SLAM: COMPARATIVE APPROACH

Khooshal Saurty

OUTLINE

- Introduction - What is SLAM?
- EKF SLAM
- FAST SLAM
- Comparison
- Cartographer
- Conclusion and References

INTRODUCTION - WHAT IS SLAM?

- Simultaneous Localization And Mapping
- Why do we need that?
    - Construct map of unknown environment and keep track of the agent's location in it
- Possible applications
    - Deep sea exploration
    - Mine Exploration
    - Search and Rescue
    - Space exploration

INTRODUCTION - WHAT IS SLAM?

- 2 tasks:
  - Mapping
  - Localization

SLAM ALGORITHMS

- EKF SLAM
- Fast SLAM
- Graph SLAM
- RatSLAM
- Several more at openslam.org

THE SLAM PROBLEM

- Given
  - Robot controls
    - $U_T = \{u_1, u_2, u_3, \ldots u_T\}$
  - Observations
    - $Z_T = \{z_1, z_2, z_3, \ldots z_T\}$
- Estimate
  - Map of the environment
    - m
  - Path of Robot
    - $X_T = \{x_0, x_1, x_2, \ldots x_T\}$

THE SLAM PROBLEM - LANDMARKS

- Essential part SLAM
- Distinct points/parts in environment
- for e.g: Walls, tables, chairs
- Assumption: Position of landmarks don’t change.

THE SLAM PROBLEM - SENSOR/ APPARATUS

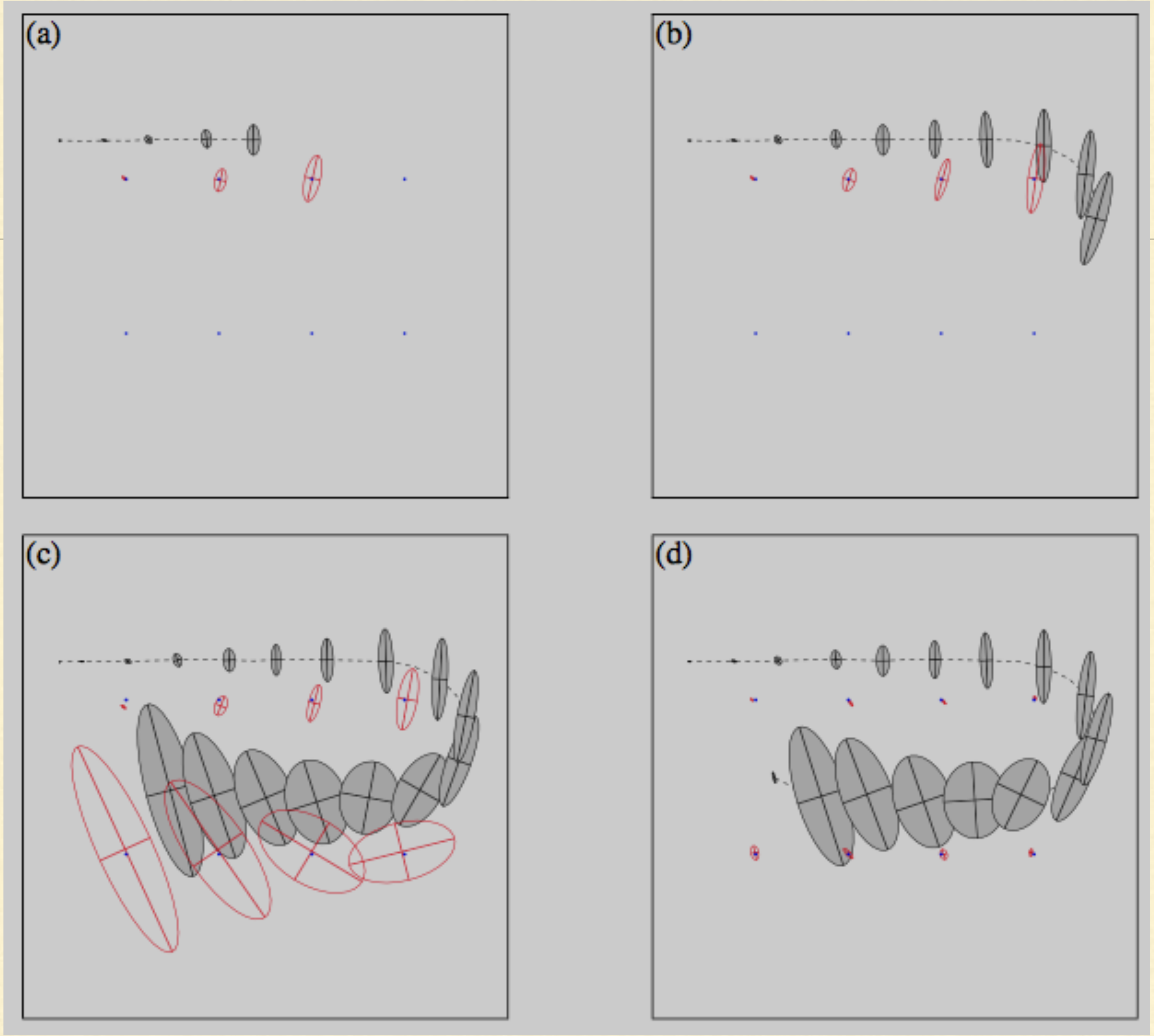
- Odometer
  - Location
- Distance Sensors
  - Sonar Sensor
  - Infrared Sensor
  - Laser range finder

EKF SLAM

- First variants of SLAM
- Based on Kalman-Filter
- Aim: Estimate the robot's position and locations of landmarks.
- State Representation - 3 Matrices
  - Position Vector - ((3+2N) x1) Matrix
  - Observation Vector - (2N x 1) Matrix
  - Covariance Matrix - (3+2N) dimensions

EKF SLAM - CYCLE

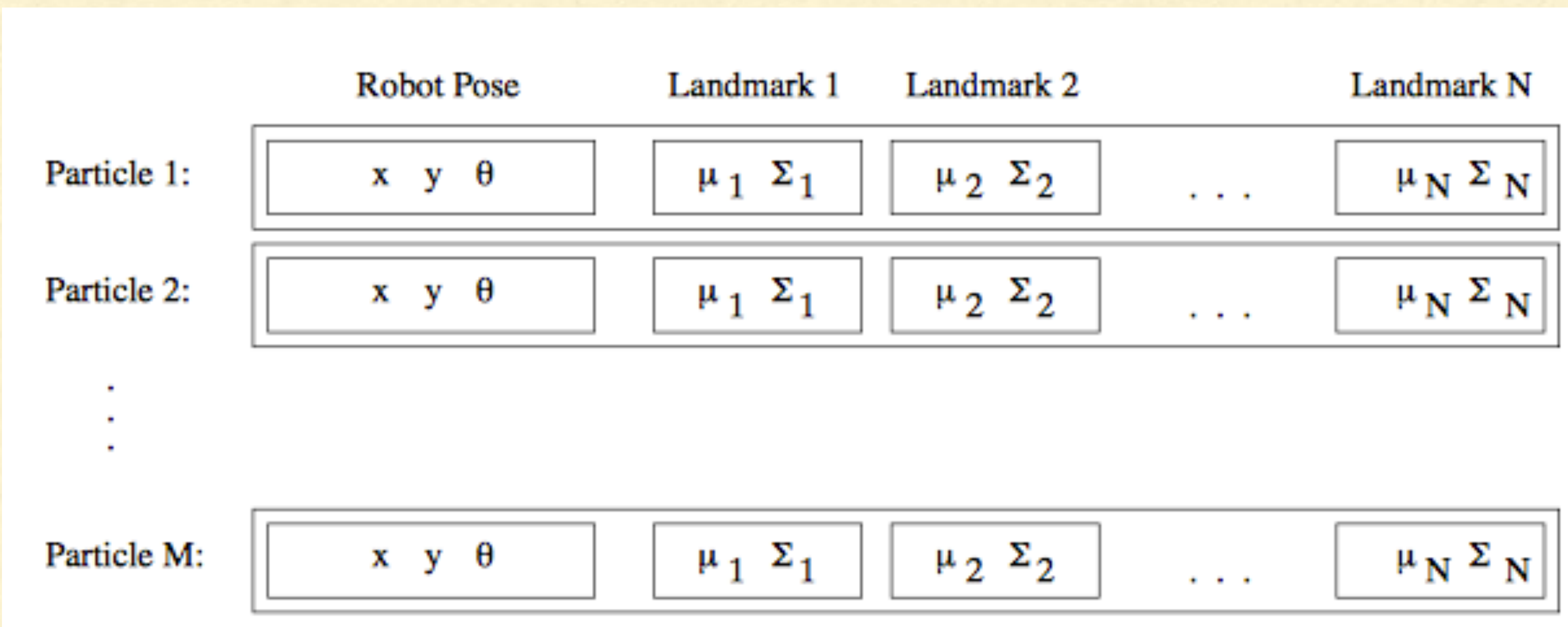
- State Prediction
- Predicted measurement (expected to observe)
- Take real measurement
- Data association
- Update



[1]

FAST SLAM

- Uses particle filter
- 1 particle -> 1 position
- Each landmark has its own EKF
- N Landmarks and M particles -> Mx(N +1) filters



[3]

FAST SLAM - CYCLE

- For each particle:
  - Sample new robot pose for each particle
    - add sample to temporary set of particles
  - Update observed landmark estimate
    - Updated values added to temporary particle set
    - each landmark is updated using the standard EKF update
- Resampling
  - draw from temporary set of particles to form new particle set

FAST SLAM

[3]

COMPARISON

- EKF SLAM
  - Covariance Matrix
  - Updated every step
  - Expensive operation
  - Complexity N^2
- FastSLAM
  - No State vector
  - Linear Complexity

COMPARISON

- EKF SLAM
  - Data Association
    - One for each landmark
- FastSLAM
  - Data Association
    - Each particle has own hypothesis to landmark
    - HOWEVER! bad sampling leads to loss of “precise” data

COMPARISON

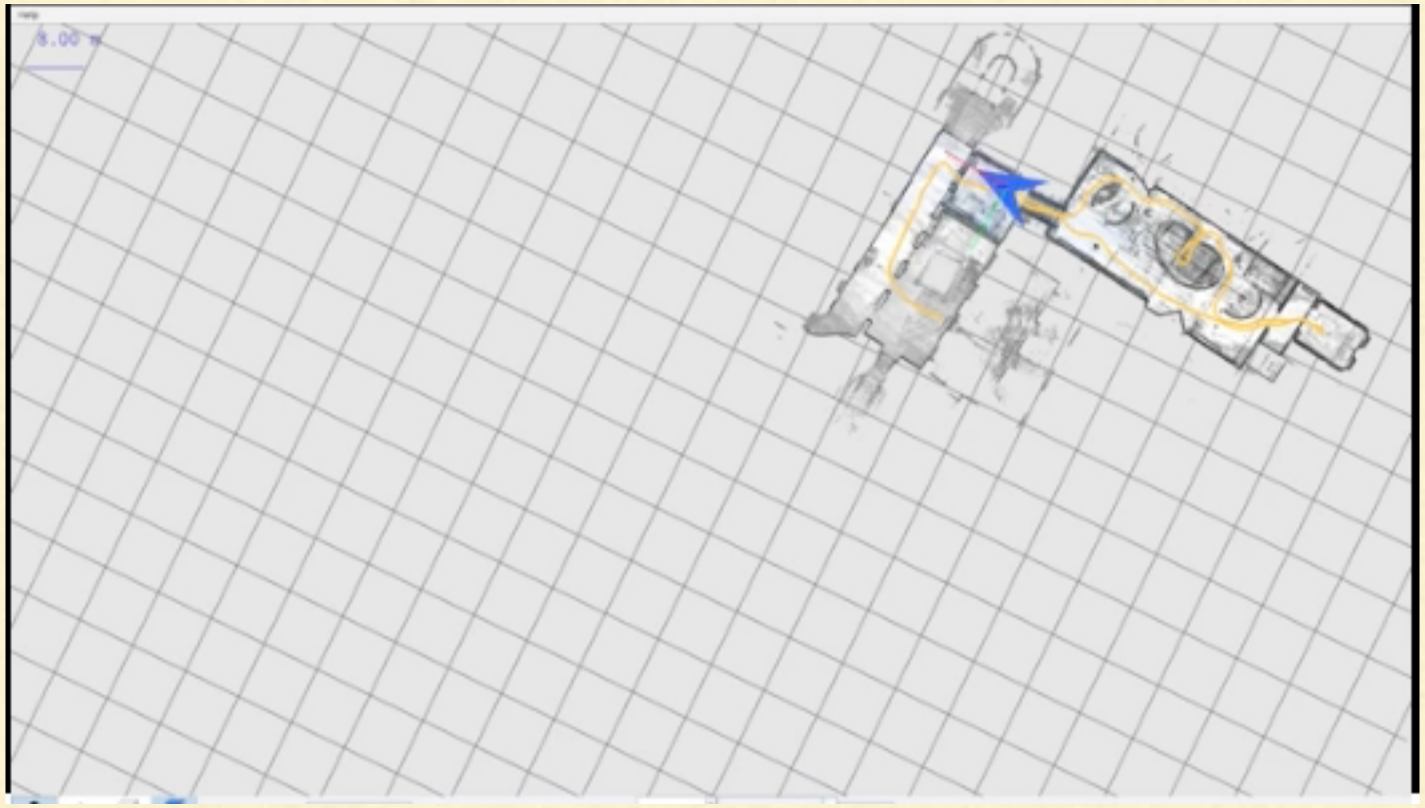
- EKF SLAM
  - Better for small areas
  - WHY? - Landmark correlations increase prediction accuracy

    The huge matrix does have a significant role!!

- FastSLAM
  - Better as we increase the number of particles
  - WHY? - More data to sample from

CARTOGRAPHER

- released in Oct 2016
- real time SLAM library



[4]

CONCLUSION

- Slam algorithms are approximate solutions
- Still need improvement
- Other factors affecting solution: quality of sensors used

REFERENCES

[1] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press, 2005.

[2] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. 2002

[3] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot "Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association." *Journal of Machine Learning Research* 4.3 (2004): 380-407.

[4] Cartographer - https://github.com/googlecartographer (2016)

[5] M. R. Naminski. ”An Analysis of Simultaneous Localization and Mapping (SLAM) Algorithms." (2013).