Summaries

Multi-Radar Inertial Odometry for 3D State Estimation using mmWave Imaging Radar LINK

Relevance: 4/5
Bullet points: Factor graph, RIO, multiple radar sensors, compensate for elevation discretization and sparsity
Summary: Good mathematical explanation of using radar & imu in a factor graph setting. Employing GTSAM with two radar sensors, one flipped by 90° to capture elevation precision.
Results:
Trick: Dual-Radar in GTSAM to compensate for elevation discretization and data sparsity

Multi-State Tightly Coupled EKF-Based Radar-Inertial Odometry with Persistent Landmarks LINK

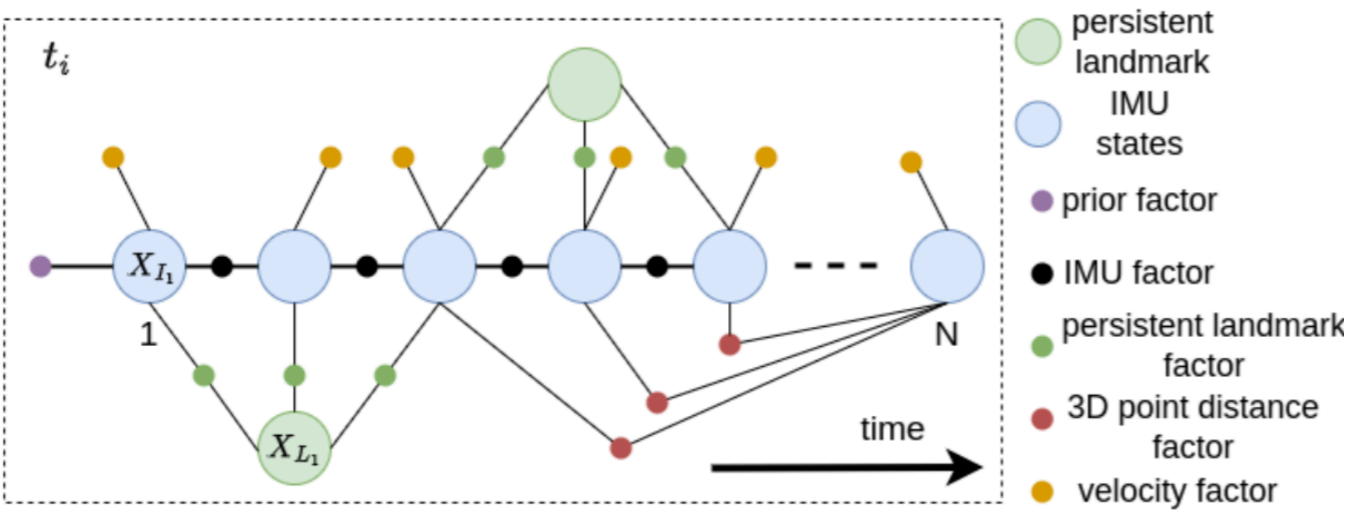
Relevance: 4/5
Bullet points: RIO with EKF, tracking past poses and radar features, creating trails of persistent landmarks using radar point-cloud matching
Summary: Using IMU as primary sensor and tracking/creating landmarks from radar scans. Trails are continuous detections of landmarks and remembering the pas position from which a landmark was detected. Once a landmark has not been detected anymore, the entire trail of this landmark becomes inactive and is deleted.
Results: Lower drift for extremely long real flights *(within one single room I believe)*
Comment: Is this really relevant? This assumes using landmarks (from FMCW).
Trick: Stochastic cloning? Multi-state update with measurement tails?

ORORA: Outlier-Robust Radar Odometry LINK

Relevance: 3/5
Bullet points: tackle radar noise such that indirect radar odometry gets better, GNC-based decoupling, ORORA = Outlier-Robust Radar
Summary: Goal is to find rotation (SO(2)) and translation ($\in R^2$) between two timesteps using radar data. For this, a novel pipeline (outlier rejection, translation estimation) is presented *(novel in the field of radar research)*.
Comment: Rotating radar, hence how relevant? Great explanation on noise in radar data (Anisotropic Uncertainty Modeling of Radar Data).
Results:
Trick:

Tightly-Coupled Factor Graph Formulation for Radar-Inertial Odometry LINK

Relevance: 5/5
Bullet points:
Summary: Compares RIO to EKF while using the same front-end *(front-end: radar point matching and noise reduction, e.g. RANSAC)*. Use of a sliding window for factor graph optimization. The factor graph uses for each radar scan a single IMU factor (integration in between). From the radar scan it uses velocity factors as well as persistent landmarks. Landmark retention follows [Multi-State EKF-Based RIO Odometry with Persistent Landmarks]().
Comment:
Results: Research often **understates the importance of front-end**.
Trick:



Milli-rio: Ego-motion estimation with low-cost millimetre-wave radar [LINK]()

Relevance: 5/5
Bullet points: 3D pose, RIO with RNN, point-association and then scan matching using NDT, indoor application, cm precision
Summary: The proposed method uses radar data by; performing point-association (e.g. creating pairs) between consecutive scans (range information). The pairs *(=Landmarks)* are then used to perform an optimization algorithm to extract pose change ([Normal Distribution Transform NDT]()). A bi-LSTM is used to learn the transition function *(learns motion model)* from one pose to another. Finally an **U**nscented **K**alman **F**ilter is used to track the 6 DoF pose.
Radar: Range information is matched between consecutive scans.
Comment: Extremely interesting how they **reduce rotational and translational error over time** - I have not quite understood why this happens yet. Some global alignment by 'adding current point-cloud to global point-cloud environment'.
Results: Experiments provide centimeter level accuracy. However, a **comparison to an alternate approach** is **missing**! (Apart from odometry & gyro method implemented on their

turtle-bot).

Trick: robust point association and then translation estimation (NDT). If LSTM is considered 'a trick' is unclear to me. Also, experiments were in 3D (x, y, θ).

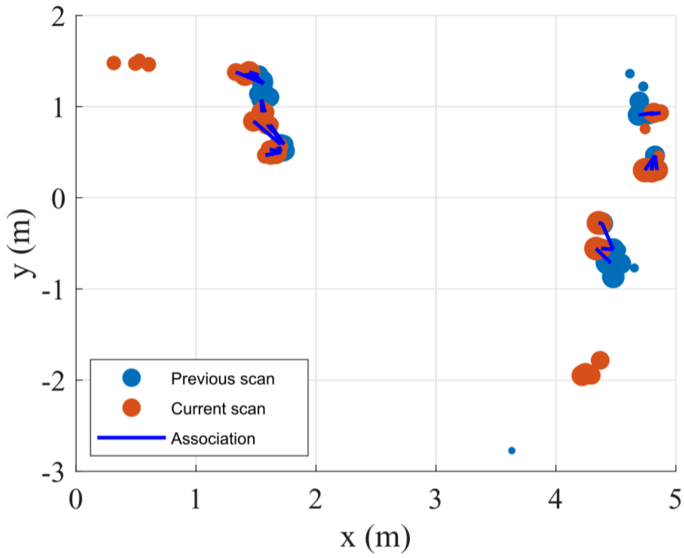


Fig. 4: **Example of point association for MMWave radar scans.** The marker sizes are proportional to the power intensity values. The proposed point association method finds the best matches, eliminating the low-intensity points and the ghost objects. It also penalizes the objects in the current scan that have no counter-part in the previous scan. The radar moves at approximately 0.50 m/s. Positions are relative to radar position.

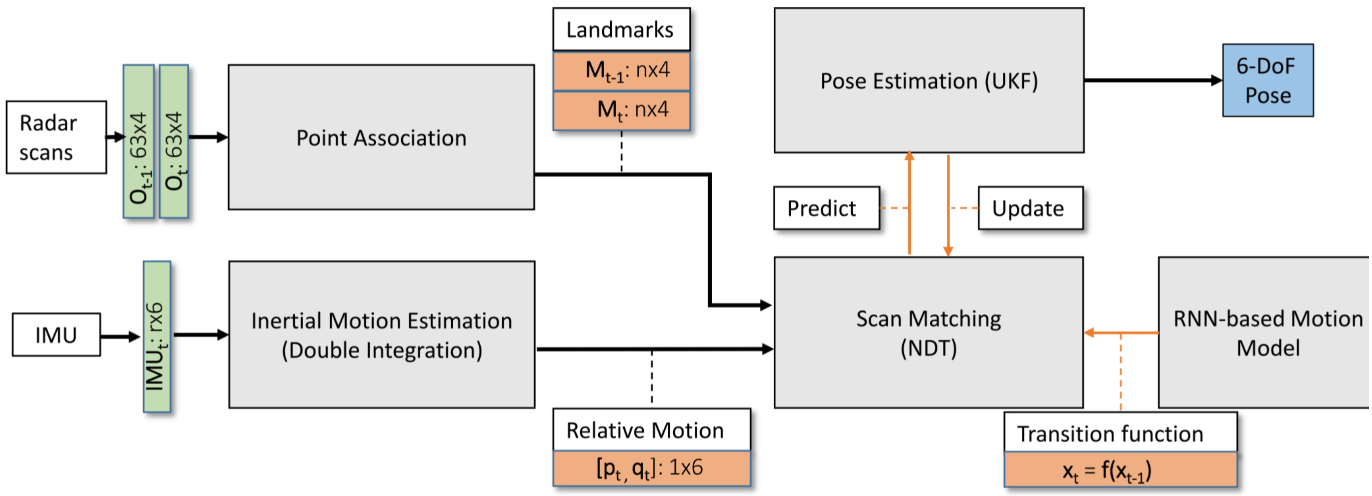


Fig. 3: **Ego-motion estimation workflow.** Raw MMWave radar point clouds are processed in the point association module t extract n landmarks, where n is the number of landmarks determined by the cost function. Scan matching module registers th landmarks using NDT scan matching algorithm, which uses a RNN-based transition model. In parallel, relative ego-motion i estimated from IMU readings using our inertial navigation system. Radar and IMU estimations are fused in the real-time pos estimation module using a UKF to regress the final 6 DoF pose values. Green, orange, gray and blue boxes represent inpu to the system, intermediate estimations, processing units, and outputs, respectively.

Radar Odometry Combining Probabilistic Estimation and Unsupervised Feature Learning LINK

Relevance: 4/5

Bullet points: Factor graph, unsupervised feature matching using EM, radar scan matching

Summary: The approach uses a factor graph and uses radar scan matching between poses. Framework for radar scan matching that outputs keypoints, keypoint detectors and weights for uncertainty. Method used is based on variational inference and an unsupervised learning method (Expectation Maximization of (negative) ELBO). Framework is 2D, but is already generalized to 3D $SE(3)$.
Radar: Rotating automotive radar **point cloud** (positional data only) where azimuth-range and raw-radar-power-return is used as input images.
Results: No IMU used!
Comment: Interesting approach for scan matching using unsupervised learning. EM explained here.
Trick: Unsupervised learning of scan-matching using Expectation Maximization. No need for manual data-labelling.

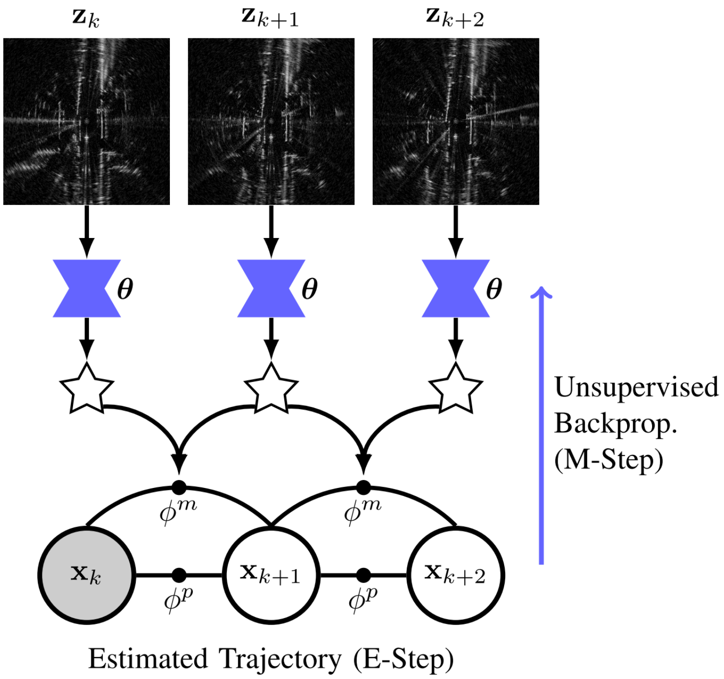


Fig. 2. This figure depicts the factor graph of our radar odometry pipeline. $\mathbf{x}_k$ and $\mathbf{z}_k$ are defined as the state of the vehicle and the radar scan at time t_k, respectively. The vehicle trajectory is estimated over a sliding window of w frames, where $w = 3$ in this figure. The deep network parameters are denoted by $\boldsymbol{\theta}$. The output of the network is a set of feature locations (x_i, y_i), their associated inverse covariance matrices, $\mathbf{W}_i$, and their learned descriptors, $\mathbf{d}_i$, which are together represented by stars in the diagram. These features are then matched between pairs of frames using a differentiable softmax matcher. The matched features are then used to form measurement factors, ϕ^m. A white-noise-on-acceleration motion prior is applied to create prior factors, ϕ^p.

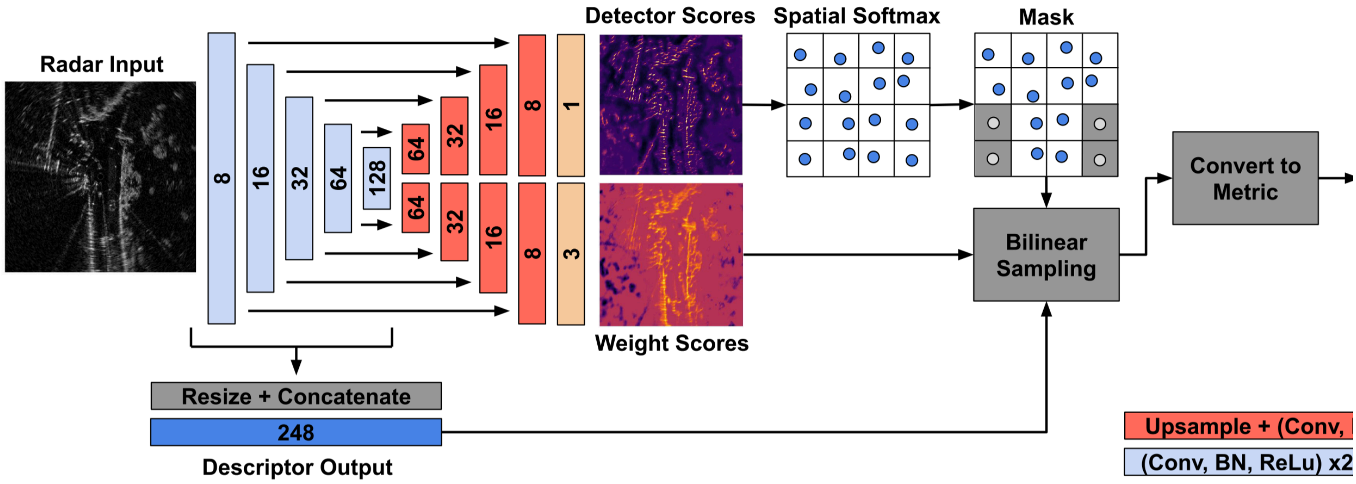


Fig. 3. We based our network on the architecture presented by Barnes and Posner [9]. The network outputs detector scores for keypo scores predicting keypoint uncertainty, and descriptors for matching. The weight scores are composed into 2×2 inverse covariance m corresponding image is the log-determinant). Descriptors are the concatenation of all encoder layer outputs after resizing via bilinear inter and decoder layers are a double application of a 3×3 convolution, batch normalization, and ReLU nonlinearity. The layer sizes vary by max-pooling (encoder) and bilinear upsampling (decoder). Note that the output size is the same as the input, and are visually smaller in An output 1×1 convolution is applied for the detector and weight scores. The detector score map is partitioned into uniform cells, where weighted summation of coordinates are applied to yield a keypoint for each cell. Corresponding weights and descriptors are obtained v

A robust baro-radar-inertial odometry m-estimator for multicopter navigation in cities and forests LINK

Relevance: 5/5

Bullet Points: Outlier rejection using m-estimation (opposed to RANSAC), barometer helps in vertical direction, factor graph optimization, Welsch loss function on radar residual

estimate
Summary: Factor graph method for odometry making use of barometric data alongside sparse 3D radar point clouds and IMU. Present extensive work on rejecting radar outliers *based on incoherent velocities* (Welsch loss).
Results:
Radar: Doppler shift is used in factor graph
Comment: Zero-velocity objects: I did not understand this...
Trick: Special loss function to reject radar outliers. Barometer for elevation precision.

Radarize: Enhancing Radar SLAM with Generalizable Doppler-Based Odometry [LINK]

Relevance: 4/5
Bullet points: a version of scan matching, radar only for odometry (+SLAM)
Summary: Propose a method for radar only SLAM/odometry. Use feature-rich raw radar data (TI + readout board). Range-azimuth heatmaps are used for feature extraction (for SLAM map). A doppler-azimuth heatmap gives translational motion. Special neural network is used for estimating rotation from radar scans at t and t-1.
Results: Exceed milli-ego.
Radar: Feature-rich raw radar data of TI SoC radar sensor (+ readout board).
Comment: Excellent use of machine learning to achieve different tasks. **Computationally too expensive for drones?**
Trick: Feature-rich raw radar data.

Radar-Inertial Ego-Velocity Estimation for Visually Degraded Environments [LINK]

Relevance: 5/5
Bullet points:
Summary:
Results:
Radar:
Comment:
Trick:

Table

Create a Note