Chapter 9
Localization and Mapping

Part 1

9.1 Representation and Issues

Outline

- 9.1 Representation and Issues
  - 9.1.1 Introduction
  - 9.1.2 Representation
  - 9.1.3 Timing and Motion issues
  - 9.1.4 Related Localization issues
  - 9.1.5 Structural Aspects
  - 9.1.6 Example: Unmanned Ground Vehicle …
  - Summary

Outline

- 9.1 Representation and Issues
  - 9.1.1 Introduction
  - 9.1.2 Representation
  - 9.1.3 Timing and Motion issues
  - 9.1.4 Related Localization issues
  - 9.1.5 Structural Aspects
  - 9.1.6 Example: Unmanned Ground Vehicle …
  - Summary

9.1.1 Introduction

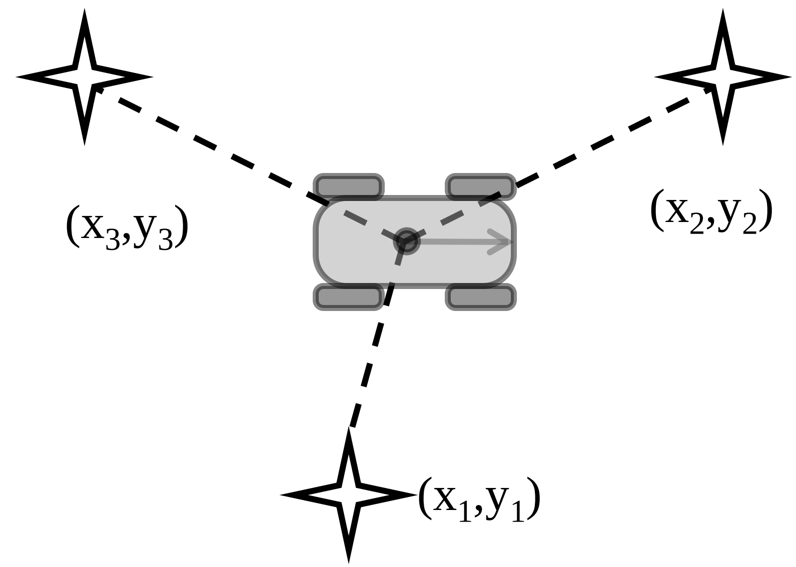
- Mapping = The process of making maps – not just using them.
- Purpose: encode knowledge over and above:
  - a) what can be seen/measured now
  - b) whatever assumptions may be encoded in software algorithms
- Why do it?
  - Memory
  - Data fusion

9.1.1 Introduction
(Organization)

- Indexing:
  - Spatially indexed (raster):
    - Properties of places
  - Object indexed (vector):
    - Locations of things
- Function:
  - predicting sensor readings (navigation).
  - predicting environment interaction (planning).

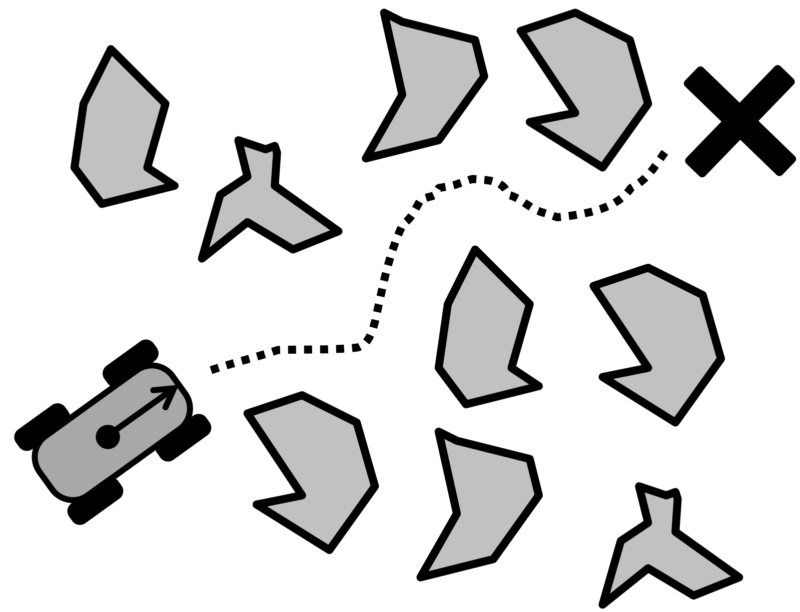
9.1.1 Introduction
(Navigation Maps)

- Use innovations to resolve pose error.
- Map stored as:
  - List of landmarks
  - List of range scans
- Great dilemma:
  - Need location for mapping.
  - Need maps for localization.



9.1.1 Introduction
(Planning Maps)

- Predict what will happen if the robot decides to go somewhere specific.
- Stored as:
  - a 2D or 3D grid
  - a list of obstacle locations, roads etc.



Outline

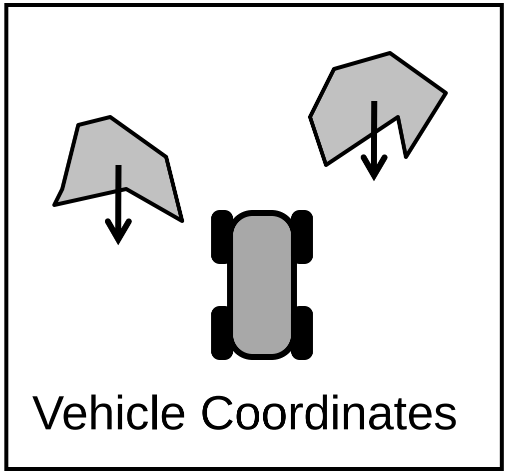
- 9.1 Representation and Issues
  - 9.1.1 Introduction
  - <u>9.1.2 Representation</u>
  - 9.1.3 Timing and Motion issues
  - 9.1.4 Related Localization issues
  - 9.1.5 Structural Aspects
  - 9.1.6 Example: Unmanned Ground Vehicle …
  - Summary

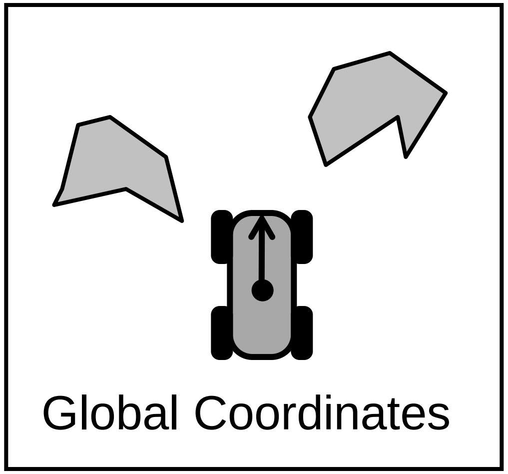
9.1.2 Representation

- When designing a map, there are a few big decisions to make.
- Design drivers are convenience or efficiency of certain (dominant) computations:
  - E.g. easy to predict slope (Terrain map)
  - E.g. minimal information loss (Image map)
  - E.g. Topological reasoning (Topological map)

9.1.2.1 Coordinate System Aspects

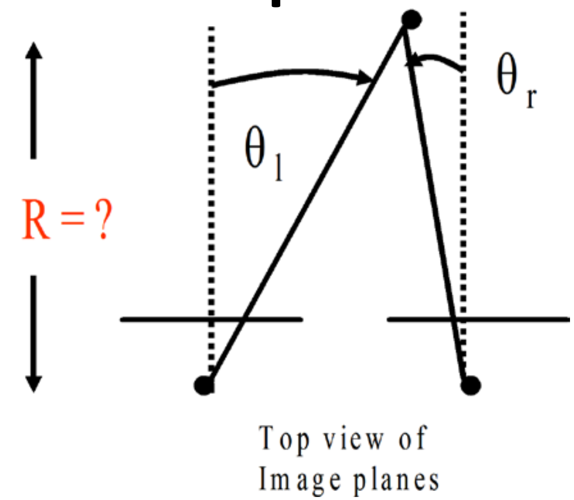
- Consider Representing Motion…..
- Vehicle fixed coordinates:
  - Move the map to reflect motion (expensive).
  - May make sense for small footprint maps.
  - Computations of obstacles relative to vehicle are convenient (free).
- Ground fixed coordinates:
  - Move the vehicle to reflect motion (free).
  - Map size does not matter.
  - Computations of vehicle relative to, waypoints, other vehicles, gravity are convenient (slightly costly).



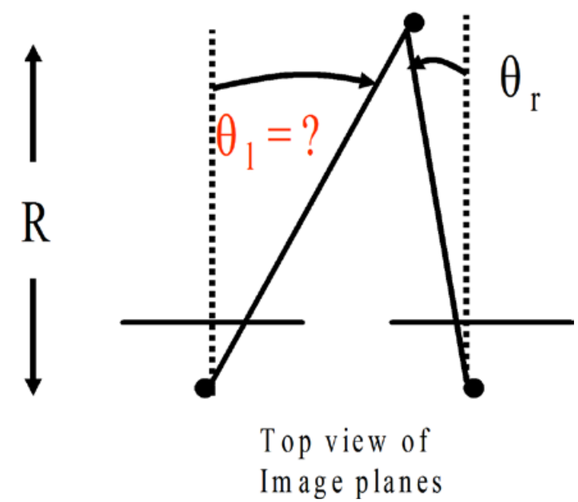


9.1.2.1 Coordinate System Aspects

- Consider Sampling Uniformity...
- Sensors usually sample uniformly in their image plane.
- Distortion is minimized by minimum change of viewpoint.
- Hence, merge pixels in some real or synthetic image plane.
  - Still have sampling and interpolation issues.
- Merging may introduce difficult correspondence issues (opposite).



Stereo: Correspondence Unknown



Reprojection: Correspondence Known

9.1.2.1 Coordinate System Aspects
(Example: Virtualized Reality)

- Range + appearance is the essence of a computer graphics model.
- Given one image and range info, many completely synthetic views can be generated
  - subject to missing parts problem.

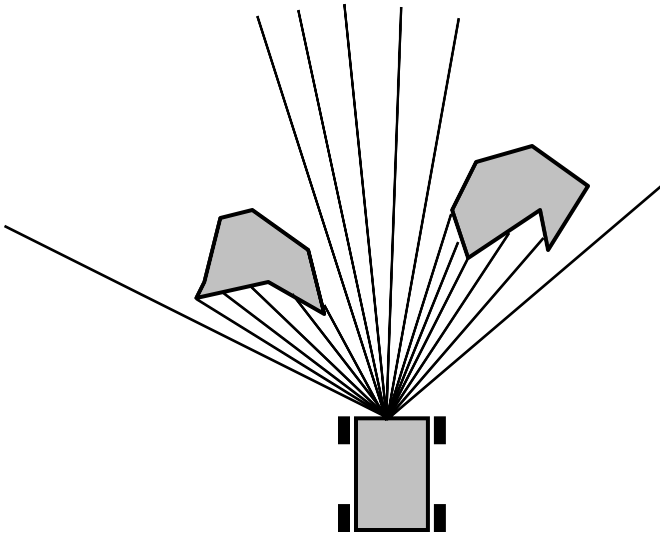
9.1.2.1 Coordinate System Aspects

(Example: 3D Video)

9.1.2.3 Sampling Issues and Missing Parts

(Missing Parts)

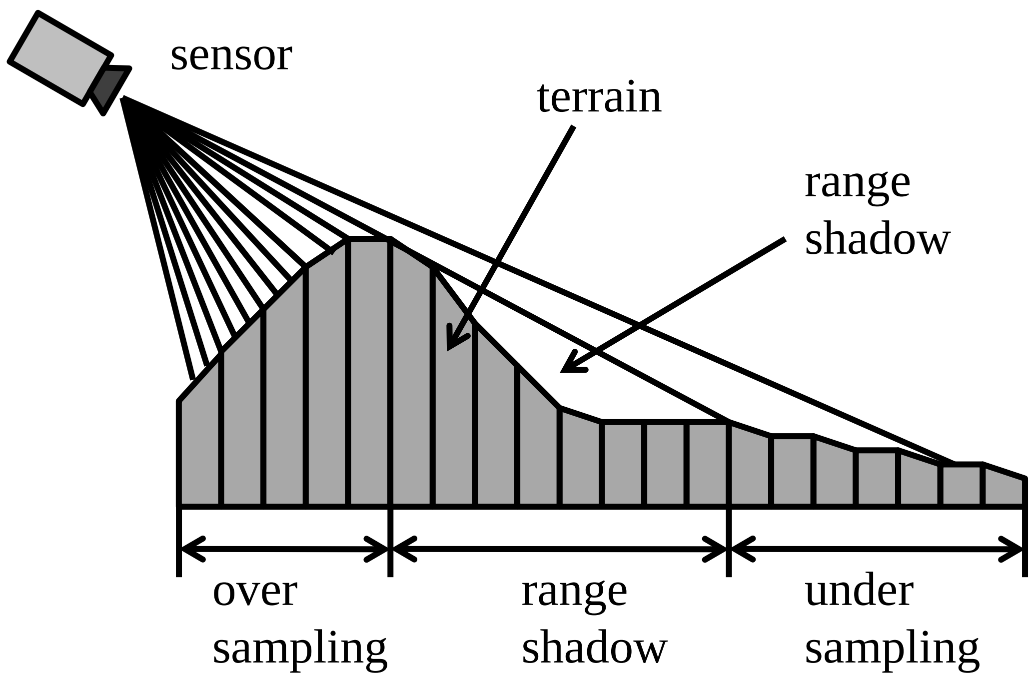
- Different viewpoints create potential for missing parts in maps.
- Core issue is:
  – viewpoint dependence of image
  – combined with environmental self occlusion.
- Leads to partial information in parts of the map.



9.1.2.3 Sampling Issues and Missing Parts

(Missing Parts)

- Most sensors have pixels equally spaced in angle.
- Nonlinear transformation to any other coordinate system will distort the sampling



9.1.2.4 Semantic Aspects
(Object Oriented)

Object oriented maps could store:

- Planning:
  - Walls
  - Objects
  - Obstacles
  - Costs
- Navigation:
  - Points
  - Lines
  - Images

9.1.2.4 Semantic Aspects
(Sampled)

Sampled maps could store:

- Terrain shape descriptors: such as elevation, slope, roughness, overhang height
- Terrain mechanical descriptors: such as stiffness/compressibility, traction, and density (grass and underbrush are low density).
- Terrain classification descriptors: such as wooded, rocky, high grass, deep or shallow water.
- Hazard descriptors: such as cost of traversal, information content (e.g. range shadows have little content).
- Tactical descriptors: such as the relative threat, cover, or recon or communications availability potential of a cell.

9.1.2.5 Meta-Informational Aspects

- How processed is the data:
  - Raw sensor readings (e.g scene attributes) → delays commitment
  - Semantic interpretation (e.g. walls / doors) → more efficient
- Fidelity:
  - Globally accurate
  - Locally smooth
  - Doing both is hard…
- Signal representation:
  - Sampled / rasterized → more common for <u>planning</u> maps
    - Some dimensions may remain continuous [e.g. z(i,j)]
  - Object Oriented → more common for <u>navigation</u> maps.

9.1.2.5 Meta-Informational Aspects

- Uncertainty……………..
- Housekeeping information:
  – time/pose tags
  – distance, position or source associated with the data in the cell,
  – backpointers associated with global planning algorithms.

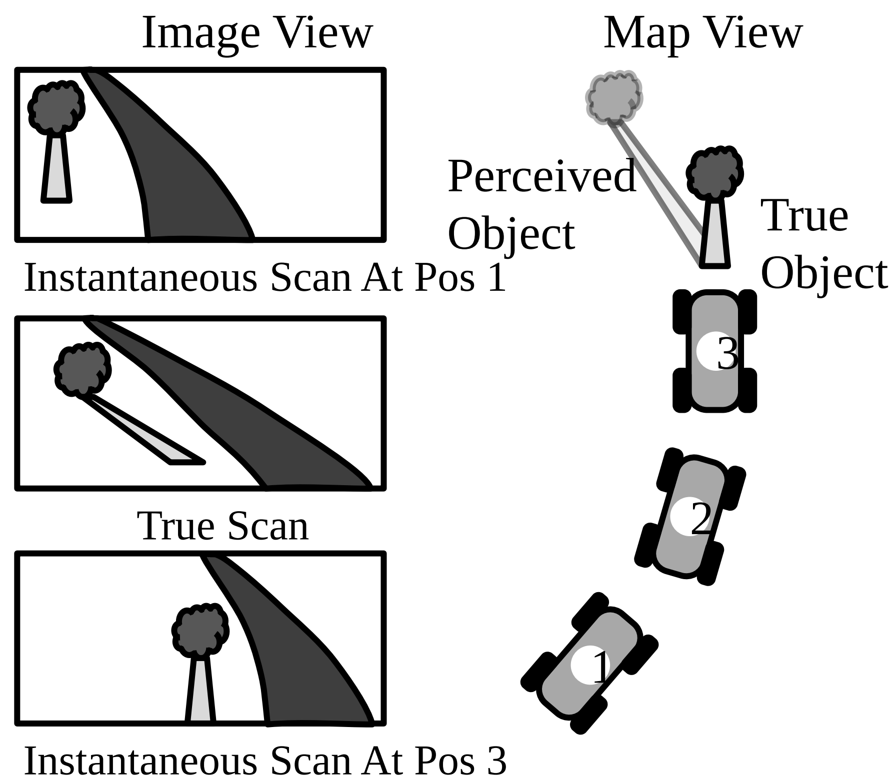
Outline

- 9.1 Representation and Issues
  - 9.1.1 Introduction
  - 9.1.2 Representation
  - 9.1.3 Timing and Motion issues
  - 9.1.4 Related Localization issues
  - 9.1.5 Structural Aspects
  - 9.1.6 Example: Unmanned Ground Vehicle …
  - Summary

9.1.3 Timing and Motion Issues

(Motion Blur)

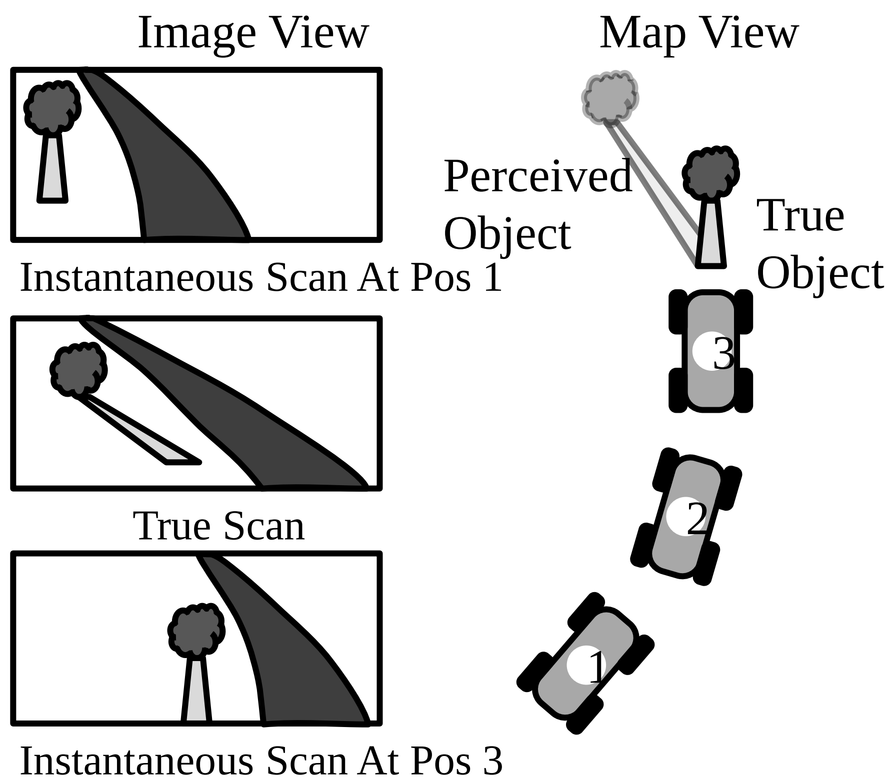
- Maps are <u>distorted</u> when the robot is moving and the timing of data acquisition is not accurate enough.
- Effect goes away when you stop moving.
- Happens <u>even over sub second</u> time windows.
  – Robots can rotate fast
- Angular measurements usually matter most.
- May only matter when perception data is <u>high fidelity</u>.



9.1.3 Timing and Motion Issues

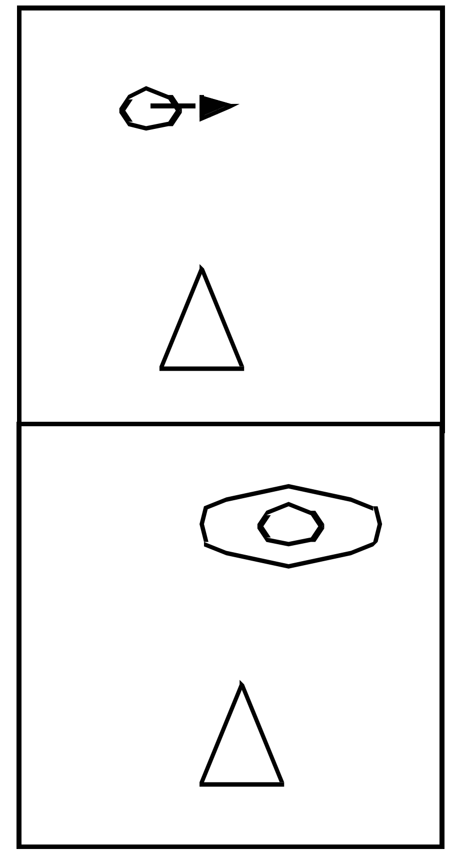
(Motion Blur)

- Precise azimuth/elevation of a 120 Hz laser beam mounted on a bouncing vehicle is a challenge to achieve.
- Timing (or pose tags) are used to synchronize the perception data stream with the localization data stream.
  - Done with interrupts deep in the system software.
- A big issue for scanning ladar.
- Less of an issue for FLIR.
- Usually not an issue for cameras.



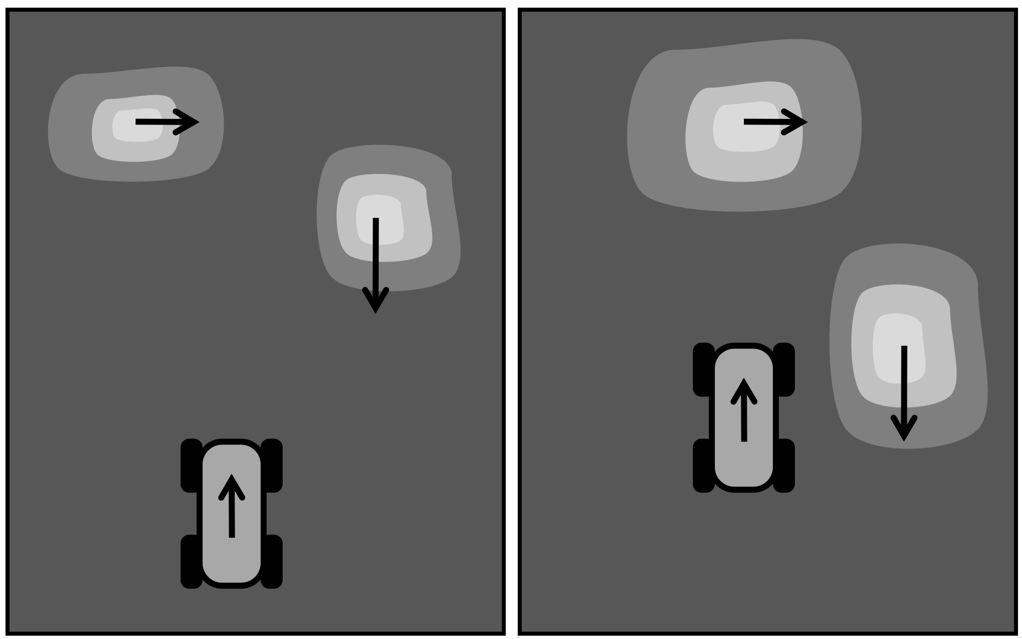
9.1.3.2 Ghosting

- Moving objects can create traces in maps.
  - When a false static environment assumption is being made.
- Bayesian maps with an integrated motion model are a great way to deal with moving objects.



9.1.3.3 Moving Object Detection and Tracking
(Motion in Evidence Grids)

- Use measurements and tracking to refine estimates of position.
- Use motion models to account for growth of uncertainty between models.



Outline

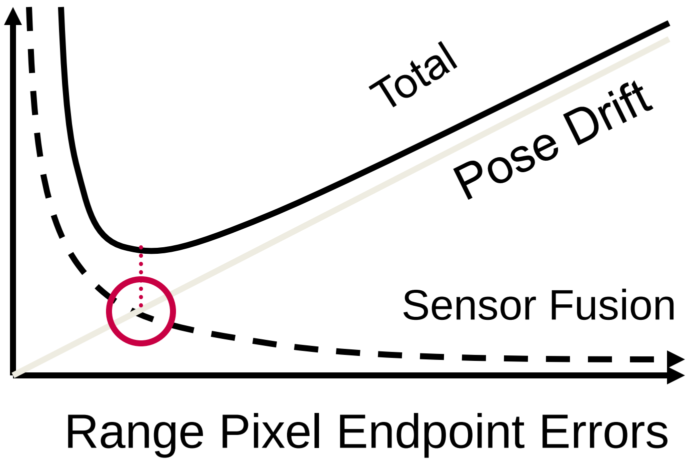
- 9.1 Representation and Issues
  - 9.1.1 Introduction
  - 9.1.2 Representation
  - 9.1.3 Timing and Motion issues
  - <u>9.1.4 Related Localization issues</u>
  - 9.1.5 Structural Aspects
  - 9.1.6 Example: Unmanned Ground Vehicle …
  - Summary

9.1.4.1 Localization Drift and Local Consistency

- Over time and/or distance, dead reckoning error of all types accumulates.
  - encoder odometry
  - gyro odometry
  - visual odometry and all related forms of determining relative pose from registration.
- Net result is that objects are both distorted and mislocated.
  - The degree of mislocation between two objects grows with the distance between them.

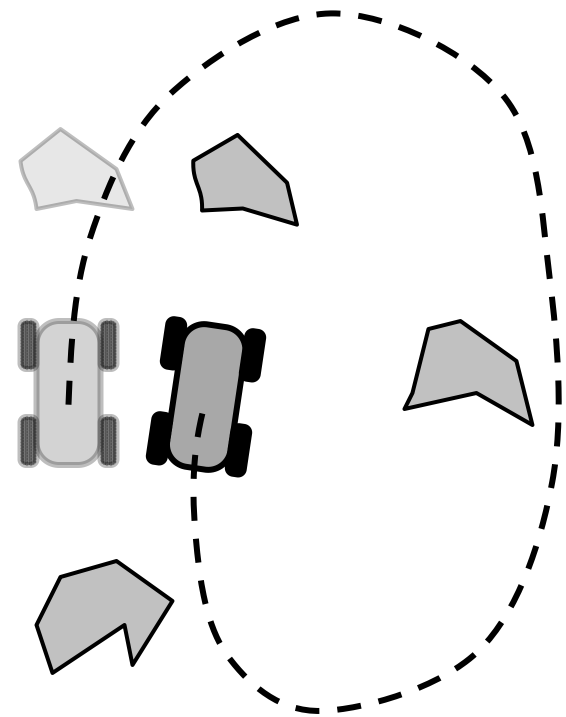
Accumulation / Distortion Tradeoff

- Evidence <u>accumulation</u> over time:
  - <u>reduces</u> effect of <u>unbiased error</u>.
  - <u>increases</u> the effect of <u>drift error</u>.
- A sweet spot exists when
  - drift = desired resolution/scale
- Approach:
  - Accumulate data until sweet spot.

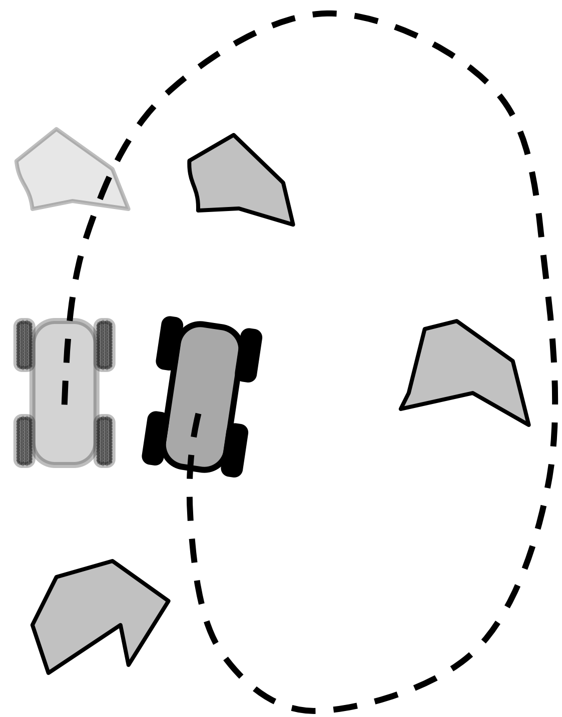


9.1.4.2 Data Aging and Global Inconsistency

- An extreme case of localization error issue.
- When the robot returns to a place visited earlier, integrating old and new data can lead to <u>two slightly displaced copies</u> of everything.
- One approach is to <u>limit the memory</u> of data which is based on a drifting estimate.



Global Inconsistency



Outline

- 9.1 Representation and Issues
  - 9.1.1 Introduction
  - 9.1.2 Representation
  - 9.1.3 Timing and Motion issues
  - 9.1.4 Related Localization issues
  - 9.1.5 Structural Aspects
  - 9.1.6 Example: Unmanned Ground Vehicle …
  - Summary

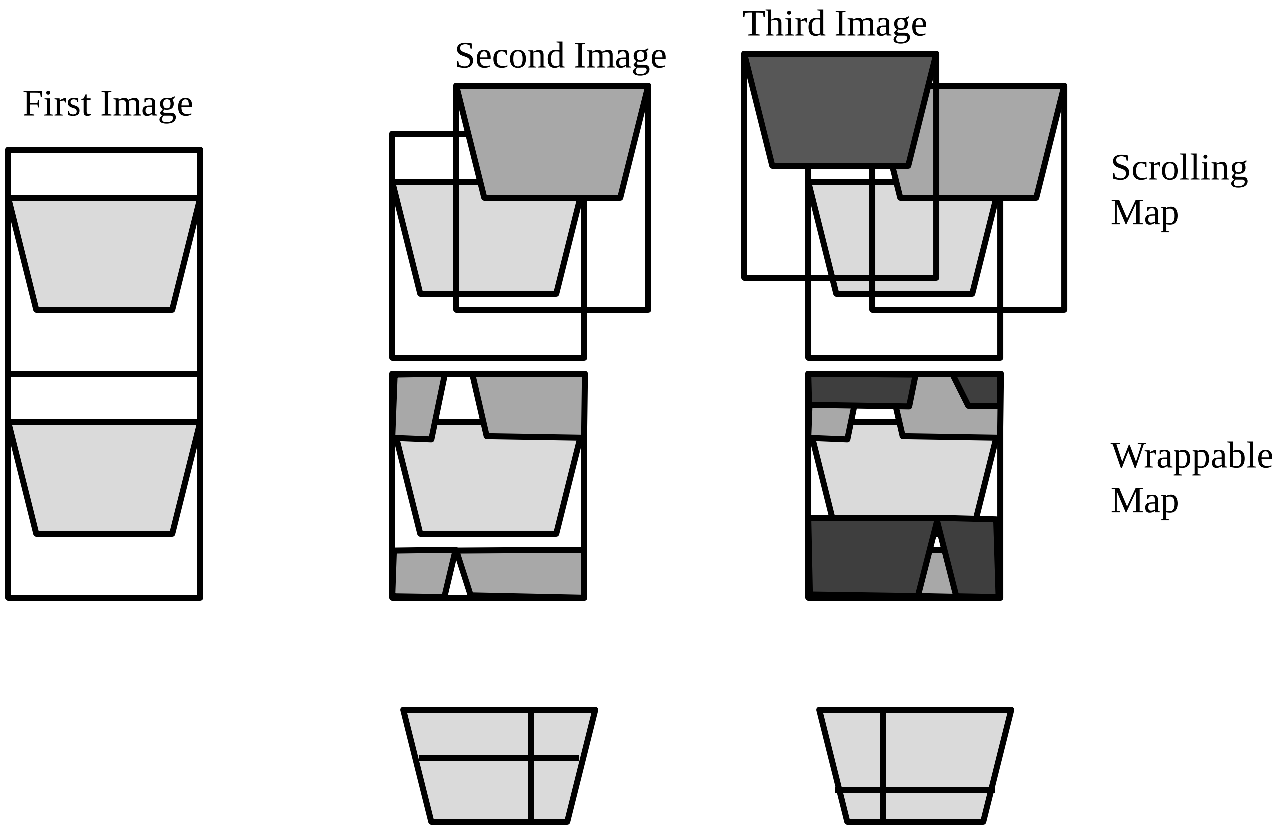
9.1.5.1 Structural Aspects

- Dimension: <u>2D</u> or <u>3D</u> storage indexing
  - Have an array of cells or a cube of voxels.
  - Does benefit justify cost of 3D?
- <u>Metric</u> or <u>Topological</u>
  - Sometimes knowing which edge you are on is enough.
  - Other times, info available is much richer.
- Topology
  - Will the map have <u>cyclic</u> or acyclic (<u>tree</u>) structure.
  - Latter avoids difficult "loop closing" problem of SLAM.

9.1.5.2 Extent

- Big maps can be too big to store. When they are too big, there are a few options.
- Cacheing - keep most of it on disk and know how to ship small pieces in and out efficiently.
- Wraparound - a multidimensional ring buffer can be used to continue to reuse the same memory while always surrounding the vehicle with the nearby data.

Fig 9.8 Scrolling and Wrapping Maps



9.1.5.2 Extent

(Duration: Short Term)

- Short Term:
  - Great way to avoid many integrated skew issues.
  - Makes sense for a map whose sole purpose is obstacle avoidance.
- "Aging" data is one good/efficient idea.
  - Render it artificially invisible after some time/distance window.
- There are two issues to deal with:
  - Since data is not explicitly deleted, need a mechanism to render old data invisible.
  - When new data is actually associated with a different place than the old data, the old data must be erased.

9.1.5.2 Extent

(Duration: Long Term)

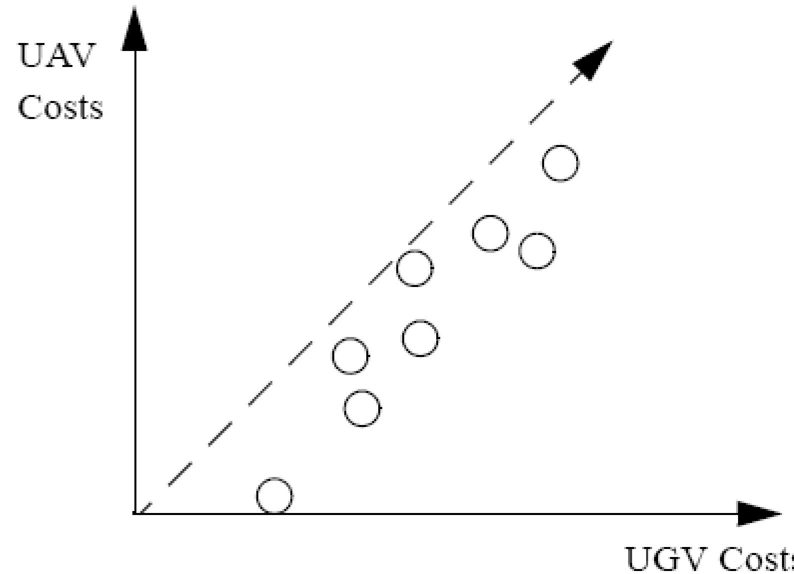
- For mapping, need to remember a lot to be able to detect loop closure.
  - and need enough resolution to do comparision.
- For global planning, need
  - Long term memory
  - Large map extent
  - but often its possible to sacrifice resolution.

9.1.5.3 Hierarchy

- Duplicated data
  - problem in planning maps
  - asset in navigation maps (helps close loops).
- A two layer hierarchy is useful here.
  - Organize map into rigid chunks.
  - The chunks remain rigid but they can be moved with respect to each other.
  - Such a structure is natural for scanning laser radar (1D or 2D) , stereo, and camera imagery.

9.1.5.4 Layers

- Maps can usefully have multiple "bit planes" or layers.
- One use of this is to <u>keep data from separate sources separate</u>, to enable:
  - Variable weighting.
  - Registration.
  - Calibration for (cost) consistency.



Outline

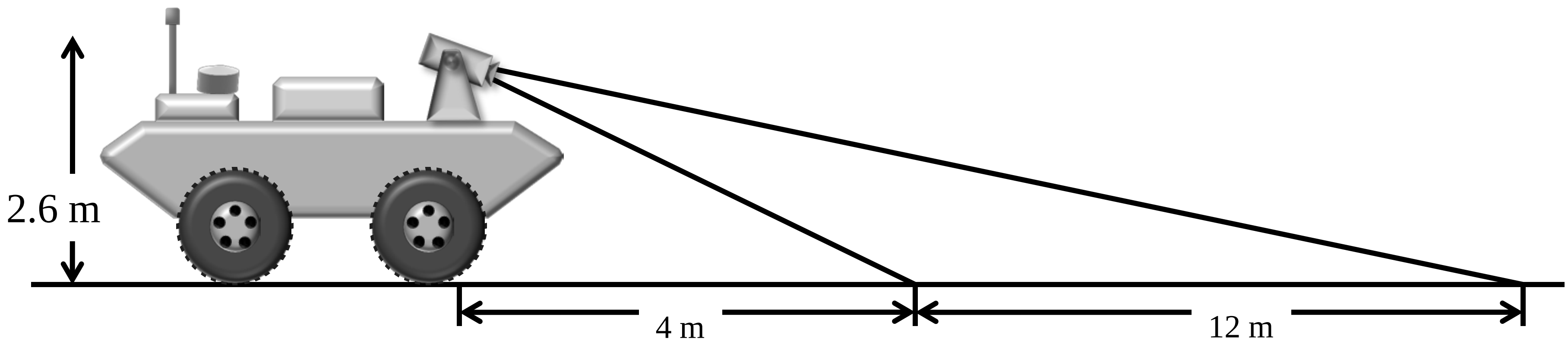
- 9.1 Representation and Issues
  - 9.1.1 Introduction
  - 9.1.2 Representation
  - 9.1.3 Timing and Motion issues
  - 9.1.4 Related Localization issues
  - 9.1.5 Structural Aspects
  - <u>9.1.6 Example: Unmanned Ground Vehicle …</u>
  - Summary

9.1.6 Example : Unmanned Ground Vehicle

(Demo II Terrain Mapping (1992))

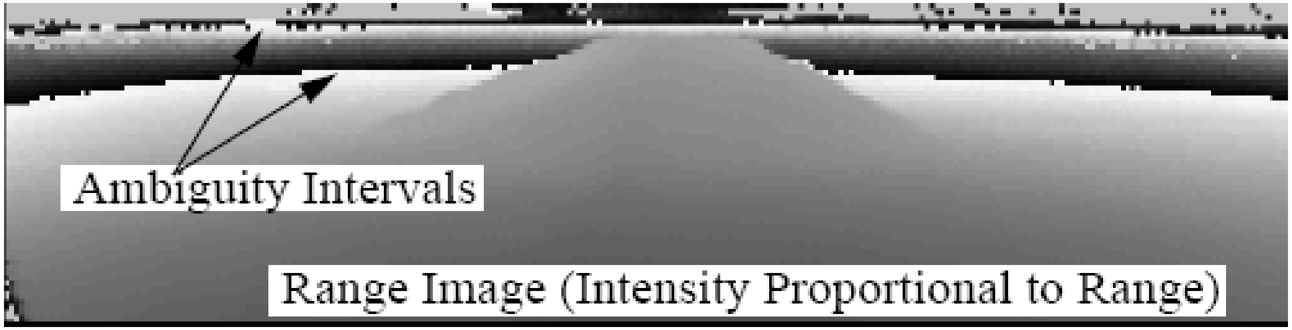
- Similar to Hughes ALV system. Reused on PerceptOR
- ERIM laser rangefinder, SICK laser rangefinder(s)
- Models terrain with elevation map encoding z(x,y) in sampled form. Cells accumulate an average and variance for z over time.
- Fills holes with interpolation (UGV)
- Supports obstacle detection and path planning.
- Designed for high speed on rough terrain.

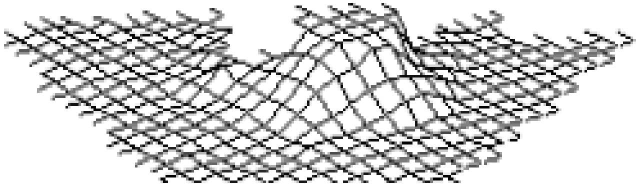
9.1.6 Example : Unmanned Ground Vehicle
(Sensor Configuration)



Scanning Lidar Specifications			
Horizontal Field of view	80°	Horizontal Range Pixels	256
Vertical Field of View	30°	Vertical Range Pixels	64
Range Resolution	10 cm	Frame Rate	2 Hz

9.1.6.1 Range Imagery and Terrain Map





Wireframe Surface
Representation of CEM

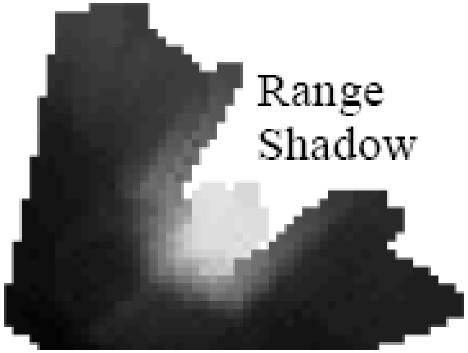
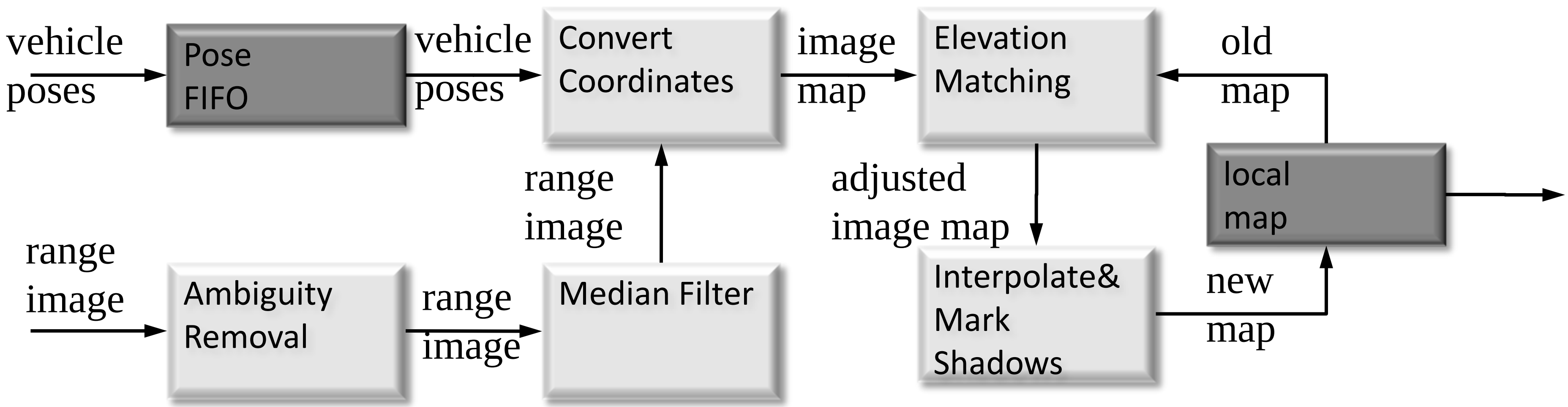


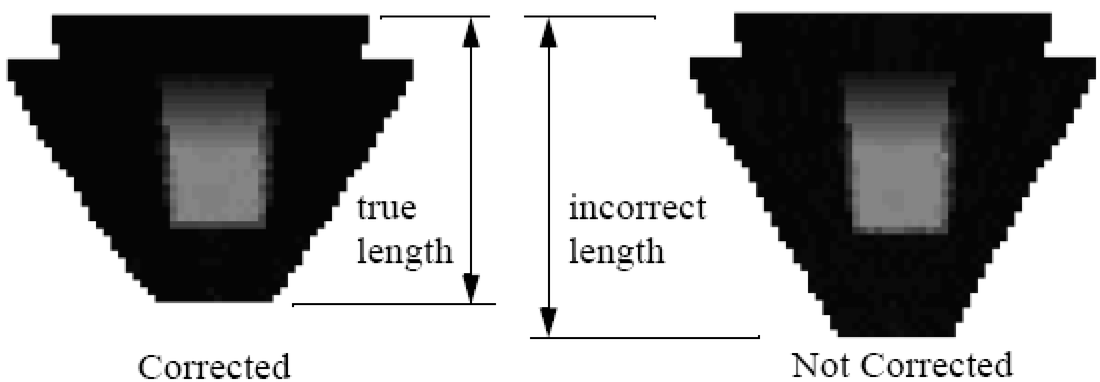
Image Representation of
CEM (Intensity proportional
to elevation)

9.1.6.2 Software Data Flow



9.1.6.3 Motion Distortion Removal

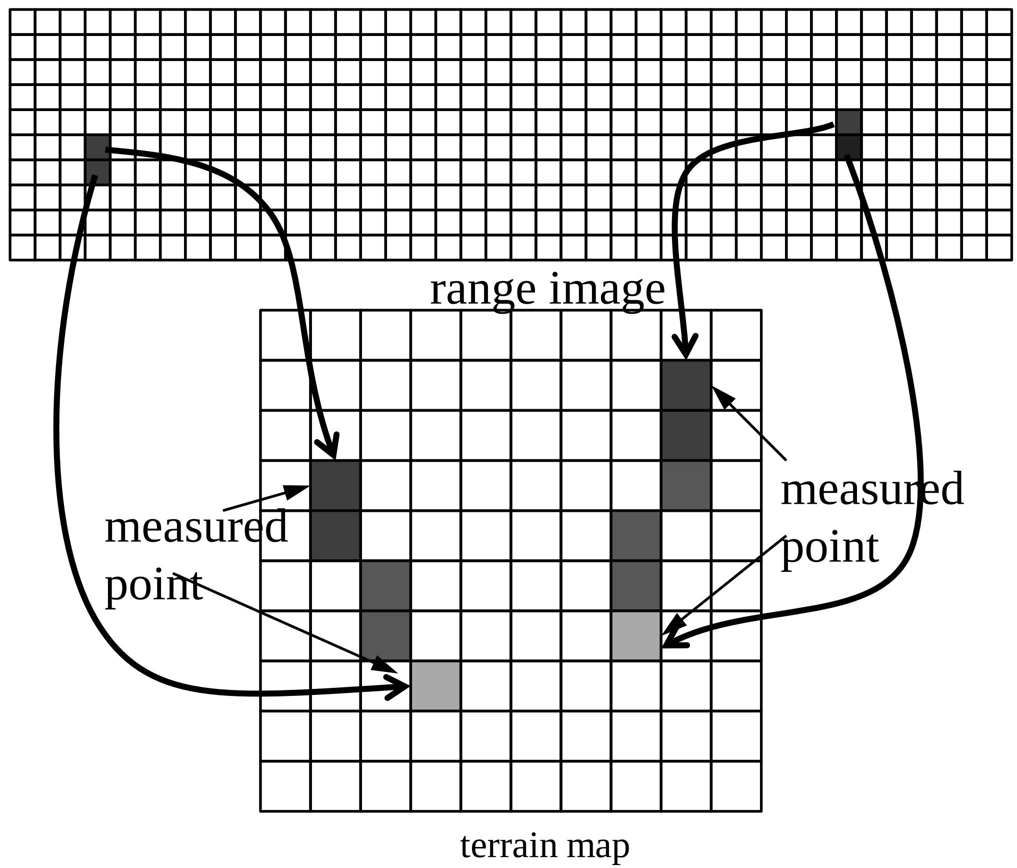
- Pose tags of each image were interpolated to approximate vehicle motion.
  - Different pose for each range pixel.



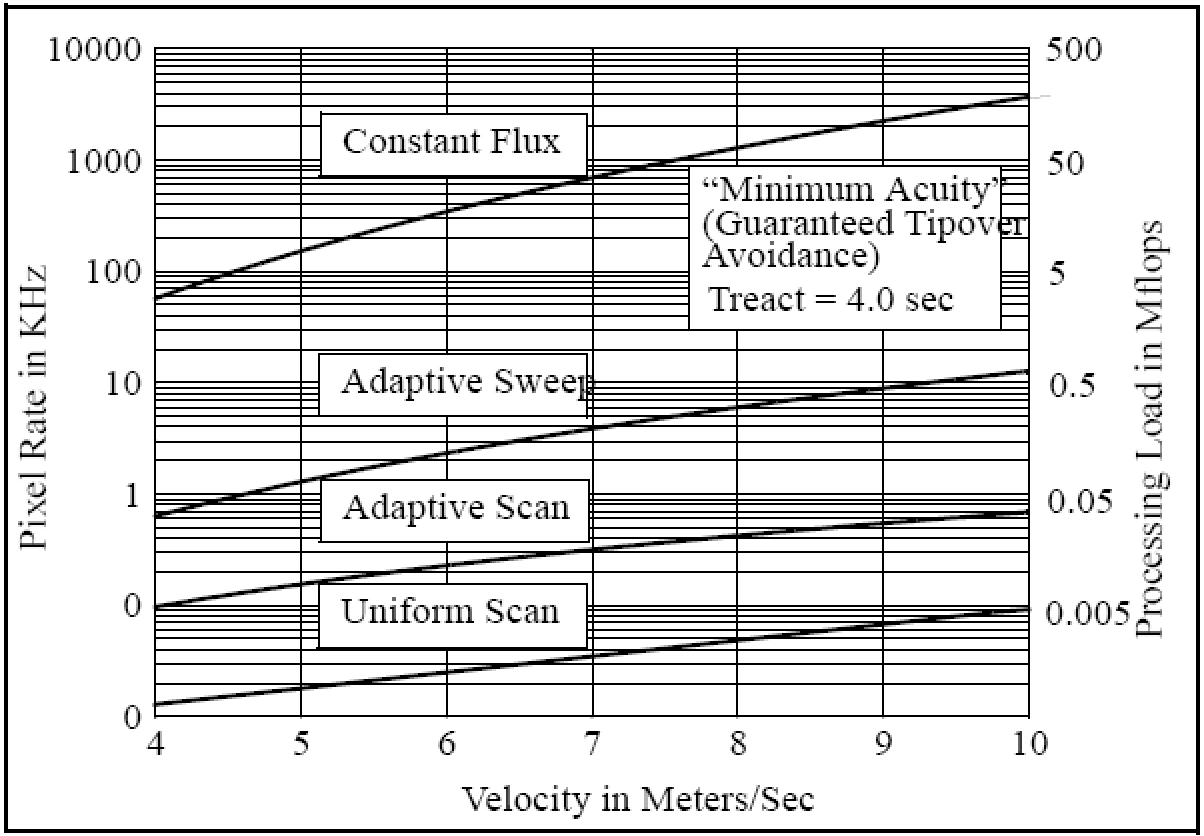
9.1.6.5 Sampling Issues

(Scan Conversion and Interpolation)

- Solution for sampling problem.
- Based on linear interpolation and Bresenham's Line algorithm
- Unified treatment of undersampling and range shadows

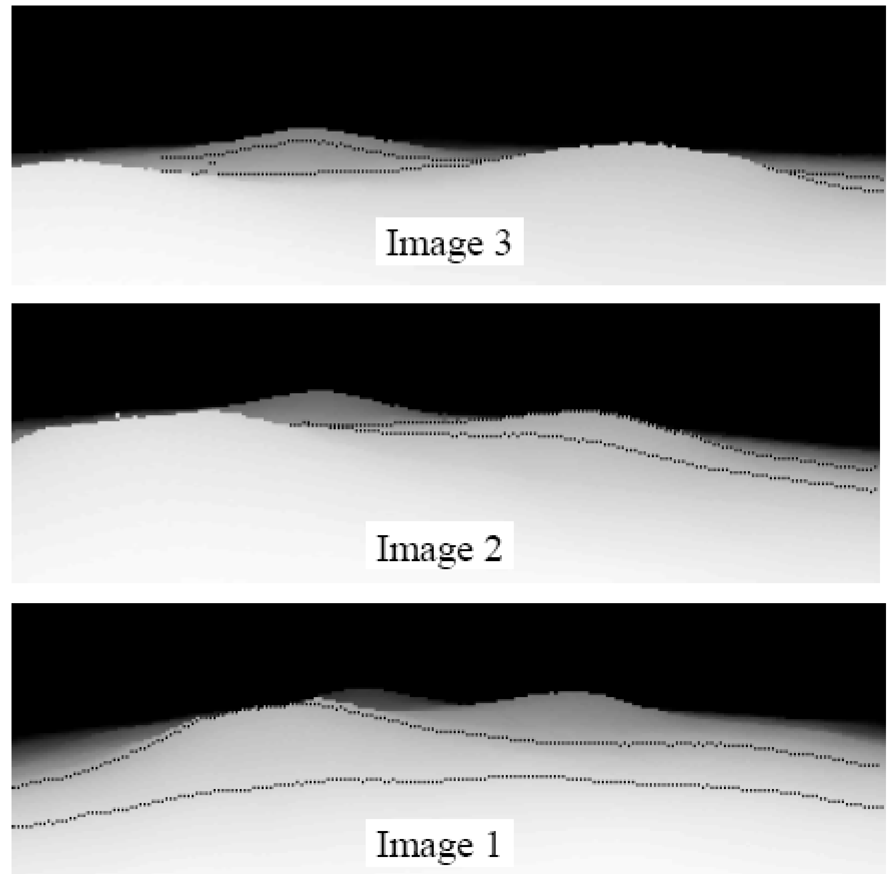


9.1.6.6 Computational Image Stabilization
(Processing Requirements)



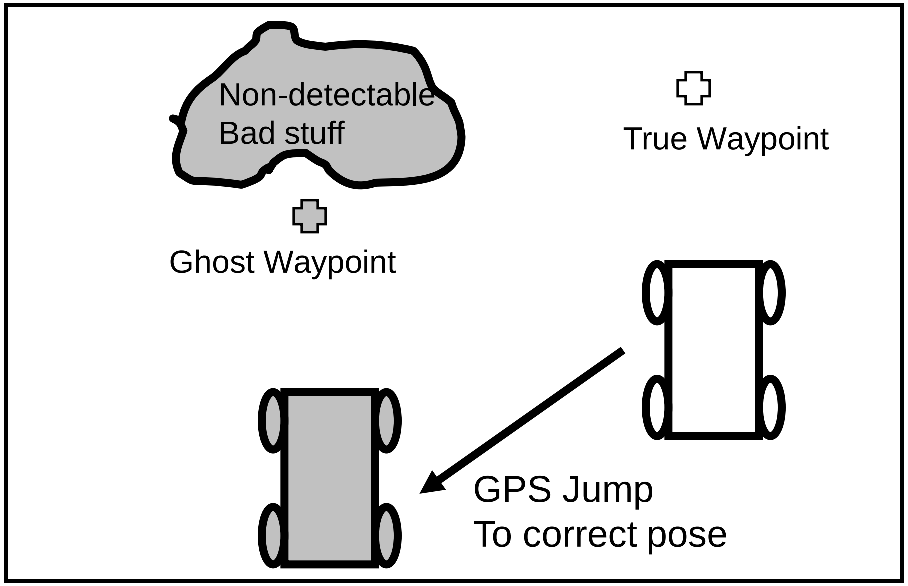
9.1.6.6 Computational Image Stabilization

- Converts ROI to polar coordinates in real time.

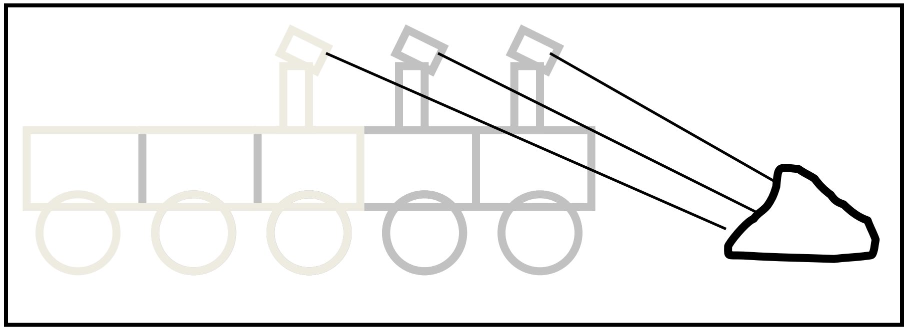


9.1.6.7 Dual Pose Estimates

- GPS "jumps" are the rule in complex natural and urban settings.
- Jumps due to intermittent high quality measurements are …
  - Great for waypoint following and map creation or processing.
  - Disaster for ladar-based obstacle detection.
- Perception data accumulation at different scales has conflicting pose quality requirements.

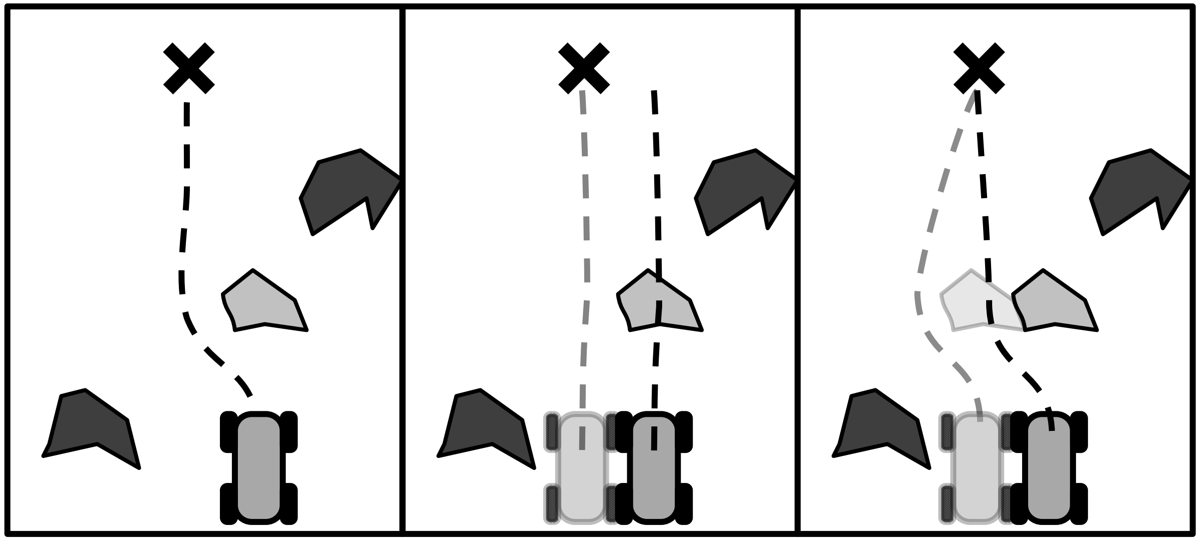


GPS Jump at satellite acquisition is <u>needed</u> for goal acquisition.



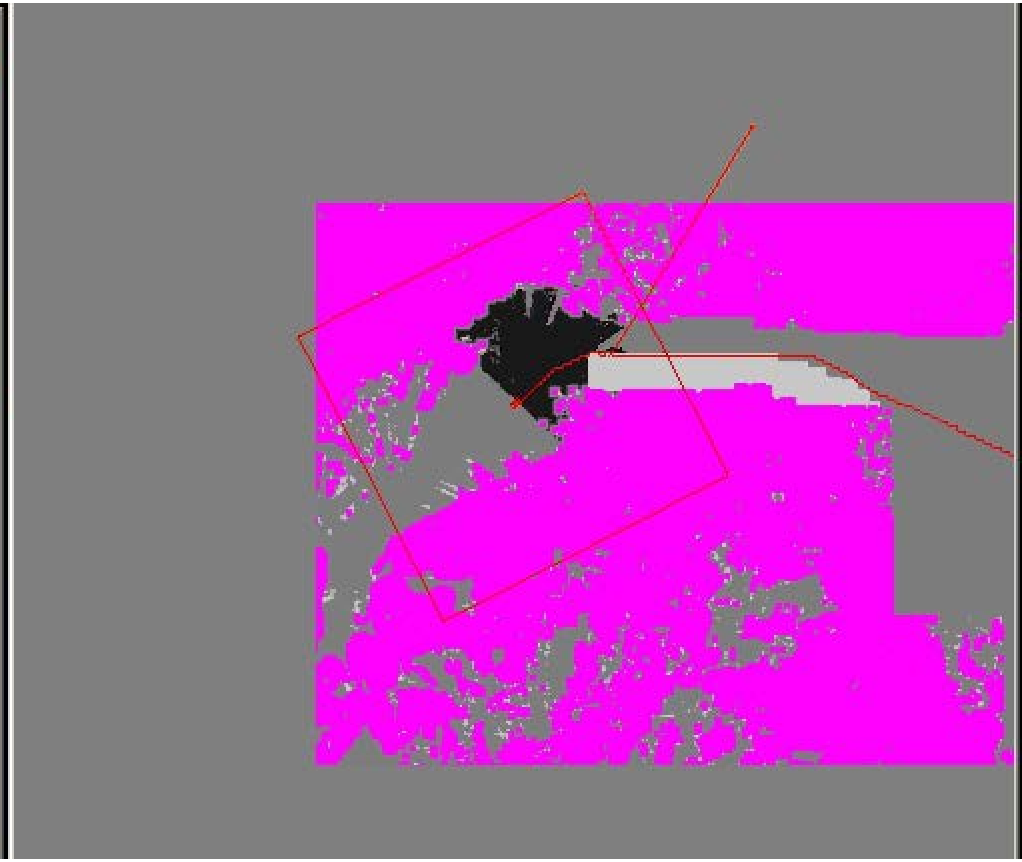
GPS Jump at satellite acquisition is <u>disaster</u> for perceptual data fusion.

9.1.6.7 Dual Pose Estimates



- 1: Dual Estimates: a) globally accurate and b)locally smooth.
- 2: Filtering: Local estimate does not process any measurement which projects directly onto position or orientation states.
- 3: Lazy Registration: Local and global obstacle data registered (lazy) whenever its needed.
- Point: Obstacle Avoidance becomes completely immune to GPS drift and jumps.

PerceptOR Cooperative Mapping



Summary

- Maps can be used for navigation or planning.
- Maps use memory to increase the amount of information available for decision making.
- Many design issues remiscent of data structures occur.