Sensors **2007**, *7*, 1001-1027

sensors

ISSN 1424-8220

www.mdpi.org/sensors

Full Paper

Distributed Peer-to-Peer Target Tracking in Wireless Sensor Networks

Xue Wang *, Sheng Wang †, Dao-Wei Bi ‡ and Jun-Jie Ma §

State Key Laboratory of Precision Measurement Technology and Instrument, Tsinghua University, Beijing 100084, P. R. China; E-mails: † wang_sheng00@mails.tsinghua.edu.cn; ‡bdw02@mails.tsinghua.edu.cn; § mjj@mails.tsinghua.edu.cn

* Author to whom correspondence should be addressed; E-mail: wangxue@mail.tsinghua.edu.cn

Received: 6 June 2007 / Accepted: 25 June 2007 / Published: 25 June 2007

Abstract: Target tracking is usually a challenging application for wireless sensor networks (WSNs) because it is always computation-intensive and requires real-time processing. This paper proposes a practical target tracking system based on the auto regressive moving average (ARMA) model in a distributed peer-to-peer (P2P) signal processing framework. In the proposed framework, wireless sensor nodes act as peers that perform target detection, feature extraction, classification and tracking, whereas target localization requires the collaboration between wireless sensor nodes for improving the accuracy and robustness. For carrying out target tracking under the constraints imposed by the limited capabilities of the wireless sensor nodes, some practically feasible algorithms, such as the ARMA model and the 2-D integer lifting wavelet transform, are adopted in single wireless sensor nodes due to their outstanding performance and light computational burden. Furthermore, a progressive multi-view localization algorithm is proposed in distributed P2P signal processing framework considering the tradeoff between the accuracy and energy consumption. Finally, a real world target tracking experiment is illustrated. Results from experimental implementations have demonstrated that the proposed target tracking system based on a distributed P2P signal processing framework can make efficient use of scarce energy and communication resources and achieve target tracking successfully.

Keywords: Wireless sensor networks, target tracking, distributed signal processing, peer-to-peer, ARMA model.

1. Introduction

Wireless sensor networks (WSNs) are being envisioned and developed for a variety of applications involving monitoring and manipulation of the physical world in a tetherless fashion Typically, each individual sensor node can sense in multiple modalities but has limited signal processing and communication capabilities. Because of its spatial coverage and multiplicity in sensing aspect and modality, WSN ideally suited for visual target tracking by conquering the disadvantages of traditional single-view tracking, such as: a limited observation window, ambient noise, interference, processing limitations at the sensor in terms of power and memory and sensor reliability issues [1]. However, visual target tracking via a WSN is also especially a very challenging, multi-faceted problem in which many challenges must be overcome. In particular, two critical problems must be addressed in this field: efficient single sensor node algorithms with low computational cost, and distributed signal processing with collaboration between wireless sensor nodes.

Recently, several research groups have tackled various aspects of target tracking in WSNs [2-5]. In this paper, we focus our research efforts on the implementation of a visual multi-view target tracking system in WSNs with a spotlight on distributed peer-to-peer (P2P) signal processing and the specific algorithms which can be successfully adopted under the constraints imposed by the limited communication and computational abilities of the sensor nodes as well as their finite battery life.

In contrast to centralized signal processing, distributed signal processing can reduce latency, wireless bandwidth and energy consumption, as well as improve the robustness of network connections. Because information quality and energy consumption are both important for WSNs, how the information is gathered and signal processing is carried out, querying and routing tasks in a distributed manner with consideration of the tradeoff between sensing accuracy and energy efficiency are among the most important issues addressed in distributed signal processing. Several methods have been proposed to implement distributed signal processing with collaboration between sensor nodes. Zhao [6] presented an information-driven approach to sensor collaboration in *ad hoc* sensor networks which considered the information utility of each sensor node and developed several approximate measures of the information utility. Qi [7] proposed a mobile agent-based distributed sensor networks (MADSNs) which adopted mobile agents to incrementally carry out data fusion. Xu [8] introduced a distributed computing framework, called MADSN, to carry out collaborative signal processing in sensor networks using mobile agents. In the previous work, information quality is measured by analyzing the predicted contribution of their sensing actions and energy consumption which is nearly proportional to bandwidth appropriation is estimated from the network structure. For implementing target tracking in WSNs, in this paper, distributed signal processing is combined with a P2P architecture. P2P is another kind of novel architecture which has the advantages of robustness and dynamic. Thus architecture can increase the performance and prolong the lifetime of WSNs because it can reduce the congestion and energy consumption.

Furthermore, a target tracking system for strictly constrained WSNs is designed, which consists of several specific signal processing algorithms for target detection, classification and tracking. In the proposed system, background subtraction based target detection, 2-D integer lifting wavelet transform (ILWT) based feature extraction, support vector machine (SVM) based target classification and auto regressive moving average (ARMA) model based target tracking are carried out in each sensor node, while multi-view localization algorithm is implemented with the collaboration between wireless sensor nodes in a distributed P2P signal processing framework.

The structure of this paper is as follows. Section II introduces the distributed peer-to-peer signal processing framework. Section III presents the target tracking system which consists of specific in-node and collaborative signal processing algorithms for target tracking in distributed P2P signal processing framework and discusses the feasibility of the proposed system. Section IV analyzes the performance, energy consumption and execution time of the proposed target tracking system. Finally, conclusions are given in section VI.

2. The Basis of Distributed Peer-to-Peer Signal Processing Framework

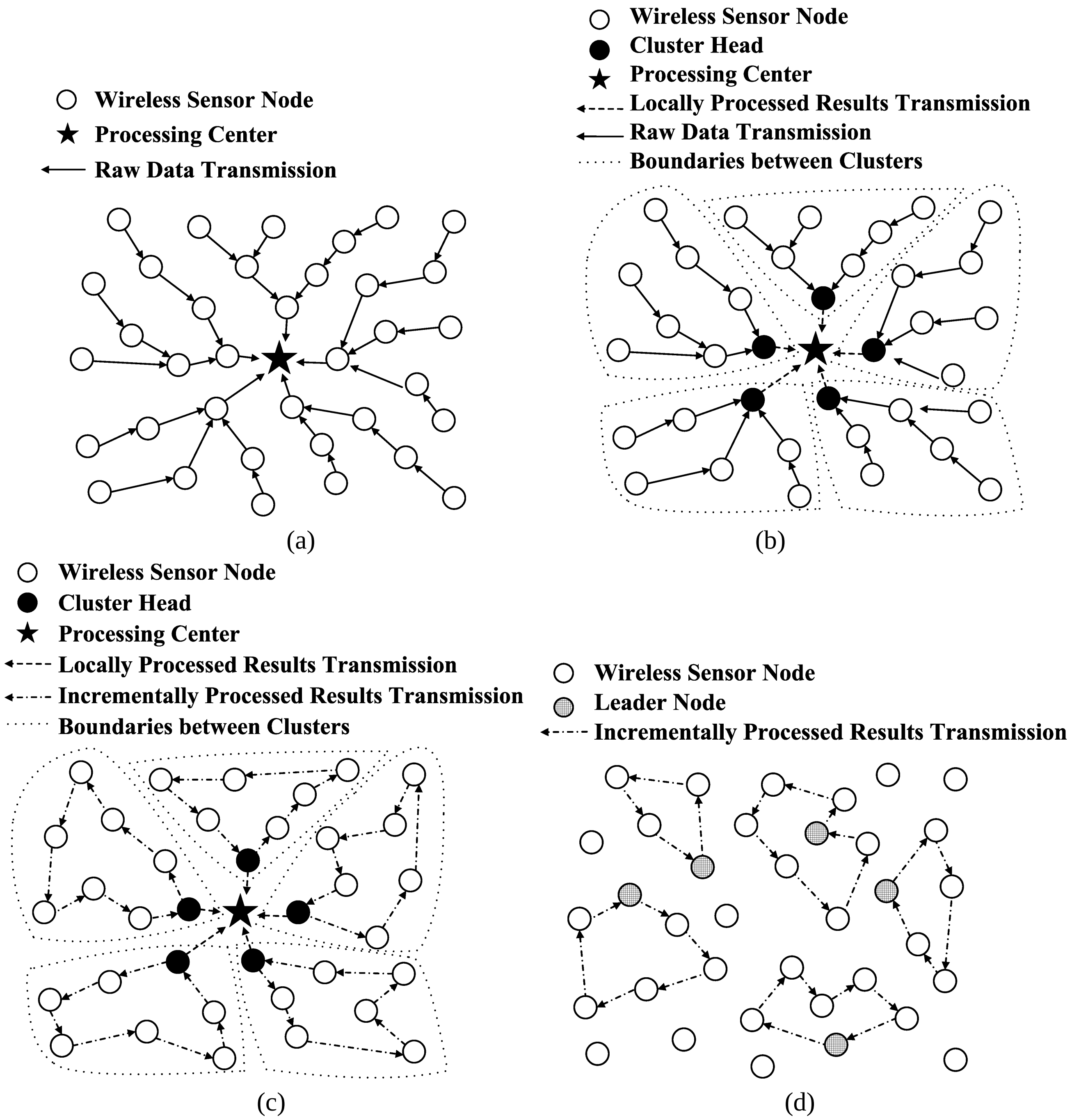
2.1. Preliminaries

In WSNs, the centralized client/server is one of the most popular signal processing frameworks. As shown in Figure 1(a), in this framework, the unique processing center sends out the commands to relative sensor nodes to acquire the information, then a selected wireless sensor node transmits raw signal to the processing center, where further signal processing is performed. Although it is widely used, client/server is not appropriate for signal processing in WSNs, because it consumes many scarce resources, such as battery power and network bandwidth, for transmitting the raw signal, and it also needs some super sensor nodes acting as processing centers, which require much higher energy, storage and computing capabilities [6].

Recently, distributed signal processing became another hotspot of signal processing in WSNs, which is always carried out in the WSNs with clusters. Distributed signal processing is always implemented in a client/server framework, where distributed local signal processing at a cluster head replaces the centralized signal processing in the processing center to decrease the workload of the latter, reducing the amount of data transmission and balancing the signal processing tasks among wireless sensor nodes. The distributed client/server signal processing framework is illustrated in Figure 1(b). However, the topology of WSNs determines that, in distributed client/server signal processing frameworks, the closer a sensor node is to the processing center, the more energy the sensor node will consume because it has to be an intermediate sensor node to route the packets from other sensor nodes. This results in sensor nodes closer to the computing center dying much more rapidly than other sensor nodes.

The structure of MADSN is illustrated in Figure 1(c). Obviously, distributed mobile agent signal processing is also built on hierarchical structure which needs a processing center and causes an imbalance between transmission and energy consumption. Furthermore, distributed mobile agent signal processing is significantly limited by the size of the mobile agent. In some specific complex application, the overhead energy consumption for transmitting mobile agent will largely increase the total energy consumption in signal processing, which can not be afforded in strictly constrained WSNs

Figure 1. Different signal processing frameworks in WSNs: (a) centralized client/server framework; (b) distributed client/server framework; (c) distributed mobile agent framework and (d) distributed P2P framework.



2.2. The Structure of Distributed P2P Signal Processing Framework

P2P networking has recently emerged as a new framework for building networked applications. P2P differs from client/server in several crucial ways. Perhaps most importantly, a peer is both a producer and a consumer of the implemented service while the clients only generate workload and workload is processed by servers in client/server. P2P computing is a collective computing environment where a peer is not only able to act as both a “client” and a “server”, but also can interact with other peers in more complex ways to accomplish the task at hand [9]. With distributed P2P signal processing seen as a prolongation of P2P computing, its objectives are as follows: collaboration, local autonomy, high performance via parallelism, resource heterogeneity management and minimum impact on local computation. The distributed P2P signal processing framework is responsible for finding the appropriate resources, executing required processes, and returning results [9].

In distributed P2P signal processing framework, wireless sensor nodes act peers which perform the data acquiring, processing and transmitting in WSNs. Limited energy resources makes it important to develop progressive signal processing to provide incremental accuracy. However, different from MADSN, some sensor nodes dynamically and autonomously become leader sensor nodes for result collecting, analyzing and storing according to the specific tracking scenario in the distributed P2P signal processing framework. During the procedure of signal processing, each wireless sensor node periodically carries out local signal pre-processing and acquires the local results. At the same time, signal processing is started from a leader sensor node and is progressively performed from one sensor node to another in a progressive distributed data fusion mechanism. That is, each sensor node integrates its result with previous results to potentially increase accuracy. After that, the sensor node transmits the partially fused results to other sensor nodes one after another. Once the predefined criterion is met, such as desired accuracy being achieved, the last sensor node terminates migration and returns the result. The scenario of distributed P2P signal processing framework is shown in Figure 1(d). If the communication latency is low, the progressive signal processing can be almost considered as a synchronous procedure, so it can be realized without any time stamp information.

The distributed P2P signal processing framework can remarkably increase the performance of WSNs, because the distributed P2P signal processing framework carries out the signal processing procedure incrementally between the selected sensor nodes according to the specific requirement and resources, and it can greatly reduce the energy consumption and network congestion because of the sequential data transmission. Furthermore, the multiformity of the wireless sensor nodes can improve the robustness of the whole network and the wireless sensor nodes may be active or go off-line in a very dynamic fashion. Compared to distributed signal processing, distributed P2P signal processing is a completely information driven framework for signal processing in WSNs; it can dynamically and autonomously carry out signal processing with proper set of wireless sensor nodes without a set processing center. Distributed P2P signal processing is an ideal framework to complete complex multi-threads tasks. One of the core challenges in building P2P systems is how to achieve high performance signal processing efficiently [10]. For distributed P2P signal processing framework, a basic strategy should be used to select leader sensor node and sensor nodes with practically feasible metrics and organize the schedules of sensor nodes for implementing the signal processing with the lowest congestion and consumption in WSNs.

In practice, many wireless sensor nodes are randomly deployed in the sensing area. When a phenomenon of interest occurs, some wireless sensor nodes will detect it and announce their leadership among nearby sensor nodes. However, the problem of simultaneous announcement between different sensor nodes may increase the number of leader sensor nodes and result in unnecessary energy consumption. When a wireless sensor node detects the phenomenon of interest, it will wait a randomized back-off time delay before announcing the leadership. Once a wireless sensor node receives a leadership announcement from a nearby sensor node before its own announcement, it will stop announcing and act as a normal wireless sensor node. If the leadership announcement from nearby sensor node is received after its announcement, the wireless sensor node will compare the timestamps of two announcements and return the result to the source sensor node. The wireless sensor node which makes the earliest announcement is determined to be the leader sensor node. As the phenomenon of interest moves or environmental conditions vary, the leadership may change hands among sensor nodes.

Besides leader sensor node selection, the key research challenge for distributed P2P signal processing is the routing problem in progressive signal processing. Because of the intrinsic properties of distributed P2P signal processing, the routing of the accessed wireless sensor nodes has a significant impact on the performance and energy consumption of signal processing. The objective of routing is to find a path to satisfy the desired signal processing accuracy while minimizing the energy consumption. A routing algorithm based on the consideration of energy consumption, path loss, and signal energy is introduced in [11]. But the proposed algorithm just focuses on energy consumption without the consideration of the information utilities of wireless sensor nodes. For balancing the energy consumption and information utility, as presented in [6], the sensor selection can be considered as an optimization problem with the following objective function:

$$M\left(S^i\right) = -\alpha \cdot \varphi_{Cost}\left(S^i\right) + (1-\alpha) \cdot \varphi_{Utility}\left(S^i\right) \tag{1}$$

where S^i is the candidate wireless sensor node, φ_{Cost} is the energy consumption metric, $\varphi_{Utility}$ is the information utility metric, α is the relative weight. Depending on applications and assumptions, φ_{Cost} and $\varphi_{Utility}$ have various forms.

2.2.1 Energy Consumption Metric

Practically, the energy consumption metric φ_{Cost} contains three basic types: sensing energy φ_s, signal processing energy φ_p and communication energy φ_c.

The sensing energy φ_s is determined by the sensing power p_s and sensing time t_s:

$$\varphi_s = p_s t_s \tag{2}$$

The signal processing energy φ_p is shown with the following equations [12]:

$$\varphi_p = \int_{t_0}^{t_0+t_p} p_p\left(t\right) dt \tag{3}$$

where t_p is the time taken for signal processing and $p_p\left(t\right)$ is the instantaneous power of the processor.

However, the overhead energy caused by fusion is much lower than the energy caused by local pre-processing, so each wireless sensor node almost consumes processing energy at same level. Because of this, the signal processing energy φ_p can be ignored in sensor selection. For the same reason, the sensing energy φ_s can be ignored too.

For communication energy φ_c, given minimum communication power p_0 along a standard distance l_0, instantaneous communication power is in direct proportion to the square of communication distance l_c between the source sensor node and destination sensor node [11]

$$p_c = \frac{l_c^2}{l_0^2}\frac{(4\pi)^2\beta}{G_t G_r \lambda^2} p_0 \tag{4}$$

where G_t, G_r are transmitting gain and receiving gain respectively. λ is the wavelength and β is the system loss factor. So the communication energy is

$$\varphi_c = \frac{p_0(4\pi)^2\beta}{G_t G_r \lambda^2}\frac{l_c^2 t_c}{l_0^2} \tag{5}$$

Obviously, the first term in Eq. (5) is a constant. Moreover, in the incremental data fusion procedure, the amount of transmitted data is almost constant, so the communication time t_c can be also considered as a constant. So, $\frac{l_c^2}{l_0^2}$ can be used as a dimensionless metric of communication energy φ_c.

Furthermore, for ensuring the sensing performance of WSNs, energy should be consumed evenly among all sensor nodes. Entropy theory is adopted here to measure the randomness of reserved energy in each sensor node. The entropy of reserved energy is as follow.

$$H\left(S^i(t)\right) = -\sum_k p\left(\tilde{E}_k^i(t)\right)\log p\left(\tilde{E}_k^i(t)\right) \tag{6}$$

where $\tilde{E}_k^i(t)$ is the estimated amount of energy reserved in *k*th sensor node when selecting *i*th sensor node for progressive signal processing at instant t. The bigger the entropy is, the more evenly the reserved energy is, so $\frac{1}{H\left(S^i(t)\right)}$ is used to scale the impact of energy consumption. The combined resource consumption metric is as follow.

$$\varphi_{Cost}\left(S^i(t)\right) = \frac{\left(l_c^i\right)^2}{l_0^2 H\left(S^i(t)\right)} \tag{7}$$

where l_c^i is the communication distance between current sensor node and *i*th sensor node.

2.2.2 Information Utility Metric

Information utility is another important metric for measuring the contribution of selected sensor nodes. In practice, the contribution of sensor nodes can be ideally predicted by analyzing the theoretical model of the specific application and the characteristics of sensor nodes. This metric can be called

$\varphi_{Predicted}$. Besides the predicted contribution, the information utility is always impacted by some other factors in the environment, such as obstacles and noise. However, it is difficult to measure their impact in practice. A more practical alternative is to estimate the confidence degree $\varphi_{Confidence}$ of the wireless sensor node, which can be measured by the contribution of the wireless sensor node in previous instants.

In conclusion, the information utility metric function can be defined as follows:

$$\varphi_{Utility}\left(S^{i}(t)\right)=\varphi_{Predicted}\left(S^{i}(t)\right)\cdot\varphi_{Confidence}\left(S^{i}(t)\right) \tag{8}$$

And then the objective function of routing in distributed P2P signal processing is:

$$M\left(S^{i}(t)\right)=(1-\alpha)\cdot\varphi_{Predicted}\left(S^{i}(t)\right)\cdot\varphi_{Confidence}\left(S^{i}(t)\right)-\alpha\cdot\left(\frac{\left(l_{c}^{i}\right)^{2}}{l_{0}^{2}H\left(S^{i}(t)\right)}\right) \tag{9}$$

Target tracking is one of the essential capabilities in WSNs. Because distributed P2P signal processing framework has advantages in dynamic, robust, energy efficiency and data transmission, target tracking in a distributed P2P signal processing framework can improve tracking performance by purposefully selecting proper set of wireless sensor nodes for progressive signal processing and data fusion. Moreover, a distributed framework decreases the energy consumption and reduces the execution time of signal processing, and the P2P architecture can improve the robustness and reduce the congestion and energy consumption in WSNs. Besides signal processing framework, in some ways, specific signal processing algorithms also have significant impact on the performance of target tracking, which will be discussed in the following section.

3. Specific In-Node and Collaborative Signal Processing Algorithms for Target Tracking

In this section, a combined tracking system is proposed and analyzed with the consideration of tradeoff between the accuracy and energy consumption, which consists of several feasible algorithms for target detection, classification, tracking and localization.

3.1. Target Detection, Classification and Tracking in Wireless Sensor Nodes

Target detection, classification and tracking are three important steps in target tracking. Because the data processing ability of wireless sensor network is extremely limited, the algorithms for target detection, classification and tracking should be lightweight with low computational complexity. The details are discussed below.

3.1.1 Target Detection Based on Background Subtraction

Background subtraction is demonstrated as a low-cost, simple, but efficient method for target detection. It is successfully used in W^4 real time visual surveillance system for tracking multiple targets [13]. Let $V^i(x)$ be the intensity of location x in the ith image of N consecutive images array V. $\sigma(x)$ and $\eta(x)$ are the standard deviation and median value of intensities. The initial background model $[\gamma_m(x), \gamma_n(x), \gamma_d(x)]$ is obtained as

$$\begin{bmatrix} \gamma_m(x) \\ \gamma_n(x) \\ \gamma_d(x) \end{bmatrix} = \begin{bmatrix} \min_i \{V^i(x)\} \\ \max_i \{V^i(x)\} \\ \max_i \{\lvert V^i(x) - V^{i-1}(x) \rvert\} \end{bmatrix}, \quad \text{where } \lvert V^i(x) - \eta(x) \rvert < 2 * \sigma(x). \tag{10}$$

After initialization, three change maps, the detection support map (*gS*), motion support map (*mS*) and change history map (*hS*), are adopted to represent the number of times a pixel location is classified as a background pixel, moving pixel and elapsed foreground pixel respectively.

$$gS(x,t) = \begin{cases} gS(x,t-1)+1 & \text{if } x \text{ is background pixel} \\ gS(x,t-1) & \text{if } x \text{ is foreground pixel} \end{cases} \tag{11}$$

$$mS(x,t) = \begin{cases} mS(x,t-1)+1 & \text{if } \left(\lvert I(x,t) - I(x,t+1) \rvert > 2*\sigma\right) \wedge \left(\lvert I(x,t-1) - I(x,t) \rvert > 2*\sigma\right) \\ mS(x,t-1) & \text{otherwise} \end{cases} \tag{12}$$

$$hS(x,t) = \begin{cases} 255 & \text{if } x \text{ is foreground pixel} \\ hS(x,t-1) - \dfrac{255}{N} & \text{otherwise} \end{cases} \tag{13}$$

Change maps are set to zero after the background model is updated.

For updating the background model, the new background model $[\gamma_m(x), \gamma_n(x), \gamma_d(x)]$ is determined as follows.

$$[\gamma_m(x), \gamma_n(x), \gamma_d(x)] = \begin{cases} [\gamma_m^b(x), \gamma_n^b(x), \gamma_d^b(x)] & \text{if } (gS(x) > r_1 * N) \text{ (pixel-based)} \\ [\gamma_m^f(x), \gamma_n^f(x), \gamma_d^f(x)] & \text{if } (gS(x) < r_1 * N \wedge mS(x) < r_2 * N) \text{ (object-based)} \\ [\gamma_m^c(x), \gamma_n^c(x), \gamma_d^c(x)] & \text{otherwise} \end{cases} \tag{14}$$

where $[\gamma_m^c(x), \gamma_n^c(x), \gamma_d^c(x)]$ is the current background model; $[\gamma_m^b(x), \gamma_n^b(x), \gamma_d^b(x)]$ presents that the pixel is classified as background pixel, while $[\gamma_m^f(x), \gamma_n^f(x), \gamma_d^f(x)]$ presents the pixel is classified as foreground pixel. As presented in [13], r_1 and r_2 are scaling factors to adjust the effect of number *N*, they are typically 0.8 and 0.1, respectively.

Giving the minimum $\gamma_m(x)$, maximum $\gamma_n(x)$, and the median of the largest interframe absolute difference $\gamma_{d'}$ images, a pixel *x* from image I^t is a foreground pixel if:

$$B(x)=\begin{cases}0 \text{ background} & \left(\left(I^t(x)-\gamma_m(x)\right)<r_3\gamma_{d'}\right)\vee\left(\left(\gamma_n(x)-I^t(x)\right)<r_3\gamma_{d'}\right)\\ 1 \text{ foreground} & \text{otherwise.}\end{cases} \tag{15}$$

where r_3 is set to 2 in our system. According to the complexity of calculation, the computational cost of this algorithm is low, and the detailed discussion can be found in [13]. It means that background subtraction is practically feasible in WSNs.

Normally, the output result of target detection is the minimum boundary rectangle (MBR) of target. Target detection is always the basic and preliminary item for target classification and tracking which will be detailedly discussed in the following sections.

3.1.2 Target Classification Based on 2-D ILWT and SVM

With the MBR results of target detection, the target information is acquired. Next, target classification is desired for further signal processing. Target classification contains two key elements: feature extraction and division. Feature extraction refers to a transformation for reducing the number of effective features of an original data set by retaining most of the intrinsic information. For images, compression can be considered an effective technique for feature extraction.

The use of the discrete wavelet transform (DWT) for embedded lossy image compression is now well established [14]. The wavelet transform of a function $f(x)\in L_2(R)$ is defined as (1-D case):

$$WT\{f(x);\delta_a,\delta_b\}=\int f(x)\psi_{\delta_a,\delta_b}(x)dx \tag{16}$$

where $\psi_{\delta_a,\delta_b}(x)$ defines the family of the wavelet functions:

$$\psi_{\delta_a,\delta_b}(x)=\frac{1}{\sqrt{|\delta_a|}}\psi\left(\frac{x-\delta_b}{\delta_a}\right) \tag{17}$$

where $\delta_a\neq 0$ is the scale of the transform and δ_b is the parameter of spatial location.

With the discrete translation of function $f(x)$ on a dyadic scale $\delta_a=2^j$ and discrete translation $\delta_b=2^jk$, the discrete wavelet transform (DWT) can be presented as follow [15]:

$$DSWT\{f(x);2^j,2^jk\}=C_{j,k}=WT\{f(x);\delta_a=2^j,\delta_b=2^jk\} \tag{18}$$

where $C_{j,k}$ is the wavelet coefficients of the function $f(x)$, and the wavelet function forms a orthogonal and complete dyadic family:

$$\psi_{j,k}(x) = 2^{-\frac{1}{2}j}\psi\left(2^{-j}x - k\right) \tag{19}$$

Thus, the function $f(x)$ may be obtained from its wavelet coefficients $C_{j,k}$:

$$f(x) = \sum_{j\in Z}\sum_{k\in Z} C_{j,k}\psi_{j,k}(x) \tag{20}$$

For discrete signals $f(n)$, $n \in Z$,the DWT is defined as:

$$DWT\left\{f(n); 2^i, 2^j k\right\} = \left\{c_{j,k}\right\} = \sum\nolimits_{n\in Z} f(n) g_j^*\left(n - 2^j k\right) \tag{21}$$

where $g_j^*\left(n-2^j k\right)$ is the discrete equivalent of the $2^{-\frac{1}{2}j}\psi\left(2^{-j}\left(x-2^j k\right)\right)$. With DWT, the $f(n)$ can be written as a multiresolution decomposition on *J* levels, $j = 1, \ldots, J$, given by [15]:

$$f(n) = \sum_{j=1}^{J}\sum_{k\in Z} c_{j,k}\tilde{g}_j\left(n-2^j k\right) + \sum_{k\in Z} e_{j,k}\tilde{h}_J\left(n-2^J k\right) \tag{22}$$

where $\tilde{g}_j\left(n-2^j k\right)$ is the synthesis wavelets and discretely equal to $\psi_{j,k}$, and the scaling coefficients $e_{j,k}$ is defined as:

$$e_{j,k} = \sum_n f(n) h_J^*\left(n-2^J k\right) \tag{23}$$

where $h_J^*\left(n-2^J k\right)$ is the scaling sequences.

For images, a wavelet descomposition is applied first to image rows and then to columns. The two dimensional wavelet transform leads to a descomposition in four sets of coefficients: approximation and details in three different orientations, vertical, horizontal and diagonal. The full descomposition is obtained by iterating the filtering on the approximation set, where the approximation at *j+1*th level can be approximately considered as the compressed image information [14].

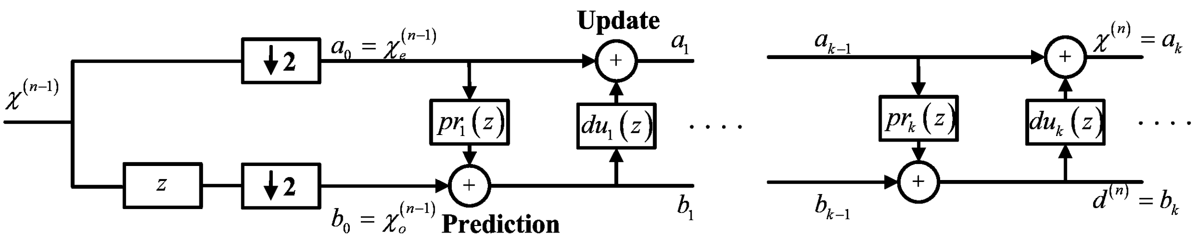
However, in WSNs, the limited processing abilities of wireless sensor nodes limit the possibility to employ DWT algorithm. To simplify the computation, a lifting scheme (LS) is introduced in DWT [16]. LS exploits the redundancy between the high pass and low pass filters necessary for perfect reconstruction and reduces the number of arithmetic operations up to a factor of two, compared to the filter-bank implementation, The LS based wavelet transform seems to be an ideal candidate for embedded image compression issues, due to its simple structure and good decorrelation properties [17].

As shown in Figure 2, in LS, the input data are split into two signals with evenly and oddly indexed samples respectively. One signal is convolved with a primal lifting filter $pr_1(z)$, and then the role of the two signals is then reversed and dual lifting filter $du_1(z)$ is applied, where $pr_1(z)$ and $du_1(z)$ are simple and short FIR filters. After k iterations of primal and dual lifting, the input signal $\chi^{(n-1)}$ is split into $d^{(n)}$ which corresponds to the details of $\chi^{(n-1)}$, and $\chi^{(n)}$ which corresponds to the approximation. In reconstruction, the reconstruction filter $pr_i'(z)$ and $du_i'(z)$ are exactly equivalent to its decomposition counterpart $pr_i(z)$ and $du_i(z)$, without its sign, i.e.,

$$pr_i'(z) = -pr_i(z) \tag{24}$$

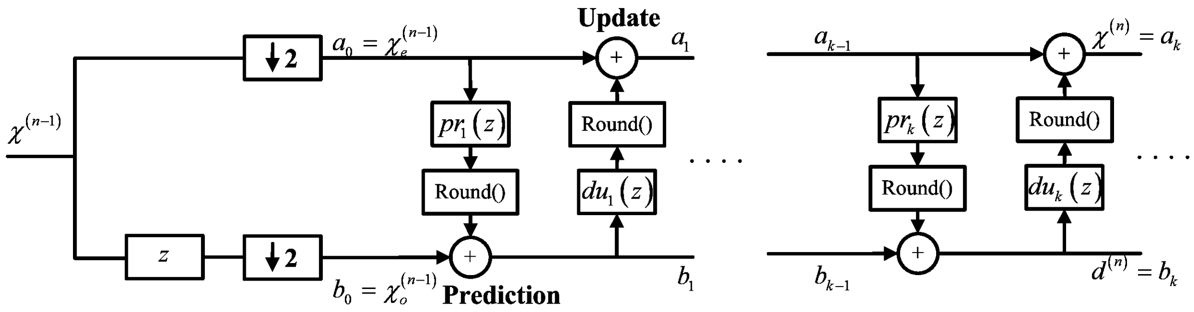
$$du_i'(z) = -du_i(z) \tag{25}$$

Figure 2. The structure of the lifting scheme in DWT.



For conquering the disadvantage of lossy image compression in LWT, a further improvement is achieved by combining the benefits offered by the integer wavelet transform (IWT) and integer LS, because IWT enables true lossless reversible transformations [18]. The structure of ILWT is illustrated in Figure 3. Because of completeness, the performance of IWT is slightly worse than DWT. However, ILWT structure can greatly reduce the computational complexity which implies that ILWT is an ideal choice for low energy systems. In practice, ILWT has been successfully used in embedded processing of wireless sensor nodes for image compression in wireless sensor networks [18]. Then the compressed image can be considered as the compact representation of original image for target classification, which can decreasing the computational complexity in the polynomial computation in each node and ensuring this selected subspace can retain enough intrinsic information of the original space.

Figure 3. The structure of lifting scheme in LWT.



After image compression, SVM is used to construct a classifier for target classification. A SVM is essentially a linear classifier operating in a higher dimensional space. For reducing computational load, a kernel function $K(\cdot)$ is used to perform input-feature transformation $\phi(\cdot)$, which is defined as follows:

$$K(m \cdot n) = \phi(m) \cdot \phi(n). \tag{26}$$

With the kernel functions, the basic form of SVM is as follow.

$$f(m) = \mathrm{sign}\left(\sum_{i=1}^{l} q_i \left(\phi(m) \cdot \phi(m_i) \right) + b \right). \tag{27}$$

where q_i is the weighting factors $(0 < q_i < \infty)$, b is a scalar threshold for adjusting the results of classification, here, it is set to 0. In this paper, a third degree polynomial kernel is used:

$$K(m,n) = \left(m^T n + 1 \right)^3 \tag{28}$$

Unfortunately, the training phase of SVM algorithm may take a long time. However, once the classifier is trained, SVM just need to estimate the posterior probability for each class, so the computational complexity is rather low. For multi-category classification, multiple classifiers can be trained and combined for more complicated classification. Moreover, some distributed training methods for SVM algorithms were proposed in many literatures. These methods are demonstrated feasible in WSNs [19]. After training, the prerequisite parameters of kernel functions are transferred to each node for classification. Each node just needs to perform some polynomial operation, which is reasonable for the processing ability of wireless sensor nodes.

3.1.3 Target Tracking and Estimation with ARMA Model

After target detection, wireless sensor node uses historical information of target location for target tracking and estimation. Target tracking is always performed by Kalman filter. However, it is extremely challenging to implement a Kalman filter to track a maneuvering target if the dynamic model of target is highly nonlinear [20]. Although a standard particle filter can solve the nonlinear non-Gaussian problem [21], it cannot solve the estimation error of cumulating problems when manoeuvers occur. Furthermore, some algorithms have been proposed for maneuvering target tracking, such as unscented particle filter [20], radial basis function based particle filter [21]. But these algorithms are computation-expensive for wireless sensor node with simple embedded data processing capabilities.

When the target is moving in Cartesian coordinates, the target motion can be described by the following state space model, which is expressed along the X-axis for simplicity.

$$\mathrm{x}(k+1)=\mathrm{F}\mathrm{x}(k)+\mathrm{G}\mathrm{w}(k) \tag{29}$$

And the measurement model is given by

$$\mathrm{y}(k+1)=\mathrm{H}\mathrm{x}(k)+\mathrm{v}(k) \tag{30}$$

where the process noise $\mathrm{w}(k)$ and the measurement noise $\mathrm{v}(k)$ are assumed to be zero mean, independent white Gaussian sequences. In the case of a maneuvering target, the state vector x is defined as $\mathrm{x}=[x \quad \dot{x} \quad \ddot{x}]^T$, where x is the target position, $\dot{x}$ is the velocity and $\ddot{x}$ is the acceleration. The corresponding matrices are

$$\mathrm{F}=\begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & T \end{bmatrix}, \quad \mathrm{G}=\begin{bmatrix} \frac{T^3}{6} \\ \frac{T^2}{2} \\ T \end{bmatrix}, \quad \mathrm{H}=\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{31}$$

where T is the measuring interval.

Here, a new state vector $\mathrm{x}'=[x_1 \quad x_2 \quad x_3]^T$ is given, where x' and x are related by the following equation

$$\mathrm{x}'=\mathrm{P}_0^{-1}\mathrm{x} \tag{32}$$

and the transformation matrix P_0^{-1} is given by

$$\mathrm{P}_0^{-1}=\begin{bmatrix} \mathrm{H} \\ \mathrm{HF} \\ \mathrm{HF}^2 \end{bmatrix}=\begin{bmatrix} 1 & 0 & 0 \\ 1 & T & \frac{T^2}{2} \\ 1 & 2T & 2T^2 \end{bmatrix} \tag{33}$$

Therefore $x_i, i = 1,2,3$, in vector x' are related to the target position, velocity and acceleration by

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \mathrm{P}_0^{-1} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} x \\ x + T\dot{x} + \dfrac{T^2}{2}\ddot{x} \\ x + 2T\dot{x} + 2T^2\ddot{x} \end{bmatrix} \tag{34}$$

The new state vector x' contains only the target position information at each subsequent time instant, and it is equivalent to x in presenting the motion of target since they are related by a linear transformation. This implies that the problem of maneuvering target tracking can be viewed as a problem of adaptive time series prediction [22].

Here, the ARMA model is adopted due to its outstanding performance in model fitting and forecasting and its light computational cost [23]. ARMA model is a widely-used model for the prediction of future values. It contains two terms, the auto regressive (AR) term and the moving average (MA) term. The AR term is a linear regression of current value against one or more prior values. It captures the dependency of current value and its nearest prior values. And the MA term is introduced to capture the influence of random shocks to the future. In general, a linear system can be derived as follow.

$$s(n) = \sum_{i=1}^{p} a_i s(n-i) + \sum_{i=0}^{q} b_i \omega(n-i) \tag{35}$$

where the system input is $\omega(n)$, output is $s(n)$, a_i and b_i are the AR coefficients and MA coefficients respectively, p is the order of the system and $q \le p$. The measurement equation is

$$y(n) = s(n) + \upsilon(n), \quad 0 \le n \le N-1 \tag{36}$$

where $\upsilon(n)$ is the measurement noise and N is the number of measurements.

Normally, the MA coefficients are estimated by minimizing the error between the actual measurements and the weighted impulse response sequence generated by the estimated denominator coefficients.

For reducing the computation complexity in estimating the AR and MA coefficients, the AR coefficients a_i is estimated by using the robust singular value decomposition (SVD) based linear predictive coding algorithm (LPCA) method, while the MA coefficients b_i are estimated by minimizing the error between the actual measurements and the weighted impulse response sequence generated by the estimated denominator coefficients as usual [22]. The SVD based LPCA method is presented as follows.

The forward prediction in LPCA is described by

$$\hat{y}(n) = \sum_{i=1}^{M} \hat{a}_i y(n-i) \tag{37}$$

where M is the order of the filter, $\hat{y}(n)$ is the estimate, $y(n)$ is the measured data, and the above equation can be written as

$$\hat{Y} = \mathrm{A}\hat{\mathrm{a}} \tag{38}$$

where $\hat{\mathrm{a}} = [\hat{a}_1, \hat{a}_2, \dots \hat{a}_M]^T$, $\hat{\mathrm{Y}}$ is a $(N-M)\times 1$ vector and A is a $(N-M)\times M$ matrix, $p \le M \le N-M$. The optimal vector $\hat{\mathrm{a}}$ is given as follows:

$$\mathrm{A}^T \mathrm{A}\hat{\mathrm{a}} = \mathrm{A}^T \mathrm{Y} \tag{39}$$

For decreasing the computational complexity of computing $\mathrm{A}^T\mathrm{A}$, the robust SVD method is applied. Let

$$\mathrm{A} = \mathrm{U}\begin{bmatrix} \rho & 0 \\ 0 & 0 \end{bmatrix}\mathrm{V}^T \tag{40}$$

where $\mathrm{U} = [\mathrm{u}_1, \mathrm{u}_2, \dots \mathrm{u}_{N-M}]$ and $\mathrm{V} = [\mathrm{v}_1, \mathrm{v}_2, \dots \mathrm{v}_M]$ are unitary matrices of dimension $(N-M)\times(N-M)$ and $M \times N$, $\rho = diag[\rho_1, \rho_2, \dots, \rho_r]$ is an $r \times r$ matrix where and r is the rank of A. So the optimal weights $\hat{a}$ as

$$\hat{a} = \sum_{k=1}^{r} \rho_k^{-1} \mathrm{v}_k \mathrm{u}_k^T \mathrm{Y} \tag{41}$$

According to the target state model, a third order filter is ideally sufficient to track a maneuvering target. However, it must be noted that the order of the proposed ARMA filter should be higher than the order of the target state model, especially for target tracking with noise. In this paper, the order of the system is set to $p = 4, q = 3$. Furthermore, tracking target with ARMA model requires a delay of N steps for the parameter estimation. However, this problem only occurs in the initialization. When sufficient measurements are available, no delay is required.

3.2. Distributed Target Localization with P2P Manner

Although each wireless sensor node can achieve target detection, classification and tracking, 3-D localization can only be achieved by fusing the information from multiple wireless sensor nodes, because single wireless sensor node can only acquire the bearings of target. Because of the requirement of dynamic, energy efficiency and low computation complexity, a multi-view localization method based on distributed P2P signal processing framework is proposed.

3.2.1 Multi-view Localization

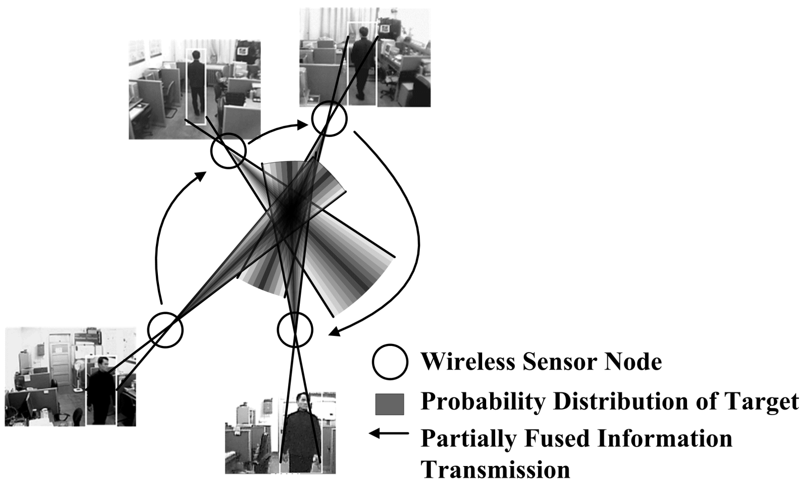
For integrating the bearing information of each wireless sensor node and achieve target localization, a simple multi-view localization method is proposed to conquer the negative effects of occlusion and obstacles by fusing MBR results of several sensor nodes. It is assumed that the measuring uncertainty of target position can be effectively approximated by a 2-D Gaussian distribution in MBR. As illustrated in Figure 4, after projecting the Gaussian distribution to the ground, multi-view localization fuses Gaussian distribution results by

$$E\left(x \middle| \tau_1, \ldots \tau_n\right) = \xi\left(\sum_{i=1}^{n} p\left(x \middle| \tau_i\right) \times p\left(\tau_i\right)\right) \tag{42}$$

where $p\left(x \middle| \tau_i\right)$ denotes Gaussian distribution acquired by *i*th wireless sensor node, $p\left(\tau_i\right)$ is the scale factor. Here, it's given by distance between wireless sensor node and target. ξ presents the normalizing operator. x is the fused result. Then the approximate position of moving target can be calculated as expectation of the distribution *X*:

$$E\left(x\right) = \sum_{i} x_i \cdot p\left(x_i\right) \tag{43}$$

Figure 4. Multi-view localization fuses Gaussian distribution results from different wireless sensor nodes.



The fused results are at most as large as the uncertainty of the most accurate individual sensor node, if all sensor nodes work normally. But if some sensor nodes are broken or blocked, the corresponding results are not credible. So, if the probability distribution becomes zero or even low after fusing the MBR results of some sensor nodes, the relative sensor nodes will be considered as useless ones, the results will not be fused and the original results will be transmitted to other nodes instead.

Furthermore, a simple voting mechanism, utilizing the status consistency and the target tracking information, is applied. For each node, one vote is cast for the resulting target class, and votes are counted for 5 neighboring frames at each step, the target class receiving the largest number of votes decides the target's class in this node. This voting mechanism reduces the number of random misclassifications in video sequence caused by the temporal occlusion.

3.2.2 Multi-view Localization with Distributed P2P Signal Processing Framework

Because of the inherent characteristic of the multi-view localization method, it can be incrementally carried out in distributed P2P signal processing framework. In this application, the resource consumption retains the general form which is described in Section II, and the application-specific information utility metric is discussed in here.

Image sensor node is a kind of bearing-only sensor node. For bearing-only sensor nodes, Wang [24] has proposed an entropy-based sensor selection heuristic, which selects an informative sensor such that the fusion of the selected sensor observation with the prior target location distribution would yield on average the greatest or nearly the greatest reduction in the entropy of the target location distribution.

The entropy of the probability distribution of the view of wireless sensor node i, H_i^v, is

$$H_i^v = -\int p\left(z_i^v\right)\log p\left(z_i^v\right)dz_i^v \tag{44}$$

where z_i^v is the view of sensor i about the sensor location. For computing the distribution in numerical method, the discrete representation of $p\left(z_i^v\right)$ with a grid δz_i^v, H_i^v is computed as

$$H_i^v = -\sum p\left(z_i^v\right)\log p\left(z_i^v\right)\delta z_i^v \tag{45}$$

And the entropy of the sensing model of wireless sensor node i for the actual target location x^t is approximated as

$$H_i^s = -\int p\left(z_i\left|\hat{x}\right.\right)\log p\left(z_i\left|\hat{x}\right.\right)dz_i \tag{46}$$

where $\hat{x}$ is the maximum likelihood estimate of the target location in the prior target location distribution in the partially fused results.

Because the wireless sensor node i with larger entropy difference $H_i^v - H_i^s$ yields on average larger reduction in the uncertainty of the posterior target location distribution [24], the sensor selection can be considered as the following problem

$$\hat{i} = \arg\max\left(H_i^v - H_i^s\right) \tag{47}$$

which implies that the $H_i^v - H_i^s$ can be used as the predicted contribution metric $\varphi_{Predicted}$.

Furthermore, the confidence degree $\varphi_{Confidence}$ of the wireless sensor nodes can be evaluated by the Mahalanobis distance between tracking result of the wireless sensor node and the final fusion result at last instant, because Mahalanobis distance can easily determine the similarity between two probability distribution sets [25]. The confidence degree is defined as:

$$\varphi_{Confidence}\left(S^i\left(t\right)\right) = -\left(X'_{s^i(t-1)} - \hat{X}'_{t-1}\right)^T \hat{\Sigma}'^{-1}_{t-1}\left(X'_{s^i(t-1)} - \hat{X}'_{t-1}\right) \tag{48}$$

where $X'_{s^i(t)}$ is the probability distribution of sensor at instant $t-1$, $\hat{X}'_{t-1}$ and $\hat{\Sigma}'_{t-1}$ is the mean and covariance of probability distribution of final fusion result at instant $t-1$ respectively.

Then the objective function of sensor node selection during information fusion is as follow.

$$M\left(S^i\left(t\right)\right) = -\left(1-\alpha\right)\cdot\left(H^v_{s^i(t)} - H^s_{s^i(t)}\right)\cdot\left(X'_{s^i(t-1)} - \hat{X}'_{t-1}\right)^T \hat{\Sigma}'^{-1}_{t-1}\left(X'_{s^i(t-1)} - \hat{X}'_{t-1}\right) - \alpha\cdot\left(\frac{\left(l^i_c\right)^2}{l_0^2 H\left(S^i\left(t\right)\right)}\right) \tag{49}$$

where the relative weight is set to $\alpha = 0.5$ in this paper.

The criterion for finalizing fusion is evaluated by the entropy of probability distribution of target position which is defined as

$$H_p\left(X\right) = -\sum_i p\left(x_i\right)\log p\left(x_i\right) \tag{50}$$

If the entropy exceeds the predefined value, the localization is considered to be finished, then the current wireless sensor node will transmit the result back.

For multi-target tracking, multiple leader sensor nodes will be dynamically and autonomously established and trigger the distributed P2P signal processing based multi-view localization for each target. The inherent characteristic of P2P network gives many benefits on distributed multithreading signal processing, which implies that distributed P2P signal processing framework will perform better in multi-target tracking.

Because the collision often occurs in multi-target tracking, it is difficult to associate the measurements of each sensor node with individual targets. The assignment between tracks and measurements is formulated as a discrete optimization problem to maximize a dimensionless global likelihood ratio of the measurements-to-tracks associations. The nearest neighbor approach [26] is adopted to associate the measurements to tracks by determining which measurement is closest to the predicted target-originated measurement. For simplifying the computation, Mahalanobis distance is also used to determine the optimal associated results for each target:

$$\hat{i} = \arg\min_{1<i<N}\left(\left(P_{predicted} - \hat{X}_i\right)^T \hat{\Sigma}_i^{-1}\left(\left(P_{predicted} - \hat{X}_i\right)\right)\right) \tag{51}$$

where $P_{predicted}$ is the target position predicted by ARMA model, $\hat{X}_i$ and $\hat{\Sigma}_i$ are the mean and covariance of probability distribution of *i*th potential position, *N* is the number of potential position.

Actually, the multi-view localization algorithm can also be carried out in traditional centralized client/server framework and distributed client/server framework. For comparing the performance of centralized and distributed client/server frameworks and the proposed distributed P2P framework and studying the tracking performance of the combined tracking system with several specific algorithms, a real world target tracking experiment is described and the results are analyzed in the following section.

4. Simulation Results

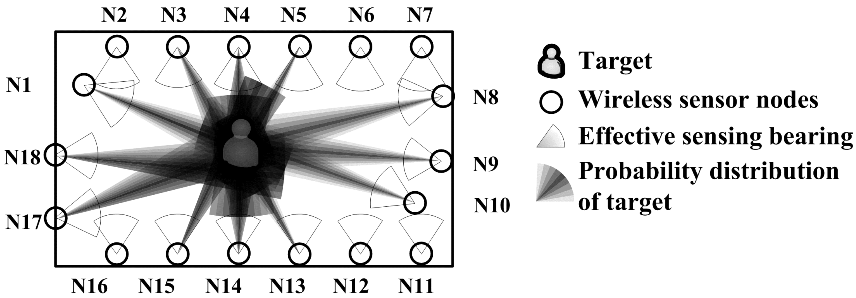
In this section, an indoor target tracking experiment is described and the tracking performance, energy consumption and communication latency of target tracking with ARMA model in a centralized client/server framework, a distributed client/server framework and a distributed P2P framework will be evaluated. The distributed mobile agent framework is ignored, because the overhead of the mobile agent is large in visual target tracking [7]. Because of the space limitation, just a small number of nodes are deployed and tested. But this scenario can also be considered as a prototype of the WSN [27].

4.1. Deployment of the Wireless Sensor Network

As illustrated in Figure 5, a wireless sensor network with 18 wireless sensor nodes is deployed in a room with a distributed P2P framework. Each wireless sensor node consists of one image/pyroelectric-infrared sensor pair which has 60° visual angle and 3.6 mm camera lens. Each wireless sensor node is working autonomously. As soon as target enters the tracking area, the correlative wireless sensor nodes are awakened by the pyroelectric-infrared sensor module and then the image acquisition, target extraction, feature extraction, target classification and target tracking are performed continuously until the target leaves.

Each node processes the input information at a frame rate of 10 Hz with video down-sampled to 160×120 pixels. Each node can locally estimate rough energy consumption and share the reserved energy information per minute. Each tracking data package is 1 kbytes and the interval is 0.1 second. It carries partially fused tracking result and a list of passed itinerary. In this experiment, the scenario of target moving is as follow: one person pushed forward a chair to the center of the room, and then walked to the other side. In some angles of views, the person and the chair was separated in the latter scenario, so it changed to multi-target tracking problem.

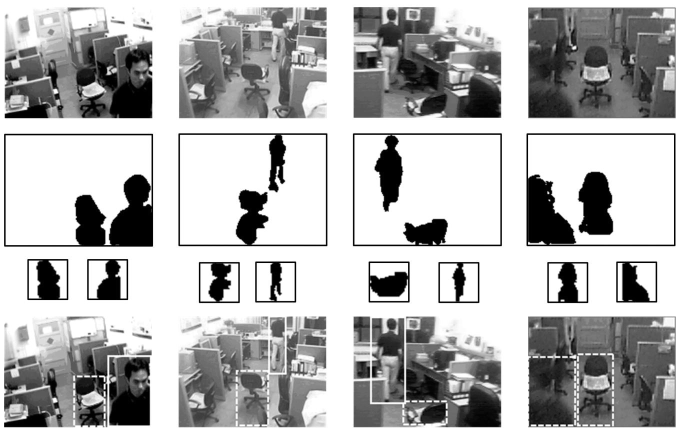
Figure 5. The setup scenario of the wireless sensor network in distributed P2P framework.



4.2. Multi-Target Tracking Results

Figure 6(a), (b), (c) and (d) respectively show the sequential signal processing results in 4 of 18 wireless sensor nodes, N1, N8, N10 and N18 as illustrated in Figure 5. Each sequential signal processing result consists of 4 intermediate results, where the first row presents the original images, the second row is the foreground target detection results, the third row shows the rescaled contour information of targets and the fourth row illustrates the target classification results.

Figure 6. Signal processing results sequences in 4 of 18 wireless sensor nodes, where the solid rectangle represents that the classification result is human type and the dashed rectangle represents that the classification result is non-human type.

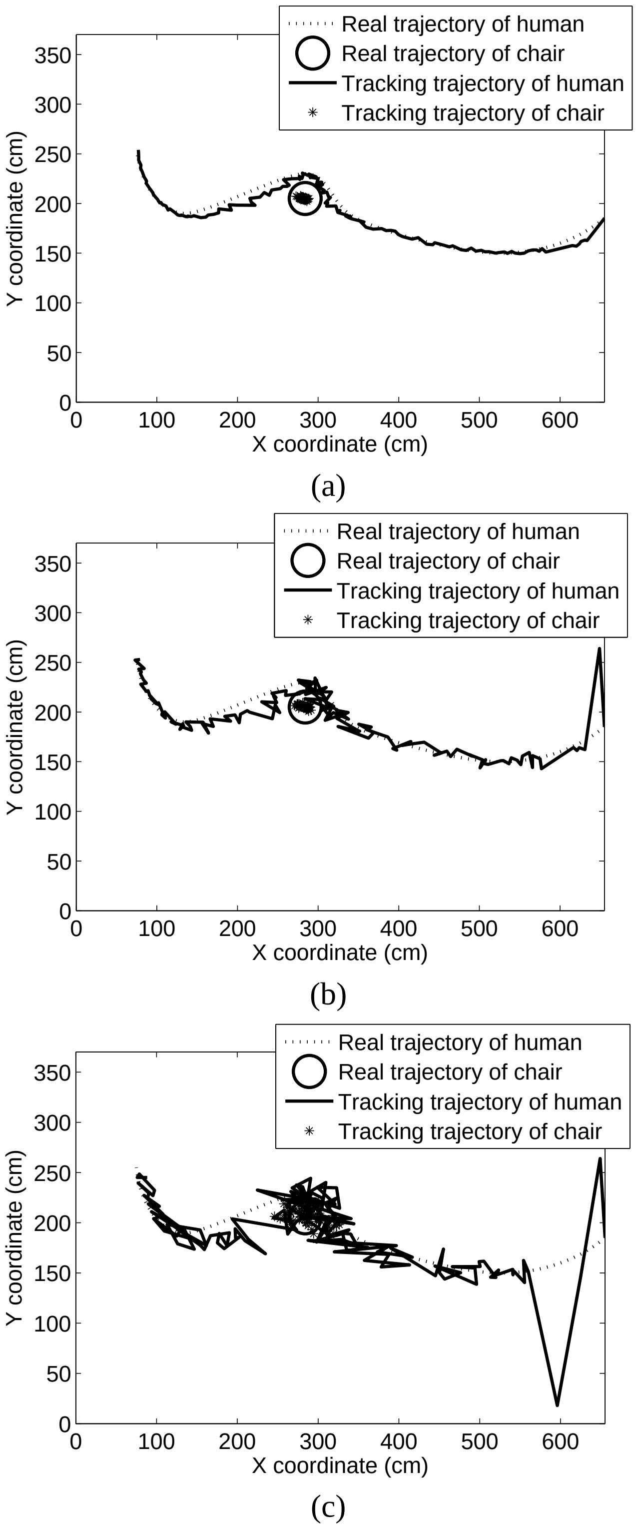


(a) (b) (c) (d)

The results illustrate that the proposed tracking system combined with background substraction algorithm and 2-D ILWT and SVM algorithm in each wireless sensor nodes can successfully achieve target detection, feature extraction and target classification, even if targets are partially blocked. Obviously, the person and chair can be clearly separated. It means that, although multi-target may involve many severe occlusions at one time, they can be automatically detected once they can be separated correctly in at least one of the wireless sensor nodes.

During target classification, a total of 5000 binary images of human and 4000 binary images of non-human, which were produced in different dates with different groups of people, are used for training, and 2000 binary images of human and 1500 binary images of non-human are used for testing. In such a extensive experiment, the classification accuracy for human and non-human are 92.4% and 95.5% respectively, although the limitation of view angles and disturbances of obstacles will also cause misclassification, i.e., in Figure 6(d).

Figure 7. Multi-target tracking results of (a) distributed P2P framework, (b) distributed client/server framework and (c) centralized client/server framework.

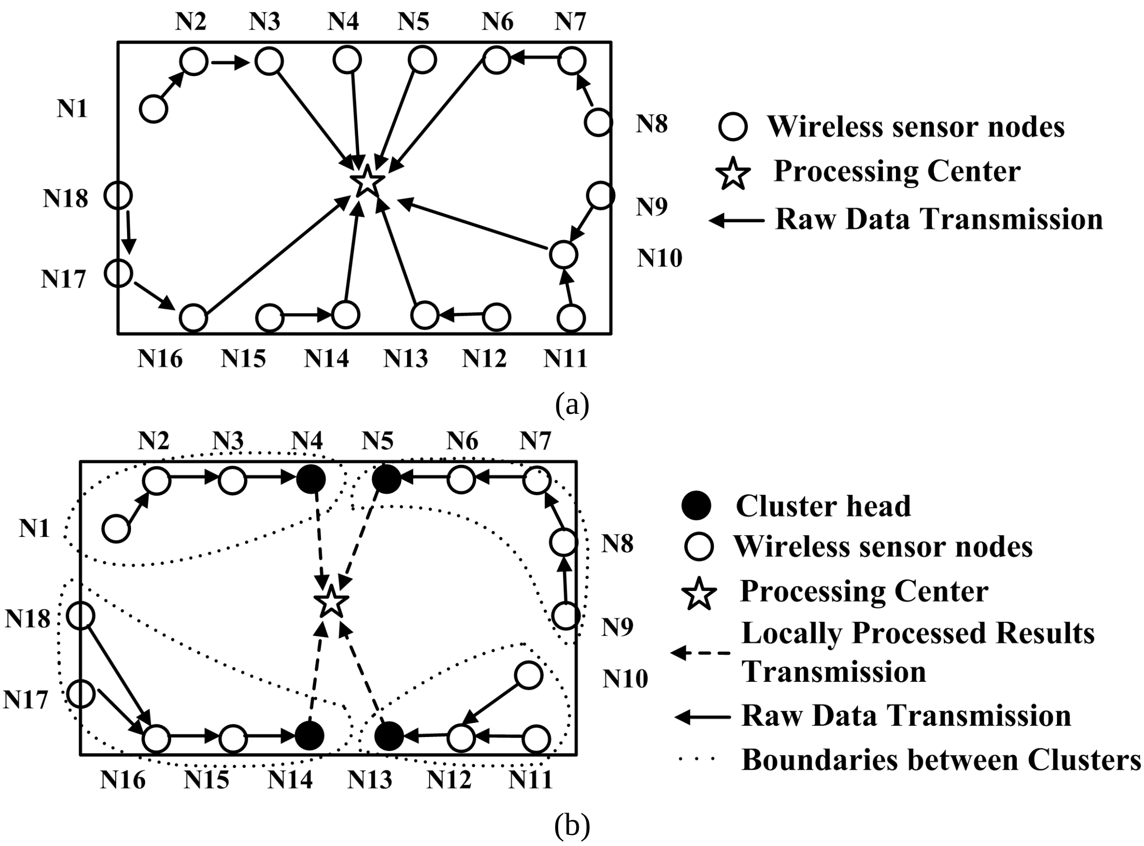


The final multi-target tracking results of the proposed tracking system based on distributed P2P framework is illustrated in Figure 7(a), where the trajectories of targets are labeled for classification. The tracking trajectories are compared to the real trajectories. The results verify that the multi-target tracking system based on distributed P2P framework can effectively realize multi-target tracking.

4.3. Performance Comparison

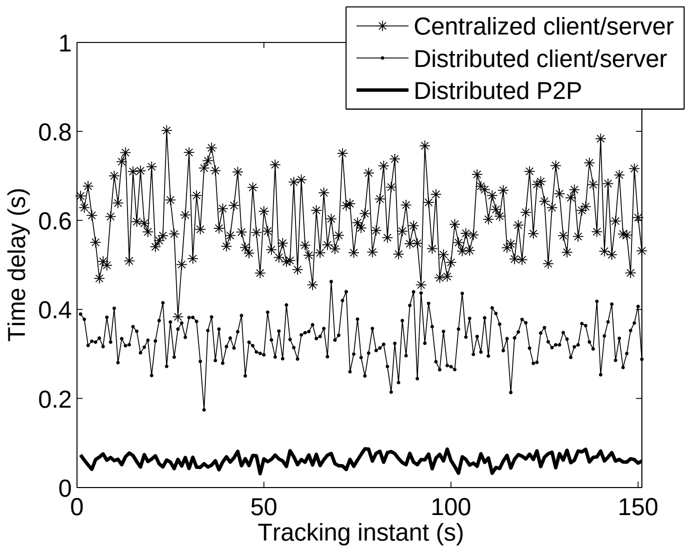
For investigating the performance of the centralized client/server framework, distributed client/server framework and distributed P2P framework, the distributed target localization algorithm is carried out in centralized client/server, distributed client/server and distributed P2P framework, respectively. The setup scenarios of the centralized client/server framework and distributed client/server framework are illustrated in Figure 8, where the wireless sensor network is divided into 4 clusters in distributed client/server framework.

Figure 8. The setup scenario of the wireless sensor network in (a) centralized client/server framework and (b) distributed client/server framework.

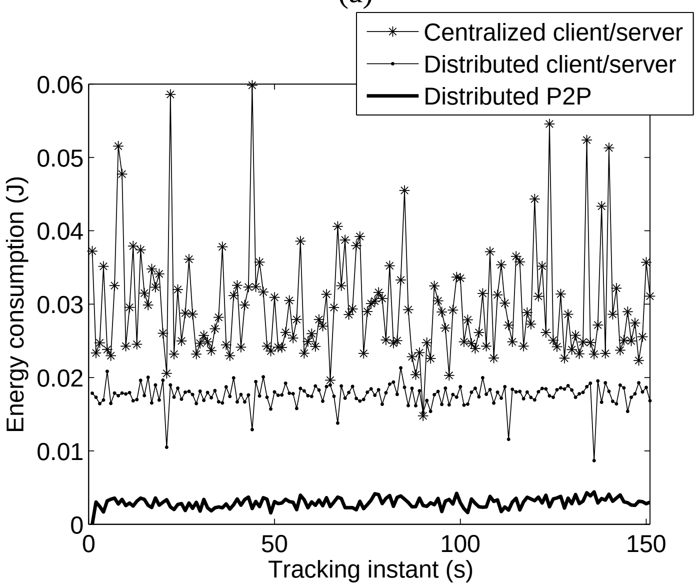


The multi-target tracking results of each framework are illustrated in Figure 7. Obviously, the performance of distributed P2P framework is much better than other two frameworks in target tracking. The reason is that raw data transmission in both client/server frameworks will shapely increase the amount of data, the confused data transmission will aggravate the congestion of wireless communication, and cause the measurements to be out-of-sequence and packet loss. Compared to the other two client/server frameworks, the distributed P2P framework dynamically chooses the proper set of wireless sensor nodes for progressive signal processing according to the current predictions of information contributions, energy consumption and lifetime of WSNs. The integrated objective function defined in Eq. (49) ensures the tradeoff between information utilities and energy consumption. Moreover, the performance of distributed client/server framework is better than the performance of centralized client/server framework, because the distributed signal processing decreases the workload and data transmission of processing center and improves the quality of wireless network service.

Figure 9. Comparison between centralized client/server framework, distributed client/server framework and distributed P2P framework in (a) time delay and (b) energy consumption.



(a)



(b)

Furthermore, the time delay and energy consumption in communication of the three mechanisms are compared for further investigation. As illustrated in Figure 9, the time delay and energy consumption of a distributed P2P framework are also much less than in a centralized client/server framework and distributed client/server framework at each time instant. The reason is that the distributed P2P framework can dynamically and sequentially carry out signal processing to attain a desired level of performance in WSNs, and ensure to exchange the least amount of data between wireless sensor nodes according to the situations of each wireless sensor nodes, such as the reserved energy, energy consumption, signal processing ability and predicted information contribution. Inversely, in two client/server frameworks, once the target is sensed by a number of sensor nodes, a significant amount of traffic is triggered, this may easily lead to congestion in the forward path, which will cause serious time delay and energy consumption.

From the comparison of tracking performance, time delay and energy consumption between distributed P2P framework, distributed client/server framework and centralized client/server framework, it is obvious that the proposed distributed P2P framework and the proposed combined tracking system based on background subtraction, 2-D ILWT, SVM, ARMA model and multi-view localization algorithms can succeed in robust multi-target tracking in WSNs.

5. Conclusions

For performing target tracking in the strictly constrained wireless sensor networks, this paper proposes a distributed P2P signal processing framework and introduces a combined target tracking system. In the distributed P2P framework, signal processing is progressively carried out in a set of selected wireless sensor nodes with an integrated criterion based on some feasible factors for achieving the tradeoff between energy consumption and information utility. The combined target tracking system consists of a series of specific in-node algorithms, such as background subtraction based target detection, ILWT and SVM based target classification, ARMA model based target tracking, and a multi-view localization algorithm based on the distributed P2P signal processing framework. Then an indoor experiment is carried out for investigating the performance of the proposed tracking system and comparing the impacts of centralized client/server framework, distributed client/server framework and distributed P2P framework in tracking, time delay and energy consumption. The experiment results demonstrate that the distributed P2P framework is an effective signal processing framework with better performance in processing, time delay and energy consumption of wireless sensor networks than centralized client/server framework and distributed client/server framework, and the proposed target tracking system based on the distributed P2P signal processing framework can be successfully achieved in strictly constrained wireless sensor networks and perform target detection, classification, tracking and localization.

Acknowledgements

This paper is sponsored National Grand Research 973 Program of China (No. 2006CB303000) and National Natural Science Foundation of China (No. 60673176, No.60373014, No. 50175056).

References and Notes

1. Meesookho, C.; Narayanan, S.; Raghavendra, C. S. Collaborative classification applications in sensor networks. *IEEE Sensor Array and Multichannel Signal Processing Workshop* **2002**, 370-374.
2. Hatzinakos, D.; Liang S. A cross-layer architecture of wireless sensor networks for target tracking. *ACM Transactions on Networking*, **2007** *15 (1)*, 145-158.
3. Yeow, W.-L.; Tham, C.-K.; Wong, W-C. Energy efficient multiple target tracking in wireless sensor networks. *IEEE Transactions on Vehicular Technology* **2007**, *56 (2)*, 918-928.
4. Hao, Q.; Brady, D. J.; Guenther, B. D.; Burchett, J.; Shankar, M.; Feller, S. Human tracking with wireless distributed pyroelectric sensors. *IEEE Sensors Journal* **2006**, *6 (6)*, 1683-1696.
5. Wang, X.; Ma, J.-J.; Wang, S.; Bi, D-.W. Prediction-based dynamic energy management in wireless sensor networks. *Sensors* **2007**, *7 (3)*, 251-266.
6. Zhao, F.; Liu, J.; Liu, J.; Guibas, L.; Reich, J. Collaborative signal and information processing: an information directed approach. *Proceedings of the IEEE, Sensor Networks and Applications* **2003**, *91 (8)*, 1100-1209.
7. Qi, H.; Iyengar, S. S.; Chakrabarty, K. Multi-resolution data integration using mobile agents in distributed sensor networks. *IEEE Trans. on Systems, Man, and Cybernetics Part C: Applications and Reviews* **2001**, *30 (3)*, 383-391.
8. Xu, Y.; Qi, H. Distributed computing paradigms for collaborative signal and information processing in sensor networks. *Journal of Parallel and Distributed Computing* **2004**, *64 (8)*, 945-959.
9. Kant, K.; Iyer, R.; Tewari, V. A framework for classifying peer-to-peer technologies. *The 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid* **2002**, 1-8.
10. Ng, T.S.E.; Chu, Y.-H.; Rao, S.G.; Sripanidkulchai, K.; Zhang, H. Measurement-based optimization techniques for bandwidth-demanding peer-to-peer systems. *IEEE INFORCOM 2003* **2003**, 2199-2209.
11. Wu, Q.; Iyengar, S. S.; Raok, NS. V.; Barhen, J.; Vaishnavi, V. K.; Qi, H.; Chakrabarty K. On computing mobile agent routes for data fusion in distributed sensor networks. *IEEE Trans. on Knowledge and Data Engineering* **2004**, *16 (6)*, 740-753.
12. Russell, J.; Jacome, M. Software power estimation and optimization for high performance, 32-bit embedded processors. *Proceedings of ICCD'98* **1998**, 328-333.
13. Haritaoglu, I.; Harwood, D.; Davis, L.S. W4: real-time surveillance of people and their activities. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **2000**, *22 (8)*, 809-830.
14. Pardo, M. B.; van der Reijden, C. T. Embedded lossy image compression based on wavelet transform. *IEEE Region 8 International Symposium on Video/Image Processing and Multimedia Communications* **2002**, 195-199.
15. Zaciu, R.; Lamba, C.; Burlacu, C.; Nicula, G. Image compression using an overcomplete discrete wavelet transform. *IEEE Trans. on Consumer Electronics* **1996**, *42 (3)*, 800-807.
16. Sweldens, W. The lifting scheme: a custom-design construction of biorthogonal wavelets, *Applied and Computational Harmonic Analysis* **1996**, *3 (2)*, 186-200.

17. Reichel, J.; Menegaz, G.; Nadenau, M. J.; Kunt, M. Integer wavelet transform for embedded lossy to lossless image compression. *IEEE Transactions on Image Processing* **2001**, *10 (3)*, 383-392.
18. Martina, M.; Masera, G.; Piccinini, G.; Vacca, F.; Zamboni, M. Embedded IWT evaluation in reconfigurable wireless sensor network. *The 9th International Conference on Electronics, Circuits and Systems* **2002**, 855-858.
19. Flouri, K.; Beferull-Lozano, B.; Tsakalides, P. Training a SVM-based classifier in distributed sensor networks. *Proc. of 14nd European Signal Processing Conference* **2006**, 1-5.
20. Van der Merwe, R.; Doucet, A.; de Freitas, N.; Wan, E. The unscented particle filter. *Technical Report*, CUED/ F-INPENG/TR 380 Department of Engineering, Cambridge University, UK **2000**.
21. Wang, X.; Wang, S.; Ma, J.-J. An improved particle filter for target tracking in sensor system. *Sensors* **2007**, *7 (1)*, 144-156.
22. Liu, W.; Farooq, M. Tracking maneuvering targets via an ARMA type filter. *Proceedings of the 33rd IEEE Conference on Decision and Control* **1994**, 3310-3311.
23. Liu, C.; Wu, K.; Tsao, M. Energy efficient information collection with the ARIMA model in wireless sensor networks. *GLOBECOM '05* **2005**, 2470-2474.
24. Wang, H.-B.; Yao, K.; Pottie, G.; Estrin, D. Entropy-based sensor selection heuristic for target localization. *Proceedings of the third international symposium on Information processing in sensor networks* **2004**, 36-45.
25. Wang, X.; Wang, Sheng Collaborative signal processing for target tracking in distributed wireless sensor networks. *Journal of Parallel and Distributed Computing* **2007**, *67 (5)*, 501-515.
26. Li, X.-R.; Bar-Shalom, Y. Tracking in clutter with nearest neighbor filters: analysis and performance. *IEEE Trans. on Aerospace and Electronic Systems* **1996**, *32 (3)*, 995-1010.
27. Wang, X.; Wang, S.; Ma, J.-J. An Improved Co-evolutionary Particle Swarm Optimization for Wireless Sensor Networks with Dynamic Deployment. *Sensors* **2007**, *7 (3)*, 354-370.