Hindawi Publishing Corporation
EURASIP Journal on Embedded Systems
Volume 2009, Article ID 352172, 9 pages
doi:10.1155/2009/352172

Research Article

Evaluation of a "Smart" Pedestrian Counting System Based on Echo State Networks

Emi Mathews[1] **and Axel Poigné**[2]

[1] *Department of Computer Science, University of Applied Sciences Bonn-Rhein-Sieg, Grantham-Allee 20, 53757 Sankt Augustin, Germany*
[2] *Knowledge Discovery Department, Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS), Schloss Birlinghoven, 53754 Sankt Augustin, Germany*

Correspondence should be addressed to Axel Poigné, axel.poigne@iais.fraunhofer.de

Received 30 August 2008; Revised 25 November 2008; Accepted 18 December 2008

Recommended by Bernhard Rinner

We have designed an inexpensive intelligent pedestrian counting system. The pedestrian counting system consists of several counters that can be connected together in a distributed fashion and communicate over the wireless channel. The motion pattern is recorded using a set of passive infrared (PIR) sensors. Each counter has one wireless sensor node that processes the PIR sensor data and transmits it to a base station. Then echo state network, a special kind of recurrent neural network, is used to predict the pedestrian count from the input pattern. The evaluation of the performance of such networks in a novel kind of application is one focus of this work. The counter gave a performance of 80.4% which is better than the commercially available low-priced pedestrian counters. The article reports the experiments we did for analyzing the counterperformance and lists the strengths and limitations of the current implementation. It will also report the preliminary test results obtained by substituting the PIR sensors with low-cost active IR distance sensors which can improve the counter performance further.

1. Introduction

A pedestrian counter has lots of applications like effective resource utilization, planning of service activities, ensuring safety and convenience, and so on. The use of pedestrian counters can be dated to few centuries back, where people used rotating gates and turnstiles for counting. With the advent of high-speed electronic devices and powerful computing, the counting techniques were automated. Most recent commercially available products use wireless data transmission for logging the people count. The software coming along with these devices can analyze the trends and patterns in the traffic and give activity profile as charts and graphs using the logged data.

The goal of this research is to design a pedestrian counter to be used, for instance, in advertising. While designing a pedestrian counter for a small private firm, we should keep in mind that the counter should be publicly usable, easily portable, available at affordable prices, and have good accuracy. The commercially available counters that perform well are those using computer vision algorithms [1]. The primary limitation of such camera-based counters is that private firms are not allowed to use them in public places. Moreover, they are very expensive and need careful control of the lighting conditions to work efficiently. We designed a counting system using off-the-shelf passive infrared sensor arrays. The PIR sensors record the motion pattern and the wireless sensor units send the sensor data to a base station for processing. The wireless sensor network enables the system to be a distributed one which is an advantage over other systems. The powerful machine learning techniques employed at the base station learn the data patterns from the noisy sensors and predict the counts. The evaluation of the pedestrian counter shows that the counter we have devised excels other counters of the same price range in performance [2].

At present, we do not yet take advantage of the distributed nature of the counting application, since we only use the communication capabilities but analyze and predict on the global level. This work is a feasible study preparing for a more distributed implementation of intelligence, in which at least some intelligent preprocessing using small echo state networks should be allocated at each counter unit and aggregated using the network.

A detailed view of the design and implementation of the "smart" pedestrian counting system can be found in our previous publication [3]. This article focuses on the evaluation of pedestrian counter system. Section 2 gives a short overview of the commercially available pedestrian counters. Section 3 briefly summarizes the system design and architecture. The brain of this smart counter is developed using a recurrent neural network. The echo state network [4], a relatively new method of training recurrent neural networks, was adopted to train the network. Section 4 describes the implementation of the echo state network and the training of the network. The analysis of the system and the evaluation of the system performance are given in Section 5.

2. Related Work

Commercially available pedestrian counters can be classified based on the sensors used for detecting motion. The most commonly used sensors for motion detection are piezoelectric sensors, microwave radar, ultrasonic sensors, infrared sensors, laser scanners, and video cameras [2]. The choice of the sensors affects the complexity and cost of the system. Some systems such as infrared barriers use simple *beam-break* principle while others use complex video processing algorithms for counting. Some counters cost few hundreds of Euros while others cost tens of thousands of Euros. The decision about which counter to use is determined by competing factors of accuracy, reliability, practicality, and cost.

Mechanical counters such as turnstiles and gate-type counters were used in old days for counting pedestrians. Due to their capability to count accurately, they are still in use with added features. Piezoelectric counters are simple and reliable counters, acknowledged as one of the most effective one in counting pedestrians [5]. Infrared counters are the most popular type of commercially available counters used in indoor settings [1]. These counters can be mounted vertically or horizontally. With the arrival of pyroelectric sensing technology that does not require expensive cooling methods, passive infrared pedestrian counters came into the market. Hashimoto et al. [6] developed a passive infrared counting system using 1-dimensional, 8-element array detectors. They placed the detector arrays 60 cm apart parallel to the moving direction and used pattern matching and data comparison algorithms for counting. Another passive infrared counter is the one manufactured by IRISYS [7] which employs a downward facing pyroelectric array of 16 × 16 format. Compared to the image processing techniques, the thermal image processing in IRISYS counter is very easy and fast because of the low resolution and binary nature of the image [8]. The moving person is mapped to a white blob formed by clusters of elements. The main disadvantage is the high selling price (4000 £ [7]) of the unit which makes it too expensive for many customers.

Laser scanners can provide highly accurate count information but are very expensive. The LD people counter (PeCo) manufactured by SICK uses double vertical laser curtain and counts people based on their height. Video cameras-based system uses image processing techniques to estimate the number of people [9]. The image processing steps can be decomposed into detecting phase, tracking phase, and interpreting phase [1]. In many cases [10], a neural network is trained to establish the nonlinearity relationship between pedestrian count and the pixels of pedestrian object. Researchers have also tried combinations of detector technologies to overcome the limitations of one technology. For example, a combination of passive infrared with ultrasonic sensors is used in ASIM infrared-ultrasonic sensor [5], where the infrared sensors detect the presence and the ultrasonic sensors measure the distance.

3. System Design and Architecture

The evaluation of the commercially available systems shows that the systems which give good performance are extremely costly and are not allowed to be used in public places and the systems that are affordable as per our customer specifications have very high error rate. Hence, we decided to design a new system that could meet these two ends. The primary consideration of the sensor to use was the PIR sensor because of its low cost. These sensors are widely available commercially and have good range of up to 12 m. We used PIR sensor units manufactured by hygrosens which costs about 13 €. For data acquisition and processing, we used wireless sensor networks [11] which help us not only to achieve our aim of making a distributed counter and a counter with wireless capabilities, but also to simplify the overall design.

We adopted an overhead mounting configuration which avoids several errors associated with the side-mounted IR counters. A linear arrangement for placing the sensor units was considered with an inter sensor unit distance of 60 cm. To get the direction of the movement of people, we need at least 2 rows of sensors. The row separation can be varied at regular intervals for testing. We chose a separation of 52 cm as optimal one and more details about the geometric configurations can be found in [3].

The pedestrian counting system consists of multiple identical counters which communicate to a base station computer or hand held device over radio channel with the aid of the wireless sensor network. The system has three main components: the hardware, the software, and the machine learning algorithm.

3.1. Hardware Architecture. The hardware consists of multiple identical counters and a base station. Each counter can cover passages of width 120 cm. Several counters can be connected together to form one unit to cover wide passages. Such units of counters can be placed at multiple entrances forming a distributed pedestrian counting system. The counter has four passive infrared sensor units and a Tmote Sky wireless sensor node. Though the counters work independently using the radio channel when they form a unit, they can be connected together using cables as shown in Figure 1. In such a unit, one sensor node is enough. This reduces the system cost considerably and also avoids the tight synchronizations overheads needed for multiple sensor nodes. The four PIR sensor units in a counter are arranged in two rows as shown in Figure 1. The PIR sensor units consist of a PIR sensor, a Fresnel lens, and a case. The Fresnel lens extends the detection range of the PIR sensors and the case selectively exposes the central segment of the Fresnel lens and thus focuses the field of view to ±5° on the horizontal and the vertical directions. The four sensor units are connected to the 8-bit quasi-bidirectional port of PCF8574, the remote I/O expander for I2C-bus. The I/O expander is interfaced to the Tmote Sky sensor node via a connector switch. This connector switch enables different counters in a unit to connect together by sharing the I2C-bus. Up to eight counters can be connected to the same I2C bus of a single Tmote node. The base station consists of a listener sensor node and a computer for processing the sensor data.

3.2. Software Architecture. The system software consists of the firmware of the beacon node and the listener node and the host software running on the base station computer.

3.2.1. The Firmware. The beacon nodes are the sensor nodes on the counter units which send the sensor data to the base station and the listener nodes are those which are attached to the base station computer, where the data is processed. The flow chart showing the working of the beacon firmware is given in Figure 2.

The top level configuration of the counter application is shown in Figure 3. After initializing components such as radio, I2C, and timer, a timer is set in the repeat mode with a time period of 125 milliseconds, which is found to be the minimum time needed to trigger the PIR sensor in determining an object. When the timer is fired, a scheduler serves the request for the bus, issued by the competing radio and I2C modules, in a round robin fashion. Once the bus is granted to I2C module, it reads the sensor data from all counters in its unit. The sensor readings are filled in a buffer to reduce the counter message rate. When the buffer containing the sensor data is full, the message packet is handed over to the radio communication module, where the header fields are added and the message is broadcasted. All Tmote sensor nodes in the vicinity will hear this message. If the current node receiving the message is not the base station, then based on the routing protocol implemented for multihopping, the message is forwarded to the base station. The Listener sensor node is programed with an application called `TOSBase` provided by the TinyOS, an open-source operating system designed for wireless embedded sensor networks. `TOSBase` acts as a simple bridge between the serial and radio links.

3.2.2. PC Host Software Architecture. The pedestrian counter host software architecture is designed based on the client/server model as depicted in Figure 4. It consists of four main parts.

(1) *The Listener interface:* the Listener is a server that acts as a proxy between the attached mote and the PC preprocessor or other client applications. It reads the data from the serial port forwarded by the Listener firmware. It separates incoming messages from different counter units based on their group IDs, adds time stamps to the data packets, and notifies all registered clients about the arrival of new packets from the serial port. The Listener also logs these unprocessed packets for offline working mode.

(2) *The PC preprocessor:* the preprocessor receives the message updates from the Listener and does the prior processing. It has the following modules:

   (a) *reader module:* this module reads messages from the TCP port in the online mode or from the files in the offline mode; it extracts the binary data from the messages and hands them over to the filter module;

   (b) *filter module:* the filter module employs a Gaussian filter or a Bayesian filter, which can be specified at run time; after filtering the sensor data, the module hands over the packet to the data logging module and server module;

   (c) *data logging and server module:* the data logging module creates a unique file and logs the filtered data packet in it; the sever module has a server that outputs each filtered packet through the TCP port (4450).

(3) *Trained neural network layer:* a trained neural network accepts the incoming packets from the TCP ports or log files and predicts the counts.

(4) *The application layer:* this is the layer where user can write his own application to display the results. It also saves the counts for future reference. The results can be shown as graphs or charts for traffic analysis.

4. Machine Learning

There are various machine learning algorithms such as concept learning, decision tree, and artificial neural networks [12]. The selection of the algorithm depends on the learning problem. The geometric configuration of the sensor units is such that each counter has two rows and each row has two sensors. The basic pattern obtained from the counter is given as follows:

$$S_{11}^{A}S_{12}^{A}\cdots S_{11}^{N}S_{12}^{N}S_{21}^{A}S_{22}^{A}\cdots S_{21}^{N}S_{22}^{N}, \qquad (1)$$

where S_{ij}^{X} is the output of the jth sensor in the ith row of the counter X. The entry and exit of a person are indicated by a sequence of 0 to 1 and 1 to 0 transitions, respectively. The motion pattern varies with the velocity, size, and point

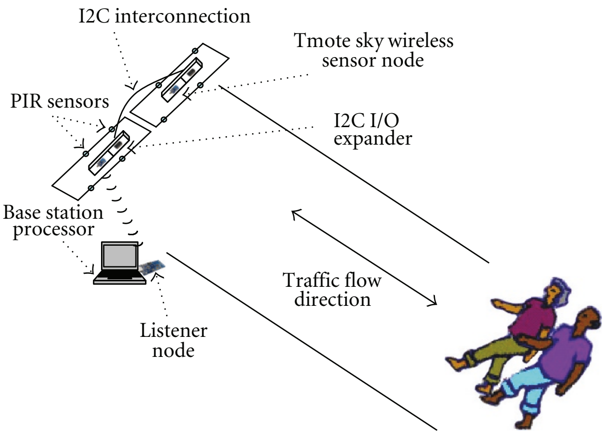

FIGURE 1: The pedestrian counter system.

of entry of the walking person. The sequence gets longer for slow walking and shorter for fast walking patterns. The sensor noise makes the problem difficult. Artificial neural networks were chosen to solve this problem as they are more effective than other machine learning algorithms especially when the sensor data is noisy and the training data contains errors.

There are two major types of multilayered artificial neural networks [13], the feed forward networks and the recurrent networks. The feed forward networks are acyclic-directed networks unlike the recurrent neural networks (RNNs). RNNs have (at least one) cyclic path of synaptic connections. In our problem of counting, we have a temporal pattern. At an instant of time, if the pedestrians were in the entry mode or in the exit mode, we could get their counts very easily. Some of the pedestrians could be in their internal states and these may last for a very long time period. We need a network that could keep the previous patterns in the memory for predicting the count. Recurrent neural networks can easily implement dynamical systems by storing the old values but the practical difficulties of implementing the networks and their algorithmic complexities hinder their use [14]. The echo state networks (ESNs) are recurrent neural networks where the training is made easier and faster by adopting certain strategies for training the weights of the network. Hence, we used echo state network for our pattern learning task.

4.1. Echo State Networks. Echo state networks are recurrent neural networks with echo state property and can be trained easily, as only the output weights need to be trained [4]. A discrete time echo state network can be described as a graph with three sets of nodes, namely, K input units, N internal network units, and L output units [13]. The interconnect edges are represented by weights $w_{ij} \in IR$, which are collected in adjacency matrices, such that $w_{ij} \neq 0$ implies that there is an edge from node $j \rightarrow i$. The real-valued connection weights are collected in an $N \times K$ input weight matrix W^{in}, $N \times N$ internal connection matrix W, $L \times (K + N + L)$ matrix W^{out} for the connections to the output units, and $N \times L$ matrix W^{back} for the connections that project back from the output to the internal units [4]. Connections directly from the input to the output units, connections between output units, and recurrent pathways between internal units are allowed.

The ESN which we used has 8 input units that read the data from the 8 sensor units. We used a simple ESN structure where there are no direct output feedback connections. Hence, the output matrix W^{out} is $L \times (K + N)$ matrix instead of $L \times (K + N + L)$ matrix. We have chosen 8 output units, four of them to represent the entry detector sensor *En* and the rest for the exit detector sensor *Ex*. The number of internal units N is selected based on the length T of the training data and the difficulty of the task. N should not exceed an order of magnitude of $T/10$ to $T/2$ to avoid over fitting [4]. The database used for training had 6000 training data and we chose N to be 1000 units. Hence, the internal weight matrix is a 1000×1000 sparse matrix.

The activation of internal units is updated according to [4]

$$x(n+1) = f\left(W^{\text{in}} \cdot u(n+1) + W \cdot x(n) + W^{\text{back}} \cdot y(n)\right), \quad (2)$$

where $f = (f_1, \ldots, f_N)$ are the output activation functions (typically sigmoid functions) of the internal units. Calculation of this new internal node vector from the current inputs, given old activation and old output according to (2), is called evaluation. The neural network computes its output activations according to

$$y(n+1) = f^{\text{out}}\left(W^{\text{out}}(u(n+1), x(n+1), y(n))\right), \quad (3)$$

where $f^{\text{out}} = (f_1^{\text{out}}, \ldots, f_L^{\text{out}})$ are the output activation functions and $(u(n+1), x(n+1), y(n))$ denote concatenation of input, internal, and previous output activation vectors. Hoevere, (3) is called exploitation [4].

In order for the ESN principle to work, the internal units must have the echo state property (ESP). An ESN with ESP can be generated by following the steps given as follows [4]:

(1) randomly generate the sparse internal weight matrix;

(2) normalize the weight matrix with the maximum absolute Eigen value;

(3) scale the weight matrix with the spectral radius α.

The echo state property will be there for this network $(W_{\text{in}}, W, W_{\text{back}})$ regardless of the choice of W_{in} or W_{back}.

4.2. System Modeling. In order to generate various motion patterns, a simulator was designed using Simulink and Virtual Reality Toolbox. The virtual world model of the pedestrian counter, created using the Virtual Reality Modelling Language (VRML), consists of 8 sensors (the spheres), placed in two rows of 4 sensors each. This model can be considered as two real counters wired together to cover a wider area. Each sphere has 60 cm diameter which maps to the sensor placement distance of 60 cm. A rectangle of size 60 cm × 25 cm is used to denote the human width and thickness. The simulator has 4 subunits, the pose sequence

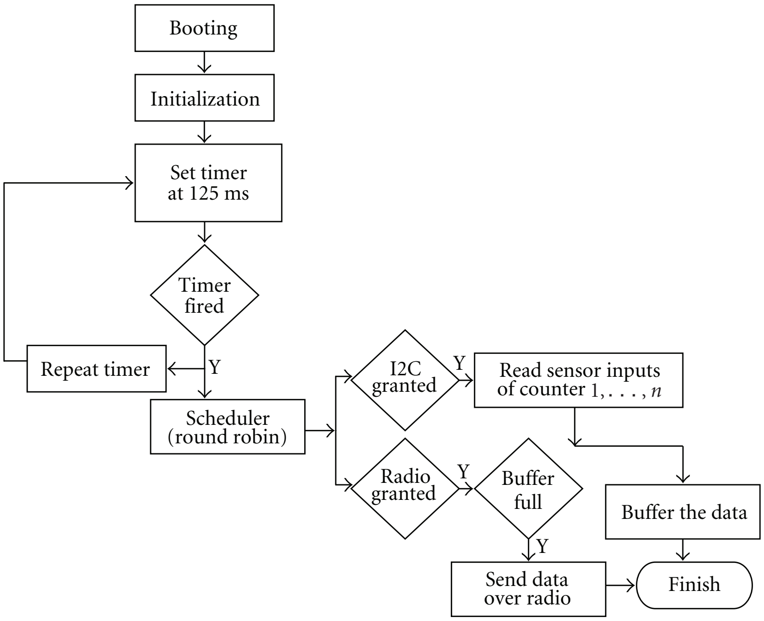

Figure 2: Flow chart showing the control sequences of firmware module.

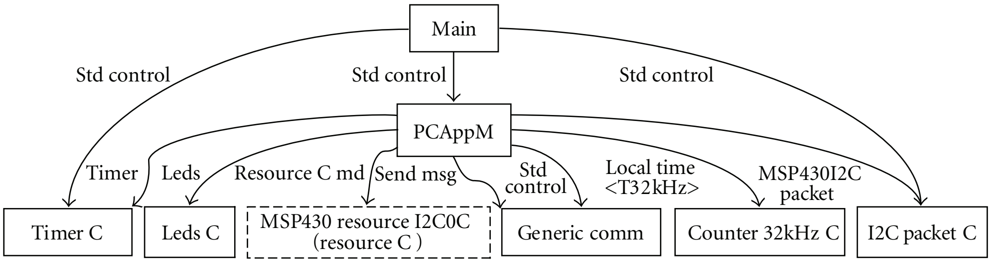

Figure 3: TinyOS top level configuration of pedestrian counting system.

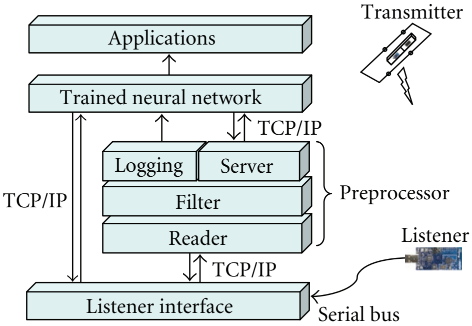

Figure 4: Host software architecture of the pedestrian counter.

generator, VR sink, detector, and output unit. The pose sequence generator generates pedestrians' positions. The VR sink block loads the virtual world model and simulates the pedestrian movements. The detector module implements the 8 sensors. It outputs 1 if the pedestrian is within the boundary of the sensors, otherwise outputs 0. In addition to these 8 sensors, it implements an entry detector, an exit detector, and a sum counter which indicate the number of pedestrians entering, exiting, and currently present in the counter area at a particular time step. The output unit receives the eleven outputs from the detector module and sends them to the display module for visualization and to the file module for logging.

4.3. *Training the Echo State Network.*

Echo state networks can be trained easily as only output weights (W^{out}) need to be trained. To train the network, we created a database having various motion patterns using the simulator. The simulated groups of people move under the counter in different directions with different velocities. The spacing among these people is also varied. The database has 11 fields corresponding to the eight PIR sensor outputs and three (*En*, *Ex*, and *T*) special sensor outputs. The training of the output weights is done by using the least square method. The steps for training the ESN [4] are given as follows.

(1) First, we initialize the network state arbitrarily to zero, $x(0) = 0$, and then drive the network with the training data for time $n = 0$ to 6000 by presenting the

teacher input $[Pir1(n)\ Pir2(n) \cdots Pir8(n)]$, and by teacher-forcing the teacher output $[En(n-1)\ Ex(n-1)]$. At time $n = 0$, where $En(n)$ and $Ex(n)$ are not defined, $En(n) = 0$ and $Ex(n) = 0$ are used.

(2) The first "nForgetPoints" specified by an initial washout time T_0, say 100, are deleted, as these states could not be relied due to the initial transients. For $T \geq T_0$, we collect the network state $x(n)$ as a new row into a state collecting matrix $M_{(T-T_0+1)\times(K+N+L)}$. This matrix is the concatenation of vectors $(u^T_{\text{teach}}(n+1), x^T(n+1), y^T_{\text{teach}})$ in rows.

(3) Similarly, for $T \geq T_0$, the sigmoid-inverted teacher outputs $\tanh^{-1}(y^T_{\text{teach}}(n))$ are collected rowwise into a teacher collection matrix $C_{(T-T_0+1)\times L}$.

(4) Finally, the output weights are computed by multiplying the pseudoinverse of M with C, $W' = M^{-1}C$, and transposing it, that is, $W^{\text{out}} = W'^{T}$.

4.4. Testing the Network. Now, the network ($W^{\text{in}}, W, W^{\text{back}}$ and W^{out}) is ready for use. We created another database of motion patterns using the same source for testing the network. This database has about 4000 data entries. The test data was given to the trained ESN and the performance of the network was evaluated. We also tried different ESNs with 700–1500 internal units and repeated the generation, initialization, training, and testing steps. We selected the ESN with 1000 units that gave the best performance among them.

5. System Analysis and Evaluation

We will now analyze the counter performance in the simulated environment and in the real environment. In the simulated environment, we are checking the ESN performance mainly. In the real environment, we conduct a number of benchmarking experiments which point out the strengths and limitations of the counting system.

5.1. Echo State Network Performance. The database used for training the ESN had motion patterns of 565 pedestrians and for testing had 465 pedestrians. The ESN outputs are given as follows:

$$\begin{aligned} &\text{Pedestrians Entered} = 476 \\ &\text{Pedestrians Exited} = 442 \\ &\text{Average} = 459 \\ &\text{Performance} = 99\%. \end{aligned} \tag{4}$$

The ESN gave a performance of 99%. To verify the performance of the ESN further, we reversed the training and the test dataset. The performance of the ESN was equally good. Then we added noise to the train and test database and repeated the tests. The network performance was 93% for first test dataset and 84% for second dataset. When we increased the level of noise, the performance became worse. From these tests, it was observed that there is a considerable degradation in the performance of ESN with the addition of noise.

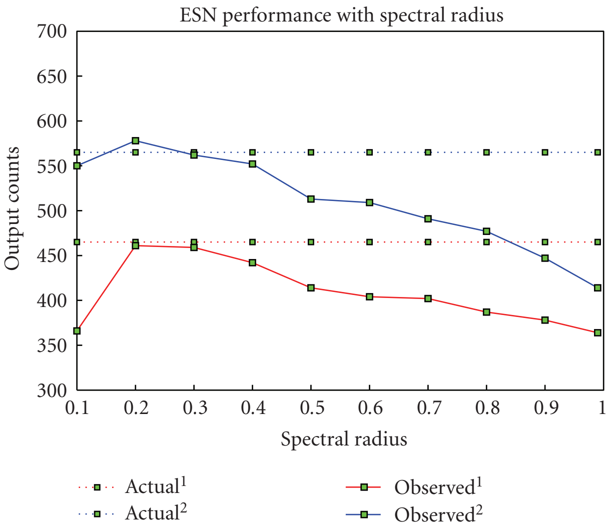

FIGURE 5: Echo state network performance with variations of spectral radius.

Another test was to find out the optimal value for the spectral radius α. It is a crucial parameter that affects the model performance [4]. It is small for the fast teacher dynamics and large for the slow teacher dynamics. Figure 5 shows the variation of the system performance with spectral radius. The solid red line indicates the average observed count outputs and the dotted red line indicates the actual counts at various α values. The blue lines indicate the results for another test database. From these two tests, it was found that the system performed well at $\alpha = 0.3$. Hence, this value was chosen as the spectral radius for the network.

A third test was conducted to evaluate the performance of echo state network with sensor geometry variations. In this test, the counter row distances were varied at 30 cm intervals. We considered six different counter environments. In the first setup, the detection zones of two rows overlap (−30 cm) while in the last the zones are separated by a distance of 120 cm. The same pedestrians move under these simulated environments to generate the train and test dataset. The results show that the errors are almost the same for all geometric configurations. Hence, we infer that even if we vary the sensor geometry, the same performance can be obtained, provided a new dataset is collected and the network is trained for that set.

5.2. Pedestrian Counter Performance. Two prototype counter systems were developed and installed at a corridor in Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS). Various test movements were repeatedly conducted using a set of people in order to evaluate the performance of the counter. The tests can be categorized as follows:

(i) *Test 1:* one person moves under the counter in both directions several times such that he is detected by all PIR sensors at least once.

(ii) *Test 2:* repeat the test 1, but with two pedestrians. They are positioned one behind the other very closely.

(iii) *Test 3:* the two pedestrians go in opposite directions, trying to enter the counter area at the same time and then diverge.

(iv) *Test 4:* this is a combination of tests 2 and 3 with four pedestrians, where each pair moves in opposite directions.

(v) *Test 5:* the pedestrians are allowed to walk freely.

Figure 6 shows the results of these tests. The simple pattern such as walking in a single file, as adopted for test 1, gave almost perfect results. When the pedestrians walked close together, as in test 2 and test 4, the performance was 85% and 64%, respectively. The last test, where the pedestrians walked freely gave 84% result. Though this can be considered as the general performance of the pedestrian counter, if we calculate the overall performance, from the ratio of total pedestrians counted by the counter to the actual value, we arrive at 80.4%.

We conducted another test to check the system performance with the sensor sampling frequency variation. A single PIR sensor unit was used for this test and the sensor output was sampled at various frequencies from 40 Hz (25 milliseconds) to 0.5 Hz (2 seconds). We found that the system's response becomes slower and misses fast moving pedestrians at lower frequencies, but the sensor noise becomes lesser at these frequencies.

5.3. Analysis of Pedestrian Counter Performance. When we analyze the test results, we observe that the more the pedestrians are separated, the better the performance is. When the pedestrians are very close together, the counter has problems in distinguishing them due to the low sampling rate of the PIR sensors. This leads to undercounting. To analyze the errors further, a table showing the sample counter output pattern when one person moves under the counter *B* is given as follows.

We expect that the PIR sensor outputs a continuous low value when there is no motion and high value when it detects motion. Due to the sensor noise, the output switches to low value even if there is motion as marked by the gray regions in Table 1. We also observe other errors as indicated by the blue and purple colors. The chief sources of errors can be classified as follows.

Sensor errors: the sensor noise as explained above is prominent for very slow movements as the pattern gets longer for slow movements. The sensor sensitivity varies with changes in the environmental temperature and this causes variations in the detection area from the predicted perfect conical geometry.

System configuration errors: geometrical configuration errors occur due to misalignment of the sensors or irregularities in the sensor casing. The irregularities of the sensor cases sometimes expose more zones of the Fresnel lens and sometimes fail to expose the required central zone. Though the pattern length varies with velocities, the configuration errors make these variations nonuniform.

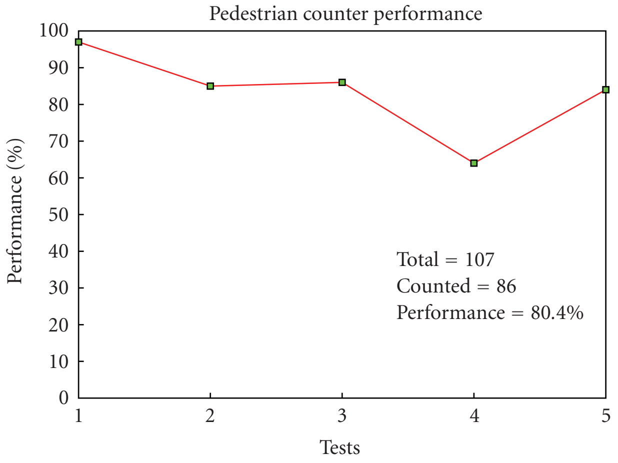

Figure 6: Pedestrian counter performance.

Table 1: Sample data sequence from pedestrian counter.

S_{11}^A	S_{12}^A	S_{11}^B	S_{12}^B	S_{21}^A	S_{22}^A	S_{21}^B	S_{22}^B
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0

Modelling error: there may be many unpredicted motion patterns occurring in the real world which were not modelled by the simulator leading to modelling errors.

ESN configuration error: the network we have chosen may not be the most optimal one. A wrong choice of network tuning parameters or number of internal neurons affects the system performance badly.

Other errors: other sources of errors include unexpected pedestrian size, unexpected motion pattern (e.g., zigzag pattern underneath the counter), pedestrians stopping exactly under the counter and executing different actions, passing of animals or luggage having a temperature different from the surrounding temperature, and so on.

5.4. Limitations of the Pedestrian Counter. We have seen various sources of errors, some of these errors could not be handled by any pedestrian counter, particularly those explained under other errors. Some errors such as sensor noise, geometric errors, and ESN error are particular to our counter. The echo state network was able to handle many unpredicted patterns but the overall system performance should be improved. The form factor of the counter is rather big which causes difficulties in transportation. Currently, the prototype counters have a fixed length of 1.2 meters which should be changed to an adjustable configuration to fit the complete width of the passages.

5.5. Suggestions for Improving the Counter Performance. We have seen that the PIR sensor noise is very high and it affects the system performance very badly. The minimum trigger time of PIR sensor is 125 milliseconds which is rather slow. If we could sample at a higher rate, we could reduce system noise. Hence, we searched for a substitute for the PIR sensor and found an active IR sensor from Sharp Corporation, Ill, USA. The most interesting series are the recent updates GP2Y3A003K0F and GP2Y0A700K. The sensor calculates the distance from the angle of the reflected light. The results of the preliminary tests show that the distance information is quite accurate. Due to its high sampling frequency of 40 Hz, it can clearly differentiate even small distance of separation like 10 cm. The low-priced sensor GP2Y0A700K has more or less the same price as that of the PIR sensors. Hence, by using this sensor, the system cost is not increased further. These sensors overcome many limitations of PIR sensors like low sample rate, inability to detect stationary persons, sensor noise, and so on. It also reduces the form factor of the counter. In addition to these advantages, the range information helps the system to predict the count more accurately.

6. Conclusion

We have developed a sc alable, publicly usable, easy to deploy, and low cost pedestrian counting system that has reasonably good accuracy. The use of passive infrared sensors helped to achieve the objectives such as public usability and low cost. The overall hardware cost is less than 200 €. The counter works in distributed mode with wireless communication facilities. We have implemented and trained an echo state network. From the performance analysis, it is found that this recurrent neural network is very successful in learning the various motion patterns, as it gave a performance of 99% while testing. In the real-world experiment, the counter gave a performance of 80.4% in spite of several limitations of the PIR sensor. This result is promising as it is better than the commercially available low-priced pedestrian counters and a system redesigned with the new active IR sensor will certainly improve the performance without increasing the price further.

The contribution of this work is not just limited to the development the pedestrian counting system. The modular layered software framework designed for processing the sensor data can be used for many other applications in the wireless sensor network field. The lowest layer, the base station server, can be used as a tool for data acquisition and extraction from the sensor network. The applications developed for interfacing external sensors to the Tmote Sky sensor nodes using ADC and I2C bus are useful contributions to the TinyOs community. The neural network layer of the software platform eases the testing of trained neural network, by providing client-server connectivity to the neural network implementation. The software developed for the training of the echo state network eases the training procedure and can be added to the ESN package.

7. Future Work

The system is to be redesigned using the new active infrared sensors from Sharp, Ill, USA. From the preliminary experiments, we expect that we can apply the same framework. Further, the simulator used to generate training examples needs more sophistication. Another task would be to acquire the counting results in real time (e.g., by porting the software to hand held devices like PDAs). Furthermore, the use of filtering and compression algorithms prior to sending messages increases the efficiency of communication. We need efficient networking protocols for ensuring proper quality of service when the counting system works in the distributed mode, that is, the counter units are placed at multiple locations.

As indicated in the introduction, the most interesting question is whether more intelligence can be implemented on the level of the sensor nodes of the counting units. So far the prospective answers are ambivalent. Though the predictive power of echo state networks when restricted to a single counter unit is very good, and some encouraging experiments have been conducted with regard to strategies to avoid overcounting at the adjoining counter areas, the computational needs of such networks still exceed the capabilities of small sensor nodes.

References

[1] B. Fanping, G.-R. Ryan, C. D. Mara, and R. David, “Estimating pedestrian accident exposure: automated pedestrian counting devices report,” Tech. Rep., UC Berkeley Traffic Safety Center, Berkeley, Calif, USA, March 2007.

[2] C. Ching-Yao and B. Fanping, “Literature review of pedestrian detection technologies and sensor survey,” Tech. Rep., Institute of Transportation Studies, UC Berkeley, Berkeley, Calif, USA, April 2005.

[3] E. Mathews and A. Poigné, “An echo state network based pedestrian counting system using wireless sensor networks,” in *Proceedings of the 6th Workshop on Intelligent Solutions in Embedded Systems (WISES ’08)*, pp. 1–14, Regensburg, Germany, July 2008.

[4] H. Jäger, “The echo state approach to analysing and training recurrent neural networks,” GMD Report 148, German National Research Center for Information Technology, Berlin, Germany, 2001.

[5] M. Peter, T. F. Yuqi, and W. Xiaodong, “Detector technology evaluation,” Tech. Rep., Department of Civil and Environmental Engineering, Traffic Lab, University of Utah, Salt Lake City, Utah, USA, November 2003.

[6] K. Hashimoto, K. Morinaka, N. Yoshiike, C. Kawaguchi, and S. Matsueda, “People count system using multi-sensing application,” in *Proceedings of the International Conference on Solid-State Sensors and Actuators*, vol. 2, pp. 1291–1294, Chicago, Ill, USA, June 1997.

[7] S. Porter, M. Mansi, N. Sumpter, and L. Galloway, “Low cost smart sensors using infrared detector arrays,” *Sensor Review*, vol. 21, no. 4, pp. 283–287, 2001.

[8] J. Kerridge, A. Armitage, D. Binnie, L. Lei, and N. Sumpter, "Using low-cost infrared detectors to monitor movement of pedestrians: initial findings," *Transportation Research Record*, vol. 1878, no. 1, pp. 11–18, 2004.

[9] J.-W. Kim, K.-S. Choi, B.-D. Choi, J.-Y. Lee, and S.-J. Ko, "Real-time system for counting the number of passing people using a single camera," in *Proceedings of the 25th Symposium of the German Association for Pattern Recognition (DAGM '03)*, pp. 466–473, Magdeburg, Germany, September 2003.

[10] T. Schlögl, B. Wachmann, W. Kropatsch, and H. Bischof, "Evaluation of people counting systems," in *Proceedings of the 25th Workshop of the Austrian Association for Pattern Recognition (AAPR '01)*, pp. 49–53, Kusadasi, Turkey, June 2001.

[11] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," *ACM SIGPLAN Notices*, vol. 35, no. 11, pp. 93–104, 2000.

[12] T. Mitchell, *Machine Learning*, McGraw-Hill, New York, NY, USA, March 1997.

[13] H. Jäger, "Tutorial on trainig recurrent neural networks covering bppt, rtrl, ekf and the echo state network approach," Tech. Rep., GMD Forschungszentrum Informationstechnik GmbH, Sankt Augustin, Germany, 2002.

[14] P. G. Plöger, A. Arghir, T. Günther, and R. Hosseiny, "Echo state networks for mobile robot modeling and control," in *Proceedings of the 7th RoboCup International Competitions and Conferences (RoboCup '03)*, pp. 157–168, Padua, Italy, July 2003.