pVoted: A Progressive On-Line Algorithm for Robust Real-Time Localization and Tracking in spite of Faulty Distance Information

Marcel Baunach

University of Würzburg, Department of Computer Engineering, D-97074 Würzburg
baunach@informatik.uni-wuerzburg.de

1 Summary

We present a progressive 3D localization algorithm for obtaining fairly precise position estimations in spite of highly imprecise and error-prone distance measurements from low cost hardware. At the same time, we achieve a high localization frequency, reduce energy for wireless data aggregation and require very little memory. We also provide a quality and trust classifier for each estimation. The low CPU load also facilitates its use on weak devices, e.g. within wireless sensor/actor networks (WSAN). For real-time tracking applications, a short position history further improves performance and precision.

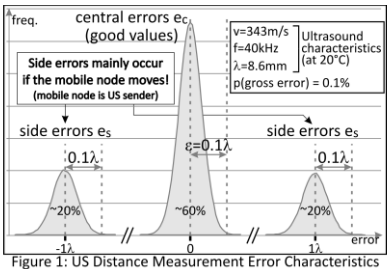

Figure 1: US Distance Measurement Error Characteristics

2 Motivation, Measurements and Goals

While implementing our real-world indoor localization and tracking system SNoWBat [1], we observed several problems concerning signal detection and distance measurements via TDoA of radio and ultrasound. Beside strong distance and angle dependencies, we encountered significant influences of the system's reactivity and timestamping reliability for related actions, events or IRQs. This was particularly obvious upon concurrent execution of several software components/tasks. By using a lightweight DSP procedure and the preemptive SmartOS operating system [2], we achieved an almost constellation independent and temperature compensated error characteristic despite of node mobility (Fig. 1). In addition to the shown *central errors* ($e_c \sim N(0)$, $e_c = \pm\varepsilon$) and *side errors* ($e_s \sim N(\pm\lambda)$), the probability for sporadic gross errors (up to 10cm) was ~0.1%. Nevertheless, the goal was to reliably achieve an absolute 3D position error $e \leq \delta = 2\varepsilon\cdot\sqrt{3} < \lambda$ and a high localization frequency f despite of only few (≈60%) "good" values.

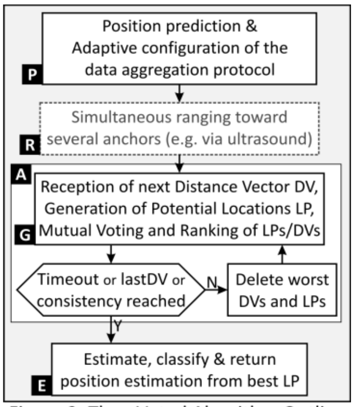

Figure 2: The pVoted Algorithm Outline

3 The pVoted Algorithm for Position Estimation

pVoted is a centralized five stage algorithm (Fig. 2) running on mobile nodes. Starting with a position ***P****rediction* p_{pred} from just the most trustworthy historic values, it computes the position dependent number m of required distance vectors (DVs toward static anchors) for the next estimation. Then it configures a self-organizing radio protocol [3] to efficiently receive this data by a collision free TDMA scheme with tightly packed slots. After invocation of any ***R****anging* process, the ***A****ggregation* stage successively collects the DVs d from the anchors and ***G****enerates* potential location points (LP) p from each consistent DV triplet, i.e. if the corresponding spheres (centers at the anchors, radii=distances) intersect in ≥1 point. For each new p_n, the *score* $s(d)$ of each involved DV d is incremented. Using a heuristic, p_n receives a *precision trust* $t_p(p_n)\in[0,1]$ inverse to its probability for an error due to inconvenient anchor constellations. Then, similar to many ranking algorithms, p_n is voted against any available former p_f: If $||p_n,p_f||=s \leq \delta$ these LPs "vote" for each other. Then, the *overall precision* $P(p_n)\text{+=}t_p(p_f)\cdot 1/s$ is updated and vice versa for p_f. The *consistencies* $\varsigma(p_n)$, $\varsigma(p_f)$ are incremented by $1/s$ reflecting the weighted number of mutual voters. In parallel, the *best location so far* p_B is updated (by $\max \varsigma(p)$, initially $p_B := p_{pred}$). Then, the worst LPs ($\min \varsigma(p)$) as well as the worst DVs ($\min s(d)$) are deleted to limit the memory usage and computation time for further iterations. The aggregation stage stops as soon as either an adjustable threshold for $\varsigma(p_B)$ or a timeout is reached or if the radio protocol signals the reception of the last data packet. Then, based on p_B the final ***E**stimation* E is computed by Weighted Centroid Localization over all LPs which voted for p_B. The individual weight equals the precision trust these points imposed on p_B. Among other metrics, a final cross classifier $X(E)$ considers p_B's reached precision and consistency compared to an expected value X_{exp} which depends on the number of received DVs and the measurement error characteristics. The algorithm computes $X(E)\in[0,½)$ for less and $X(E)\in[½,1]$ for more reliable estimations. Commonly, in case of sufficient DVs, $X(E) \sim 1/\text{absoluteError}$. While this information is an advantage for many applications, the next prediction also relies on it to avoid the use of weak estimations and to define its impact on the next localization.

4 Performance Analysis Overview

For a short overview, a mobile node was tracked along four traces within an industrial hall of 30x20x7m (Fig. 3a). The anchors at the ceiling granted a 99% chance for ≥4 good measurements (ε=±0.86mm) for each estimation. For some algorithms (see [4] for Eckert's with Kalman filtering), Fig. 3b/c show the RMSE and the number of estimations within the requested accuracy $e \leq \delta$=±2.98mm. For comparison, the Multilateration marked with "*" was fed with good values only. While producing better results than this, pVoted can reliably distinguish its own good and bad estimations at runtime (Fig. 3d). Since we run the *A*/*G*/*E* stages in parallel to the *P*/*R* stages of the next iteration, a localization frequency of $f \approx 2.9\text{Hz}$ was achieved within our setup (m_{max}=9) based on MSP430 CPUs at 8MHz. Note, that *R*+*A* already take 337ms (CPU independent) while *G*+*E*+*P* just fill CPU idle times.

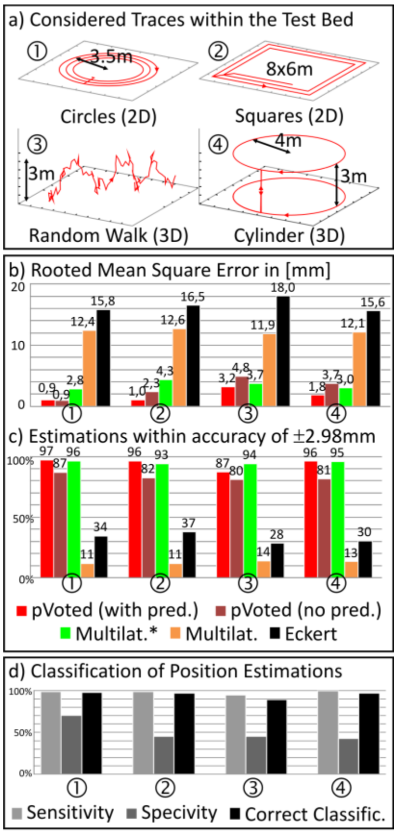

Figure 3: pVoted Performance Analysis

5 Conclusion and Outlook

Beside this very brief overview, further work considers the adaptive configuration of the radio protocol and the classification/estimation scheme. Both significantly reduce time and memory consumption (precise bounds can be shown) and even allow the detection of faulty sensors. We also compared more sophisticated algorithms and address anchor installation and self-calibration. Therefore, we currently research an extended pVoted scheme for SNoW Bat to observe its real-world performance when deploying setups of 50 anchors and concurrently operating mobile nodes.

6 References

[1] Baunach, Kolla, Mühlberger: "*SNoW Bat: A high precise WSN based location system*", Univ. of Würzburg, Tech. Rep. 2007
[2] Baunach: "*Dynamic Hinting: Real-Time Resource Management in Wireless Sensor/Actor Networks*", RTCSA 2009
[3] Baunach: "*Speed, Reliability and Energy Efficiency of HashSlot Comm. in WSN Based Localization Systems*", EWSN 2008
[4] Eckert, Dressler, German: "*An Indoor Localization Framework for Four-rotor Flying Robots Using Low-power Sensor Nodes*", Univ. of Erlangen, Tech. Rep. 2009