**SLAM System**

1. **Overview**

Goal: estimate 6DOF state

2 components

|  |  |
| --- | --- |
| Navigation filter  fuses info from inertial measurement unit/ other sensors → consistent 3D solution | 2D SLAM system  provides position/ heading info within ground plane |

Navigation coordinate system:

Right-handed system, origin at starting point

3D state represented by





roll, pith, yaw Euler angles

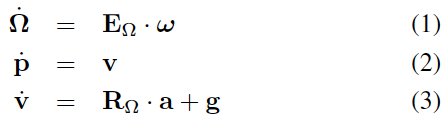
position

velocity

Inertial measurements

Input vector

Angular rates w and accelerations a

Nonlinear differential equation describing the motion of a body:

E maps angular rates to derivatives of Euler angles

R is direction cosine matrix (mapping vector in body frame to navigation frame)

Sensor noise → further sensor info: scan matcher, wheel odometry ect.

2D SLAM

Occupancy grid map

Scan → transformed into local stabilized coordinate frame (using attitude of LIDAR)

Scan → using platform orientation and joint values → point cloud of scan endpoints

**2. Map access**

Discrete occupancy grid map

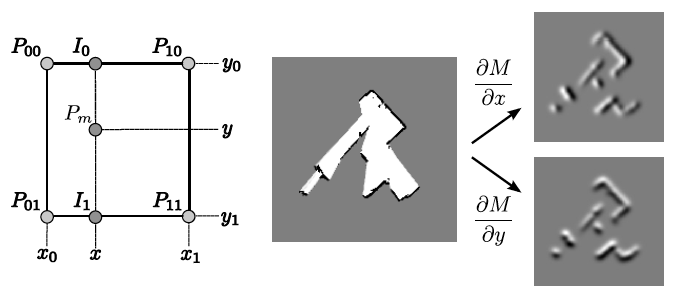
→ limited precision

→ no direct computation of interpolated values/derivatives

Therefore: estimating occupancy probabilities and derivatives of points within one cell in an interpolation scheme through bilinear filtering

Grid map cells values: samples of underlying continuous probability distribution

Given: continuous map coordinate → estimate occupancy value M(P) and gradient nabla M by using four closest integer coordinates → linear interpolation



**3. Scan Matching**

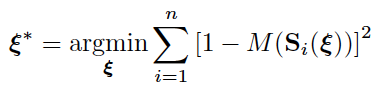
= process of aligning laser scans with each other or with existing map

Alignment of beam endpoints with map learnt so far (Gauss-Newton approach):

as scans get aligned with existing map, matching is implicitly performed with all preceding scans

Find: transformation that gives best alignment of laser scan with map or

Transformation that minimizes

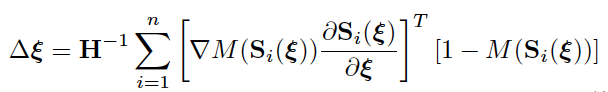


S(E): world coordinates of scan endpoints

E: pose of robot in world coordinates

M(S(E)): map values at coordinates given by S(E)

Starting estimate of E → estimate delta E optimizing error measure



Using map gradient nabla M(S(E)) and derivative of scan endpoints→ evaluation of Gaussian-Newton equation→ step delta E towards minimum

*What is the minimum? What is the error measure?*

*Ah: error = discrepancy bt scan coordinates and map coordiantes*

*Man will nicht E minimieren! Das ist die position des Robots und dieser wieder immer ein stückchen weiter bewegt solange bis die Koordinaten des scans am besten mit den Koordinaten der map übereinstimmen*

!convergence towards minimum cannot be guaranteed (nevertheless sufficient accuracy)

Match uncertainty

Covariance estimate: sampling pose estimates close to scan matching pose→ covariance

**4. Multi-Resolution Map Representation**

Risk: getting stuck at local minima

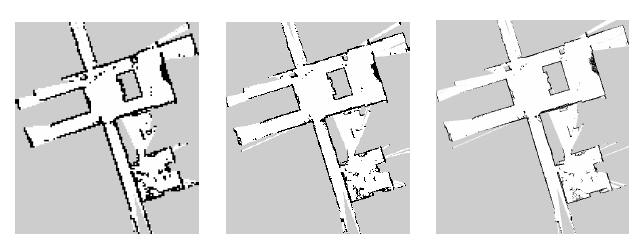
Mitigated by using multi-resolution map representation

Multiple occupancy grid maps (each coarser map having half the resolution of previous one)

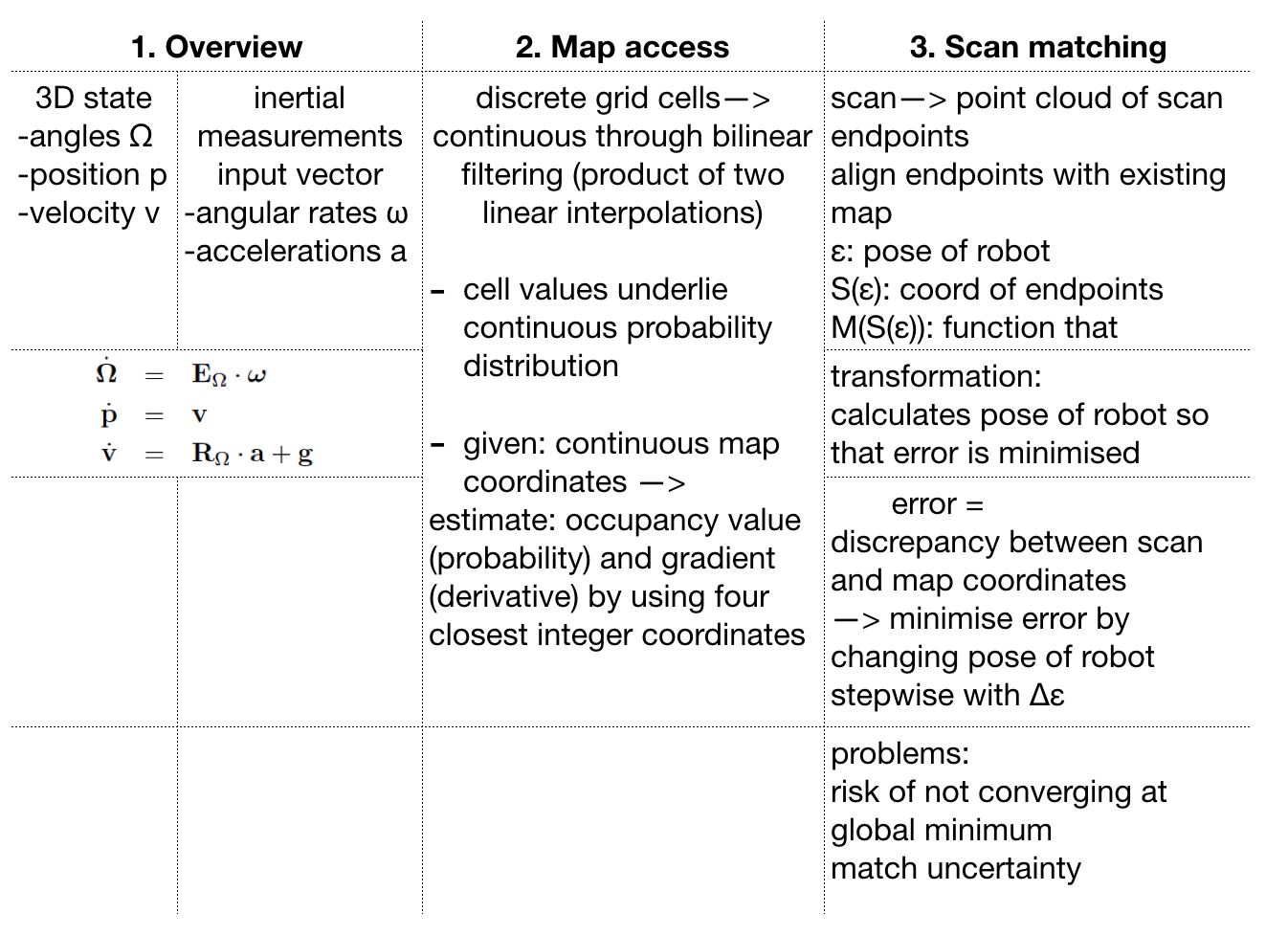
Multiple map levels generated from different maps (simultaneously updated)

Scan alignment process started at coarsest level:

estimated pose = start estimate for next level



⇒ summarized:



Search algorithms for occupancy grids

Issue: search strategy based on occupancy grid with cell values expressing the probability that target is located in cell ( → goal is to converge at cells with high values)

But in our case the cell values express the probability that cell is occupied/free

Alternatives:

* Raster scan
* Lawn mower
* Voronoi diagram