

# Chapter 2

## Architectures for Target Localization and Tracking

### 2.1 Introduction

This chapter summarizes the main architectures used for target localization and tracking in ubiquitous computing systems. Existing approaches can be coarsely classified into peer-to-peer and hierarchical schemes. In peer-to-peer architectures each node exchanges sensing information with its neighbors in order to reach a consensus on the target status. In hierarchical architectures nodes organize following some hierarchy. Three main hierarchical schemes have been adopted: tree-based, cluster-based and hybrid schemes. Cluster-based schemes are the most widely employed and researched approaches. Clustering naturally solves scalability and efficiency since the measurement flow and the packet interchange are kept within the cluster, simplifying computations and transmissions.

This chapter also presents a general architecture for cluster-based tracking that is used as the main conductor of this book. This architecture comprises three main modules that deal with typical problems in cluster-based tracking. The first one is responsible for measurement integration and target estimation. The cluster evolves as the target moves and it is necessary to dynamically decide the nodes that are included in or excluded from the cluster. Nodes included in the cluster are activated, while nodes excluded from the cluster are deactivated, saving energy. The second module is responsible for deciding which nodes should be included or excluded from the cluster. Clustering-based tracking schemes rely on a cluster head (CH) that acts as the main scheduler of the cluster. The third module of the architecture is responsible for dynamically deciding which node acts as CH.

The experimentation of the schemes and modules shown in this book was performed in UBILOC (Ubiquitous Localization Testbed), a testbed specifically designed for multi-sensor target localization and tracking in ubiquitous computing systems. This chapter briefly presents its main characteristics and components.

This chapter is structured as follows. The main hierarchical architectures developed for target tracking are presented in Sect. 2.2. The main peer-to-peer architectures are described in Sect. 2.3. The general cluster-based architecture is summarized in Sect. 2.4. UBILOC is briefly described in Sect. 2.5.

## 2.2 Hierarchical Networks

In the simplest tracking scheme each node in the network is ready for tracking all the time. When a node gathers a measurement of the target, it transmits the measurement to the sink node or to base station, which estimates the target location using the received measurements. This centralized scheme has clear drawbacks. First, it is not efficient in terms of energy because all the nodes in the network are required to be always in a mode ready to track the target. Also, it involves heavy communicational burden at the surroundings of the sink node. Furthermore, robustness is compromised in case of channel congestion or failure of the sink node.

Different schemes have been developed to mitigate the drawbacks of this baseline method. This section briefly summarizes the main hierarchy-based architectures: tree-based, cluster-based and hybrid schemes.

### 2.2.1 *Tree-Based Architectures*

The nodes of the network are organized in a hierarchical tree that can be represented as a graph. The nodes that sense the target communicate with each other in order to select a specific node that acts as the root of the tree and collects the information. The root can dynamically change to adapt to the target motion. Although more robust than the aforementioned baseline approach, tree-based schemes still result in high energy consumptions and heavy communications.

One example is STUN (Scalable Tracking Using Networked Sensors) [1]. In STUN each link between two anchor nodes in the tree is assigned with a cost that is proportional to the Euclidean distance between the two nodes. The leaf nodes track the target and send the collected data to the root through intermediate nodes of the tree. The intermediate nodes register the detected targets and send updated information to the root when there is a change.

In DCTC (Dynamic Convoy Tree-Based Collaboration) [2] the tree is rooted at the anchor node that is the closest to the target. The target position is estimated by the location of the root node. In DAT (Deviation Avoidance Tree) [3] tracking is performed in two steps: update and query. Updates are initiated when the target moves to a new location. Update cost is reduced by the deviation avoidance tree algorithm and query cost is reduced by the query cost reduction algorithm in a second step.

DOT (Dynamic Object Tracking) [4] is a scheme that reports the tracking information of moving targets to a moving source. The first stage is to identify the neighbors.

In the second stage, called target discovery, the source sends requests to sensor nodes and the nodes that are close to the target reply. Finally, in the target tracking step, the source sends a query to the beacon node (the node keeping track of the information), which replies with the target next location. Then, the source moves towards the next beacon node.

### **2.2.2 Cluster-Based Schemes**

In these schemes anchor nodes sensing the same target organize into clusters. Each cluster has one node acting as cluster head (CH). Clustering techniques are energetically efficient: nodes not sensing the target are kept in sleep mode, saving energy, during most of the time. Clustering allows several targets to be tracked simultaneously, with a cluster for each target. Cluster-based tracking is particularly interesting in applications that require scalability. There are two main approaches depending on how clusters are created: static and dynamic clustering.

#### **2.2.2.1 Static Clustering**

Clusters are formed during the deployment of the network. Nodes are assumed static and for each cluster their members and coverage areas never change [5]. Static clustering is very simple and convenient in many cases but has several problems. It is not robust to failures in cluster head (CH) nodes. Also, different clusters cannot share information or collaborate in measurement integration and processing.

#### **2.2.2.2 Dynamic Clustering**

Dynamic clustering offers many interesting features for target localization and tracking. The formation of clusters can be triggered by external events, e.g. the detection of a target. If a node with sufficient battery and computational resources detects the target, it volunteers as cluster head. Other nodes near the CH are invited to become members of the cluster and to report their measurements to the CH.

One example is the so-called information-driven sensor querying technique IDSQ [6] in which the most suitable node to perform the sensing task is determined. This approach assumes that each node can locally estimate the cost of sensing, processing and communicating data with other nodes. It is energy efficient because only a few nodes are active simultaneously at any time.

DELTA [7] is an algorithm for tracking a person moving at constant speed that dynamically organizes a cluster and selects the cluster head, which will be responsible for monitoring the target and for collaborating with the rest of the sensor nodes.

RARE [8] is an energy efficient tracking protocol based on two algorithms, RARE-Area (Reduced Area Reporting) and RARE-Node (Reduction of Active Node

Redundancy). RARE-Area inhibits nodes that are far from the target reducing the number of nodes participating in tracking. RARE-Node identifies overlapping sensors in order to reduce the redundant information. The cluster is created dynamically using predictions computed during target tracking.

In DTSC [9] clusters are created by interchanging packets between the nodes that have detected the target. If a node has received packets from at least 4 nodes it declares itself as CH. DSTC can fail if the density is too low or if the cluster size is too large.

### 2.2.3 Hybrid Architectures

Hybrid schemes have characteristics from the two aforementioned approaches.

One example is DPT (Distributed Predictive Tracking) [10], which adopts a clustering approach and a prediction-based tracking technique. DPT is robust against prediction failures and target losses.

DCAT (Dynamic Clustering for Acoustic Tracking) [11] forms clusters using Voronoi diagrams. The CH asks its neighbors to join the cluster by broadcasting invitation packets. Nodes decide if they should reply to the CH after analyzing the distance to the target, which is estimated probabilistically. The CH determines the location using their replies and transmits the results to the sink.

In HPS (Hierarchical Prediction Strategy) [12] clusters are also created using Voronoi diagrams and the next location of the target is predicted using a technique based on Least Squares.

## 2.3 Peer-to-Peer Networks

In hierarchical networks many nodes are involved in sensing at the same time but only one acts as scheduler. The scheduler takes the greater part of the communication and computational burden resulting in inhomogeneous consumption of resources. In contrast, in peer-to-peer architectures all nodes have the same role and target tracking estimation is performed using consensus techniques in which each node interchanges its measurements with its neighbors [13].

Tracking in peer-to-peer networks is often based on average consensus algorithms. These algorithms perform successive refinements of local estimates maintained by individual nodes. The main objective of consensus filters is to estimate the global information contribution using only local and neighboring information [14]. Each iteration in the consensus filter has two main steps. In the first one, called communication step, each node interchanges information with its neighbors. In the second, called update step, each node uses the information gathered in the communication step in order to refine its previous estimate. Consensus filters are completely distributed and can be applied in large-scale sensor networks.

A scalable distributed target location and routing architecture for wide-area peer-to-peer applications is presented in [15]. Work [16] proposes a target tracking system based on the auto regressive moving average model in a distributed peer-to-peer signal processing framework. Sensor nodes act as peers that perform target detection, feature extraction, classification and tracking. It also includes a distributed peer-to-peer signal processing framework that considers the trade-off between tracking accuracy and energy consumption.

In general Recursive Bayesian Filters (RBFs), which are widely-applied in hierarchical-based tracking, are not applied to peer-to-peer networks due to their implementation complexity. However, a number of schemes that combine Kalman Filters and consensus filters have been developed, see e.g. [17, 18].

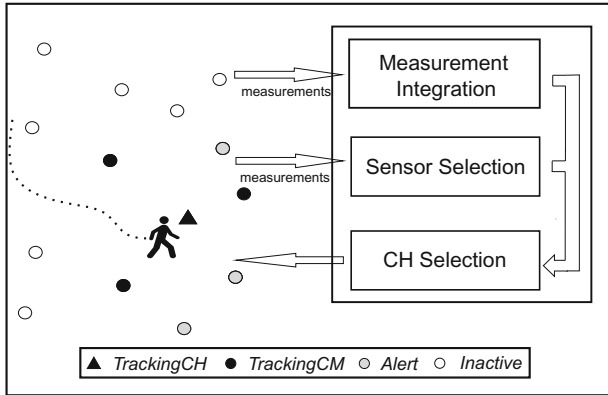
## 2.4 A General Cluster-Based Scheme for Localization and Tracking

This chapter presents a general architecture for cluster-based tracking that is used as the main conductor of this book.

Cluster-based schemes is by far the most widely employed and researched approach in target localization and tracking in ubiquitous computing systems. All the nodes that sense the same target organize into a cluster with a cluster head (CH) that schedules its operation. The tracking measurement flow and packet interchange are kept within the cluster, simplifying packet transmission. Each target being tracked has its own cluster and a number of targets can be tracked simultaneously. Of course, only the nodes that participate in target tracking are active: the rest are inactive—in low-energy mode—not consuming resources.

Cluster-based tracking requires methods to integrate the measurements gathered by the cluster nodes. Using different types of sensors for target tracking originates synergies of interest in many applications. The estimation and measurement fusion techniques adopted should allow flexible integration of measurements from sensors of different types. Measurement fusion is usually resource demanding, hence the method adopted should be efficient and prevent inhomogeneous consumption of computational, energy or bandwidth resources between the cluster nodes. Scalability, robustness to sensor failures and packet loss are other properties of interest in measurement integration methods.

Of course, the cluster should track the target as it moves. Then, methods should deal also with the task of deciding the inclusion/exclusion of nodes from the cluster. The objective is to keep within the cluster only the nodes that provide useful information for target tracking trying to reduce the active sensors to the minimum. Nodes are activated when they are invited to participate in the cluster and, they are deactivated when they are excluded from the cluster.



**Fig. 2.1** General cluster-based architecture for target tracking

The performance of cluster-based target tracking highly depends on which node acts as cluster head. Cluster-based tracking schemes should also deal with the dynamic selection of the most suitable cluster head.

### 2.4.1 Architecture

A general cluster-based architecture for target localization and tracking is shown in Fig. 2.1. The nodes can be at different modes depending on the level of involvement in tracking. At time  $t$  any node is at one of the following tracking modes:

- *TrackingCH*, the node acts as cluster head—represented by a triangle in Fig. 2.1;
- *TrackingCM*, represented as a black circle, the node participates in the cluster gathering measurement but not as head;
- *Alert*, represented as a gray circle, the node is not currently participating in the cluster but it is at single-hop distance from the head and could be included in the cluster at that time if necessary;
- *Inactive*, it is not involved in tracking and cannot be included in the cluster at that time.

Nodes in modes *trackingCH* and *trackingCM* participate in the cluster and altogether perform the target location and tracking estimation. Nodes in *alert* and *inactive* are not currently participating in tracking but those in *alert* could be invited to enter the cluster at that time  $t$ .

This architecture is composed of the following main modules:

- *Module I*: multi-sensor measurement integration. It integrates the measurements of the target gathered by the different cluster nodes. It should allow distributed

implementation. The main existing measurement integration techniques for target localization and tracking are presented in Chap. 3.

- *Module2*: sensor node activation/deactivation. Its objective is to keep active only the nodes that provide useful information for tracking trying to reduce to the minimum the number of active sensors. The main existing techniques are presented in Chap. 4.
- *Module3*: cluster head selection. The objective is to dynamically select which of the nodes within the cluster is the most suitable to perform the cluster head role. The main existing techniques are presented in Chap. 5.

Below, the main requirements of the architecture and its modules are summarized:

- **Computational distribution.** Each node within the cluster should actively participate in tracking. All nodes should gather measurements and perform computations that are used for measurement integration, node inclusion/exclusion and cluster head selection.
- **Efficiency.** The nodes should employ efficient techniques and exploit computation reuse resulting in low computational burden and energy consumption.
- **Scalability.** The consumption of resources from each cluster node should be constant or almost constant regardless of the cluster size.
- **Flexibility.** Each module of the architecture is responsible for addressing a main issue in the cluster-based scheme. Each module can be implemented with different techniques without affecting the rest of the modules. The architecture is flexible also in the sensors used for tracking.
- **Extensibility.** The architecture can be extended with other modules such as sensor calibration methods, which benefit from the synergies between measurements in order to supervise if sensors are calibrated and recalibrate them if necessary.

### 2.4.2 Schemes

The general architecture depicted in Fig. 2.1 can be adapted to different tracking schemes that use different types of sensors:

- *Scheme1*: Efficient tracking using only camera measurements,
- *Scheme2*: Efficient tracking using only RSSI measurements,
- *Scheme3*: Efficient tracking using camera and RSSI measurements.

The main difference between them is the sensors employed. In *Scheme1* each node is assumed equipped with a camera and tracking uses only camera measurements. Each camera node is assumed with sufficient computational capacity to execute simple image processing methods and measure the coordinates of the target center in its image plane.

*Scheme2* performs tracking employing only RSSI measurements. It assumes that the target is tagged with a mobile node and that each anchor node is capable of

**Fig. 2.2** Picture of the UBILOC testbed during an experiment



collecting RSSI measurements from the target node. *Module1* integrates RSSI measurements and *Module2* decides which nodes should gather and integrate RSSI measurement for localization and tracking.

*Scheme3* is an extension of *Scheme1* that also integrates RSSI measurements. *Scheme3* naturally employs the RSSI measurements that can be measured from the packets actually interchanged in *Scheme1*, without requiring any more transmissions. *Scheme3* assumes that the target is tagged. Nodes sensing the same target—either using the camera or with RSSI—organize autonomously into clusters. In *Scheme3*, *Module1* integrates camera and RSSI measurements and *Module2* decides which camera nodes are included/excluded in/from the cluster and with which measurements these camera nodes contribute to target tracking.

## 2.5 UBILOC: Ubiquitous Localization Testbed

The experimentation of the aforementioned schemes and modules was performed in UBILOC (Ubiquitous Localization Testbed), a testbed specifically designed for localization and tracking in ubiquitous computing systems, see Fig. 2.2. UBILOC was developed on top of the *CONET Integrated Testbed* [19],<sup>1</sup> a testbed designed for remote experiments with Cooperating Objects [20], and inherits from it its main characteristics.

UBILOC emulates a typical office smart environment scenario and includes sensors widely used for localization in smart environments: a network of cameras for camera-based localization, a Wireless Sensor Network for radio-based localization and a network of Time-of-Flight (ToF) sensors. These heterogeneous devices are integrated using a modular and flexible architecture. It also includes a set of mobile

<sup>1</sup><http://conet.us.es>.

**Fig. 2.3** Nodes of the WCN: each node is composed of a CMUcam3 module and a WSN node



robots that are used as targets to be localized and tracked in the experiments. UBILOC is set in a room of more than  $500\text{ m}^2$  ( $22\text{ m} \times 24\text{ m}$ ) in the basement of the School of Engineering of Seville.

Among others, UBILOC includes a Wireless Camera Network (WCN). Each node of the WCN is composed by a *CMUcam3* camera module [21] connected to a *TelosB* WSN node, see Fig. 2.3. *CMUcam3* camera modules are endowed with embedded programmable image processing capabilities. Each *CMUcam3* module captures RGB images of  $352 \times 288$  pixels and applies simple image processing methods. The results of the local image processing methods executed at each camera are transmitted to the *TelosB* node using a simple bidirectional protocol.

In UBILOC robots are used as targets to be localized and tracked. The advantages of using robots as targets instead of humans are: (a) they allow fully unattended experiments; (b) higher repeatability of experiments; and (c) robots include tools to determine their ground truth location, which is necessary to measure the errors provided by the localization techniques tested. A total of 5 Pioneer 3-AT robots are used. Each robot is equipped with one 2D laser range finder and one Microsoft Kinect and an IEEE 802.11 Wireless bridge. Each robot is capable of accurately computing its own location and orientation using the Adaptive Monte Carlo Localization algorithm (AMCL) [22]. These estimates are taken as the ground truth for indoor experiments. In outdoors the ground truth pose is obtained using GPS receivers and Inertial Measurement Unit (IMU).

UBILOC inherits its architecture based on Player [23] from the *CONET Integrated Testbed*. Player makes available user-transparent inter-module communication using standard interfaces. It is based on a modular client/server architecture. The Player Server interacts with the hardware elements and uses abstract interfaces to communicate with the Player Client, which provides access to all the system elements through device-independent APIs.

UBILOC provides infrastructure for experimentation support including high-level abstract interfaces to hardware and also the communication between processors. Users only have to program the modules they want to test. This architecture allows

a high range of localization experiments following centralized and decentralized schemes.

The experiments of *Scheme1* and *Scheme3* were performed with the WCN. The experiments of *Scheme2* were performed using the WSN.

### 2.5.1 Main Characteristics of the Hardware Components

*CMUcam3* uses an ARM7TDMI based fully programmable embedded computer vision sensor. The main processor is a Philips LPC2106 connected to an Omnivision CMOS camera sensor module. Custom C codes can be developed using GNU tool-chain along with a set of open source libraries. Executables can be flashed onto the board using the serial port with no external hardware required. Their main energy consumption characteristics are summarized in Table 2.1.

*TelosB* is a well-known ultra low-power WSN module. *TelosB* uses industry standards like USB and IEEE 802.15.4 to interoperate with other devices. It integrates humidity, temperature and light sensors; provides flexible interconnection with peripherals; and enables a wide range of applications. *TelosB* Revision B is a replacement for *Moteivs Revision A* design. The main operating conditions and energy consumptions of *TelosB* are summarized in Table 2.2.

**Table 2.1** Energy consumption of several components of the CMUcam3 module

| Power state | Active current (mA) | Idle current | Voltage (V) |
|-------------|---------------------|--------------|-------------|
| All active  | 130                 | 25           | 5           |
| CPU (60MHz) | 30                  | 10 $\mu$ A   | 1.8         |
| CMOS camera | 25                  | 10 $\mu$ A   | 5           |

**Table 2.2** Typical operating conditions and energy consumptions of *TelosB* nodes

|                                          | Min | Nominal | Max  | Unit    |
|------------------------------------------|-----|---------|------|---------|
| Supply voltage                           | 2.1 |         | 3.6  | V       |
| Current consumption: MCU on, Radio RX    |     | 21.8    | 23   | mA      |
| Current consumption: MCU on, Radio TX    |     | 19.5    | 21   | mA      |
| Current consumption: MCU on, Radio off   |     | 1800    | 2400 | $\mu$ A |
| Current consumption: MCU idle, Radio off |     | 54.5    | 1200 | $\mu$ A |
| Current consumption: MCU standby         |     | 5.1     | 21.0 | $\mu$ A |

## 2.6 Conclusions

This chapter presented the main architectures for target localization and tracking in ubiquitous computing systems. They can be coarsely classified into hierarchical-based and peer-to-peer based. Clustering-based architecture is the most widely used scheme for localization and tracking.

This chapter also presented a general cluster-based architecture that will be used as the main conductor of the book. This architecture comprises three basic modules that deal with three issues of cluster-based localization and tracking: (1) measurement integration and estimation, (2) dynamic inclusion/exclusion of nodes from the cluster and (3) dynamic selection of the cluster head. These three modules are presented respectively in Chaps. 3, 4 and 5.

The presented architecture is instantiated in three different schemes for target localization and tracking using: only camera measurements (*Scheme1*), only RSSI measurements (*Scheme2*) and integrating camera and RSSI measurements (*Scheme3*).

This section also briefly presented UBILOC, a testbed designed for localization in ubiquitous computing systems, where the experiments described in this book were performed.

## References

1. Kung HT, Vlah D (2003) Efficient location tracking using sensor networks. In: Proceedings of the IEEE wireless communications and networking (WCNC'2003), vol 3, pp 1954–1961
2. Zhang W, Cao G (2004) Dctc: dynamic convoy tree-based collaboration for target tracking in sensor networks. *IEEE Trans Wireless Commun* 3(5):1689–1701
3. Lin CY, Peng WC, Tseng YC (2006) Efficient in-network moving object tracking in wireless sensor networks. *IEEE Trans Mob Comput* 5(8):1044–1056
4. Tsai HW, Chu CP, Chen TS (2007) Mobile object tracking in wireless sensor networks. *Comput Commun* 30(8):1811–1825
5. Fayyaz M (2011) Classification of object tracking techniques in wireless sensor networks. *Wirel Sens Netw* 3:121
6. Zhao F, Shin J, Reich J (2002) Information-driven dynamic sensor collaboration. *IEEE Sig Process Mag* 19:61–72
7. Wälcchli M, Skoczylas P, Meer M, Braun T (2007) Distributed event localization and tracking with wireless sensors. In: Proceedings of the wired/wireless internet communications. Springer, pp 247–258
8. Olule E, Wang G, Guo M, Dong M (2007) Rare: an energy-efficient target tracking protocol for wireless sensor networks. In: Proceedings of the IEEE international conference on parallel processing workshops (ICPPW 2007), p 76
9. Phoha S, Jacobson N, Friedlander D, Brooks R (2003) Sensor network based localization and target tracking through hybridization in the operational domains of beamforming and dynamic space-time clustering. In: Proceedings of the IEEE global telecommunications conference (GLOBECOM'03), vol 5, pp 2952–2956
10. Yang H, Sikdar B (2003) A protocol for tracking mobile targets using sensor networks. In: Proceedings of the IEEE international workshop on sensor network protocols and applications, pp 71–81

11. Chen WP, Hou JC, Sha L (2004) Dynamic clustering for acoustic target tracking in wireless sensor networks. *IEEE Trans Mob Comput* 3(3):258–271
12. Wang Z, Li H, Shen X, Sun X, Wang Z (2008) Tracking and predicting moving targets in hierarchical sensor networks. In: *IEEE proceedings of the international conference on networking, sensing and control (ICNSC'2008)*, pp 1169–1173
13. Scherber DS, Papadopoulos HC (2005) Distributed computation of averages over ad hoc networks. *IEEE J Sel Areas Commun* 23(4):776–787
14. Xiao L, Boyd S (2004) Fast linear iterations for distributed averaging. *Syst Control Lett* 53(1):65–78
15. Rowstron A, Druschel P (2001) Pastry: scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *Lecture Notes in Computer Science*, vol 2218. Springer, Heidelberg, pp 329–350
16. Wang X, Wang S, Bi DW, Ma JJ (2007) Distributed peer-to-peer target tracking in wireless sensor networks. *Sensors* 7(6):1001–1027
17. Zhou Y, Li J (2009) Distributed sigma-point kalman filtering for sensor networks: dynamic consensus approach. In: *Proceedings of the IEEE international conference on systems, man and cybernetics (SMC'2009)*, pp 5178–5183
18. Olfati-Saber R (2005) Distributed kalman filter with embedded consensus filters. In: *Proceedings of the IEEE conference on decision and control and european control conference (CDC-ECC'05)*, pp 8179–8184
19. Jiménez-González A, Martínez-de Dios JR, Ollero A (2011a) An integrated testbed for cooperative perception with heterogeneous mobile and static sensors. *Sensors* 11(12):11,516–11,543
20. Marrón PJ, Karnouskos S, Minder D, Ollero A (2011) *The emerging domain of Cooperating Objects*. Springer
21. Rowe A (2012) Cmucam3 datasheet. <http://www.cmucam.org/>
22. Fox D, Burgard W, Dellaert F, Thrun S (1999) Monte carlo localization: efficient position estimation for mobile robots. *AAAI/IAAI* 1999:343–349
23. Gerkey B, Vaughan RT, Howard A (2003) The player/stage project: tools for multi-robot and distributed sensor systems. In: *Proceedings of the international conference on advanced robotics*, vol 1, pp 317–323

Cluster-based Localization and Tracking in Ubiquitous  
Computing Systems

Martínez-de Dios, J.R.; Bernabé-Clemente, A. de S.;

Torres González, A.; Ollero, A.

2017, XVI, 82 p. 32 illus., 6 illus. in color., Softcover

ISBN: 978-3-662-54759-5