Practical Vision-Based Monte Carlo Localization on a Legged Robot

Mohan Sridharan   Gregory Kuhlmann   Peter Stone

Learning Agents Research Group
Department of Computer Sciences
The University of Texas at Austin

IEEE International Conference on Robotics and Automation, 2005

The Problem

Mobile Robot Localization

Maintain estimate of global position and orientation over time

- Given map of fixed landmark locations
- Not SLAM

The Problem

Mobile Robot Localization

Maintain estimate of global position and orientation over time

- Given map of fixed landmark locations
- Not SLAM

Challenging Platform

Typical Platform

- Wheeled robot
- Range-finding sensors

Sony Aibo ERS-7

- Color CMOS Camera in nose
  - Narrow field-of-view (56^o)
  - 30 YCrCb frames per second
- Quadruped
- 576MHz processor
  - All on-board processing

Challenging Platform

Our Platform

- Legged robot
- Vision-based sensors

Sony Aibo ERS-7

- Color CMOS Camera in nose
  - Narrow field-of-view (56^o)
  - 30 YCrCb frames per second
- Quadruped
- 576MHz processor
  - All on-board processing

Goal

Desiderata

- Navigate to specific point quickly
- Remain localized while colliding
- Recover quickly from kidnappings

Approach

- Begin with baseline MCL algorithm
- Add set of practical enhancements

Large improvement over baseline

Goal

Desiderata

- Navigate to specific point quickly
- Remain localized while colliding
- Recover quickly from kidnappings

Approach

- Begin with baseline MCL algorithm
- Add set of practical enhancements

Large improvement over baseline

Goal

Desiderata

- Navigate to specific point quickly
- Remain localized while colliding
- Recover quickly from kidnappings

Approach

- Begin with baseline MCL algorithm
- Add set of practical enhancements

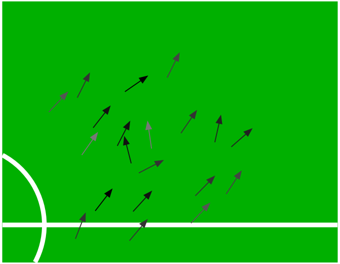
Large improvement over baseline

Method: Particle Filtering

- Estimate $p({\color{red}h_T}|o_T, a_{T-1}, o_{T-1}, a_{T-2}, \ldots, a_0)$: Distribution of poses given observations and actions
- Represented by finite set of samples: particles
  - Each is a hypothesis: $\langle\langle x, y, \theta\rangle, p\rangle$
- Average to get single estimate of pose and confidence

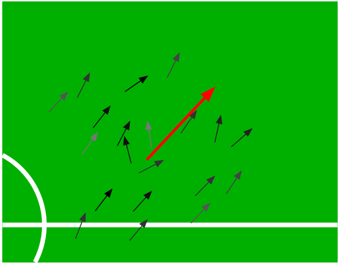
Method: Particle Filtering

- Estimate $p(h_T | o_T, a_{T-1}, o_{T-1}, a_{T-2}, \ldots, a_0)$:
  Distribution of poses given observations and actions
- Represented by finite set of samples: particles
  - Each is a hypothesis: $\langle\langle x, y, \theta\rangle, p\rangle$
- Average to get single estimate of pose and confidence



Method: Particle Filtering

- Estimate $p(h_T | o_T, a_{T-1}, o_{T-1}, a_{T-2}, \ldots, a_0)$:
  Distribution of poses given observations and actions
- Represented by finite set of samples: particles
  - Each is a hypothesis: $\langle \langle x, y, \theta \rangle, p \rangle$
- Average to get single estimate of pose and confidence



Outline

1. Practical Enhancements
   - Distance-Based Updates
   - Landmark Histories
   - Extended Motion Model
2. Empirical Results
   - Physical Robot Experiments
   - Simulation Experiments

Outline

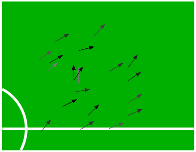
1. Practical Enhancements
   - Distance-Based Updates
   - Landmark Histories
   - Extended Motion Model
2. Empirical Results
   - Physical Robot Experiments
   - Simulation Experiments

Baseline: Observation Update

- Need sensor model: $p(o|h)$
  - Predicts observations given pose hypothesis using map
- Update each particle when robot sees something
  - Compute similarity for each observed landmark in frame
    - Use angles only [Rofer and Jungel, 2003]
    - Measured and expected angle difference
  - Compute product of similarities
  - Adjust probability closer to new value

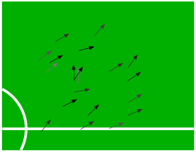
Baseline: Observation Update

- Need sensor model: $p(o|h)$
  - Predicts observations given pose hypothesis using map
- Update each particle when robot sees something
  - Compute similarity for each observed landmark in frame
    - Use angles only [Rofer and Jungel, 2003]
    - Measured and expected angle difference
  - Compute product of similarities
  - Adjust probability closer to new value



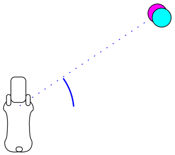
Baseline: Observation Update

- Need sensor model: $p(o|h)$
  - Predicts observations given pose hypothesis using map
- Update each particle when robot sees something
  - Compute similarity for each observed landmark in frame
    - Use angles only [Rofer and Jungel, 2003]
    - Measured and expected angle difference
  - Compute product of similarities
  - Adjust probability closer to new value



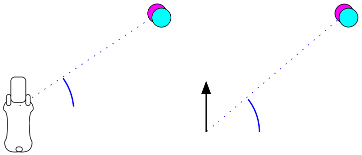
Baseline: Observation Update

- Need sensor model: $p(o|h)$
  - Predicts observations given pose hypothesis using map
- Update each particle when robot sees something
  - Compute similarity for each observed landmark in frame
    - Use angles only [Rofer and Jungel, 2003]
    - Measured and expected angle difference
  - Compute product of similarities
  - Adjust probability closer to new value



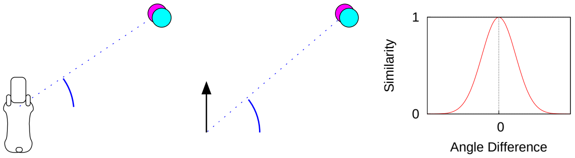
Baseline: Observation Update

- Need sensor model: $p(o|h)$
  - Predicts observations given pose hypothesis using map
- Update each particle when robot sees something
  - Compute similarity for each observed landmark in frame
    - Use angles only [Rofer and Jungel, 2003]
    - Measured and expected angle difference
  - Compute product of similarities
  - Adjust probability closer to new value



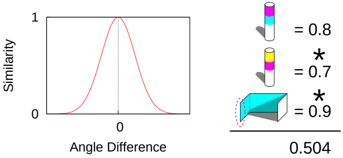
Baseline: Observation Update

- Need sensor model: $p(o|h)$
  - Predicts observations given pose hypothesis using map
- Update each particle when robot sees something
  - Compute similarity for each observed landmark in frame
    - Use angles only [Rofer and Jungel, 2003]
    - Measured and expected angle difference
  - Compute product of similarities
  - Adjust probability closer to new value



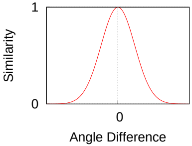
Baseline: Observation Update

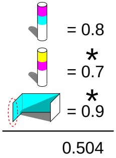
- Need sensor model: $p(o|h)$
  - Predicts observations given pose hypothesis using map
- Update each particle when robot sees something
  - Compute similarity for each observed landmark in frame
    - Use angles only [Rofer and Jungel, 2003]
    - Measured and expected angle difference
  - Compute product of similarities
  - Adjust probability closer to new value

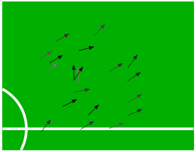


Baseline: Observation Update

- Need sensor model: $p(o|h)$
  - Predicts observations given pose hypothesis using map
- Update each particle when robot sees something
  - Compute similarity for each observed landmark in frame
    - Use angles only [Rofer and Jungel, 2003]
    - Measured and expected angle difference
  - Compute product of similarities
  - Adjust probability closer to new value

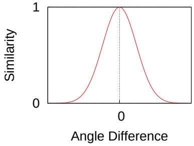


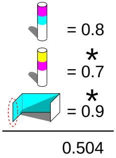


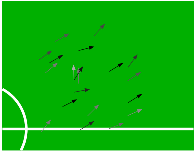


Baseline: Observation Update

- Need sensor model: $p(o|h)$
    - Predicts observations given pose hypothesis using map
- Update each particle when robot sees something
    - Compute similarity for each observed landmark in frame
        - Use angles only [Rofer and Jungel, 2003]
        - Measured and expected angle difference
    - Compute product of similarities
    - Adjust probability closer to new value





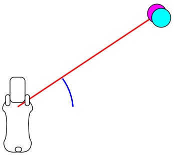


Enhancement: Distance-Based Updates

- Enhancement to observation update
  - Use distance in addition to angle
- Update each particle
  - Difference between measured and expected distance
  - Use average of distance and angle similarities
- Distances must be very accurate

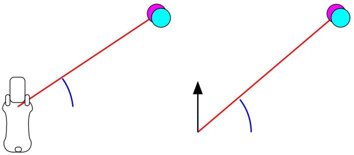
Enhancement: Distance-Based Updates

- Enhancement to observation update
  - Use distance in addition to angle
- Update each particle
  - Difference between measured and expected distance
  - Use average of distance and angle similarities
- Distances must be very accurate



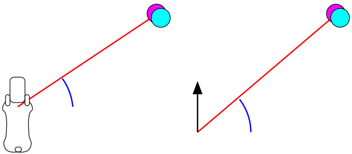
Enhancement: Distance-Based Updates

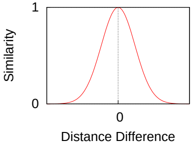
- Enhancement to observation update
  - Use distance in addition to angle
- Update each particle
  - Difference between measured and expected distance
  - Use average of distance and angle similarities
- Distances must be very accurate



Enhancement: Distance-Based Updates

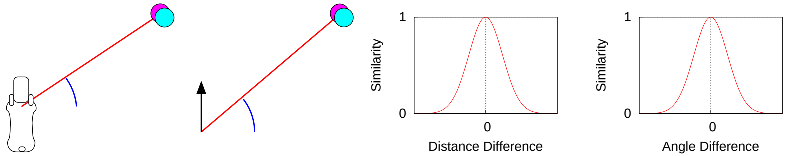
- Enhancement to observation update
  - Use distance in addition to angle
- Update each particle
  - Difference between measured and expected distance
  - Use average of distance and angle similarities
- Distances must be very accurate





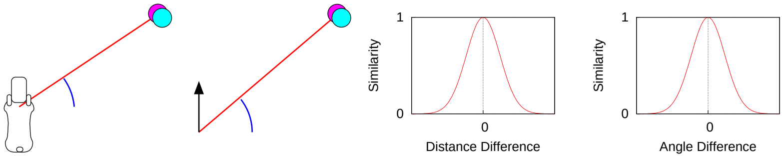
Enhancement: Distance-Based Updates

- Enhancement to observation update
  - Use distance in addition to angle
- Update each particle
  - Difference between measured and expected distance
  - Use average of distance and angle similarities
- Distances must be very accurate



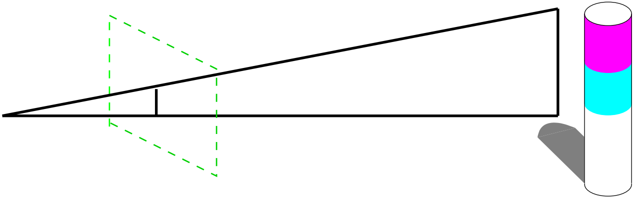
Enhancement: Distance-Based Updates

- Enhancement to observation update
  - Use distance in addition to angle
- Update each particle
  - Difference between measured and expected distance
  - Use average of distance and angle similarities
- Distances must be very accurate



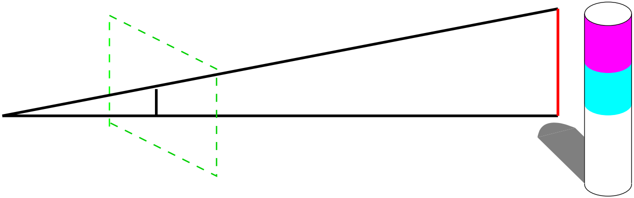
Estimating Landmark Distances

- Know actual height of beacon and focal length of camera
- Measure height of beacon in image
- Use similar triangles to find distance
- Error due to pixelized segmentation, distortion, etc.



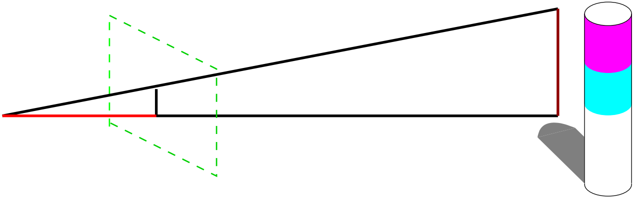
Estimating Landmark Distances

- Know actual height of beacon and focal length of camera
- Measure height of beacon in image
- Use similar triangles to find distance
- Error due to pixelized segmentation, distortion, etc.



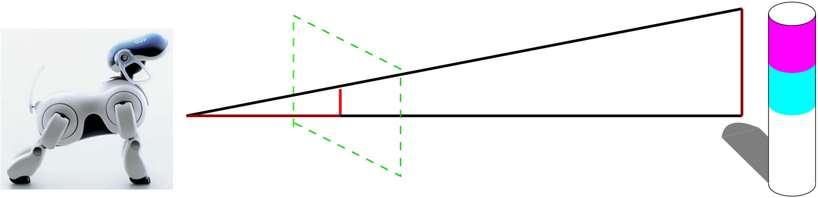
Estimating Landmark Distances

- Know actual height of beacon and focal length of camera
- Measure height of beacon in image
- Use similar triangles to find distance
- Error due to pixelized segmentation, distortion, etc.



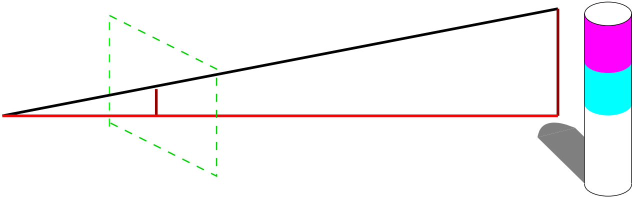
Estimating Landmark Distances

- Know actual height of beacon and focal length of camera
- Measure height of beacon in image
- Use similar triangles to find distance
- Error due to pixelized segmentation, distortion, etc.



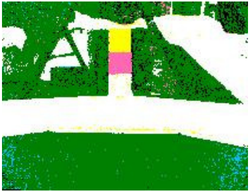
Estimating Landmark Distances

- Know actual height of beacon and focal length of camera
- Measure height of beacon in image
- Use similar triangles to find distance
- Error due to pixelized segmentation, distortion, etc.



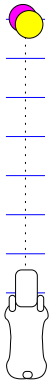
Estimating Landmark Distances

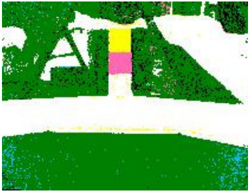
- Know actual height of beacon and focal length of camera
- Measure height of beacon in image
- Use similar triangles to find distance
- Error due to pixelized segmentation, distortion, etc.

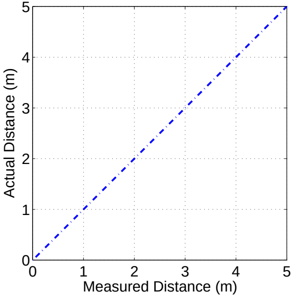


Function Approximation

- Place robot at known distances
- Actual and Measured don't match (Nonlinear relationship)
- Approximate function using cubic regression for each landmark
- Maximum error reduced to 5%

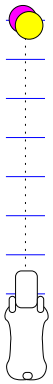


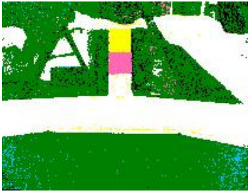


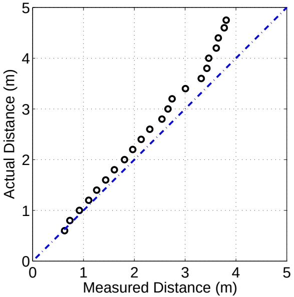


Function Approximation

- Place robot at known distances
- Actual and Measured don't match (Nonlinear relationship)
- Approximate function using cubic regression for each landmark
- Maximum error reduced to 5%

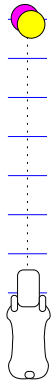


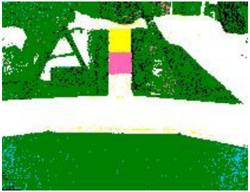


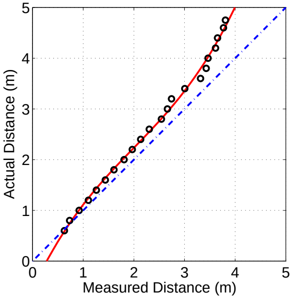


Function Approximation

- Place robot at known distances
- Actual and Measured don't match (Nonlinear relationship)
- Approximate function using cubic regression for each landmark
- Maximum error reduced to 5%

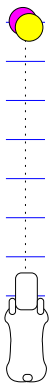


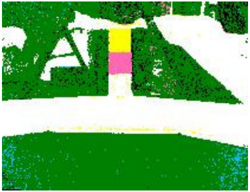


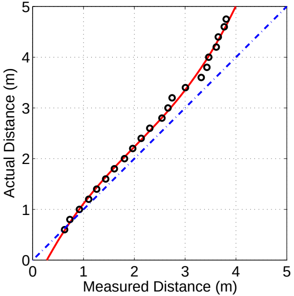


Function Approximation

- Place robot at known distances
- Actual and Measured don't match (Nonlinear relationship)
- Approximate function using cubic regression for each landmark
- Maximum error reduced to 5%





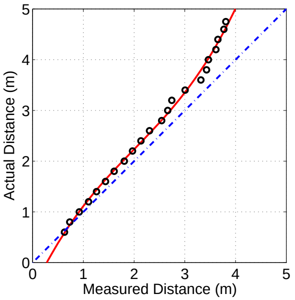


Function Approximation

- Place robot at known distances
- Actual and Measured don't match (Nonlinear relationship)
- Approximate function using cubic regression for each landmark
- Maximum error reduced to 5%

Result

Distances safe to use.



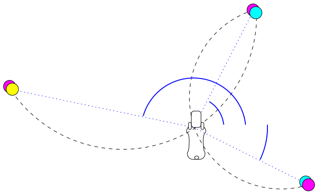
Outline

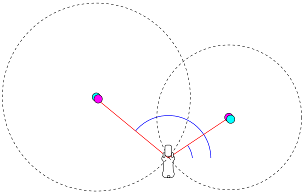
Baseline: Reseeding

- Based on Sensor Resetting MCL [Lenser et al., 2000]
    - Helps recovery when lost
- Triangulate position using multiple landmarks
    - Three landmarks using just angles
    - Two landmarks using distances and angles
- Add new hypotheses before resampling step

Baseline: Reseeding

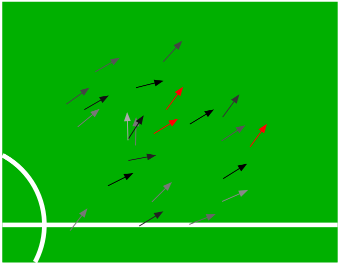
- Based on Sensor Resetting MCL [Lenser et al., 2000]
  - Helps recovery when lost
- Triangulate position using multiple landmarks
  - Three landmarks using just angles
  - Two landmarks using distances and angles
- Add new hypotheses before resampling step





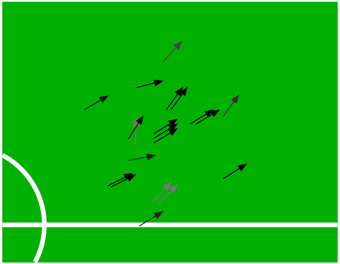
Baseline: Reseeding

- Based on Sensor Resetting MCL [Lenser et al., 2000]
  - Helps recovery when lost
- Triangulate position using multiple landmarks
  - Three landmarks using just angles
  - Two landmarks using distances and angles
- Add new hypotheses before resampling step



Baseline: Reseeding

- Based on Sensor Resetting MCL [Lenser et al., 2000]
  - Helps recovery when lost
- Triangulate position using multiple landmarks
  - Three landmarks using just angles
  - Two landmarks using distances and angles
- Add new hypotheses before resampling step



Baseline: Reseeding

- Based on Sensor Resetting MCL [Lenser et al., 2000]
  - Helps recovery when lost
- Triangulate position using multiple landmarks
  - Three landmarks using just angles
  - Two landmarks using distances and angles
- Add new hypotheses before resampling step

Shortcoming

- Robot must see multiple landmarks in the same frame
  - Infrequent with narrow field-of-view camera

Enhancement: Landmark Histories

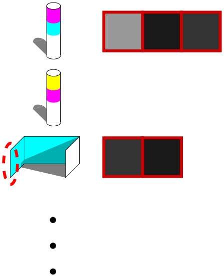
- Want more reseeding values
  - Maintain “history” of recent observations
- Observation list for each landmark
  - Record: Dist, Ang, Conf, Timestamp, Odometer

- Motion update
- Confidence decay
- Remove old
- Weighted average
- Combine for reseed

Enhancement: Landmark Histories

- Want more reseeding values
  - Maintain “history” of recent observations
- Observation list for each landmark
  - Record: Dist, Ang, Conf, Timestamp, Odometer

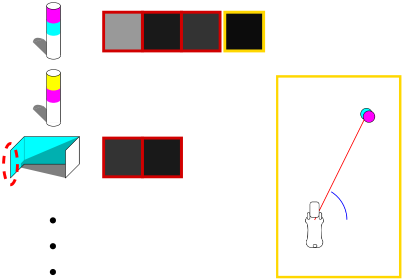
- Motion update
- Confidence decay
- Remove old
- Weighted average
- Combine for reseed



Enhancement: Landmark Histories

- Want more reseeding values
  - Maintain “history” of recent observations
- Observation list for each landmark
  - Record: Dist, Ang, Conf, Timestamp, Odometer

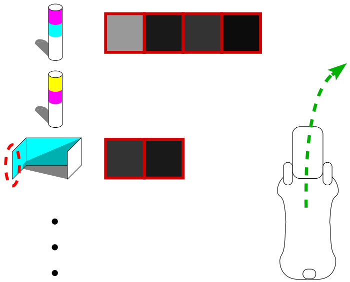
- Motion update
- Confidence decay
- Remove old
- Weighted average
- Combine for reseed



Enhancement: Landmark Histories

- Want more reseeding values
  - Maintain “history” of recent observations
- Observation list for each landmark
  - Record: Dist, Ang, Conf, Timestamp, Odometer

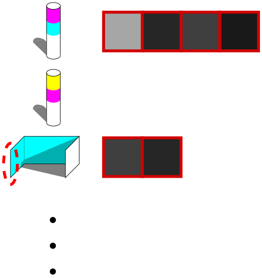
- Motion update
- Confidence decay
- Remove old
- Weighted average
- Combine for reseed



Enhancement: Landmark Histories

- Want more reseeding values
  - Maintain “history” of recent observations
- Observation list for each landmark
  - Record: Dist, Ang, Conf, Timestamp, Odometer

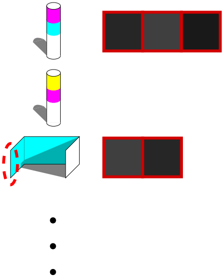
- Motion update
- Confidence decay
- Remove old
- Weighted average
- Combine for reseed



Enhancement: Landmark Histories

- Want more reseeding values
  - Maintain “history” of recent observations
- Observation list for each landmark
  - Record: Dist, Ang, Conf, Timestamp, Odometer

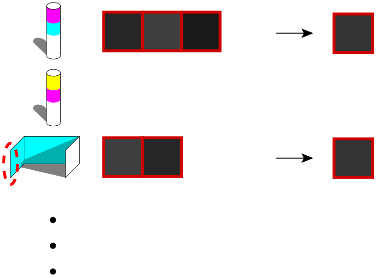
- Motion update
- Confidence decay
- Remove old
- Weighted average
- Combine for reseed



Enhancement: Landmark Histories

- Want more reseeding values
  - Maintain “history” of recent observations
- Observation list for each landmark
  - Record: Dist, Ang, Conf, Timestamp, Odometer

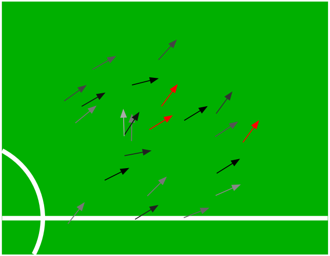
- Motion update
- Confidence decay
- Remove old
- Weighted average
- Combine for reseed



Enhancement: Landmark Histories

- Want more reseeding values
  - Maintain “history” of recent observations
- Observation list for each landmark
  - Record: Dist, Ang, Conf, Timestamp, Odometer

- Motion update
- Confidence decay
- Remove old
- Weighted average
- Combine for reseed



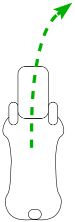
Outline

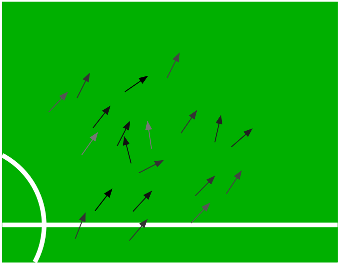
1. Practical Enhancements
   - Distance-Based Updates
   - Landmark Histories
   - Extended Motion Model

2. Empirical Results
   - Physical Robot Experiments
   - Simulation Experiments

Baseline: Motion Update

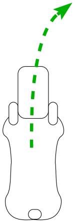
- Need motion model: $p(h'|h, a)$
  - Predict new pose given previous hypothesis and action
- Update each particle when robot moves
  - Use odometry velocities to translate particles

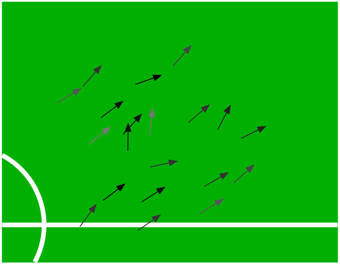




Baseline: Motion Update

- Need motion model: $p(h'|h, a)$
  - Predict new pose given previous hypothesis and action
- Update each particle when robot moves
  - Use odometry velocities to translate particles





Enchancement: Extended Motion Model

Problem

- Tradeoff between speed and motion model accuracy
  - Large steps over small distances inaccurate
  - Unable to navigate to specific point

Solution: Change Behavior

- Use accurate but slower walk near target
  - Step size reduced to 10% within 300mm of target

Enchancement: Extended Motion Model

Problem

- Tradeoff between speed and motion model accuracy
  - Large steps over small distances inaccurate
  - Unable to navigate to specific point

Solution: Change Behavior

- Use accurate but slower walk near target
  - Step size reduced to 10% within 300*mm* of target

Enchancement: Extended Motion Model

Problem

- Tradeoff between speed and motion model accuracy
  - Large steps over small distances inaccurate
  - Unable to navigate to specific point

Solution: Change Behavior

- Use accurate but slower walk near target
  - Step size reduced to 10% within 300mm of target

Enchancement: Extended Motion Model

Problem

- Tradeoff between speed and motion model accuracy
  - Large steps over small distances inaccurate
  - Unable to navigate to specific point

Solution: Change Behavior

- Use accurate but slower walk near target
  - Step size reduced to 10% within 300*mm* of target

Enchancement: Extended Motion Model

Problem

- Tradeoff between speed and motion model accuracy
  - Large steps over small distances inaccurate
  - Unable to navigate to specific point

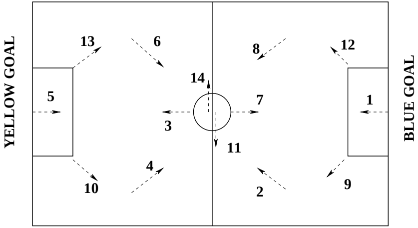
Solution: Change Behavior

- Use accurate but slower walk near target
  - Step size reduced to 10% within 300*mm* of target

Outline

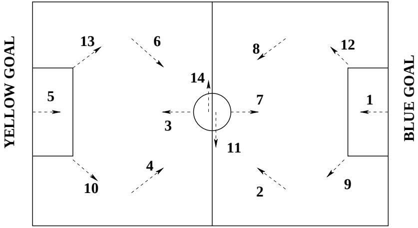
Test for Accuracy and Time

- Environment: RoboCup Legged League field
  - Size: roughly $3m \times 5m$
  - Landmarks: 4 beacons, 4 goal edges
- Visit sequence of 14 points and headings
- After stabilizing at a point, measure
  - Time taken
  - Position and orientation error



Test for Accuracy and Time

- Environment: RoboCup Legged League field
  - Size: roughly $3m \times 5m$
  - Landmarks: 4 beacons, 4 goal edges
- Visit sequence of 14 points and headings
- After stabilizing at a point, measure
  - Time taken
  - Position and orientation error



Test for Accuracy and Time

Six Localization Conditions

1. Baseline (None)
2. Landmark Histories (HST)
3. Distance-based probability updates (DST)
4. Function approximation of distances (FA)
5. Function approx. + distance-based updates (FA+DST)
6. All enhancements (All)

- Extended Motion Model present in all
- Average across 10 runs for each

Test for Accuracy and Time

Six Localization Conditions

1. Baseline (None)
2. Landmark Histories (HST)
3. Distance-based probability updates (DST)
4. Function approximation of distances (FA)
5. Function approx. + distance-based updates (FA+DST)
6. All enhancements (All)

- Extended Motion Model present in all
- Average across 10 runs for each

Results

Enhan.	Dist Err (cm)	Ang Err (deg)	Total Time (s)
None	19.75±12.0	17.75±11.48	161.25±3.43
HST	17.92±9.88	10.68±5.97	161.26±5.96
DST	25.07±13.73	9.14±5.46	196.18±12.18
FA	15.19±8.59	10.21±6.11	171.85±15.19
DST+FA	13.72±8.07	9.5±5.27	151.28±48.06
All	9.65±7.69	3.43±4.49	162.54±4.38

- With all enhancements
  - 50% reduction in position error
  - 80% reduction in orientation error
  - No significant change in time

Results

Enhan.	Dist Err (cm)	Ang Err (deg)	Total Time (s)
None	19.75±12.0	17.75±11.48	161.25±3.43
HST	17.92±9.88	10.68±5.97	161.26±5.96
DST	25.07±13.73	9.14±5.46	196.18±12.18
FA	15.19±8.59	10.21±6.11	171.85±15.19
DST+FA	13.72±8.07	9.5±5.27	151.28±48.06
All	9.65±7.69	3.43±4.49	162.54±4.38

- With all enhancements
  - 50% reduction in position error
  - 80% reduction in orientation error
  - No significant change in time

Results

Enhan.	Dist Err (cm)	Ang Err (deg)	Total Time (s)
None	19.75±12.0	17.75±11.48	161.25±3.43
HST	17.92±9.88	10.68±5.97	161.26±5.96
DST	25.07±13.73	9.14±5.46	196.18±12.18
FA	15.19±8.59	10.21±6.11	171.85±15.19
DST+FA	13.72±8.07	9.5±5.27	151.28±48.06
All	9.65±7.69	3.43±4.49	162.54±4.38

- With all enhancements
  - 50% reduction in position error
  - 80% reduction in orientation error
  - No significant change in time

Results

Enhan.	Dist Err (cm)	Ang Err (deg)	Total Time (s)
None	19.75±12.0	17.75±11.48	161.25±3.43
HST	17.92±9.88	10.68±5.97	161.26±5.96
DST	25.07±13.73	9.14±5.46	196.18±12.18
FA	15.19±8.59	10.21±6.11	171.85±15.19
DST+FA	13.72±8.07	9.5±5.27	151.28±48.06
All	9.65±7.69	3.43±4.49	162.54±4.38

- Additional findings
  - Bad distance updates hurt (25% increase in error)
  - Func. Approx. largest contributor
  - Combined better than in isolation

Results

Enhan.	Dist Err (cm)	Ang Err (deg)	Total Time (s)
None	19.75±12.0	17.75±11.48	161.25±3.43
HST	17.92±9.88	10.68±5.97	161.26±5.96
DST	25.07±13.73	9.14±5.46	196.18±12.18
FA	15.19±8.59	10.21±6.11	171.85±15.19
DST+FA	13.72±8.07	9.5±5.27	151.28±48.06
All	9.65±7.69	3.43±4.49	162.54±4.38

- Additional findings
  - Bad distance updates hurt (25% increase in error)
  - Func. Approx. largest contributor
  - Combined better than in isolation

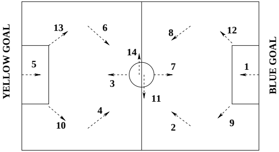
Results

Enhan.	Dist Err (cm)	Ang Err (deg)	Total Time (s)
None	19.75±12.0	17.75±11.48	161.25±3.43
HST	17.92±9.88	10.68±5.97	161.26±5.96
DST	25.07±13.73	9.14±5.46	196.18±12.18
FA	15.19±8.59	10.21±6.11	171.85±15.19
DST+FA	13.72±8.07	9.5±5.27	151.28±48.06
All	9.65±7.69	3.43±4.49	162.54±4.38

- Additional findings
  - Bad distance updates hurt (25% increase in error)
  - Func. Approx. largest contributor
  - Combined better than in isolation

Test for Stability

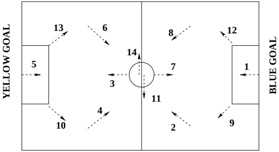
- Test ability to stay localized once at target
- Robot stationary at each of 14 points



1. Attempt to localize for 10 seconds
2. Record deviation of pose estimate for 20 seconds

Test for Stability

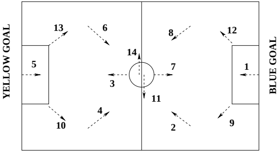
- Test ability to stay localized once at target
- Robot stationary at each of 14 points



1. Attempt to localize for 10 seconds
2. Record deviation of pose estimate for 20 seconds

Test for Stability

- Test ability to stay localized once at target
- Robot stationary at each of 14 points



1. Attempt to localize for 10 seconds
2. Record deviation of pose estimate for 20 seconds

Results

Enhan.	Dist Dev (cm)	Ang Dev (deg)
None	2.63	0.678
HST	1.97	0.345
DST	9.26	3.05
FA	1.46	0.338
DST+FA	4.07	1.30
All	1.32	0.332

- Significant improvement in stability
- Bad distance updates again perform worst
- Func. Approx. alone does as well as All
  - Distance information useful in reseed estimates

Results

Enhan.	Dist Dev (cm)	Ang Dev (deg)
None	2.63	0.678
HST	1.97	0.345
DST	9.26	3.05
FA	1.46	0.338
DST+FA	4.07	1.30
All	1.32	0.332

- Significant improvement in stability
- Bad distance updates again perform worst
- Func. Approx. alone does as well as All
  - Distance information useful in reseed estimates

Results

Enhan.	Dist Dev (cm)	Ang Dev (deg)
None	2.63	0.678
HST	1.97	0.345
DST	9.26	3.05
FA	1.46	0.338
DST+FA	4.07	1.30
All	1.32	0.332

- Significant improvement in stability
- Bad distance updates again perform worst
- Func. Approx. alone does as well as All
  - Distance information useful in reseed estimates

Evaluating Extended Motion Model

- Test impact of extended MM in isolation
- Evaluate ability to navigate to a point
  - Used “keeper” home position
  - Displace robot by hand a fixed distance
  - Allow to return to home position
  - Measure position and orientation error and time
- Average of ten runs

Results

Enhan.	Dist Err (cm)	Ang Err (deg)	Time (s)
None	12.89	15.0	17.21
Extended MM	7.50	5.5	18.14

- 40% reduction in position error
- 60% reduction in orientation error
- Only a small increase in time

Results

Enhan.	Dist Err (cm)	Ang Err (deg)	Time (s)
None	12.89	15.0	17.21
Extended MM	7.50	5.5	18.14

- 40% reduction in position error
- 60% reduction in orientation error
- Only a small increase in time

Results

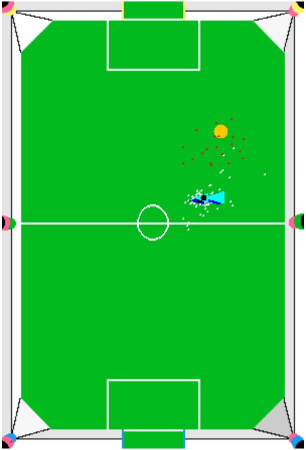
Enhan.	Dist Err (cm)	Ang Err (deg)	Time (s)
None	12.89	15.0	17.21
Extended MM	7.50	5.5	18.14

- 40% reduction in position error
- 60% reduction in orientation error
- Only a small increase in time

Outline

Simulator

- Abstract noisy observations and movements
- Always know ground truth
- Perturbations repeatable



Test for Recovery

- Robot follows figure 8 path
  - Perturbed once every 30 seconds
- Two types of interference
  - Collisions (stop for 5s)
  - Kidnappings (teleported 1.2m)
- Measure position and angle error on subset of conditions
  - Averaged over 2 hours (about 50 laps)

Test for Recovery

- Robot follows figure 8 path
  - Perturbed once every 30 seconds
- Two types of interference
  - Collisions (stop for 5s)
  - Kidnappings (teleported 1.2m)
- Measure position and angle error on subset of conditions
  - Averaged over 2 hours (about 50 laps)

Test for Recovery

- Robot follows figure 8 path
  - Perturbed once every 30 seconds
- Two types of interference
  - Collisions (stop for 5s)
  - Kidnappings (teleported 1.2m)
- Measure position and angle error on subset of conditions
  - Averaged over 2 hours (about 50 laps)

Results

Enhan.	Distance Error (cm)		
	Undisturbed	Colliding	Kidnapped
None	8.03	27.7	74.3
HST	17.6	25.3	27.3
DST+FA	7.83	16.2	31.5
All	8.67	14.4	13.5

- As expected, performance worse in presence of perturbations
- Enhancements mitigate performance degradation
  - Over 900% error increase for kidnappings without enhancements
  - Reduced to 56% increase with all enhancements
- Orientation error results similar

Results

Enhan.	Distance Error (cm)		
	Undisturbed	Colliding	Kidnapped
None	8.03	27.7	74.3
HST	17.6	25.3	27.3
DST+FA	7.83	16.2	31.5
All	8.67	14.4	13.5

- As expected, performance worse in presence of perturbations
- Enhancements mitigate performance degradation
  - Over 900% error increase for kidnappings without enhancements
  - Reduced to 56% increase with all enhancements
- Orientation error results similar

Results

Enhan.	Distance Error (cm)		
	Undisturbed	Colliding	Kidnapped
None	8.03	27.7	74.3
HST	17.6	25.3	27.3
DST+FA	7.83	16.2	31.5
All	8.67	14.4	13.5

- As expected, performance worse in presence of perturbations
- Enhancements mitigate performance degradation
  - Over 900% error increase for kidnappings without enhancements
  - Reduced to 56% increase with all enhancements
- Orientation error results similar

Results

Enhan.	Distance Error (cm)		
	Undisturbed	Colliding	Kidnapped
None	8.03	27.7	74.3
HST	17.6	25.3	27.3
DST+FA	7.83	16.2	31.5
All	8.67	14.4	13.5

- As expected, performance worse in presence of perturbations
- Enhancements mitigate performance degradation
  - Over 900% error increase for kidnappings without enhancements
  - Reduced to 56% increase with all enhancements
- Orientation error results similar

Results

Enhan.	Distance Error (cm)		
	Undisturbed	Colliding	Kidnapped
None	8.03	27.7	74.3
HST	17.6	25.3	27.3
DST+FA	7.83	16.2	31.5
All	8.67	14.4	13.5

- As expected, performance worse in presence of perturbations
- Enhancements mitigate performance degradation
  - Over 900% error increase for kidnappings without enhancements
  - Reduced to 56% increase with all enhancements
- Orientation error results similar

Summary

- Monte Carlo Localization works well in theory
- Practical implementation issues
  - Especially using vision-based legged robots
- Three Enhancements
  - Significant improvement over baseline
  - More dramatic for unmodeled movements
- Help others avoid potential pitfalls