Authors' accepted paper - NEWCAS 2020 - 18th IEEE International NEWCAS Conference (Online, June 16-19, 2020)

Collaborative Localization and Tracking with Minimal Infrastructure

Yanjun Cao
Department of Computer and Software Engineering,
École Polytechnique de Montréal,
Email: yanjun.cao@polymtl.ca

David St-Onge
Department of Mechanical Engineering,
Ecole de technologie superieure,
Email: david.st-onge@etsmtl.ca

Giovanni Beltrame
Department of Computer and Software Engineering,
École Polytechnique de Montréal,
Email: giovanni.beltrame@polymtl.ca

***Abstract*—Localization and tracking are two very active areas of research for robotics, automation, and the Internet-of-Things. However, accurate tracking of a large number of devices in large areas with many rooms is very challenging: generally, one needs substantial infrastructure (infrared tracking systems, cameras, wireless antennas, etc.) for each room. This paper aims at covering a large number of devices distributed in many of small rooms, with minimal localization infrastructure. We use Ultra-wideband (UWB) technology in a device-2-device collaborative setting to develop a localization solution requiring a minimal number of fixed anchors. We present a strategy that autonomously shares the UWB network among devices and allows fast and accurate localization and tracking. We show results from an experimental campaign tracking visitors in the Chambord castle in France.**

I. INTRODUCTION

Indoor localization and tracking have the potential to unlock a plethora of new concepts and applications, both for autonomous system research and for public space enhancement. For instance, the main scenarios envisioned by our team are: 1) autonomous commercial cleaners, 2) audio-guides tracking, and 3) material handling in complex storage facility. While many paths have been already explored, it is very challenging to deploy a flexible and affordable absolute indoor positioning system for large numbers of devices. Technologies such as RFID, can cover a large area, but with limited accuracy. Commercial products try to fill the gap (GiPStech, AccuWare, Locbee, etc.) with solutions coupling inertial measurements, geomagnetic data, and available WiFi/Bluetooth beacons or routers. These are expected to reach at best a one-meter accuracy. In this paper we use Ultra-wideband (UWB) network-based measurements, which have much higher potential accuracy [1]. In addition, UWB transceivers are making their way into the smartphone market (e.g. in the latest iPhone). The general concept is that each device tracked has full or partial knowledge of its own position and this can be used to help the localization of other devices with a minimal number of fixed UWB beacons (a.k.a. anchors). We leverage principles from swarm robotics to allow devices to achieve consensus on the time sharing of the UWB network.

In literature, we can find many works using UWB sensors for localization. However, most are used for tracking one object with a fixed anchor setup [2], [3] in an open area that minimize occlusions. Cooperative localization was also proposed by Prorok et al. [4] with multiple robots. However, the authors still require the use of several fixed UWB anchors plus a customized sensor for relative position estimation. More recently, Guo et al. [5] propose a method for tracking three robots using UWB in combination with an odometer. It is comparable to our work, but uses a centralized system to manage the UWB network (therefore avoiding the complexity of a distributed solution). Their solution is not scalable over a large area without significant infrastructure investment.

To share the UWB network without the need for centralized supervision, we used two strategies: gradient-based synchronization [6], [7] and distributed dynamic Time-Division Multiple Access (TDMA). Flooding-time and gradient-based techniques both try to minimize the average clock offsets, but, gradient-based synchronization grants more importance to the closest nodes. To the best of our knowledge, none of these techniques were applied to the distributed usage of a UWB network. As for the second strategy, TDMA, it is a technique allowing multiple users to share the same transmission medium by splitting its usage into time slots. The Distributed Multichannel TDMA Slot Assignment Protocol (USAP) [8] and its extension USAP-MA [9] allow nodes to get conflict-free slot assignment in a decentralized way using local topology information in dynamic configurations. However, Dastango [10] showed that the convergence of USAP-MA is sensitive to the number of slots per frame, requiring careful tuning. In our implementation, we share only two values within the control packets; it is simpler than the Net Manager Operational Packet (NMOP) used in the original USAP [8], and decreases bandwidth usage.

In the following, Sec. II details our strategy to synchronize devices and share the UWB network among them; Sec. III summarizes the measurement types and presents our tracking system. We put everything together and discuss the performance of the system in Sec. IV by presenting full-scale field experiments conducted at the Chambord castle, in France.

II. SHARING THE NETWORK

First off, we postulate the following: 1) the localization system will be used daily in a large public space, i.e. devices can be alone or hundreds altogether, moving from room to room; 2) the building management cannot afford to set up a large fixed infrastructure.

This scenario leads to two characteristics influencing the design of our system: 1) a highly dynamic environment and 2) a local network usage. Furthermore, we target a system supporting a large number of moving devices with a minimal amount of anchors in large area. As the UWB network cannot be used by multiple devices simultaneously, we have to schedule network usage. Considering the number of active devices is dynamic and unknown, we cannot use a fixed schedule (which would also be inefficient). Therefore, our system first locally synchronizes clusters of nearby devices, and then automatically assigns TDMA communication slots to each device. The whole transmission mechanism runs in cycles, as shown in Fig. 1: first the synchronization, then the slots attribution by TDMA, and finally the slots execution.

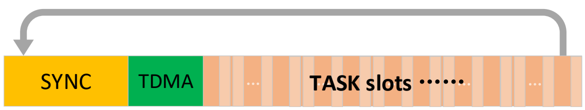

Fig. 1: System operation cycle, including synchronization (SYNC), slot allocation (TDMA) and slot execution (TASK slots).

A. Gradient-based Synchronization

From the scenario presented above, we extract another key characteristic: a group of devices in the same room likely represents a fully connected network, while devices in other rooms are connected through more hops. It would be rather difficult to keep all devices in the network graph synchronized. Besides, only local synchronization is required to communication with neighboring devices for localization.

Synchronization uses the hardware clock h_i of the device in order to maintain a logical clock, defined as:

$$L_i(t) = \int_{t_0}^{t} h_i(\alpha) l_i(\alpha) d\alpha + \theta_i(t_0) \quad (1)$$

where l_i is the relative rate compensating for the drift of the logical clock relative to the hardware clock h_i and the logical clock offset $\theta_i(t_0)$. If the hardware clock is considered accurate enough for the application, the compensation rate l can be set to 1. The objective is for each device to maintain its logical clock as close as possible to the ones of its neighbors. The messages sent in this phase of the transmission cycle, `syn msg`, contain a single piece of information: the logical clock value when the message was sent. Each device receives these messages from all direct neighbors N_i, and adjusts its own clock with the average offset, given by:

$$\theta_i(t_{k+1}) = \frac{\sum_{j \in N_i} (L_j(t_k) - L_i(t_k) + \delta)}{|N_i| + 1} \quad (2)$$

where $(L_j(t_k) - L_i(t_k) + \delta)$ is the clock difference between device i and its neighbor j with an estimated delay δ.

All the mobile devices adjust their clocks towards the average of their neighbors. When the difference between the clock and the neighbors' average is smaller than a threshold, or if the synchronization times out, the mobile devices are considered synchronized and keep their logical clock until the next cycle.

B. Distributed dynamic TDMA

Once all mobile devices are considered synchronized, a time-based schedule will be computed so all mobile devices have sequential access the UWB network for their measurements. Our distributed TDMA algorithm integrates two processes: scheduling and execution. In the scheduling process, all neighboring mobile devices negotiate which slots of the execution phase they can have, converging to a consensus on the final sequence. To reach consensus, they exchange `tdma msg`, serialized packets including three types of information: the sender id (usually provided by the UWB device controller), an action code, and a requested slot id. Every device maintains two TDMA-related tables for scheduling: a `sent list` and a `listen list`. From these tables, the device can derive a list of free slots IDs. When broadcasting a requested slot, the device selects one of these free slots. While in the listening state, the mobile devices update their lists with the new messages coming in.

Algorithm 1: TDMA Schedule to reach broadcast sequence consensus

```
input :  MsgReceived(actCode, slotId)
output:  SendList, ListenList.
SendList, ListenList ← −1;
blockList ← False;
while len(SendList) < Threshold do
    if time == myBroadcastTime then
        msgSend = msgQueue(−1);
        self.broadcast(msgSend);
    else
        msgRecv = self.listenChannel();
        (actCode, slotId) = msgRecv.decode();
        if actCode == −1 and
          blockList(slotId) == False then
            ListenList(slotId) == senderId
        else if actCode == −1 and
          blockList(slotId) == True then
            nextMsg = (sender.Id, slotId);
            msgQueue.append(nextMsg);
        else if actCode == self.Id then
            SendList(slotId).disable;
            nextMsg = (self.Id, slotId);
            msgQueue.append(nextMsg);
        else if actCode == ListenList(slotId) then
            ListenList(slotId).reset;
    end
    blockList.update(SendList, ListenList)
end
```

An example helps clarify the algorithm detailed in Alg. 1 and show how we avoid scheduling conflicts. The five small circles illustrated in Fig. 2 are five mobile devices. For simplicity, we only draw the communication ranges of mobile devices B and C. A, C, and D are within the communication range of B. Also, B, D, E are within the range of C.

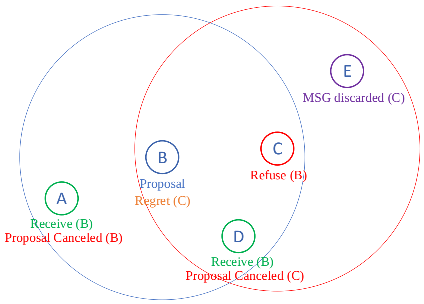

Fig. 2: TDMA schedule example.

While negotiating, a device can do any of three actions: slot proposal, proposal rejection, and proposal cancellation. The text under the circles represents the device's chosen actions with the letter in parenthesis representing the source of the incoming message. The first device B starts broadcast with a slot proposal and listeners (A, D) update their `listen list`, as in lines 10-11 of Alg. 1, unless this slot is already flagged as attributed in their list, like C in this case. Device C sees the conflict and then uses the source device ID as the action code to show the conflict, as in lines 12-14 of Alg. 1. When a device gets this type of message, it checks if it is the source of the conflict and in this case, it emits a proposal cancellation message, as in lines 15-18 of Alg. 1. Otherwise, if the conflicting ID fits the one attributed in the `listen list`, then it knows the device with this conflicting ID causes a conflict and the `listen list` entry is erased, as in lines 19-20 of Alg. 1. As shown in Fig. 2, both A and D canceled the proposal of B, but following different message flows. D got the proposal rejection directly from device C, but A is not a neighbor of C, so it got only the cancellation when broadcast by B. This logic prevents conflicts to emerge in the schedule, even with *hidden nodes*. It should be noted that for a device that is not a neighbor of device B, like E, the proposal rejection message is ignored since E never had a relative entry in the listen list.

III. LOCALIZATION AND TRACKING

With the above mechanisms, all UWB devices in range collaborate to share the band and minimize packet loss. Trilateration is thus possible when enough anchors are visible in a device's scheduled slot. However, most commercial systems require a minimum of $n+1$ anchors in sight to converge on a stable pose in a n-dimensional space. Public spaces as the ones target by this work (museum, schools, malls) are made of hundreds of rooms, that would each need to host 4 anchors for 3D positioning. To decrease their number as well as the footprint of the tracking infrastructure, we use ranging measurements for both anchors and other devices.

At each TDMA slot, the state estimation update follows the decision flow of Fig. 3, based on the number of available anchors. As the figure shows, whenever a device sees enough anchors to do trilateration, this option is preferred to get full pose estimation directly.

For both measurement types, the anchors are calibrated and their absolute position is provided to each tracked device.

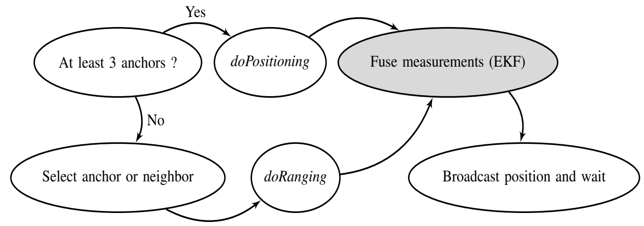

Fig. 3: The decision flow: range to a single node, and trilateration from 3 anchors or more inside a TDMA slot.

Trilateration computes the root square minimal solution given by the 4 anchor positions and their ranging measurements. The range between two devices is computed with two-way ranging (TWR), because its accuracy is independent of each device's clock.

While the raw trilateration can use common filtering to parse the outliers from reflection and other disturbances, the integration of range-only measurement to update the full pose is more complex. We implemented an Extended Kalman filter, following the common approach from [11], to compute the three degrees of freedom in translation and their rate of change, $\mathbf{x} = \begin{bmatrix} x & \dot{x} & y & \dot{y} & z & \dot{z} \end{bmatrix}^T$ based on a changing observation vector $\mathbf{z}$. We expect a smooth continuous movement and implement a discrete constant velocity model of the system. However, the observation model changes following the input: trilateration outputs direct observations of the system states (x, y, x) while range-only gives a non-linear relation between these states. For ranging, the measurement $d = \sqrt{(x-x_a)^2 + (y-y_a)^2 + (z-z_a)^2}$ is differentiated with respect to each independent variable and the residual is computed from the predicted state to estimate the distance measured, $\tilde{y} = d - \hat{d}$. The resulting EKF smooths the raw input from the UWB measurements and merge both information into a single state estimator.

IV. EXPERIMENTS

The system was tested in a realistic deployment scenario, out of the laboratory, in a large public space. The museums rooms covered are shown in the map of Fig. 4. To test the performance of our solution in various conditions, we set with four areas with different amount of anchors (4, 3, 2 or 0). Seven mobile devices (audio guides in our case) were moving around, following pre-defined goals in each zone, marked on the floor.

We start all mobile devices together in the green zone. They first synchronize, then coordinate. In all cases, as shown in Fig. 1, synchronization occurs again at the beginning of the next cycle, most likely with a different configuration of neighbors. The schedule phase had slots pre-attributed following ascending IDs and until reaching consensus on the scheduling, after which the mobile devices wait for their slots.

Fig. 5 shows the resulting scheduling tables of one run with the seven mobile tags, follwoing with their ID in the figure. The total number of slots available for scheduling is 25, selected empirically to ensure all seven mobile devices get at least a slot, demonstrated in the first schedule table `Start`

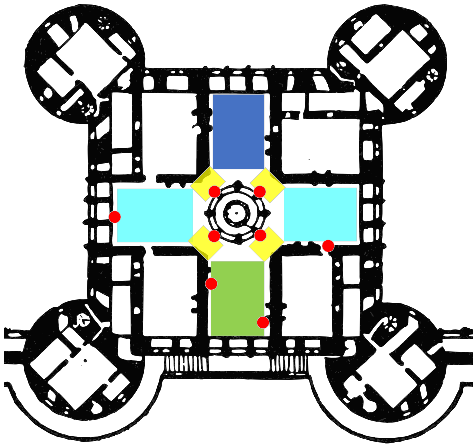

Fig. 4: Position of the 8 anchors (red dots) on the Chambord castle 2nd floor plan for the experiments. The zones color are: blue for 3 anchors, green for 4 anchors, purple for 2 anchors, and yellow for zero anchors visible (a cornice prevented measurement from right under the anchor).

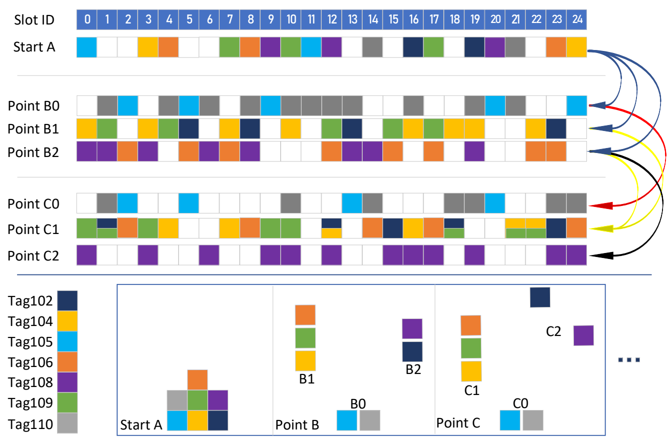

Fig. 5: TDMA schedule resulting allocation tables at three checkpoints. All mobile devices started together and split in groups toward a first checkpoint (Bx), then a second (Cx).

`A` of Fig. 5. It should be noted a large number of slots will not slow down the frequency of the localization phase, since each device will be attributed multiple slots until the schedule is nearly filled. This is also visible at all checkpoints (Bx, Cx) of Fig. 5, in which the mobile devices get more slots than the initial scheduling. An interesting behavior can be observed between the schedule of `B0` and `B1`: one member of the B1 group received messages from tag 105 and blocked most slots selected by this tag in group B1. This observation relates to the *hidden node* concept discussed in Sec. II. The arrows represent the TDMA transition process from device groups.

After all mobile devices get an approved schedule table, they start positioning themselves, following the decision tree shown in Fig.3. The result of one visitor's run is shown in Fig. 6, walking from the lower red diamond to the right one and finally to the top one.

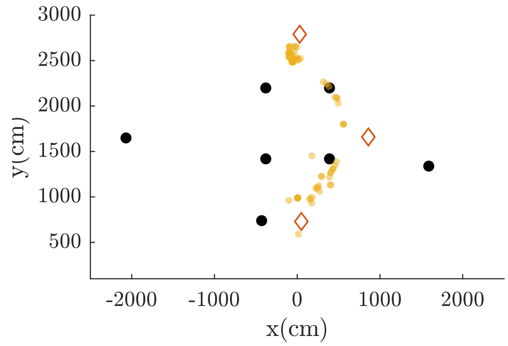

Fig. 6: Output positions from a visitor in the Castle (in yellow) with the anchors (in black) and the goals (in red).

V. Conclusions

With this relatively limited (in terms of number of devices and ground truth comparison) but large scale (in terms of area) deployment experiment, we confirmed the two contributions of this localization strategy: none of the mobile devices experienced collision while conducting range measurements or trilateration, and we obtained a scalable dynamic localization and tracking system.

References

[1] D. Zito and D. Morche, "UWB Radios — The maturity age?" in *2016 14th IEEE International New Circuits and Systems Conference (NEWCAS)*, Jun. 2016, pp. 1–4.

[2] J. Tiemann, F. Eckermann, and C. Wietfeld, "Atlas-an open-source tdoa-based ultra-wideband localization system," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2016, pp. 1–6.

[3] V. Mai, M. Kamel, M. Krebs, A. Schaffner, D. Meier, L. Paull, and R. Siegwart, "Local positioning system using uwb range measurements for an unmanned blimp," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2971–2978, 2018.

[4] A. Prorok and A. Martinoli, "Accurate indoor localization with ultra-wideband using spatial models and collaboration," *International Journal of Robotics Research*, vol. 33, no. 4, pp. 547–568, 2014.

[5] K. Guo, Z. Qiu, W. Meng, L. Xie, and R. Teo, "Ultra-wideband based cooperative relative localization algorithm and experiments for multiple unmanned aerial vehicles in gps denied environments," *International Journal of Micro Air Vehicles*, vol. 9, no. 3, pp. 169–186, 2017.

[6] R. Fan and N. Lynch, "Gradient clock synchronization," *Distributed Computing*, vol. 18, no. 4, pp. 255–266, 2006.

[7] P. Sommer and R. Wattenhofer, "Gradient Clock Synchronization in Wireless Sensor Networks," in *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, ser. IPSN '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 37–48.

[8] C. D. Young, "USAP: a unifying dynamic distributed multichannel TDMA slot assignment protocol," in *MILCOM '96, Conference Proceedings, IEEE Military Communications Conference, 1996*, vol. 1, Oct. 1996, pp. 235–239 vol.1.

[9] ——, "USAP multiple broadcast access: transmitter- and receiver-directed dynamic resource allocation for mobile, multihop, multichannel, wireless networking," in *MILCOM 2000 Proceedings. 21st Century Military Communications. Architectures and Technologies for Information Superiority (Cat. No.00CH37155)*, vol. 1, 2000, pp. 549–553 vol.1.

[10] S. Dastangoo, T. G. Macdonald, D. Reinharth, and C. Burns, "Performance analysis of distributed time division multiple access protocols in mobile ad hoc environments," in *MILCOM 2009 - 2009 IEEE Military Communications Conference*, Oct. 2009, pp. 1–7.

[11] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. USA: MIT Press, 2006.