On the Representation and Estimation of Spatial Uncertainty

Abstract

Estimating relationship/error between coordinate frames that represent relative locations of objects (position and orientation)

Covariance = expected error

Types of errors:
- Positioning errors
- Measurement errors
- Tolerances in part dimensions?

Frames known only indirectly through spatial relationships with error

Examples:
- If a camera has an object in its field of view
- If a robot can go through a door

Results agree with independent Monte-carlo simulation

Estimate how accurate the relationship is

Estimate how much a sensor would improve it

They show for 3 DOF (x,y,theta) but can be generalized to 6 DOF

Introduction

Location of one object relative to another may only be known through a sequence of reference frames with uncertainty

Relational map: network of relationships

Global reference frame unnecessary

Set of local reference frames linked through a series of uncertain transformations is better

Reduction of uncertainty due to sensing can be mapped to any frame, no matter where sensing performed

As robot moves, uncertainty with respect to initial location grows

Assuming moves are discrete, quantitative estimate made

Use sensors to make relative measurements, which improves global locations of robot and objects

Assumptions:
- Errors small at first order approximation à simple fast matrix computations
- Sensor and location errors independent

Although example here is mobile robot, can also be used for other applications, such as manipulators — originally motivated for industrial manipulators

2. The Basic Estimation Operations

AT = approximate transformation
- Mean relation and covariance matrix
- Arises from relative motions and sensing
- $\text{AT A} =$ uncertain location (A) of an object wrt some reference frame
- represented by an arrow going from frame to location
- Ellipse = contour of constant probability in 2d for a multivariate gaussian distribution
  - Centered about nominal estimate
  - Bounds a high-confidence region
  - appendix A: covariance matrix $\rightarrow$ ellipse parameters
- Procedures here don’t depend on Gaussian distribution, but something must be picked when making calculations
- Compounding = combining a chain of AT’s recursively à greater uncertainty
- Merging = combining parallel AT’s à less uncertainty
- Fig 1
  - Robot starts at position w
  - Solid ellipse = uncertainty wrt last position ?
  - Dashed ellipse = uncertainty wrt w ?
  - Solid ellipses grow? But it says uncertainty wrt world frame grows?
- $\text{AT G}$ is calculated through compounding
  - $\text{AT S} =$ inverse $\text{AT G}$, given by sensor
- combine two AT’s using weighted averaging/kalman filter equations (merging)
- 3. Compounding
  - Brooks used max/min representation of error
    - worst case – large overestimation down the sequence
    - can worst case even be absolutely defined
  - Chatila and Laumond ('85) more similar, but used scalar error
- 3.1
  - refer to written notes
  - function approximated by 1st order taylor expansions about means of random vars
  - if no error in theta, then can remove these rows and cols, and system is linear, so exact, not approximation
  - recursive compounding...
  - look at written notes for inverse stuff
- 3.2 Assumptions for compounding
  - 1st order approx accurate
    - std's small, so std$^2$ very small ?
- errors for compounding AT's independent
  - off-diagonal submatrices = 0 (eq 4)
  - if errors dependent, a compound AT w nonzero submatrices can be given, and treat pair as a unit
  - errors have zero mean
  - systematic errors removed by calibration

- 4. Merging
  - G = L4 wrt W (recursive compounding)
  - S = W wrt L4
  - parallel transformations
  - use Kalman filter equations
  - 4.1 Formulas for compounding
    - compute G through recursive compounding,
    - reverse S, with eq's (5) and (6)
    - now G and S are pointing in same direction
    - merging combines transforms in pairs, recursively
    - Let
      - C1 and C2 be covariances of transformations to be merged
      - C3 be resulting covariance after merging
      - Xhat1, Xhat2, Xhat3 → estimated means
    - doesn't derive formulas → sites Kalman filter equations
      - look at underlined stuff
  - 5. Combined Estimation Procedures
    - explains how to combine larger maps
    - come back to the rules for writeup....
      - ? middle node should not be connected to any other node?
        - Does this mean besides the two on the chain of 3?
    - fig 3: Wheatstone bridge problem: inner loops
      - again, analogy to electrical circuit theory
      - can reduce from t to v, but not from w to u
      - approximate estimate by deleting certain edges
    - fig 4 – electric circuit theory suggestion for this problem
      - replace triangle (delta) of AT's (A, B, C) by a “Y” (X, Y, Z)
6. A Mobile Robot example

- Mobile robot uses sensors to build and update a world map
  - Known environment: localization
  - Unknown environment: mapping
- Initial position might be world frame
- As it moves, uncertainty grows, as shown with ellipses in Fig 1
  - Each relative move represented by an uncertain transformation
- Procedures above allow robot to estimate uncertainty, so it knows if it has enough accuracy for success
  - Maybe decides sensing is necessary before action
6.1 Robot sensing

- Mobile robot uses sensors to determine locations of objects in the environment
  - Accuracy of location determined by sensor resolution
- Sensed relationship between robot and object can also be represented as another AT in the network
  - If map is known, then location of object is known, and there is a known transformation between object and world (i.e., initial position node)
  - If map unknown, then sensing the object would add it to the network, or map, with some uncertainty ellipse
- Reduction of uncertainty through sensing can be done ahead of time
  - This would be a merging step
  - In a compounding step, eq 4,
    - \( C_3 \) comes from \( C_1, C_2, H, \) and \( K \)
    - \( H \) and \( K \) jacobians come from functions \( f, g, \) and \( h \)
    - Inputs to these functions are \( x, y, \) theta values
    - So here, the location of the sensed object would have to be known
    - And it must be sensed first in order to determine its location
  - In a merging step, eq's 8, 9, 10
    - \( C_3 \) come from \( C_1 \) and \( C_2 \)
      - \( K \) just comes from \( C_1 \) and \( C_2 \) as well
    - So it doesn't require \( H \) and \( K \)
    - But \( C_1 \) and \( C_2 \) come from delta \( x, \) ...
      - Delta \( x \) comes from \( (x – \text{mean of } x) \)
      - \( x \) is a variable, but \( \text{mean of } x \) is the location input
because it can be calculated ahead of time
- can decide if sensing will reduce uncertainty before performing the sensing
- alternative sensing strategies can be explored
- ? this means AT estimation can be used for offline programming
- but merging can't be done offline, so the whole network can't be done offline
- important to sense location of as many reference objects as possible while still in initial position (world coordinates)
- then these objects can help the robot once it starts moving to keep an accurate estimate of its location
- even if the robot doesn't sense these referenced objects once it starts moving, it might sense other objects that are connected in the network to the original objects
  - through accurately sensed relationships? (look at notes)

6.2 The Sensing Procedure
- 1. Determine whether sensing is possible?
- 2. Make actual observation and decide if reasonable (data association)
  - given:
    - uncertain prior location of feature
      - is this \( p(h \text{ given } x) \) ?
    - Error in sensor
  - get: \( p(z \text{ given } x) \)
    - probability of actual location
    - decide if this \( p \) is below a threshold
- 3. Merge
  - if sensing info is in diff coord system, then merge in sensor coords
  - map final AT back to cartesian (see appendix B)

6.3 Monte Carlo simulation
- robot location calculated many times
  - using Gaussian distributions for the errors in the given relations?
- Fig 5
  - point cloud with 90% confidence ellipse is robot position
    - contour ellipse assumes Gaussian distribution for result
    - ok, as long as angular error std is under 6 degrees
    - what are the three point clouds?
- 7. Discussion
- extension to 6 dof hard because of singularities with three angles
  - extreme sensitivity to singularity near pole?
- Limitation: only two moments, mean and covariance estimated
  - not all the probability information