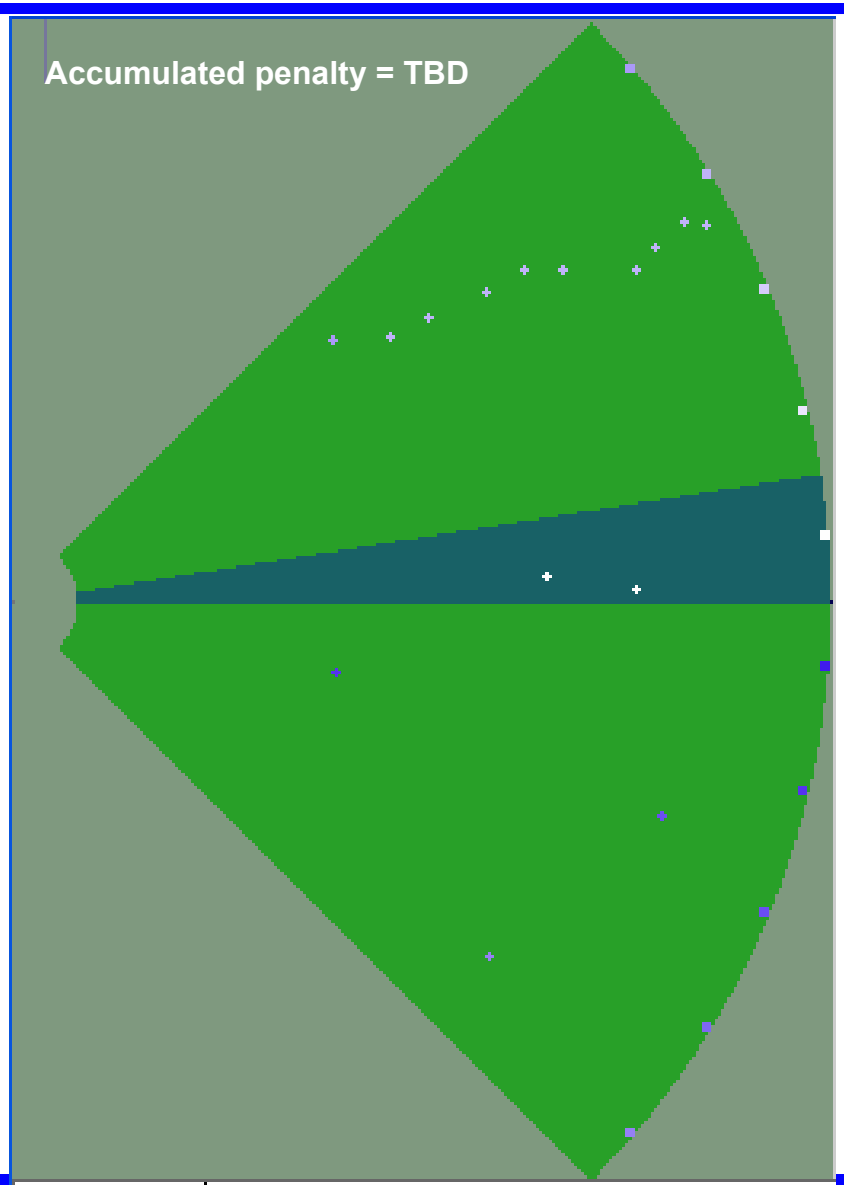
Scheduling Algorithm Comparisons

- **Schedule Comparison Heuristic**
  - Scheduling each CPI at some time has an associated (possibly zero) penalty.
  - The sum of these penalties over the scheduling time horizon provides a relative measure of the merit of different scheduling algorithms.

- **Example penalty function:**
  - Total penalty = $\Sigma\ (\text{Overdue}_i)^{\text{Priority}}$
  - For targets with higher priorities, the amount contributed to the total penalty grows exponentially.

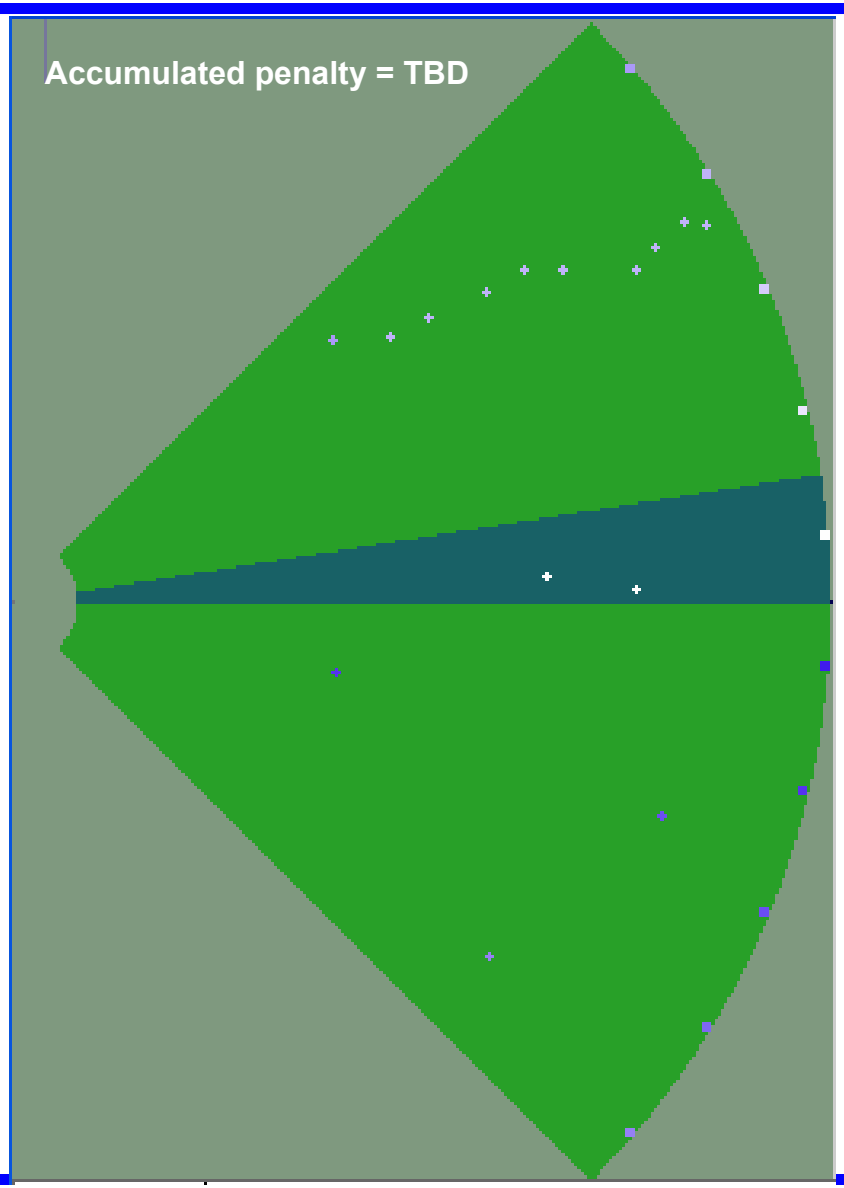
A "good" scheduling algorithm will make sure the most important work is always done (i.e. time critical targets, etc).

Linear Sweep Schedule Simulation



- Sectors are simply revisited in sequence.
  - Simplest possible scheduling approach
  - Implemented as a “point of reference” for more advanced algorithms
  - Provides sub-optimal performance
- Note that time critical targets are not visited often enough.

Greedy Heuristic Scheduler Simulation



- **Exploring heuristics to evaluate the worth of scanning each of the world's sectors.**
  - **Computationally much less expensive than an exhaustive search**
  - **Probably provides sub-optimal performance**
- **Example heuristic:**
  - **Value (sector) = - timeToDeadline * priority** ***(for positive values)***
  - **Value (sector) = - timeToDeadline / priority** ***(for negative values)***

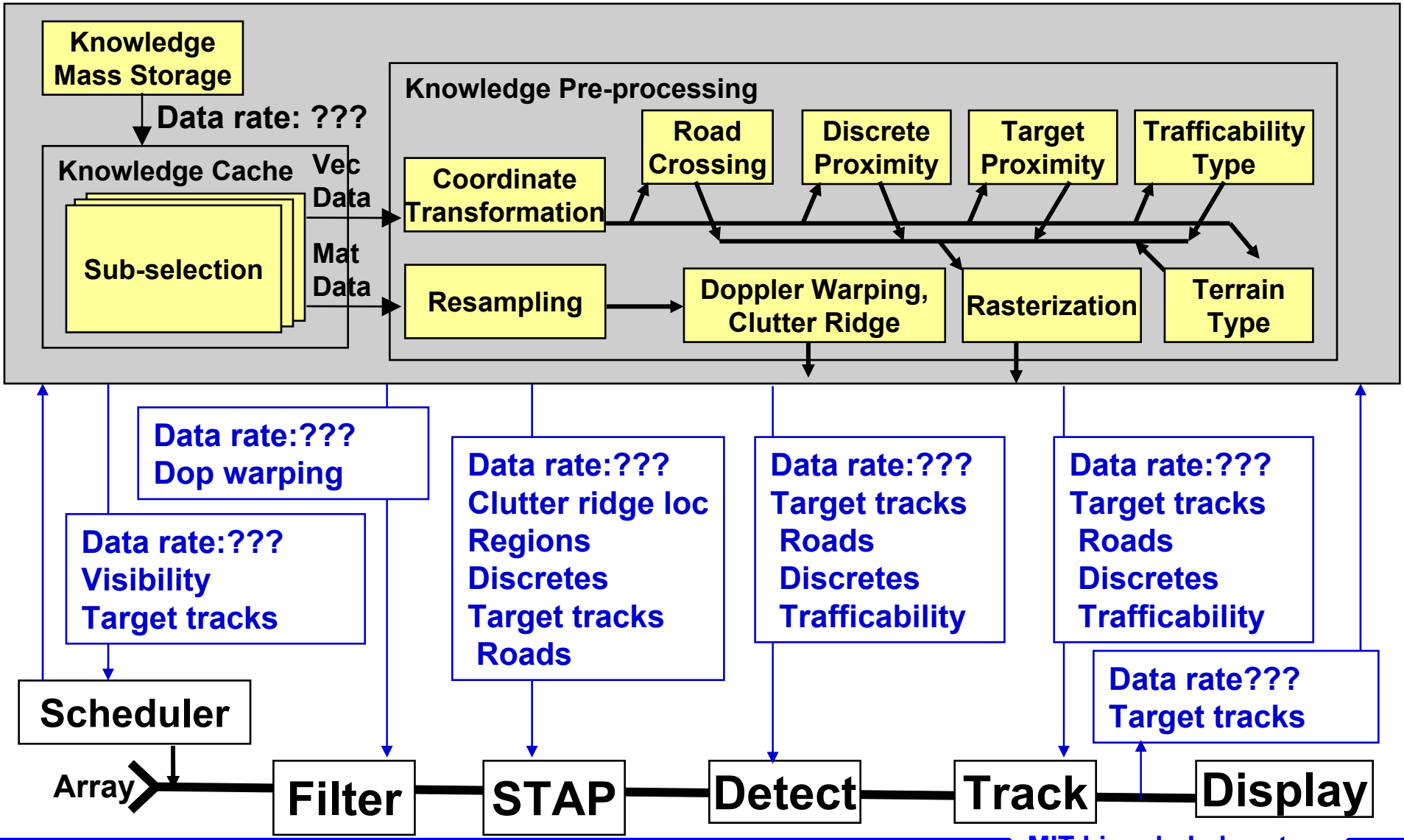
Outline

- System Architecture
- Knowledge Database
- Look-ahead Scheduling
- Knowledge Data Case Study
- Summary

Knowledge Aided Processing Case Study

- The KASSPER testbed baseline GMTI algorithm is used as a starting point
  - PRI staggered post-doppler STAP
  - Radar parameters are configurable
- Knowledge processing was added to the baseline GMTI algorithm to create a knowledge aided algorithm
- This case study examines the knowledge data volume and transfer bandwidth requirements for a particular

GMTI Processing Pipeline Block Diagram

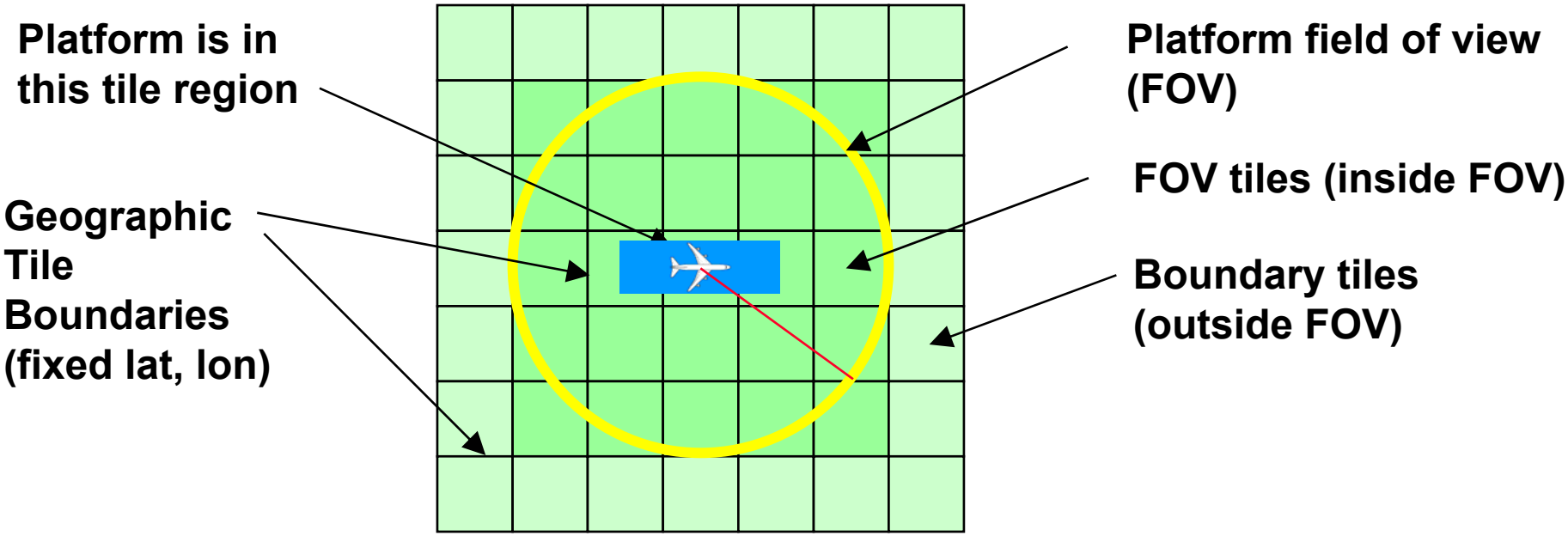


Knowledge Metadata

Knowledge Data	Source Type	Data Product Type & resolution
Visibility	Elevation Matrix	Byte Flag Vector (10M)
Doppler Warping	Elevation Matrix	Float Matrix (100M x 1Dop)
Target Tracks	Track List	Track Matrix ({lat,lon,vel,priority} x 1track)
Clutter Ridge	Elevation Matrix	Float Matrix (100M x 1Dop)
Terrain Regions	Terrain Areas	Terrain Enum Matrix (25M x 1Dop)
Clutter Discretes	Terrain Points	Byte Flag Matrix (10M x 1Dop)
Roads	Terrain Lines	Byte Flag Matrix (10M x 1Dop)
Target Tracks	Track List	Byte Flag Matrix (10M x 1Dop)
Trafficability	Terrain Areas	Terrain Enum Matrix (25M x 1Dop)

All data is transferred in vectors or matrices where each element corresponds to some number of range cells

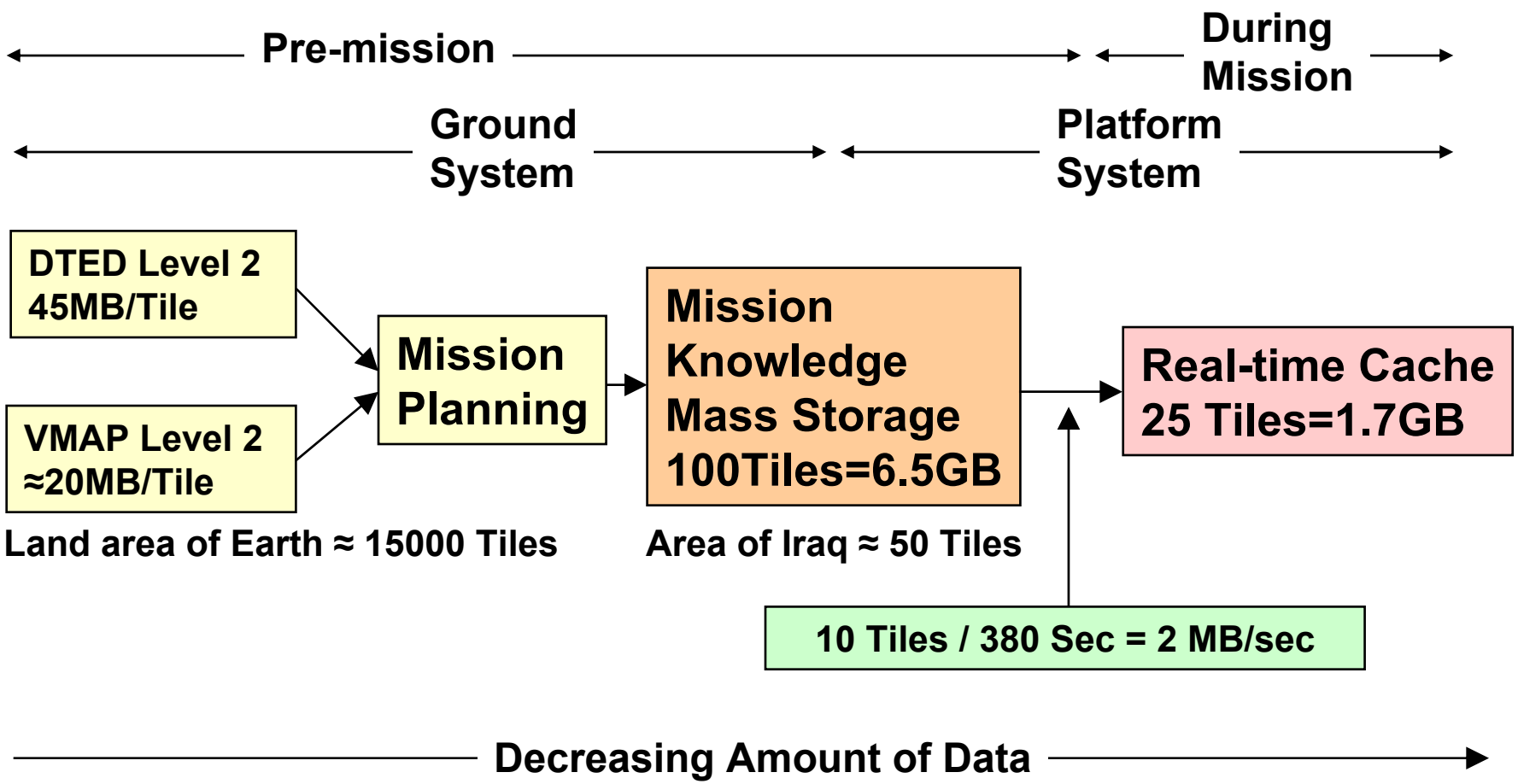
Real-time Knowledge Cache



- Tile size is based on the per tile memory, processing, and communication constraints (assume 100KM x 100KM)
- Assume cache size = 5 x 5 = 25 tiles
- Tile traversal time = 100KM @ ≈ 500 KTS = 380 sec

Geographic Knowledge can be segmented, processed in parallel, and the results combined before being presented to a processing algorithm

Knowledge Storage

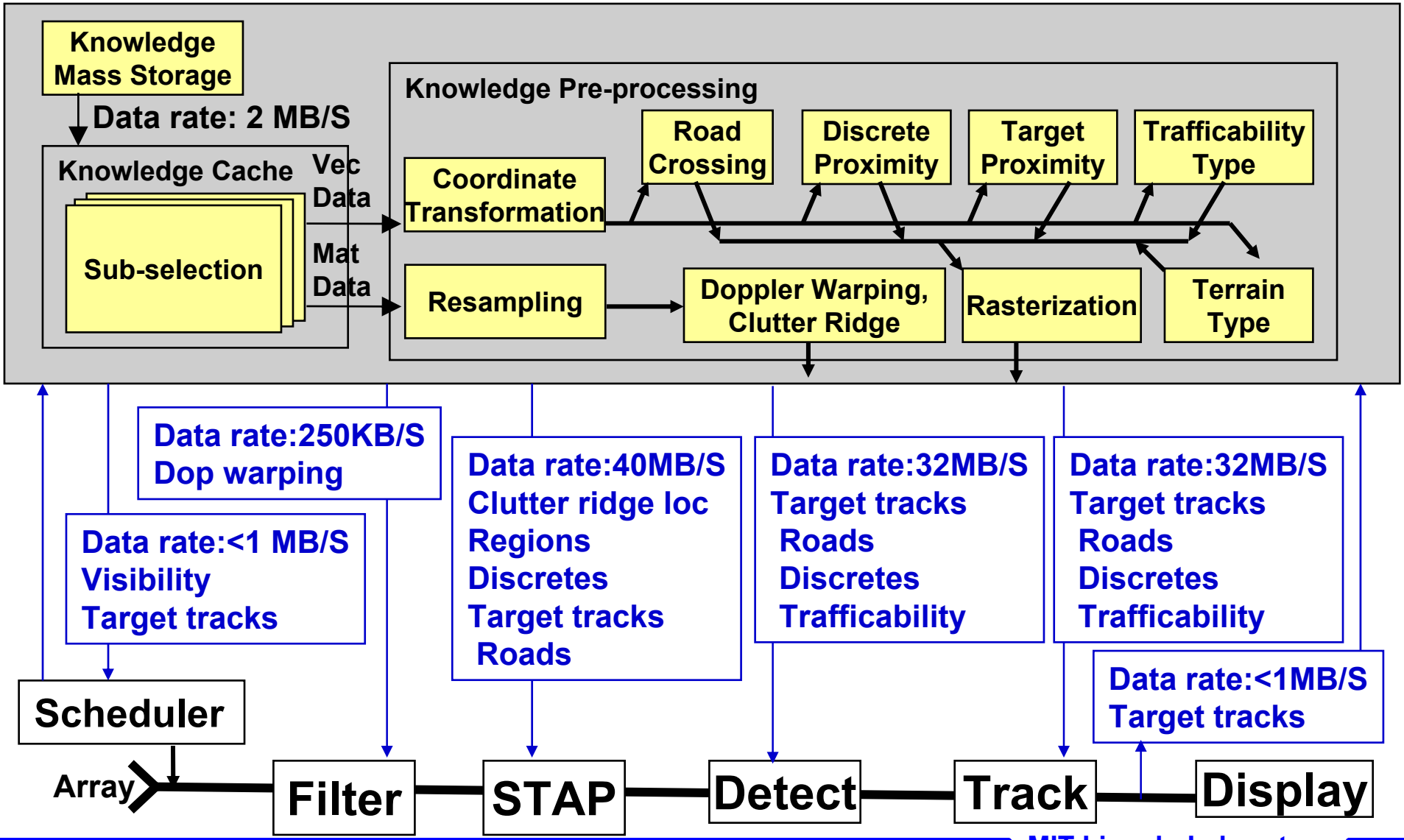


Signal Processing Knowledge Data Rates

- Scheduler
  - Visibility Flags – Range * <CPI-RATE> / granularity * <element-size>
    200KM * 31.25Hz / 10 M/value * 1 B/value = 625 KB/S
  - Target List - #targets * <CPI-RATE> * <element-size>
    1000 Targets * 31.25Hz * 4 B/Target = 125 KB/S
- Filter
  - Doppler Warping - Range * <CPI-RATE> / granularity * <element-size>
    200KM * 31.25Hz / 100 M/value * 4 B/value = 250 KB/S
- STAP
  - Clutter Ridge - Range * PRF / granularity * <element-size>
    200KM * 1Khz / 100 M/value * 4 B/value = 8 MB/S
  - Terrain Types - Range * PRF / granularity * <element-size>
    200KM * 1Khz / 25 M/value * 1 B/value = 8 MB/S
  - Discrete, Road, Target flags - Range * PRF / granularity * <element-size>
    200KM * 1Khz/ 10 M/value * 1 B/value = 24 MB/S
  - Total = 40 MB/S
- Detect + Track
  - Trafficability Types - Range * PRF / granularity * <element-size>
    200KM * 1Khz / 25 M/value * 1 B/value = 8 MB/S
  - Discrete, Road, Target flags - Range * PRF / granularity * <element-size>
    200KM * 1Khz/ 10 M/value * 1 B/value = 24 MB/S
  - Total = 32 MB/S

Assume max range = 200KM, PRF=1Khz, Shortest CPI = 1Khz/32 = 31.25Hz

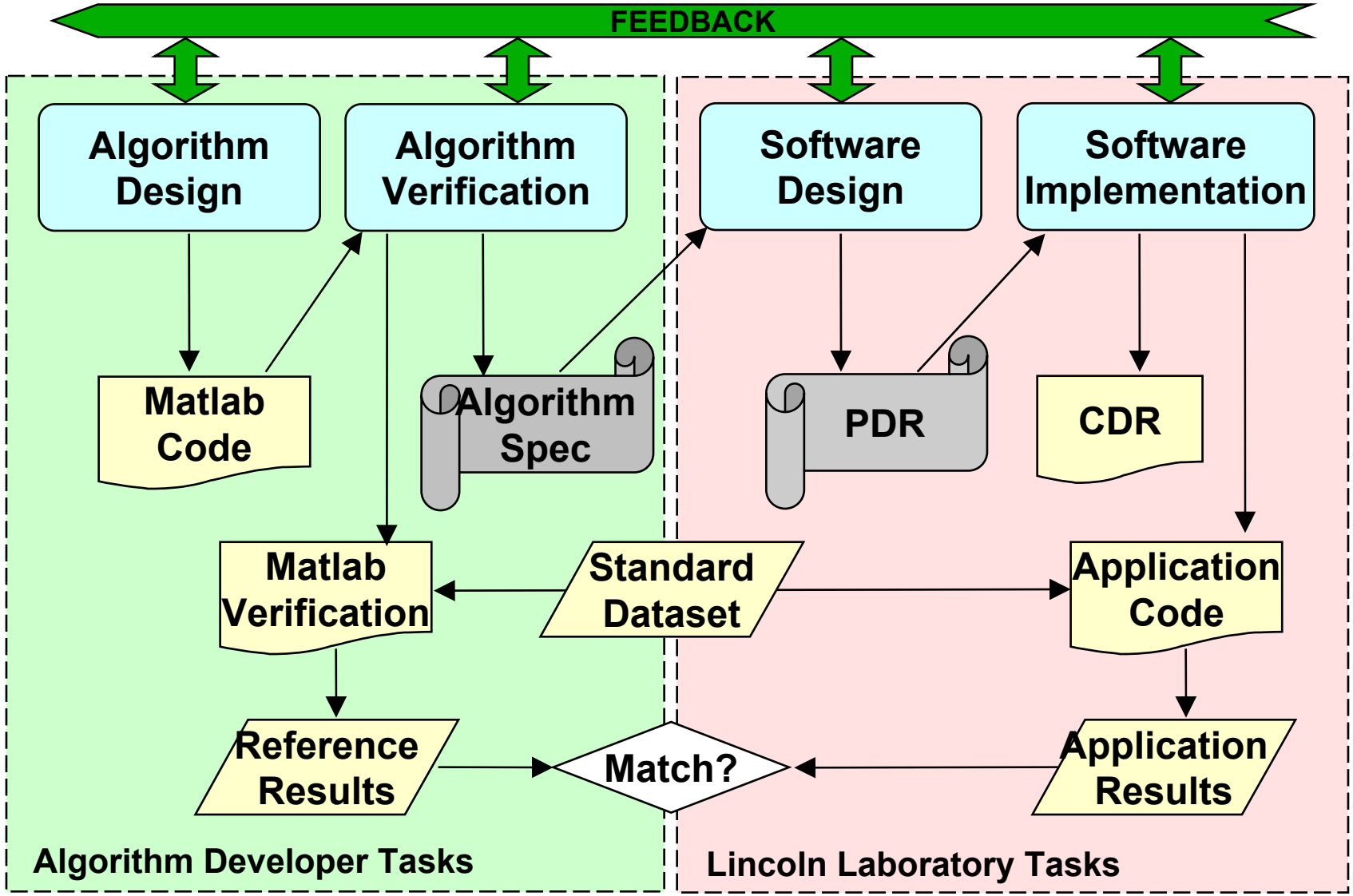
GMTI Processing Pipeline Block Diagram



Outline

- System Architecture
- Knowledge Database
- Look-ahead Scheduling
- Knowledge Data Case Study
- Summary

KASSPER Team Processor Testbed Algorithm Insertion Process



Summary

- Defined a modular knowledge-aided architecture
- Defined testbed concept to support knowledge-aided processing implementation & evaluation
- Considered key technical issues associated with Look Ahead Scheduling and Knowledge Database operation
- Developed a baseline multi-mode application
- Currently implementing scheduler and knowledge database KASSPER architecture components
  - Exploring expert system techniques which can be used for a dynamic scheduling infrastructure
- Refining real-time performance of signal processing algorithms