Proceedings of IMECE2004
2004 ASME International Mechanical Engineering Congress and RD&D Expo
November 13-19, 2004, Anaheim, California USA

IMECE2004-60964

FUZZY VALIDATION AND FUSION FOR WIRELESS SENSOR NETWORKS

Yao-Jung Wen
University of California, Berkeley
Dept. of Mechanical Engineering
6102 Etcheverry Hall
University of California
Berkeley, CA 94720-1740
USA
(510) 643-8146
rio@me.berkeley.edu

Alice M. Agogino
University of California, Berkeley
Dept. of Mechanical Engineering
5136 Etcheverry Hall
University of California
Berkeley, CA 94720-1740
USA
(510) 642-6450
aagogino@me.berkeley.edu

Kai Goebel
GE Global Research
Information & Decision Technologies
Service Algorithms Laboratory
K1-5C4A, One Research Circle
Niskayuna, NY 12309
USA
(518) 387-4194
goebelk@crd.ge.com

ABSTRACT

Miniaturized, distributed, networked sensors - called motes - promise to be smaller, less expensive and more versatile than other sensing alternatives. While these motes may have less individual reliability, high accuracy for the overall system is still desirable. Sensor validation and fusion algorithms provide a mechanism to extract pertinent information from massively sensed data and identify incipient sensor failures. Fuzzy approaches have proven to be effective and robust in challenging sensor validation and fusion applications. The algorithm developed in this paper – called mote-FVF (fuzzy validation and fusion) – uses a fuzzy approach to define the correlation among sensor readings, assign a confidence value to each of them, and perform a fused weighted average. A sensor network implementing mote-FVF for monitoring the illuminance in a dimmable fluorescent lighting environment empirically demonstrates the timely response of the algorithm to sudden changes in normal operating conditions while correctly isolating faulty sensor readings.

Keywords: fuzzy, validation, fusion, sensor network, mote.

INTRODUCTION

The development of smart buildings is one of the most promising nonmilitary applications of multi-sensor data fusion [1]. The sensor validation and fusion algorithm for sensor networks reported herein is applied to an intelligence lighting control system for commercial buildings that focuses on balancing conflicting personal preferences of occupants sharing a common lighting switch and achieving the ultimate goal of energy conservation [2]. Previous studies and surveys show that approximately 50 percent of energy consumption could be saved by introducing this control system for large commercial office buildings with shared workspaces [3]. In these applications illuminance in a room is different at every workspace and a single or several high fidelity lighting sensors will not effectively measure the lighting condition around individual occupants in addition to the whole environment. Smart motes, distributed at all potential workspaces, provide a promising solution for the problem, while sensor validation and fusion algorithms help improve the overall integrity of the mote sensor network [2].

This paper first describes the architecture and design of the mote sensor network for the smart lighting application and the characteristics that motivate the need for validation and fusion algorithms. The mote-FVF (fuzzy validation and fusion) algorithm is then proposed along with discussion of its three major components: *validation*, *fusion* and *prediction*. The results of tests on a small-scaled mote sensor network are used to demonstrate the robustness and efficiency of the mote-FVF algorithm. The conclusion section summarizes the superiority of mote-FVF, and argues that the algorithm could be applied to not only motes but also many other general sensor network applications.

DISTRIBUTED MEMS SENSORS - SMART DUST

"Smart Dust" is proposed as a futuristic dust-sized sensing and communication unit based on MEMS (microelectronic mechanical systems) technology. Millimeter-scale "motes" are available today as prototypes and can be configured with a variety of sensors in high density distributed sensor networks.

Architecture and Operation of Smart Dust Motes

Several types of smart motes have been developed by different research organizations; all consist of a microcontroller, a communication unit and onboard sensors or integrable sensor board module. The communication unit can be one of the following: an RF transceiver, a laser module, or a corner cube reflector. Standardized or customized sensor modules that measure a number of physical or chemical stimuli such as illuminance, temperature, humidity, acceleration, magnetism, or pressure, can easily be integrated onto the motes [4].

The operating system on the mote is stored in flash memory and is programmed to perform customized tasks. The microcontroller acquires a reading periodically from one of the sensors, processes the data, and stores it in memory or sends it out via the transmitter. If a receiver receives incoming communications from other motes or the base station, the microcontroller determines whether to process it or to forward it. The data packet transmitted among the motes or between the motes and the base station may contain the destination and source mote identification number, message length, payload and other useful application-oriented information [5-7].

Characteristics of Mote Sensor Network

While a mote sensor network promises to provide more useful information about the monitored environment than other sensing alternatives, the scope of each single miniature sensor node is much smaller and easily affected by local disturbances. The accuracy and efficiency of the mote sensor network depends on how the pertinent global information is extracted from the local information of each sensor node, highlighting the need for an effective sensor fusion algorithm.

In practical applications, sensor information is always corrupted to some degree by noise and sensor degradation varying with operating conditions, environmental conditions, and other factors. The inexpensive, less reliable and massively distributed MEMS mote sensors will be even more prone to individual failure. Moreover, massively transmitted data packages in the sensor network increase the chance of data collision and interference, and thus are more likely to cause loss of data or incorrect data. Efficiently detecting and isolating bad sensor readings will boost the reliability of the mote sensor network, and hence motivates the use of sensor validation.

FUZZY SENSOR VALIDATION AND FUSION (MOTE-FVF) ALGORITHM FOR MOTE SENSOR NETWORKS

Fuzzy approaches have proven to be effective and robust in a number of challenging sensor validation and fusion applications. They are particularly useful in applications where it is difficult, if not impossible, to construct a precise mathematical model, such as is the case for self-organizing wireless sensor networks that dynamically change over time. The algorithm proposed here is based on the Fuzzy Sensor Validation and Fusion (FUSVAF) algorithm, which has been tested on a range of applications, including sensor validation and fusion in autonomous vehicles and gas turbine power plants [8,9].

The fuzzy validation and fusion (mote-FVF) algorithm for mote sensor networks makes use of a Fuzzy Exponential Weighted Moving Average (FEWMA) time series predictor, dynamic validation curves that are determined by sensor characteristics, and a fusion scheme which uses confidence values for the measurements, the predicted value, the measurements, and the system state [10]. Input to the mote-FVF algorithm depends on the configuration of the sensor network and could draw on raw sensor measurements or intra-network processed data. The output can be used for other intra-network data fusion, the machine level controller, or supervisory control. Fig.1 shows the architecture of the mote-FVF algorithm. There are three units in the mote-FVF algorithm: *validation*, *fusion* and *prediction*.

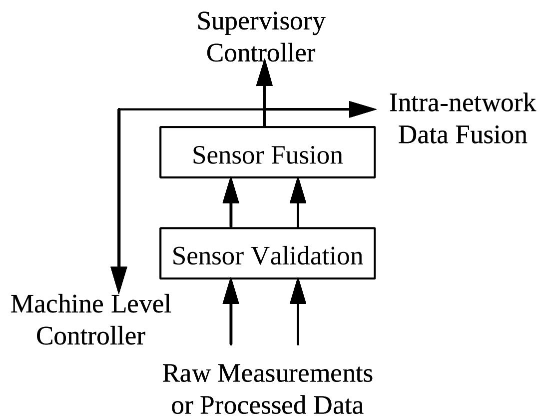

Fig. 1: Architecture for mote-FVF

Validation Unit

The sensor validation part of the mote-FVF validates each incoming reading by assigning it a confidence value, which is determined by a dynamic validation curve. The validation curve is a Gaussian curve generated from the specific sensor characteristics, the predicted value, the correlation among incoming readings, and the physical limitation of the sensor value. The assignment takes place in a validation gate. If sensor readings show a change beyond the gate, the readings are flagged as erroneous and are assigned a confidence value of ‘0’. A maximum confidence value of ‘1’ will be assigned to readings according to their coincidence with the center of the gate, which is determined by considering the predicted value and the correlation among all readings. The correlation among sensor readings is a majority-voting scheme, which can be achieved by the following two approaches: median value or Gaussian correlation.

Median value approach

This method of analyzing the behavior of a cluster of sensor readings first filters out the obvious false readings based on the physical limitation of the sensors, and then takes the median of the rest of the readings. Taking the median rather than the mean value prevents bias induced by unreasonable readings which are far away from the majority but not filtered out at the first step. Generally, the median value should reveal a reasonable estimation of the majority of sensor readings.

Gaussian correlation approach

This approach is more robust than the majority median value approach described above, but requires more computational power. The first step of this approach is also to filter out the obvious false readings based on the physical limitation of the sensors. For the remaining readings, this method first generates a Gaussian function centered on the reading with an appropriately fine-tuned standard deviation for each sensor readings, designated as $PDF_n(x)$, and then computes the reading corresponding to the maximum value of the normalized summation of all Gaussian functions as the estimate as shown in Fig. 2. The normalized summation of all Gaussian functions is calculated by:

$$\frac{\sum_{k=1}^{n} PDF_k(x)}{n}, \; x = 0, \ldots\ldots, \text{maximum sensor output}\,,$$

where n is the total number of reasonable readings (i.e., readings remaining after filtering out obviously false readings) [11].

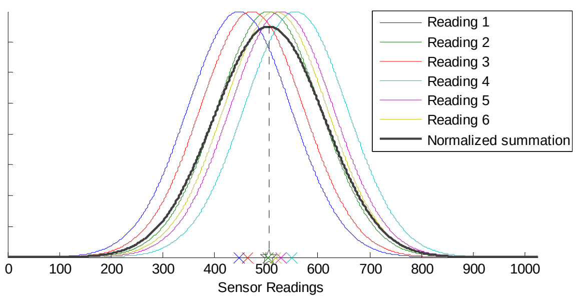

Fig. 2 Gaussian correlation curve

Validation Curve

There are a number of suitable validation curves; one useful curve is a piecewise Gaussian curve of the form $v(z) = e^{-\left(\frac{\tilde{x}-z}{a_w}\right)^2}$, where parameter a_w is chosen separately for the left and right curves according to the characteristics of sensors, z is the sensor reading, and $\tilde{x}$ is the center or precisely the split point of the left and right sections of the validation curves. The curves should be normalized in order to scale the confidence values from 0 to 1. The confidence values are then computed as following,

$$\sigma = \begin{cases} 0 & z < v_{left} \\ \dfrac{e^{-\left(\frac{\tilde{x}-z}{a_{left}}\right)^2} - e^{-\left(\frac{\tilde{x}-v_{left}}{a_{left}}\right)^2}}{1 - e^{-\left(\frac{\tilde{x}-v_{left}}{a_{left}}\right)^2}} & v_{left} < z \le \tilde{x} \\ \dfrac{e^{-\left(\frac{\tilde{x}-z}{a_{right}}\right)^2} - e^{-\left(\frac{\tilde{x}-v_{right}}{a_{right}}\right)^2}}{1 - e^{-\left(\frac{\tilde{x}-v_{right}}{a_{right}}\right)^2}} & \tilde{x} < z \le v_{right} \\ 0 & z > v_{right} \end{cases},$$

where σ is the confidence value corresponding to a particular sensor reading, v_{left} and v_{right} are the left and right validation gate borders respectively, a_{left} and a_{right} are the parameters for the left and right validation curves [9].

$\tilde{x}$ is determined by a fuzzy rule and provides a trade-off between the predicated value and the performance of the majority of sensors. Denoting the estimated majority sensor readings from either method above as $Cor(x)$, and the variation between $Cor(x)$ and the predicted reading $\hat{x}$ as $Var(x)$, the fuzzy rules are applied as follows, starting from the initial condition that $\tilde{x}$ coincides with $\hat{x}$,

IF $Var(x)$ small THEN move $\tilde{x}$ toward the $Cor(x)$ a small amount,
IF $Var(x)$ medium THEN move $\tilde{x}$ toward the $Cor(x)$ a medium amount,
IF $Var(x)$ large THEN move $\tilde{x}$ toward the $Cor(x)$ a large amount.

The fuzzy membership function is designed using standard triangular shaped functions and maximum overlap as shown in Fig. 3 [12]. There are two parameters to be tuned, m_{var} for fuzzification and m_{mov} for defuzzification. Fig. 4 depicts how the center of the validation curve shifts between the value of majority voting and the predicted reading. Note that the offset from the predicted reading takes the relationship among sensor readings into account.

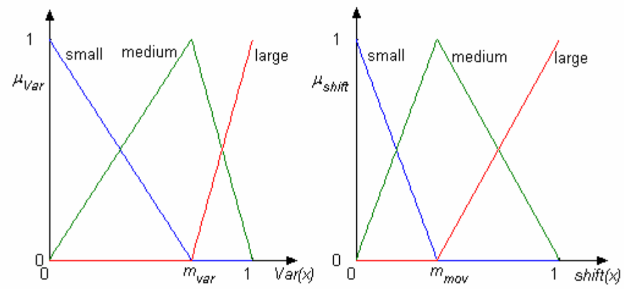

Fig. 3 Membership functions for fuzzy rules defining the center of validation curves

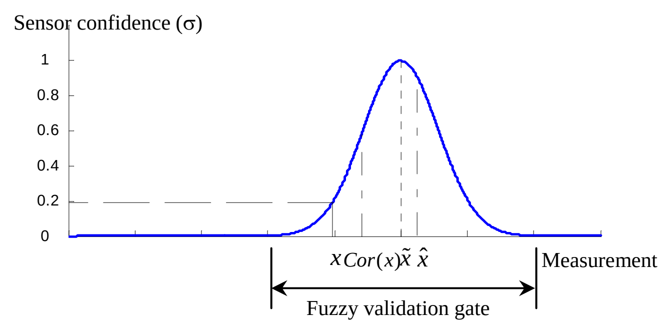

Fig. 4 Fuzzy dynamic validation curve

Fusion Unit

Fusion is performed by taking the average of measurements weighted by corresponding confidence values plus the predicted value weighted by α, an adaptive parameter representing the system state, and a constant scaling factor ω. The equation is

$$x_f = \frac{\sum_{1}^{n} z_i \sigma(z_i) + \frac{\alpha \hat{x}}{\omega}}{\sum_{1}^{n} \sigma(z_i) + \frac{\alpha}{\omega}}.$$

The scaling factor ω is introduced to include a fraction of the predicted value to prevent the system from becoming unstable when no valid readings remain after the validation procedure, and the algorithm will maintain its robustness for a temporary failure of sensors. Since the purpose of the term containing ω is only to deal with the situation of sensor failure, ω is typically large to prevent the predicted value from dominating the fused value and has to be tuned to the system at hand. In addition, the adaptive parameter α carries information about the state of the system and is used in both the fusion unit and the prediction unit. If the system is in steady state, α is set to a large value in order to weight the past history more as the variation in measurements are very likely caused by noises; on the other hand, if system is in a transient state, α is set to a small value in order to weight the predicted value less so as to reduce the lag induced by past history. A mechanism that distinguishes transient from steady state operations, i.e., that can adjust α dynamically according to the system state is given by the set of fuzzy rules below [13]:

IF *change of readings* small THEN α large,
IF *change of readings* medium THEN α medium,
IF *change of readings* large THEN α small.

The membership functions are also designed using triangular shaped functions with maximum overlap such that only two parameters have to be specified: m_e for the fuzzification and m_α for the defuzzification. The functions are shown in Fig. 5.

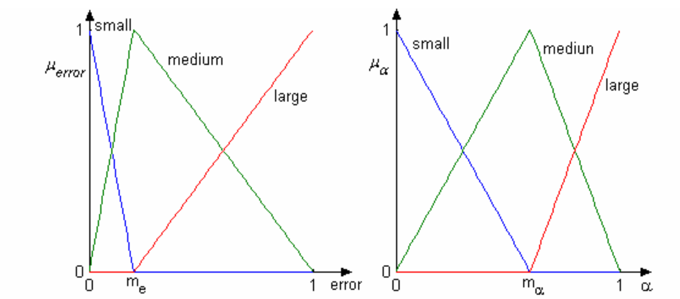

Fig. 5 Membership functions for fuzzy rules of the adaptive parameter α

Prediction Unit

The predicted value for next time step is generated by a time series predictor with the adaptive parameter α tuned to optimize the trade-offs between responsiveness, smoothness, stability, and lag of the predictor. The standard exponential weighted moving average predictor has the form $\hat{x}(k+1) = \alpha\hat{x}(k) + (1-\alpha)x(k)$, where $\hat{x}(k+1)$ is the predicted value of next time step, $\hat{x}(k)$ is the predicted value of the current step, and $x(k)$ is the upgraded current state. Combining the equation with the output of the fusion unit, the predictor in the mote-FVF algorithm is of the form $\hat{x}(k+1) = \alpha\hat{x}(k) + (1-\alpha)x_f(k)$, where $x_f(k)$ is the current upgraded fused value [8,9].

PERFORMANCE TEST

To test the mote-FVF algorithm, a small-scaled sensor network consisting of six motes was constructed. The sensed target was the illuminance on the test bed in a dimmable lighting environment. The photo sensor on the standard Crossbow MICA multi-sensor board [5] was used with the MICA mote processor [6] to sense the illuminance. The lighting environment possesses the following characteristics:

- There could be a large change in illuminance, such as when lights are turned on and off. Thus it is key to test the algorithm's response under such a large discontinuous change.
- The algorithm should be tested over the operating range of the lighting environment – from 0 lux (completely dark) to 1000 lux (bright lighting condition).
- Although zero-valued states are usually considered to be an error or rest condition for many systems, in the lighting environment zero can be either an error or a meaningful state when the environment is completely dark. The algorithm should be capable of distinguishing between them.

Hardware and Environment Setup

The test environment consisted of two double-tube dimmable fluorescent light fixture hanging about 1.5 meters above the test bed, allowing the ballast to dim the light from 1000 lux to 50 lux in 18 intervals. The true illuminance was measured by a high fidelity illuminance meter. Six MICA motes equipped with standard MICA sensor boards were arranged in a 3-by-2 matrix; each was programmed to acquire 10 illuminance readings every 5 seconds at the rate of 50 mini-seconds per sample and to send the data packet containing the readings, along with the source mote ID and the destination mote ID, to the base station. To simplify the test, the base mote was placed in the middle of the matrix to form a centralized sensor network configuration. In addition to the base mote, the base station consisted of a computer, running a Matlab program with a Java I/O interface to receive the incoming data packet from the base mote, processes the data and carry out the mote-FVF algorithm [7]. Although it would be possible to embed portions of the mote-FVF algorithm in each mote to make the sensor network more versatile, for purposes of testing the algorithm, the mote sensors were only responsible for sensing illuminance and sending back data to the base station where the mote-FVF algorithm was processed. The sensed illuminance values transmitted in the mote network were digital values from the A/D (analog-to-digital) converters on the mote. A prior auxiliary calibration was conducted in order to map the digital values to lux units.

Tuning the Parameters of Mote-FVF Algorithm

The parameters requiring tuning are a_{left} and a_{right} for the validation curve, the fuzzification parameter m_{var} and the defuzzification parameter m_{mov} for the shift of the center of the validation curve, the constant scaling factor ω, the fuzzification parameter m_e and the defuzzification parameter m_α for generating the adaptive parameter α. Without loss of generality, a_{left} and a_{right} were assigned to be the same to get a symmetric validation curve as suggested by prior experiments. Moreover, the values of a for each mote sensor were assigned identically, as they were all of the same type and were expected to have similar characteristics. m_{var} and m_{mov} were adjusted so that the validation curve could shift enough to catch at least some information of the possible largest change in a single time step without becoming too sensitive to failures. Parameters m_e and m_α here were tuned so as to optimize the response of the algorithm to the lighting environment. m_e, m_α, m_{var} and m_{mov} were tuned separately for the two majority voting schemes. The constant scaling factor ω was chosen to be large in order not to result in an obvious lag in the fused value by weighting the previous predicted value too heavily.

Real-time Implementation and Simulation Result

Fig. 6 shows the real-time implementation of the mote-FVF algorithm using the *median value majority voting* scheme on the six-mote sensor network. The small cross signs are the raw sensed data from each mote, the dashed line is the true illuminance measured by the high fidelity illuminance meter, and the solid line is the fused value. The figure reveals that the algorithm accurately followed the sensed data and reflected the illuminance pertinently with a maximum error of 3.36% regardless of the lag in the transient mode. Similarly, the maximum error using the Gaussian correlation approach (not shown in the graph) is 4.01%. These errors include calibration errors when mapping raw digital readings to units of lux and the variation of the illuminance for small changes in the spatial position on the test bed. From a physical point of view, the 4.01% error represents about a 30 lux difference to which a human being is insensitive.

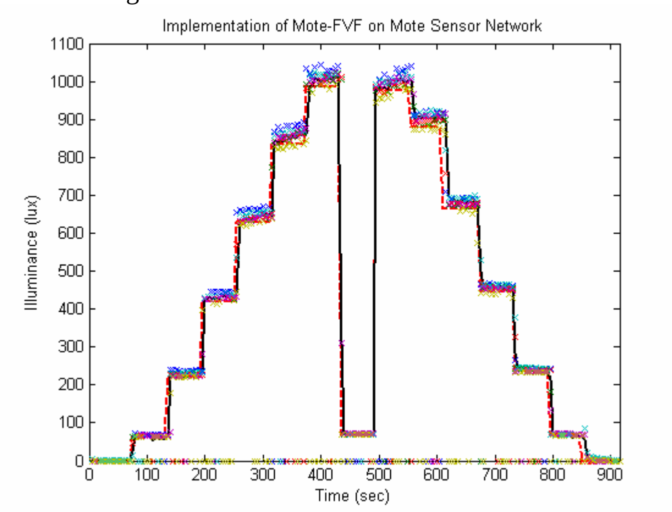

Fig. 6 Real-time implementation of mote-FVF

A comparison of several different configurations of the sensor validation and fusion algorithm is shown in Fig. 7. Part of the data set gathered in the real-time testing was used to run off-line simulations for four configurations, and some modifications were introduced to simulate possible sensor failures. Plot (a) shows the sensed data points as cross signs and the true illuminance with a dashed line. In plots (b) to (e), the red dashed line displays the true illuminance and the black solid line shows how the fusion algorithms follow the true illuminance. Plot (b) and (c) are the mote-FVF algorithms with the two majority voting schemes, median value and the Gaussian correlation, respectively. Both show good performances even when facing sensor failures. Plot (d) shows that a majority voting algorithm without involving validation and prediction is much more sensitive to sensor failures than the mote-FVF. Plot (e) is the basic FUSVAF algorithm, which is clearly not capable of dealing with large discontinuous changes by fixing the center of the validation curve at the predicted value.

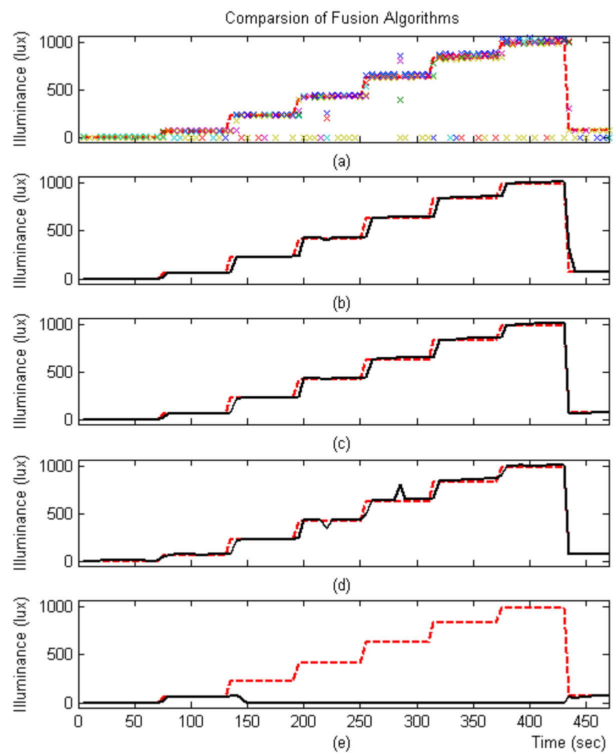

Fig. 7 (a) raw measurement data and true illuminance; (b) mote-FVF algorithm with median value approach; (c) mote-FVF algorithm with Gaussian correlation approach; (d) Gaussian majority voting; (e) FUSVAF algorithm.

CONCLUSION

"Smart lighting" is a promising energy-saving approach in the control of commercial buildings which has the challenging properties of large operating ranges and generally uneven distributions in the environment. The mote-FVF algorithm proposed here has shown its effectiveness in extracting informative illuminance information and robustly rejecting failures as needed for commercial lighting applications. Smart dust motes, equipped with computational and memory capabilities, allow more sophisticated configurations of validation and fusion algorithms for future research. The design of the mote-FVF algorithm does not involve any specific data type and only needs to be tuned to fit the system, implying that the algorithm has potential for fusion of intra-network sensor-cluster data and even different data types from disparate sensors. One possibility would be to distribute the mote-FVF calculations among clusters of motes, using local illuminance information. The fused cluster data would then be transmitted to the base station to arrive at a global calculation. This architecture lends itself to distributed actuation and control as well.

ACKNOWLEDGMENTS

We would like to acknowledge Jessica Granderson and Johnnie Kim, students in Mechanical Engineering at UC Berkeley, for building the dimmable lighting testbed; Francis Rubinstein of Lawrence Berkeley National Laboratory and Stuart Berjansky of Advance Transformer Company, Rosemont IL, for the donation of dimmable ballasts; Professor David E. Culler of Department of Computer Science, UC Berkeley for providing smart motes; and Christine Condon of Pacific Energy Center, San Francisco CA, for lending us the high fidelity illuminance meters.

REFERENCES

[1] Hall, D. L. and Llinas, J., 1997, "An Introduction to Multisensor Data fusion," Proceedings of the IEEE, **85**, No.1, pp. 6-23.

[2] Agogino, A. M., Granderson, J. and Qiu, S., 2002, "Sensor Validation and Fusion with Distributed 'Smart Dust' Motes for Monitoring and Enabling Efficient Energy Use," *Proc., AAAI 2002 Spring Symposium*, Stanford, CA, pp. 51-58.

[3] Yozell-Epstein, R., 2003, "Economic, Energy and User Needs Analysis of Intelligent Lighting Systems," Master's report, University of California at Berkeley, Berkeley, CA.

[4] Hollar, S., 2000, "COTS Dust," Master's thesis, University of California at Berkeley, Berkeley, CA.

[5] Crossbow Technology, 2003, "MTS/MDA Sensor and Data Acquisition Boards User's Manual," Rev. B, San Jose, CA.

[6] Crossbow Technology, 2003, "MPR- Mote Processor Radio Board, MIB- Mote Interface/ Programming Board User's Manual," Rev. A, San Jose, CA.

[7] TinyOS Community, 2003, "TinyOS Documentation," http://www.tinyos.net/tinyos-1.x/doc.

[8] Goebel, K. and Agogino, A. M., 1996, "An Architecture for Fuzzy Sensor Validation and Fusion for Vehicle Following in Automated Highways," *Proc., 29th International Symposium on Automotive Technology and Automation (ISATA), Dedicated Conference on Fuzzy Systems/Soft Computing in the Automotive and Transportation Industry*, Florence, Italy, pp. 203-209.

[9] Goebel, K. and Agogino, A.M., 1999, "Fuzzy Sensor Fusion for Gas Turbine Power Plants," *Proc., SPIE Conference on Sensor Fusion: Architectures, Algorithms, and Applications III*, Orlando, FL, Vol. 3719, pp. 52-61.

[10] Prajitno, P. and Mort, N., 2001, "A Fuzzy Model-Based Multi-Sensor Data Fusion System," *Proc., SPIE Conference on Sensor Fusion: Architectures, Algorithms, and Applications V*, Orlando, FL**,** Vol. 4385, pp. 301-312.

[11] Abdelrahman, M., Kandasamy, P. and Frolik, J., 2000, "A Methodology of Fusion for Redundant Sensors," *Proc., 2000 American Control Conference*, Chicago, IL, Vol. 4, pp. 2922-2966.

[12] Harris, J., 2000, *An introduction to fuzzy logic applications*, Kluwer Academic Publishers, Dordrecht.

[13] Khedkar, P. and Keshav, S., 1992, "Fuzzy Prediction of Time Series," *Proc., IEEE International Conference on Fuzzy Systems*, San Diego, CA.