Tracking a Vehicle Moving in a Wireless Sensor Network

Fernand S. Cohen, Salah Abushariefeh, Gregory Bruton,
Marcus Matthews, and Kuriakose Varghese

Department of Electrical and Computer Engineering
DREXEL UNIVERSITY, Philadelphia, PA 19104, USA
fscohen@coe.drexel.edu

Abstract. This paper deals with the problem of tracking a radio controlled vehicle (RCV) fitted with LED's moving within a wireless sensor network (WSN). The WSN test bed is made up of a number of Mica2 motes with fixed and know locations and equipped with light sensors. To reduce the overload of communications, power consumption, and allow for scalability, the network is divided into clusters, where members of each cluster communicate with the designated cluster head, which in turn communicates with a sink or another cluster head. When the RCV moves within the motes, a mote in the immediate vicinity of the vehicle becomes 'hot', i.e., its intensity exceeds a predetermined threshold, and it relays to its cluster head its ID and the intensity of its reading. The cluster head, in turns, establishes the location of 'hottest' mote, and uses this information to dynamically update the new location of the vehicle along with its velocity and acceleration.

Keywords: Wireless sensor networks, target detection, target tracking, motion update, cluster head.

1 Introduction

The need and deployment of large scale wireless sensor networks (WSN) is imminent for environmental monitoring, disaster relief, and for home land security applications in surveillance, in monitoring for nuclear, biological and chemical hazard, reconnaissance, and/or attack [1]. A common theme to all these situations is the rapid and reliable tracking over of a phenomenon (e.g., a fire spreading over California, monitoring of crop infestation, or monitoring a threatening moving target, etc.) over time and space in such a WSN environment. This WSN environment presents its own challenges ranging from the sheer number of motes; the mote density; the fact that sensor motes are prone to failure; the fact that sensor motes are limited in power, computational capacity, and memory. These factors present a major hindrance towards a rapid and reliable decision and early monitoring and tracking of a phenomenon in the vicinity of such WSN [2-5].

This paper deals with the problem of rapid and reliable tracking and ultimately prediction of the phenomena as it propagates amongst the sensor modes or motes over time. This is demonstrated by considering a test bed, which consists of a moving

J. Zheng et al. (Eds.): ADHOCNETS 2009, LNICST 28, pp. 438–449, 2010.

remote control toy vehicle in a lab setting where a large number of sensor network motes at fixed positions with known coordinates.

The first issue which needed to be addressed is nodal communication. WSN can be comprised of thousands of motes spaced over a span of miles. It isn't efficient, for such a large scale, to have each individual mote talking directly to a sink. There would result in an overloaded traffic causing dropped and missed messages and bring to a halt any attempt of fast and reliable monitoring and tracking. It is best to group the motes into clusters and to have designated cluster heads communicating with the sink or themselves (allowing scalability). Members of a cluster communicate with their cluster head only. LEACH [6], TEEN [7], Energy Aware Routing for Cluster-Based Sensor Networks [8], and self-organizing protocol [9] are examples of hierarchical protocols using cluster heads organization. In such protocols, motes are grouped into clusters and a "cluster-head" is selected, sometimes randomly and sometimes as a specialized mote with less power constraints. The cluster-head is responsible for aggregating the cluster data, doing some high level processing and delivering it to the sink or to another cluster head. Further research needs to be performed in order to minimize the overhead when forming clusters, to optimize cluster-head communication, and to integrate data and decision and fuse them with cluster formation.

The paper is organized as follows. Section 2 introduces the mote topology and the communication between them and the sink and cluster heads. Section 3 deals with the designation of 'hot' motes as a function of proximity to target and with mote resolution. In section 4, we show how we update the vehicle position over time within a cluster and from cluster to cluster. Section 5 shows in details the test bed experiment with a moving vehicle topped with a wing with attached LEDs that shines light as it moves. Conclusions and discussions are given in section 6.

2 Mote Topology and Communication

We assume that the topology of the motes is fixed with the coordinates of each mote in the cluster being either known or has been determined before the tracking takes place. The determination of the mote coordinates could be known if a GPS system is in place or if a trilateration mechanism of the set in place where the cluster head or anchor points (with known coordinates) are used to calculate the coordinates of the other motes.

It is very prohibitive for all of the motes to have direct communication to the sink, which receives the data sent from the network, while performing real time tracking. Great distances between the motes and the sink, and/or the shear number of motes that can be in these networks make direct communication ineffective and problematic. In addition, there would be so much traffic causing dropped and missed messages making the monitoring of these messages inefficient and hard. Therefore the motes are grouped into clusters with designated cluster heads, which communicate with the sink or with each other cluster heads in their vicinity as shown in Figure 1.

Members of a cluster communicate with their cluster head only. This allows scalability resulting into hundreds of clusters instead of thousands of motes communicating with the sink or themselves, in addition to a processing load that is limited to the active cluster heads. Further speed can be achieved by having messages be transmitted in a multi-hop fashion from cluster head to cluster head until it reaches the sink (see Figure. 2). In the context of our tracking problem, the type of information that is needed to be communicated between cluster heads and to/from sink are the current estimated position of the vehicle, its current velocity and acceleration. This information is passed on to the appropriate cluster head where the vehicle is moving in at this point of time. The sink also alerts the neighboring cluster to wake up as the predicted motion moving towards that new cluster. This has the effect of keeping the clusters that are not in the immediate vicinity of the moving vehicle to go to sleep, leading to saving in battery power of both the motes and the cluster heads. We can also have the cluster head alert its contiguous appropriate neighbor when it predicts the vehicle motion to enter the contiguous cluster neighbor.

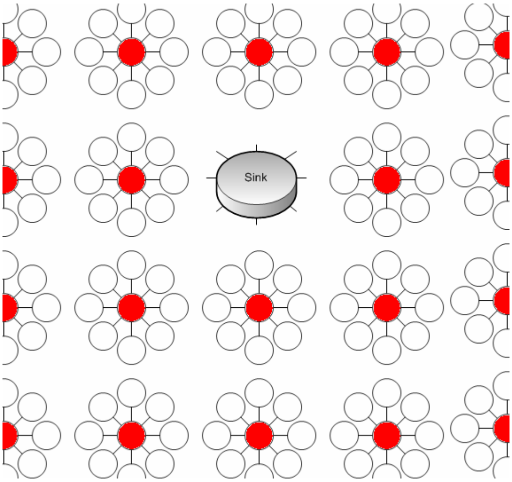


Fig. 1. Grouping the motes into clusters

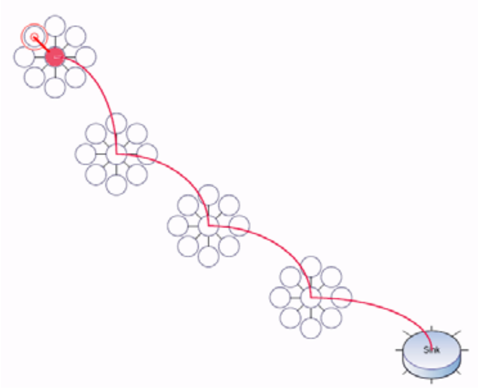


Fig. 2. Multi-hop communication using cluster heads

3 Hot Motes and Network Resolution

The first and important step in the tracking problem is to establish a resolution cell around each mote. This resolution cell triggers a mote to a ‘hot’ state, ushering the presence of the target in its vicinity. This hot state is determined by the value of the ‘target’ sensor used. The rule of thumb is an increase in the measured sensor value as a function of the proximity of the target to the mote, i.e., the closer the target to the mote the higher the sensed value. In our test bed experiment, the sensor used is a light sensor on the sensor board (MTS-300) mounted on top of the Mica2 mote functioning as the detecting unit, whereas LED’s mounted on the vehicle is used as light source. The hardware components used are given in Figure 3. The used platform and software are TinyOS and NesC language [10].

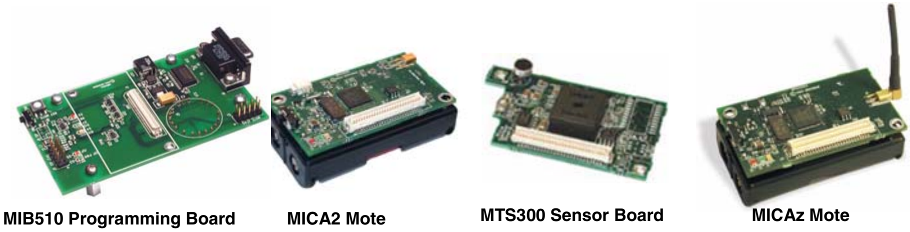
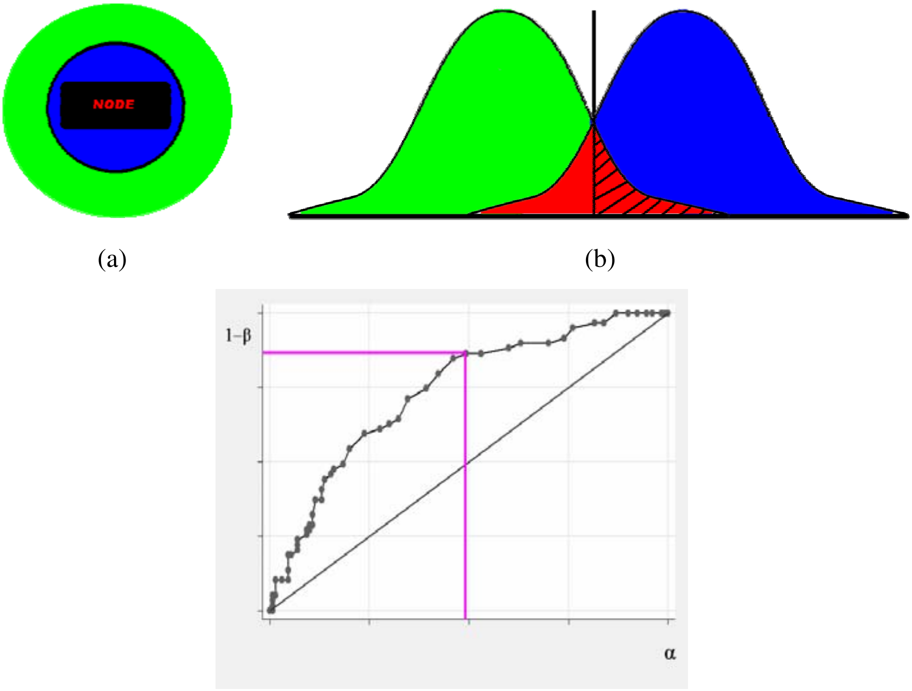


Fig. 3. Hardware used for the experiments (Crossbow Technology Inc. [11])

3.1 Hot Mote

A mote is declared to be in a ‘hot’ state if the light sensed by its light sensor exceeds a given threshold. In this section we show how we determine that threshold under a given lighting condition. For a given lighting condition and a chosen resolution, which is determined by the topology and distribution of the sensor in the WSN, we designate a tracking circle (Figure 4a) centered at the mote with a radius that determines the detecting resolution. We sample the sensor intensity inside and outside of it. The set of returns inside the circle constitutes the distribution of ‘hot’ values, whereas the returns outside the circle are the cold (no target) values. From these two sets we obtain intensity histograms (Figure 4b) of tracking and no tracking region, from which an empirical ROC curve is obtained (Figure 4c), which basically plots the false alarm probability α against the prediction probability 1-β. For each (α, 1-β) there is exists a threshold for a mote to be declared as ‘hot’, i.e., sensing the target. This also determines the false alarm rate at which the target is declared. A mote that becomes ‘hot’ sends its ID and intensity value to its cluster head. The ID determines the position of that mote in the WSN; whereas the intensity value, which exceeds the threshold, is used to update the position of the tracked vehicle in a mechanism that the cluster head uses to weighs between the position of the new declared ‘hot’ mote with its predicted target position computed from the estimated target velocity and acceleration and position of the previously found ‘hot’ target. This update, which is based on the strength of the intensity of the light sensed by the ‘hot’ mote, is described in the next section.



(a) (b)

(c)

Fig. 4. Hot mote region and associated threshold on the ROC curve

4 Updating the Target Location over Time

In this section we describe the process that the cluster head follows after receiving a message from the ‘hot’ mote in its own neighborhood to update its estimate of the location of the target at a new time t+dt, given the target location, its velocity and acceleration all at time t. If there exists more than one mote that is ‘hot’ in the neighborhood, the cluster head chooses the mote with the highest light intensity value return as the observed target position at time t+dt, and uses it along the predicted position based on the history of the target movement estimated from previous times to update the position of the target. Let

- **n**(t) – position of hot mote detected at time t.
- **r**(t) – estimated position of vehicle at time t.
- **v**(t) – velocity of vehicle at time t.
- **a**(t) – vehicle acceleration at time t.
- **p**(t+dt) – predicted vehicle position at time t.
- **n**(t+dt) – position of hot mote detected at time t+dt.

The predicted position **p**(t+dt) is evaluated as

$$\mathbf{p}(t+dt) = \mathbf{r}(t) + \mathbf{v}(t)\, dt + \mathbf{a}(t)\, dt^2 \qquad (1)$$

The estimated target position at time t+dt is then computed from (1) and $\mathbf{n}$(t+dt) in accordance with

$$\mathbf{r}(t+dt) = \gamma\, \mathbf{n}(t+dt) + (1 - \gamma)\, \mathbf{p}(t+dt) \tag{2}$$

γ is a weight between 0 and 1. It takes values close to 1 for very high return values of $\mathbf{n}$(t+dt) and values close to 0 for low return values (close to the threshold for declaring a mote as hot). The value of γ determines how much weigh we want to allocate to the new hot mote position as compared to the predicted position based on our estimate of the target motion. If the intensity of the hot mote at time t is $I_{hot}(t)$, then the assigned γ value is in accordance with

$$\gamma = I_{hot}(t) \,/\, \mathrm{range}(I_{hot}) \tag{3}$$

Where range(I_{hot}) is the range of the intensity of any hot mote determined from the set of returns inside the circle that constitutes the distribution of 'hot' values as explained in section 3.1. Updates for the velocity and acceleration vectors at time t+dt are in accordance with

$$\mathbf{v}(t+dt) = [\mathbf{r}(t+dt) - \mathbf{r}(t)]/dt \tag{4}$$

and

$$\mathbf{a}(t+dt) = [\mathbf{v}(t+dt) - \mathbf{v}(t)]/dt \tag{5}$$

5 Test Bed Experiment

A RCV moves within a network of motes as shown in the snapshot in Figure 5. The vehicle navigates and passes from one cluster to another following the shown track, which is unknown to the sensor cluster. The estimated vehicle track will be compared to the true vehicle to assess the accuracy of the WSN tracking. To trigger the sensors, a wing with attached LEDs is mounted on the vehicle.

One such cluster is shown in Figure 6. The motes are spaced out in a known lay out (positions of the motes are known). The topology of the cluster has a 3x4 matrix layout (12 motes). The motes have a 62 cm separation in the horizontal and vertical direction as shown on the cluster grid. In each cluster, children motes relay data to parent mote (cluster head) which transmits data to the base station/sink or to another cluster head when necessary.

The vehicle navigates and passes from one cluster to another following the shown track, which is unknown to the sensor cluster. The estimated vehicle track will be compared to the true vehicle to assess the accuracy of the WSN tracking. To trigger the sensors, a wing with attached LEDs is mounted on the vehicle.

When the vehicle starts entering the first cluster, the light threshold value is exceeded at mote 1 (position (1,1) in the cluster lattice) and it becomes hot (time dt0=0). The initial vehicle position is $\mathbf{r}$(dt0)=$\mathbf{n}$(dt0) (mote 1 position), $\mathbf{v}$(dt0)=$\mathbf{0}$, $\mathbf{a}$(dt0)=$\mathbf{0}$ as shown in Figure 6. If this cluster was not the initial cluster, the velocity and acceleration vectors will be the ones imported from the last cluster contiguous to the current cluster. Note that the data collected by the cluster head are time, mote ID, and light value.

At time dt1 a second mote at position **n**(dt1) detects the vehicle (position (1,2) in the cluster lattice) as shown in Figure 7. The predicted position **p**(dt1) = **r**(dt0) (since **v**(dt0) and **a**(dt0) are both **0**). The overall estimate of the vehicle position at time dt1 is **r**(dt1) = γ **n**(dt1) + (1-γ) **p**(dt1), where $\gamma = I_{hot}$(dt1) / Range (I_{hot}) in accordance with Eqt. (3), and where I_{hot}(dt1) is the observed light intensity of the hot more at time dt1. Note that time values are determined by the internal clock of the cluster mote as to when it receives a message that a mote became hot. A first estimate for the velocity of the vehicle can be obtained **v**(dt1)= [**r**(dt1) – **r**(dt0)]/ [dt1-dt0], **a**(dt1) = **0**.

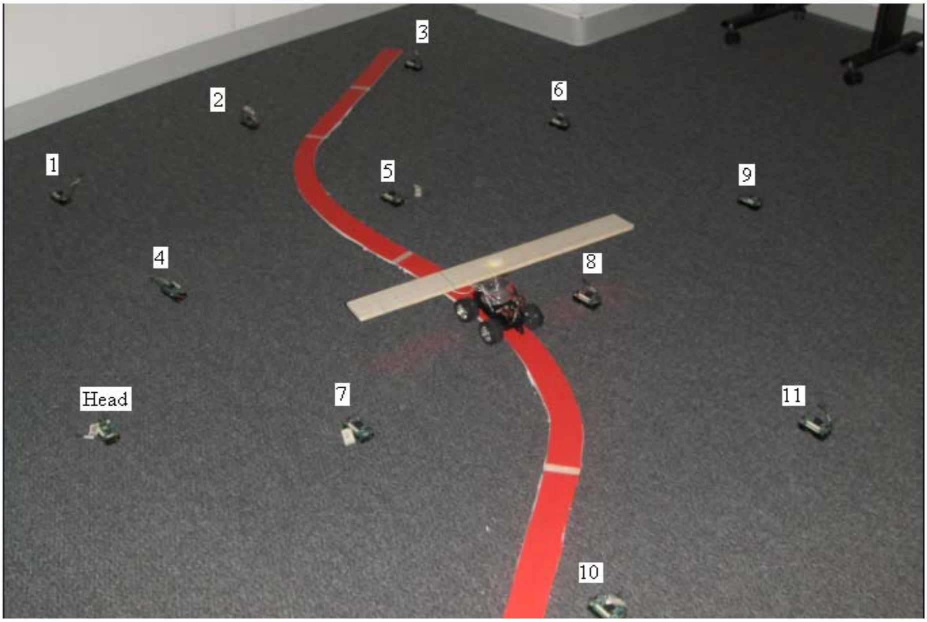


Fig. 5. A snapshot of a RV moving in a WSN. The vehicle trajectory is shown as a track.

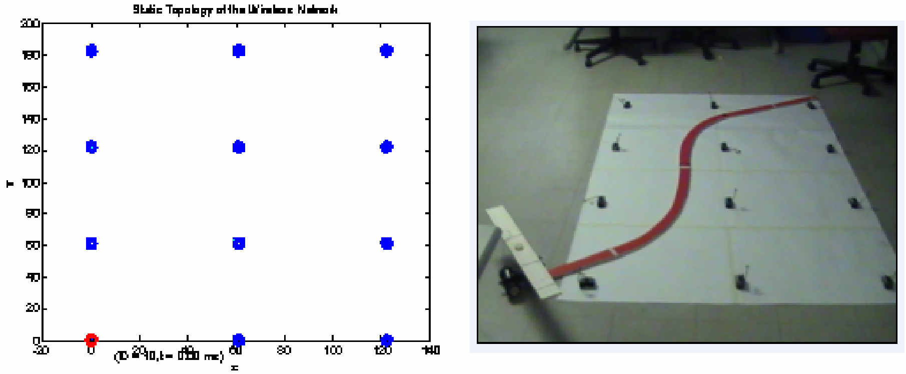


Fig. 6. Cluster of motes with 2 feet separation. The vehicle enters the initial cluster and at time dt0=0 mote 1 becomes hot.

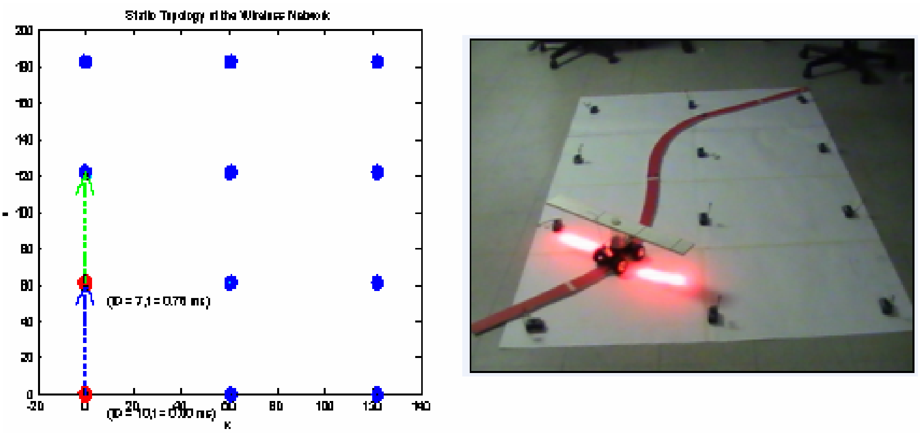


Fig. 7. A new mote becomes hot resulting in an update of the velocity vector at time dt1 but no acceleration update as of yet.

At time dt2, a 3rd mote at position **n**(dt2) becomes hot (vehicle is detected by 3rd mote). The predicted position gets now updated to **p**(dt2) = **r**(dt1) + **v**(dt1) dt1 (no acceleration term present as **a**(dt1)= **0**), and the estimate of the vehicle position at time dt2 is now updated to **r**(dt2) = γ **n**(dt2) + (1-γ) **p**(dt2), where γ = I_{hot}(dt2) / Range (I_{hot}), where I_{hot}(dt2) is the observed light intensity of the hot more at time dt2. We know have an update to the velocity **v**(dt2)= [**r**(dt2) – **r**(dt1)]/ [dt2-dt1] and we have now a first estimate for the vehicle acceleration **a**(dt2)= [**v**(dt2) – **v**(dt1)]/ [dt2-dt1]. The various updates are show in Figure 8.

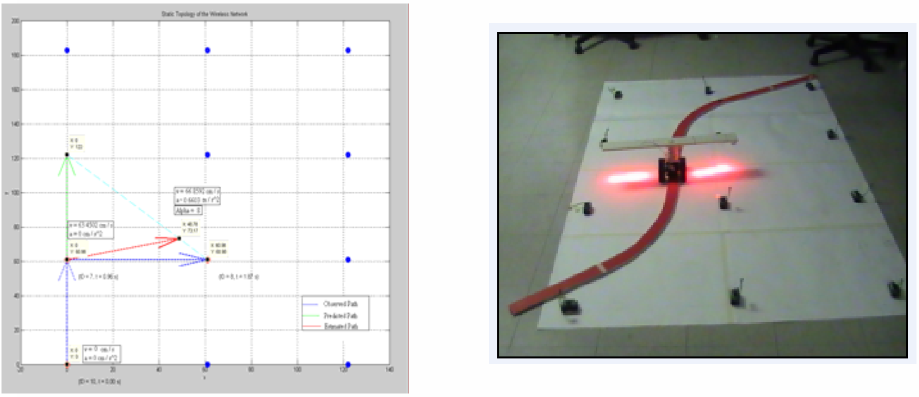


Fig. 8. Update at time dt2, with velocity and acceleration updates

The next movement is shown in Figure 9 with estimated position point shown in red. This follows exactly the same steps as in dt2 except now it is at time dt3.

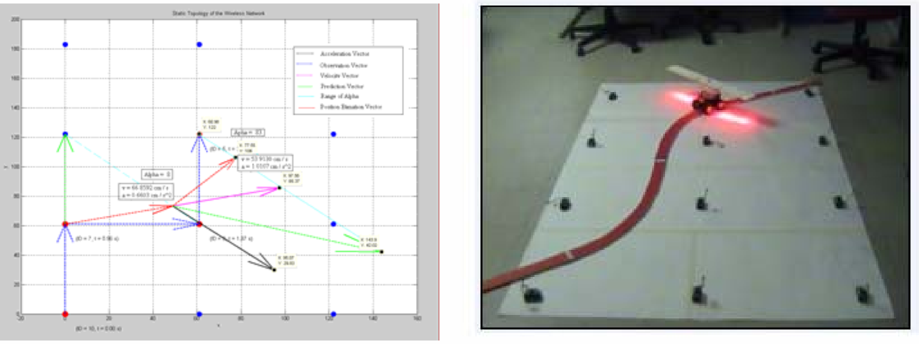


Fig. 9. Update at time dt3 during which the vehicle is accelerating

At the vehicle turns the corner following the track, it decelerates and is picked by the last mote at time dt4 in that cluster before it leaves the cluster and moves into a new one. The movement is shown in Figure 10 where the vehicle is decelerating.

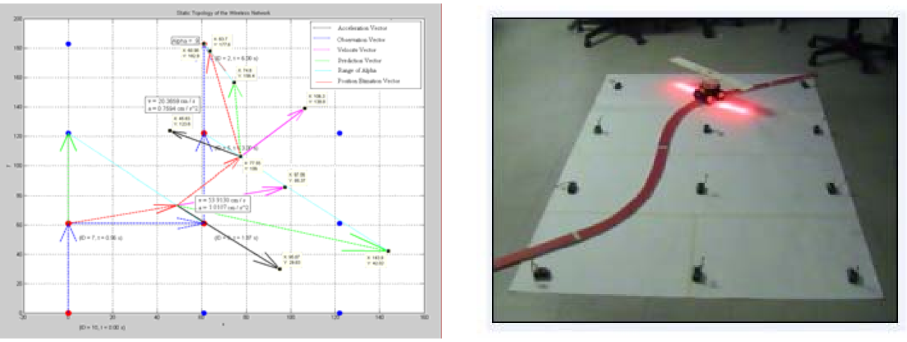


Fig. 10. Update as the vehicle decelerates

At the next movement the vehicle leaves that cluster and starts navigated into a new one. At this point, the cluster head communicates with the new cluster head information about the estimated velocity and acceleration and final position of the vehicle and the time where that occurred.

The accuracy of the tracking is judged based on the error distances between the estimated path relative to the true path. This is shown in Figure 11 for the moving vehicle within the cluster analyzed above. The errors in cm are shown in Table 1. The distances found were the ones from the estimated path perpendicular on the actual path. We found the average (around 26cms), min (around 8 cms), and max (58cms) distances to compare them to the square lattice of the sensors which for our case was of a 62 x 62 cm resolution cell.

Table 1. Distances collected from figure 5 to find the accuracy of the tracking algorithm

	X1, cm	Y1, cm	X2, cm	Y2, cm	Distance, cm
	0	38	8.5	19.8	20.08706
	1	61.5	17.8	28.3	37.2086
	0	57	32.5	47.5	33.86001
	12.4	64	26.8	37	30.6
	7.5	63	33.2	56	26.63625
	25.8	67.5	34.2	66	8.532878
	34.6	72.8	42.8	71.9	8.249242
	35	77.8	53	77.8	18
	35.4	93	64	89.8	28.77846
	44.9	137	65.5	92.8	48.76474
	54.8	152.5	76.4	105	52.18055
	63.5	159.9	74.5	103	57.95352
	65.1	178.9	86.1	163.8	25.86523
	67.4	180	70.9	165.8	14.62498
	73.1	183.5	76.7	169.8	14.1651
	79.4	187	82.2	173.3	13.9832
	84.4	190	87.2	176.4	13.88524
	89.3	192.8	91.7	178.6	14.40139
Average					25.98758
Min					8.249242
Max					57.95352

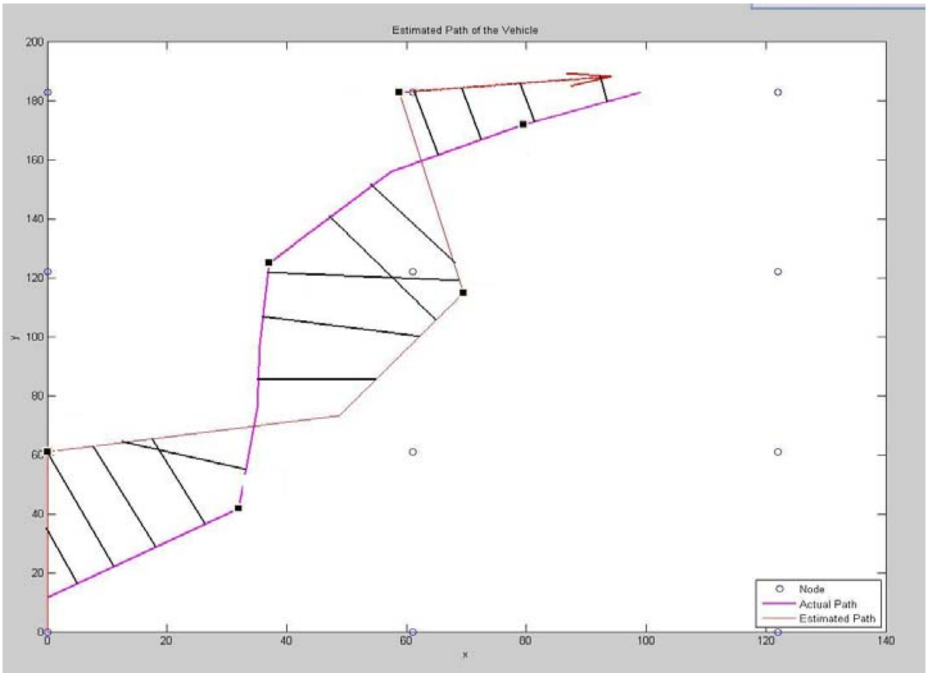


Fig. 11. Estimated vehicle as compared to true one with perpendicular error distances showing

6 Conclusions

We have demonstrated a fast and reliable tracking of a moving vehicle in A WSN made of motes with light sensors. At the forefront of that system is a partitioned of the network into clusters with each cluster having a cluster head and its offspring. Any mote in a cluster can only communicate with the head and does so when it senses that there is a strong light in its vicinity that exceeds a predetermined threshold, making it hot and ushering the passing of the vehicle. In such case, the hot mote relays to the cluster head the light intensity value it senses in addition to its ID, which identifies its position within the cluster. From this information, the cluster head updates the current position of the vehicle as well as its velocity and acceleration vectors. The latter are used predicting where the vehicle will be heading next over time. When a cluster predicts the vehicle to move out of its boundaries in the direction of the cluster contiguous to it, it alerts the new cluster head (wakes it up) and passes to it the latest estimated position, velocity and acceleration of the moving vehicle. This allows for a meaningful and quick transition and tracking from cluster to cluster, in addition to scalability of the tracking problem to bigger and bigger WSN. Although the tracking is demonstrated for a moving vehicle using light sensors, the concepts and solutions are extendable to a variety for other tracking problems, as for example the tracking of moving fire using motes equipped with temperature sensors. Finally, we have shown the accuracy of the tracking by computing the error distances between the true trajectory and the estimated one that showed quite accurate tracking over time.

Acknowledgments. This work is supported by the National Science Foundation, under grant # 0633576.

References

1. Akyildiz, I.F., Su, W., Sanakarasubramaniam, Y., Cayirci, E.: Wireless Sensor Networks: A Survey. Computer Networks Journal 38(4), 393–422 (2002)
2. Brooks, R.R., Ramanathan, P., Sayeed, A.: Distributed Target Tracking and Classification in Sensor Networks. Proceedings of the IEEE (September 2002)
3. Krishnamachari, B.: Energy-Quality Tradeoffs for Target Tracking in Wireless Sensor Networks. In: Zhao, F., Guibas, L.J. (eds.) IPSN 2003. LNCS, vol. 2634, pp. 32–46. Springer, Heidelberg (2003)
4. Yang, H., Sikdar, B.: A Protocol for Tracking Mobile Targets using Sensor Networks. In: Proceedings of IEEE Workshop on Sensor Network Protocols and Applications (2003)
5. Aslam, J., Butler, Z., Constantin, F., Cresp, V., Cybenko, G., Rus, D.: Tracking a Moving Object with a Binary Sensor Network. In: SenSys 2003, Los Angeles, CA, USA, November 5-7 (2003)
6. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: Energy-efficient Communication Protocols for Wireless Microsensor Networks. In: Proceedings of the Hawaaian International Conference on Systems Science (January 2000)
7. Manjeshwar, A., Agrawal, D.P.: TEEN: A Protocol for Enhanced Efficiency in Wireless Sensor Networks. In: Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, CA (April 2001)
8. Younis, M., Youssef, M., Arisha, K.: Energy-Aware Routing in Cluster-Based Sensor Networks. In: Proceedings of the 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2002), Fort Worth, TX (October 2002)
9. Subramanian, L., Katz, R.H.: An Architecture for Building Self Configurable Systems. In: Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing, Boston, MA (August 2000)
10. TinyOS, http://www.tinyos.net
11. Crossbow Technology. Wireless Sensor Networks (2007), http://www.xbow.com/Products/productdetails.aspx?sid=226#pricebox (accessed November 30, 2007)