Collaborative Synchronization Mechanism in Wireless Multimedia Sensor Networks

Sheng Feng¹ · Chengdong Wu¹ · Yunzhou Zhang¹

Published online: 27 April 2017

© The Author(s) 2017. This article is an open access publication

Abstract The synchronization problem of the wireless transmission in real-time visual surveillance applications was addressed. The distributed architecture of wireless multimedia sensor networks was adopted to obtain the key frames from multi-views synchronously, which has low computational cost and memory requirements. A good strategy for distributed computing was designed to assign tasks to specific smart devices. The main contribution of this architecture was a combination of wireless sensor networks and visual sensors. Image coordinates of detected objects are encapsulated in wireless data transmission. To track 3D localization in compute vision, the collaborative synchronization mechanism was presented to realize the synchronization of the key frames and asynchronization of the packet transmission from multi-views. The received data were divided into groups by the real-time clustering algorithm for 3D localization. Experimental results showed the proposed architecture can implement reliable and efficient indoor monitoring.

Keywords Indoor environment · Wireless multimedia sensor networks · XBee sensors network · Distributed architecture · Collaborative synchronization

1 Introduction

For the disaster relief system, a reliable architecture of wireless multimedia sensor networks (WMSN) is important to various applications, such as people tracking, robot navigation etc. Because of parallel distributed sensor networks with wireless communication and redundant deployment, WMSN is one of the most promising and potential applications of self-organization, rapid deployment and information accuracy [1].

✉ Sheng Feng
fengsheng_13@aliyun.com

¹ College of Information Science and Engineering, Northeastern University, Shenyang 110819, China

Owing to the limitation of wireless communication, the transmission of compressed images is troublesome and hard to receive real-time videos of multi-views synchronously [13]. In addition, privacy is a factor in a monitoring system, with no photographing. In this paper, smart visual sensors CMUcam5 (PIXY) are used to detect objects independently [17]. 3D localization in computer vision demands detected results from multi-views synchronously [2, 3, 8–10, 14–16, 18, 19]. Wireless sensor networks (WSN) are good at cluster-based communication and ad-hoc network [7, 12]. Therefore, the combination of visual sensors PIXY and WSN is reliable to realize 3D localization. Owing to the high rate of false alarms in foreground detection [11], the proposed architecture employs color algorithm to detect objects [4].

The 3D localization in computer vision demands the time stamps of key frames from multi-views are synchronized. To the best of my knowledge, the synchronization of key frames and asynchronization of the packet transmission from multi-views are not well studied in WMSN. First, the coarse synchronization at ms level makes sure the local time of all sensors is simultaneous [3]. Second, to minimize the size of the packet, it encapsulates the image coordinates of the detected objects in the key frames [4] instead of the compressed images [13]. Third, the coordinator can receive the packets transmitted from multi-views at different time stamps. Fourth, to realize the synchronization of key frames obtained from multi-views, it employs a collaborative synchronization mechanism to sample the key frames at the visual sensors synchronously, and cluster the real-time received data obtained from multi-views in PC.

The main contributions of this Letter can be summarized as follows: First, for the sake of real-time data aggregation from multi-views synchronously, a combination of distributed XBee sensors network and visual sensors PIXY in CMUcam5 project has been proposed. Second, the collaborative synchronization mechanism was employed to sample and cluster the detected results synchronously in real time. Some experiments have been carried out in the real world scenario. Experimental results showed that the proposed architecture realized real-time data aggregation from multi-views synchronously and reliably.

2 3D Localization

XBee sensors network is used to establish the mesh network in Fig. 1. Eight visual sensors in two rooms comprise PIXY, Arduino UNO and XBee S2 sensors to establish the distributed architecture of WSN. The coordinator is connected directly to PC to exchange data through the serial port. All visual sensors are directed to the center of the rooms. The overlapping area of visual sensors can be extended accordingly. About $1.5 \times 1.5 \text{ m}^2$ area of each room are overlapped by 4 visual sensors. The rest rooms are overlapped by less than 4 visual sensors.

2.1 Distributed Computing

Real-time localization of detected objects is a daunting task, since it requires 2D image coordinates from multi-views synchronously. The time intervals among the 2D coordinates obtained from multiple visual sensors need to be limited to a few milliseconds. It is not

Fig. 1 The mesh network of the proposed WMSN

possible in [13], because it transmits compressed images in wireless communication. One sensor needs to exclude others in order to complete video transmission in a period of time. Distributed computing is put forward and depicted in Fig. 2. PIXY takes the responsibility of object detection [4]. The size of API package, which includes 2D image coordinates of detected objects, can be reduced to only a few bytes. Therefore, Arduino UNO can complete the mission of encapsulating 2D location of detected objects into Application

Fig. 2 Distributed computing among smart devices and PC

Programming Interface (API) package instantaneously. XBee S2 is responsible for the analysis and transmission of API packages. PC software is responsible for unpacking API packages, implementing real-time 3D common perpendicular centroid (CPC) localization [6] and calibrating cameras.

2D image coordinate of the detected target in each view of monitoring cameras is the most important evidence to localize the target. The experimental platform of this Letter is shown in Fig. 1, the straight line between the 2D image coordinate and the target is the specific ray of one view through the optical center of the camera [6]. Between two rays of two views, the common perpendicular is the shortest distance. Furthermore, center point of the common perpendicular is the closest point between two rays. According to the theory of permutation and combination, four rays can be combined into 6 pairs of the rays. Six pairs of the rays can obtain 6 center points. The centroid of 6 center points is the closest point among 4 rays. Therefore, the centroid is closest to the theoretical intersection point of the rays. It is the estimated 3D location of the target, closest to the ground truth.

2.2 API Package Encapsulation

Depending on the configuration, XBee S2 can work in application transparent (AT) or API operating mode [5]. In AT mode, data is sent out through the serial interface when radio frequency (RF) data is received by the module. API mode is an alternative to AT mode. Data is transmitted in API frames. API specifies how commands, commands responses, and module status messages are sent and received from the module using serial interface.

The drawback of AT mode is that it is vulnerable to security attacks by sniffers like multi-channel packet analyser. Figure 7 depicts captured packets by multi-channel packet analyser. It clearly indicates the destination address, source address and mac payload, etc. However, API mode is a security mode for secure WSN routing protocols to ensure that the connectivity is maintained from security attacks.

To implement RF transmission in API frames, the algorithms for API mode are indispensable for smart devices and PC. A radio firmware of XBee Zigbee must be installed in XBee modules. And, the software of XCTU is utilized to configure XBee modules to work in API mode and establish the XBee sensors network.

An algorithm of API package encapsulation is developed for the smart device of Arduino UNO in Fig. 2. The main process of encapsulation is illustrated in Algorithm I. PIXY sends the block information of detected results to Arduino at 1 Mbits/s over SPI. It should be 50 frames/s. The most recent and largest objects are sent first. SPI is set to slave mode and rely on polling to receive updates. Due to RF transmission delay, all detected results cannot be transmitted to coordinator instantaneously. A threshold is employed to balance the speed of PIXY and the limitation of RF transmission. In order to meet the requirement of the algorithm in PC, the encapsulated data need to be formatted in advance. Checksum is also calculated to ensure data integrity and accuracy.

Algorithm I: API package encapsulation**Input:** detected results**Output:** API packages

```

1. for each monitoring interval do
2.   send block information with highest priority
3.   set slave mode to SPI
4.   if (block != null) and (counter == Threshold) then
5.     encapsulate ID of sensor
6.     for each block do
7.       format and encapsulate block information
8.     end for
9.     compute and fill in data and checksum bits
10.  end if
11. end for

```

Algorithm II: API package unpacking**Input:** API packages**Output:** semantic data

```

1. for each monitoring interval do
2.   add packet listener
3.   if response == ZNET_RX_RESPONSE then
4.     send synchronous AT command to request RSSI value
5.   end if
6.   if response == AT_RESPONSE then
7.     for each monitoring sensor do
8.       if monitoring address == remote address of response then
9.         update data of monitoring sensor
10.        break
11.      end if
12.    end for
13.  end if
14.  format received data to semantic data
15. end for

```

3 Collaborative Synchronization

The collaborative synchronization mechanism is performed on two levels: (1) coarse synchronization to synchronize sensors in the network and (2) collaborative synchronization to samples and clusters the detected results synchronously in real time.

At the coarse synchronization level, the coordinator sends the synchronization unicast packet to predefine visual sensors with a fixed period of 5 seconds. Compared with the broadcast packet in [3], the packet delivery acknowledgement can guarantee the communication quality. Because of the instability of wireless communication in the flood mechanism, visual sensors have to restart automatically sometimes. Periodic synchronization by the coordinator can make sure time stamps of all visual sensors are synchronized.

At the collaborative synchronization level, visual sensors are responsible for the synchronization of sampling time for object detection and asynchronization of packet transmission by wireless communication. PC is responsible for real-time clustering the received data into groups.

Since PIXY and Arduino are independent devices, it is a random local time when Arduino receives detected results from PIXY. If Arduino wants to detect messages delivery from PIXY in every time interval α , it requires a tolerance time 2σ . Furthermore, in order to keep only one message delivery from PIXY for each sampling, the time interval between θ and $\theta - 1$, successful detection messages delivery from PIXY, must be greater than 2σ . The time stamp of sampling by Arduino for detected results from PIXY is described as follows:

$$T_{sa}(\theta) = T_{sy} \quad \text{if } T_{sy} - T_{sa}(\theta - 1) > 2\sigma \cap (T_{sa} \% \alpha < \sigma \cup |\alpha - T_{sy} \% \alpha| < \sigma) \quad (1)$$

where T_{sy} is synchronized time at the coarse synchronization level; θ is a successful detection of message delivery from PIXY; $\sigma = 5$ is the tolerance time; $\alpha = 500$ is the time interval for sampling by Arduino, the threshold α is a trade-off between the size of the transmit packet and capability of wireless communication; $|\cdot|$ is an absolute value sign.

Because of the efforts above, all visual sensors are synchronized. In the flood mechanism, visual sensors transmit packets to the coordinator synchronously. It causes the packet loss at the coordinator. Therefore, each visual sensor is allowed to transmit a packet in every 1 s. It means one packet content contains detected results of $1000/\alpha$ key frames in the time interval α . Therefore, key frames of each view in every 1 s form a video stream, which can guarantee the real time in a short time interval α . A unique time slot is assigned to each visual sensor in a predefined order. Each visual sensor is allowed to transmit a packet at its own time slot. The time stamp of transmission for l th visual sensor is defined as follows:

$$\begin{aligned} T_t(l) &= T_{sa}(\theta) + r(\omega) + l \times \beta \\ &\text{if } T_{sa}(\theta) \% (2 \times \alpha) \cup (2 \times \alpha) - T_{sa}(\theta) \% (2 \times \alpha) < \sigma \end{aligned} \quad (2)$$

where $r(\omega)$ is a random value between 0 and 30, the upper limit of $r(\omega)$ is 30; $\beta = 50$ is the span for each time slot; l from 1 to 4 is the identity of visual sensors to label each sensor in a room. Therefore, the gap between two time slot is $\beta - \omega = 20$. In this order, the time stamp of received packets at the coordinator can be sorted in an array for all visual sensors; $\%$ is modulo operation.

At the PC side, the received data need to be grouped by clustering algorithm in real time. In order to handle the fresh received data on time, the length of the segment for data clustering is defined as $g = 20$. When the fresh received data are cumulative to $s = 10$, a new segment is ready for clustering. The real-time data of a new segment is defined as follows:

$$\begin{aligned} D_{ri}(i) &= D(i + m) \quad \text{if } i \in [1, g], \quad e(t) - e(t - 1) = s \\ m &= e(t) - g \end{aligned} \quad (3)$$

where D is the received data by the coordinator; $m + 1$ is the first index of D in the t th segment; $e(t)$ is the last index of D in the t th segment.

When a new segment is ready for clustering, BFSN clustering algorithm is employed to cluster the real-time data into different groups, based on the time stamp of sampling by Arduino [8]. The clustering result is achieved by the following formula:

$$A = f_{BFSN}(f_{SimiR}(T_s(D_{rt})\%(10 \times \alpha)), \gamma, \lambda) \quad (4)$$

where f_{BFSN} is BFSN clustering algorithm; f_{SimiR} is used to create a normalized similarity matrix; T_s is the time stamp of sampling in the real-time data. The synchronization period of the coordinator is 5 s, which is $10 \times \alpha$. As long as the time stamp of samplings in real-time data is a multiple of $10 \times \alpha$, it means they are synchronized accordingly, in case that visual sensors fail to receive synchronization messages from the coordinator. The parameter which will derive the normalized similarity matrix is $T_s(D_{rt})\%(10 \times \alpha)$. It means that when the newly received data are cumulative to $s = 10$, a new segment is ready for clustering, the length of the segment for data clustering is $g = 20$ in (3). The clustering data is T_s , which is the time stamp of sampling. In order to filter out the unsynchronized data, T_s needs to use the modulus operator to find out the synchronized data in the synchronization period of the coordinator, 5 s.

$\gamma = 0.98$ is a neighborhood threshold, if $f_{SimiR}(a, b) > \gamma$, a and b are neighbors. $\gamma = 0.98$ is the classification threshold, if the number of elements in class B is num. There is a new element c need to be classified. If more than $(\gamma \times 100)$ percent of elements num in class B are the neighbors of c , c is classified to class B .

However, the clustering result by BFSN algorithm is not following the rules of spatial and temporal correlation. In the practical scenarios, the rule of spatial correlation is that data received from different rooms need to be classified into different groups. The rule of temporal correlation is the data received at different periods of time that need to be classified into different groups, and the data received from the same visual sensor cannot appear at the same group. The real-time clustering algorithm based on spatial and temporal correlation is illustrated in Algorithm III:

Algorithm III: Real-time clustering algorithm

Input: D **Output:** G_τ

```

1.  $G_{rt}(1 + m) = id$ ;     $id = 1$     %% Initialization
2. for  $i = 2$  to  $g$  do    %% Iterate through the array  $D_{rt}$ 
3.     for  $j = 1$  to  $sz(A)$  do %% Iterate through the vector  $A$ 
4.         if  $f(A(j)) == i$  then %% Find the class identity in BFSN algorithm
5.              $Gi_A(i) = j$ 
6.         end if
7.     end for
8.     if  $Gi_A(i) \neq Gi_A(i - 1)$  then %% If neighbors have different class identities
9.          $id = id + 1$  %% The group identity will increase by itself
10.     end if
11.      $G_{rt}(i + m) = id$  %% Assign the group identity based on  $id$ 
12. end for
13. for  $i = 1$  to  $g$  do %% Iterate through the array  $D_{rt}$ 
14.     if  $vs(i) \in Ri_\tau$  then %% If the identity of  $i$ th visual sensor is in  $Ri_\tau$ 
15.          $G_\tau(ct_\tau) = [G_{rt}(i + m) \quad D(i + m)]$  %% Cluster the data for the  $\tau$ th room
16.          $ct_\tau = ct_\tau + 1$ 
17.     end if
18. end for

```

where $Gi_A(i)$ is the class identity of i th real-time data defined by BFSN clustering algorithm A ; $f()$ returns the indices that satisfy the conditional expression; $sz()$ returns the size of a matrix; $G_{rt}(i + m) = id$ is the group identity of the i th real-time data in the data of D ; vs is the identity of the visual sensor in real-time data of D_{rt} ; Ri_τ is a predefined vector of vs for the τ th room; G_τ is the final clustered data with the group identity and the τ th room for the i th real-time data; ct_τ is the last index of G_τ , ct_τ is increasing according to the real-time clustering algorithm and $ct_\tau = 1$ is the initial value.

According to the rule of temporal correlation, if the class identity of the i th and $(i - 1)$ th real-time data are different, or the identities of visual sensors for the id th group do not contain the identity of visual sensor for the i th real-time data, the group identity id increase 1. Otherwise, id remains still. Based on the rule of spatial correlation, if the identity of visual sensor for the i th real-time data belongs to Ri_τ of the τ th room, the group should be divided accordingly. Finally, each clustered group contains one or multiple unique views of a specific room.

The mean value of time differences is the key indicator for the performance evaluation of synchronization. To track 3D localization in computer vision, the mean value of time differences among the time stamp of sampling in each group should be as small as possible, which is defined as follows:

$$t^\mu = \frac{\sum_{i=1}^{n>1} \sum_{j=i+1}^{n>1} \sum_{k=1}^{\delta} \sqrt{(T_s(G_\tau(G_i(i)), k) - T_s(G_\tau(G_i(j)), k))^2}}{\delta \times C_n^2} \quad (5)$$

if $G_i = f(G_\tau == id)$

where n is the number of received data in a group, if $n = 1$, the mean value is 0; δ is the number of key frames in a received data, which is determined by the parameter α ; G_i is the indices of received data that belongs to the id th group.

4 Experimental Results

The proposed WMSN is compared with [13]. For real-time 3D localization, image size of 320×200 with the resolution of 640×400 is adopted. Since PIXY requires only 20 ms to fulfil object detection, the proposed architecture is 2 times faster than that of [13] in terms of time consumption. The results on processing time and transmission data are summarized in Table 1. It describes the fundamental performance of the architecture. Since the location of detected targets are the only content transmitted and 2 times faster than that of [13], the time stamps of sampling for data received from PIXY by Arduino can be more accurate than that of [13]. The first two rows of Table 1 depict the quantitative improvement in the hardware. The last three rows of the table present the qualitative analysis in the software.

Table 1 The comparison of processing time and transmission data

System	Reference [13]	Proposed
Number of clock cycles	2,457,600	1,536,000
Time (ms)	49.15	20
Transmission data type	Image	Location
Views in any moment	One view	Multi-views
Data in any second	Video stream	Key frames

To minimize the packet size, 2D image coordinates of the detected objects are encapsulated in the packet instead of the compressed images. The synchronous clustered data from multi-views are aggregated in PC instead of one view in any moment. Further, the key frames in 1 s from multi-views can realize 3D localization and tracking instead of the video stream in 1 s from one view.

In order to evaluate the real-time performance of proposed architecture, the demonstration system has deployed as shown in Fig. 1. The coordinator with a PC is located in the room 1. Two objects are moving in two rooms at the same time. Therefore, eight visual sensors can transmit detected results to the coordinator in the meantime.

As described in [3], the time differences at the coarse synchronization level are limited to a few milliseconds. At the collaborative synchronization level, the time differences among each group in the data received from visual sensors by the coordinator are shown in Fig. 3. Figure 3a, b show the mean value of time differences among each group. Figure 3c presents the cumulative distribution of t^μ , which is greater than 0.

Comparing the performance of the collaborative synchronization in the two rooms, results can be concluded as follows:

First, accuracy of the mean value: more than 90% of t^μ are less than 12 ms in both rooms. The mean values for all t^μ in all clustered groups are around 10 ms. It is acceptable for 3D localization, since any objects like pedestrians or mobile robot can move a short distance in such a period. For example, the average speed of an adult is about 1.1–1.5 m/s.

Second, interference caused by distance: the maximum value of t^μ in room 2 is dramatically increased to 100 ms. It is due to the distance between visual sensors and coordinator are increasing. Compared with room 1, the uncertainties of sensors network are also increasing in the meantime. The interference in the network causes the packet loss and the transmission delay, which increase the cost for collaborative synchronization.

Figure 4 indicates the interference caused by the distance in details for $t^\mu = 102.5$ in room 2. The group identity is $G_n = 43$. The time stamp of sampling at α and 2α are listed accordingly. The identity of visual sensor $vs = 23$ means the 3th visual sensor in the room 2. Raw data is the time stamp of sampling T_s in the real-time data. Cooked data is the remainder of $T_s \% (10 \times \alpha)$ in (4). It depicts that T_s is dramatically changed from 4794 to 482 for $vs = 23$ caused by the coarse synchronization level. However, t^μ between 4794 and 4994 for $vs = 23$ and 24 is 200. The reason is that the cost for collaborative synchronization is increasing along with the distance between the coordinator and visual sensors. The solution can be achieved by two steps. First, the location of the coordinator is justified to the most important room in the monitoring area, in order to optimize synchronization results in other conditions unchanged. Second, a threshold 4σ is introduced to eliminate any unsuccessful clustered group, the rule is $t^\mu > 4\sigma$.

The possible steady states are (1) a fully synchronized network; (2) a combination of synchronized clusters without synchronized nodes; and (3) not synchronized nodes (a node has no synchronized partner). Figure 2a, b show the steady states. $t^\mu > 0$ and $t^\mu \leq 4\sigma$ means the first state that the time stamps of key frames from multi-views are synchronized and well clustered into the groups. $t^\mu = 0$ means the second state that there is only one node in the group, and the node is a cluster. Because of the self-organization in WMSN, any node stands a chance to become a cluster. $t^\mu > 4\sigma$ means the third state that the clustered data in the group are not synchronized.

Figure 5 depicts the steady synchronization states over network size. In the first clustered column, there are only 4 visual sensors to communicate with the coordinator. If the visual sensors are deployed in the room 1 R1, the distance between the visual sensors and

Fig. 3 Measurement in two rooms for collaborative synchronization. **a** t^μ in room 1, **b** t^μ in room 2, **c** the cumulative distribution of t^μ

Fig. 4 Interference caused by distance in room 2

Fig. 5 Histogram of the steady synchronization states after coarse synchronization over network size. *Left bars [3]. Middle bars room 1. Right bars room 2*

coordinator is very short. The nodes are fully synchronized with only one exception. Because the distance is increasing in the room 2 R2, the percentage of the fully synchronous nodes are decreasing. Since the network size is increased to 10 visual sensors, which are placed in two rooms separately, the communication quality is decreasing obviously. The percentage of the fully synchronized nodes in both rooms is less than the situation of 4 visual sensors. However, all the situations are much better than the coarse synchronization method [3].

Figure 6 shows the synchronization results over network size in two rooms. It indicates the distance between the visual sensors and coordinator has obvious influence on the synchronization results. The mean values of t^μ in room 1 are always better than those in room 2. The network size also affects the synchronization results. The larger network size has the greater mean value of t^μ . However, the network size of 10 visual sensors in the

Fig. 6 Mean value of t^μ over network size in two rooms

room 2 can limit the mean value of t^μ to around 10 ms, which is acceptable to the most application of WMSN.

On account of security measurement employed by the proposed architecture, it is very hard to analyze the practical performance of wireless communication. WMSNs in this experiment have to be set in AT mode so that VinnoTech Multi-channel Packet Analyser can be utilized to capture packets for analysis. Figure 7 describes the packet list of AT mode. As shown in the first 5 indexes, AT packets from two source addresses, 0x3D6A and 0x5EB1, have arrived at the coordinator within 0.56 s. The maximal time interval between different packets is 0.36 s. These packets own only a few bytes. However, the packet content is the location of detected objects, which is a substitute of images in [13]. It indicates that the proposed architecture can realize real-time data aggregation from multi-views of visual sensors.

Figure 8 depicts the time-channel diagram of AT mode. 9483 packets in channel 09 have been captured in 8 min. It indicates the tremendous communication traffic of XBee sensors network. Even though there are still some disturbance in channel 07 and 08, the communication status of channel 09 remains stable and fluent along the timeline. It shows excellent anti-interference performance.

Fig. 7 Packet list of multi-channel packet analyser

Fig. 8 Time-Channel diagram of multi-channel packet analyser

5 Conclusions

According to the requirement of the disaster relief system, an architecture in indoor WMSN was presented. It established XBee sensors network and fulfilled the requirement of real-time data aggregation and RF transmission instantaneously. The detected results of multi-views can be transmitted to the coordinator synchronously and clustered in real time. The experimental results showed that the proposed architecture can realize real-time data aggregation of multi-views reliably and synchronously.

Acknowledgements This work is supported by National Natural Science Foundation of China (Nos. 61273078, 61471110).

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Akyildiz, I. F., Melodia, T., & Chowdhury, K. R. (2007). A survey on wireless multimedia sensor networks. *Computer Networks*, 51(4), 921–960.
2. Berclaz, J., Fleuret, F., Turetken, E., & Fua, P. (2011). Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9), 1806–1819.
3. Berger, A., Pichler, M., Klinglmayr, J., Pötsch, A., & Springer, A. (2015). Low-complex synchronization algorithms for embedded wireless sensor networks. *IEEE Transactions on Instrumentation and Measurement*, 64(4), 1032–1042.
4. Dinh, H., & Inanc, T. (2009). Low cost mobile robotics experiment with camera and sonar sensors. In: *American control conference, 2009. ACC'09* (pp. 3793–3798), IEEE.
5. Faludi, R. (2010). *Building wireless sensor networks: with ZigBee, XBee, arduino, and processing*. Sebastopol: O'Reilly Media, Inc.
6. Feng, S., Wu, C., & Zhang, Y. (2016). Passive target localization in multi-view system. In *Control conference (CCC), 2016 35th Chinese* (pp. 8444–8447). IEEE.
7. Feng, S., Wu, C., Zhang, Y., & Jia, Z. (2014). Grid-based improved maximum likelihood estimation for dynamic localization of mobile robots. *International Journal of Distributed Sensor Networks*, 10(3), 1–16.
8. Jiang-bo, Q., & Yi-sheng, D. (2004). A clustering algorithm based on broad first searching neighbors. *Journal of Southeast University: Natural Science Edition*, 34(1), 109–112.

9. Karakaya, M., & Qi, H. (2014). Collaborative localization in visual sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 10(2), 18.
10. Li, A., Lin, M., Wu, Y., Yang, M. H., & Yan, S. (2016). Nus-pro: A new visual tracking challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2), 335–349.
11. Lim, C. H., Vats, E., & Chan, C. S. (2015). Fuzzy human motion analysis: A review. *Pattern Recognition*, 48(5), 1773–1796.
12. Mantri, D. S., Prasad, N. R., & Prasad, R. (2016). Mobility and heterogeneity aware cluster-based data aggregation for wireless sensor network. *Wireless Personal Communications*, 86(2), 975–993.
13. Pham, D. M., & Aziz, S. M. (2013). Object extraction scheme and protocol for energy efficient image communication over wireless sensor networks. *Computer Networks*, 57(15), 2949–2960.
14. Shitrit, H. B., Berclaz, J., Fleuret, F., & Fua, P. (2014). Multi-commodity network flow for tracking multiple people. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8), 1614–1627.
15. Smeulders, A. W., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A., & Shah, M. (2014). Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7), 1442–1468.
16. Toni, L., Maugey, T., & Frossard, P. (2015). Optimized packet scheduling in multiview video navigation systems. *IEEE Transactions on Multimedia*, 17(9), 1604–1616.
17. Wibowo, F. W., & Purwacandra, P. P. (2015). Object tracking using initial data to count object image based-on wireless sensor network. *Advanced Science Letters*, 21(1), 112–116.
18. Yoon, J. H., Yang, M. H., & Yoon, K. J. (2016). Interacting multiview tracker. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(5), 903–917.
19. You, K., Yang, W., & Han, R. (2015). The video collaborative localization of a miners lamp based on wireless multimedia sensor networks for underground coal mines. *Sensors*, 15(10), 25103–25122.

Sheng Feng received the B.S. in Computing from Greenwich University in 2004, the M.S. degree in software engineering and the Ph.D. degree in pattern recognition and intelligent system from Northeastern University, Shenyang, China, in 2013 and 2017. His research interests are in the area of intelligent robot, computer vision and wireless sensor networks.

ChengDong Wu was born in June 1960. He received the B.Sc. degree in electrical automation from the University of Shenyang Architectural and Civil Engineering, Shenyang, China, in 1983, the M.Sc. degree in control theory and its application from Tsinghua University, Beijing, China, in 1988, and the Ph.D. degree in industrial automation from Northeastern University, Shenyang, in 1994. He is currently the Director of the Institute of Artificial Intelligence and Robotics, Northeastern University. He was the Head of more than 20 foundation projects in the fields such as robot control and path planning, image processing, intelligent traffic system, and wireless network communication. He has authored or coauthored nine books and more than 150 of journal and conference papers. His current research interests include the pattern recognition and image processing, robot intelligent control, and wireless sensor network.

YunZhou Zhang received B.S. degree and M.S. degree in Mechanical and Electronic engineering from National University of Defense Technology, Changsha, China in 1997 and 2000, respectively. He received Ph.D. degree in pattern recognition and intelligent system from Northeastern University, Shenyang, China, in 2009. He is now an professor with Northeastern University, China. His research interests include image processing, intelligent robot and wireless sensor networks. He is a member of the IEEE society.