DEMPSTER-SHAFER BASED INFORMATION QUALITY PROCESSING IN SMART ENVIRONMENTS

Soukaina Messaoudi, Kamilia Messaoudi and Serhan Dagtas
Department of Information Science
University of Arkansas at Little Rock
2801 S. University Avenue
Little Rock AR 72204
sxmessaoudi@ualr.edu, kxmessaoudi@ualr.edu, sxdagtas@ualr.edu

Abstract: *Smart environments* refer to buildings or locations equipped with a multitude of sensors and processing mechanisms for improved security, efficiency or functionality. Often, these sensors serve distinct purposes and their data may be processed separately by entirely separate systems. We argue that integrated processing of data available from multiple types of sensors can benefit a variety of decision making processes. For example, smart building sensors such as occupancy or temperature sensors used for lighting or heating efficiency can benefit the security system, or vice versa. Recent industry standards in sensor networks such as ZigBee make it possible to collect and aggregate data from multiple, heterogeneous sensors efficiently. However, integrated information processing with a diverse set of sensor data is still a challenge. We provide means to use a data fusion scheme that offers efficient processing of information collected from multiple sensors such as temperature sensors or motion detectors and visual sensors such as security cameras. The broader goal of multi-sensor data fusion in this context is to enhance security systems, improve energy efficiency by supporting the decision making process based on relevant and accurate information gathered from different sensors. In particular, we investigate the use of Dempster-Shafer based data fusion model and present techniques for processing of visual sensor data to facilitate the use of Dempster-Shafer model.

Key Words: Data and information fusion, Bayesian, Dempster-Shafer, Fuzzy logic, Neural Networks, Visual sensors, Non-visual sensors, Sensor networks, Motion segmentation, OpenCV, Cricket, beacon, listener.

INTRODUCTION

One of the outcomes of data fusion is the improved information quality that assists various decision making processes in a "smart environment". Our focus here is the integration of sensors information into the real-time decision making process in a surveillance context. We use data fusion in a fashion where different types of information are collected from a heterogeneous set of visual and non-visual sensors. The process of integrating data from different sources requires designing an appropriate data fusion model that would take the sensor data, integrate them following a certain model, and transform it to a set of useful and relevant decisions. The anticipation is for the resulting decisions to be more accurate and efficient than those resulting from a single source. In a broader sense, we expect data fusion to lead to a virtual collaboration between the different collected information.
Towards this goal, we first investigate the usefulness of data fusion in a smart environment equipped with visual and non-visual sensors and design a convenient data fusion model. Then, we provide an overview of data fusion methods, present our data fusion algorithm and discuss our data fusion engine. Our particular focus is to provide proper input into the Dempster-Shafer model. This is followed by a

The research in this paper was supported in part by the US Air Force Research Laboratory Sensors and Human Effectiveness Directorates. Some of the research activities were carried out with the technical support by the Tec^Edge Discovery Laboratory at the Wright Brothers Institute in Dayton, OH.

description of our smart environment simulation tool which is used to test some of the hypotheses, visualize the environment with the sensors and their spatial relationships and to allow us to build some of the case scenarios which is discussed last. We then discuss the importance of accurate, complete, and consistent information in a wireless sensor network and present the results of our research being conducted at Tec^Edge in Dayton, OH. In the last section, we summarize our findings and conclusions with a set of ideas for ongoing work.

INFORMATION ACCURACY IN SMART ENVIRONMENTS

There are three important data quality dimensions that require most focus in a smart environment: *completeness*, *consistency*, and *accuracy*. Completeness refers to the degree to which a reading from a sensor is complete. In other words, the completeness dimension measures the degree to which data is missing. Consistency, on the other hand, refers to how consistent sensory data is with respect to their scheduled data transmission. Consistent sensors are sensors that report a reading every time they are used to measure the likelihood of a given event. Accuracy, which is covered in more detailed here, is the most interesting data quality dimension in smart environments with an information quality perspective. Accuracy implies that sensors need to report the real characteristics of the environment, which makes it a critical dimension that needs much analysis and care.

In practice, these three dimensions are very related to each other. In other words, consistency of information is a requirement for the information to be accurate. In a similar manner, completeness of information is essential to conclude that the information is in fact accurate. In this sense, treating consistency and completeness lead to treating most of the accuracy aspects.

Tracking completeness of information requires analyzing the data stream for any missing data. In case the whole information is missing, methods that use the sensor's statistical data are usually used to predict the correct value. Consistency, on the other hand, is usually caused when the sensor reports quite different readings about the same event in the same environment. This situation is usually due to sensor malfunctioning or battery, and can be detected through the use of neighboring sensors' readings. Readings from neighboring sensors can effectively contribute in detecting the erroneous pattern of the sensor in question and help us build a prediction scheme to overcome the inconsistent readings. Doing this helps also the accuracy dimension where the use of neighboring sensors' readings greatly improves the accuracy and correctness of information. As a result, our work is focused mainly on addressing accuracy, which also ensures completeness and consistency of information.

Methods for Accuracy as an Information Quality Dimension

Data accuracy can be very critical in many applications such as surveillance where it is crucial to avoid false decisions such as triggering the alarms. Accuracy issues are usually due to faulty nodes that manifest two types of faults: function fault and data fault. Function fault results in the crash of nodes, and usually this problem is treated through using distributed approaches such as neighbor coordination or through using centralized approaches such as status updates. On the other hand, data fault implies erroneous node sensing, which leads to eventual erroneous decisions **Error! Reference source not found.**.

One of the most commonly used methods to address sensing errors is the outlier detection where readings are compared and analyzed to identify those that are distant from the rest. However, the correctness of the outlier detection relies heavily on the level to which data follows a certain distribution. In [9], the authors suggest that the outlier detection method would work rather with temperatures that are usually known to follow a normal distribution. However, other sensing readings such as thermal radiation do not really follow a distribution, which makes the outlier detection method prone to errors.

The after-deployment calibration is another way to deal with this issue such as the development of a mapping function that maps erroneous readings to correct ones. The parameters of the function can be obtained through many ways, but assumptions about the sensing model, dense deployment, similarity of readings among neighbors, and availability of ground truth result are usually required. The authors in [9] suggest a new method to target data faults called FIND. This method is a sequence-based detection approach that assumes no distribution of readings. Since no distribution of readings is assumed, FIND accomplishes the detection through identifying ranking violations in node sequences, where a sequence is obtained through ordering IDs of the nodes according to their readings for a given event. FIND has the objective of creating a backlist that contains all possible faulty nodes. Correcting erroneous readings or replacing malfunctioning sensors with better ones can be easily accomplished through using the backlist.

In **Error! Reference source not found.**, the authors suggest an algorithm that uses data predictions to filter out errors caused by soft failures (failures caused by a deviation from the normal behavior). The suggested algorithm allows for a delayed reporting of data, which helps them use the observed values in the next samples and find the correct choice of value that lies between the predicted and the observed value. All error corrections are carried out at the receiver that is less resource-constrained than the sensor nodes. The framework of this work is composed of three main processes. The first process is a model of data generation that is constructed through identifying the correlations observed in sample of sensory data. The second process is another model that is used for online prediction of data, and the third process is a correction block that uses the prediction history in order to correct errors detected in data.

Experiments on Information Accuracy

In order to test and improve methods that address the accuracy of information in a wireless sensor network, we have mounted a Cricket location system at Tec^Edge in Dayton, OH. The cricket system is a system that consists of a number of beacons and a listener attached on a host device. Both beacons and listeners are similar motes, and they only need to be configured either way. The way the Cricket works is simple: the beacons periodically broadcast their space identifiers and position coordinates on a radio frequency channel that can be received by the listener. For more information about the Cricket system, users/readers can access a detailed Cricket manual on their website that is hosted by MIT [11].

As mentioned, our focus is on the quality of information. Therefore, analysis of data received by the listener is the focus of our work. Data obtained from the listener can be processed on Linux computers through the use of cricketed, which is a daemon used to access the command interface over the network and allows for the processing of information to get the different location properties [11].

Figure 5. Snapshot of the Cricket system at Tec^Edge, Dayton, OH.

The data coming from the beacons is in the form of a stream where every beacon's reading can be identified through the beacon ID and the space identifier. A typical output looks like the following:

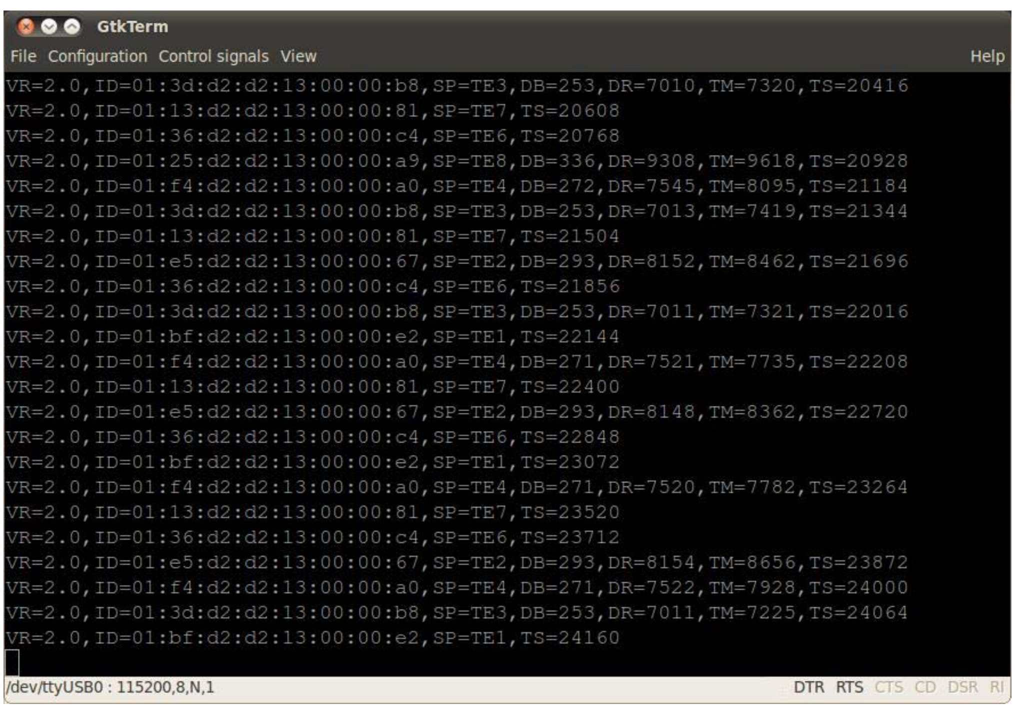

Figure 6. Snapshot of data stream received by the listener

On figure 6, ID refers to the beacon's ID which is similar to a MAC address. This is very helpful to identify the motes that are broadcasting a message, and differentiate between the different motes. Similarly, SP refers to the name or the space chosen for every sensor, which also helps in differentiating between information coming from the different sensors. DB, on the other hand, refers to the distance (information needed) between the sensor and the listener that is hooked to a device capable of getting serial data such as computer, PDA, or a similar device.

In fact, the distances are estimated with an average of two centimeter accuracy and a range of ten meters. In the Cricket System, the distance estimation starts when the beacons mounted on the ceiling first transmit a RF signal along with a message that contains information such as the beacon ID, the space ID, the beacon's coordinates, and the measured ambient temperature. In addition, the beacons also transmit a narrow ultrasonic pulse at the beginning of the RF message. The listener, on the other hand, receives the RF signal first as its speed is greater than the speed of the ultrasonic signal. When the RF signal is received (at time T_{rf}), the listener activates the ultrasonic receiver and the timer. The activation results in receiving an ultrasonic pulse at time T_{us}. The listener then calculates the time difference of arrival $\Delta T = T_{us} - T_{rf}$ **Error! Reference source not found.**. The distance is, therefore, computed as follows:

$$\Delta T = \frac{d}{V_{us}} - \frac{d}{V_{rf}} \qquad (1)$$

$$d = \frac{\Delta T(V_{us} \cdot V_{rf})}{(V_{rf} - V_{us})} \qquad (2)$$

V_{rf} is approximately 3X10^8 m/s while V_{us} changes depending on both temperature and humidity (344 m/s in normal environments). Clearly, V_{rf} is always much larger than V_{us} which changes equation 2 to the following:

$$d = \frac{\Delta T(V_{us} \cdot V_{rf})}{V_{rf}} \quad \text{or simply} \quad d = \Delta T \cdot V_{us} \qquad (3)$$

In our project, we focus on the analysis of the data streams that come from the listener and measure their quality. Sensors usually transmit erroneous information or don't even transmit it; therefore, we need to develop methods to detect the errors and correct them. The initial step is the analysis of the data coming from the motes and definition of the types of errors that appear in a Cricket system. Once the types of errors are defined and determined, they need to be categorized in order to choose the most vital ones that seriously affect the accuracy dimension. Since this is an ongoing research, results for experimentation and methods suggested to target accuracy as an information quality dimension will be provided later.

TECHNIQUES FOR DATA FUSION

Data fusion is "the theory, techniques and tools which are used to combine sensor data, or data derived from sensory data, into a common representational format." Fusing data from different sources can improve the quality and the utility of information and help improve efficiency, security and functionality. The critical problem in multi-sensor data fusion is to determine the best method to combine information from different sensors [4].

Most of the reported work in data fusion uses a statistical approach in order to describe different relationships between sensors taking into account the underlying uncertainties [4]. E. Waltz and J. Llinas summarize the methods that implement data fusion as follows: decision or detection, estimation, association, and uncertainty management theories. In decision or detection theory "measurements are compared with alternative hypotheses to decide which hypothesis best describes the measurement." Basically, the decision theory assumes "the probability descriptions of the measurement values and prior knowledge to compute a probability value for each hypothesis." [8].

Uncertainty management stems from classical methods that represent uncertainty in measurements using the Bayesian probability model to express the degree of belief in each hypothesis as a probability. The hypothesis must be mutually exclusive and this requires that all hypotheses must form a complete set of possibilities and the probabilities must sum to one. Because the Bayesian model cannot present uncertainty along with the fact that probabilities must be assigned to each hypothesis, Dempster-Shafer introduced the concept of probability intervals to provide means to express uncertainty. Other heuristic models and fuzzy calculus have also been applied to uncertainty representation for fusion applications [8]. Fuzzy logic, neural networks, Bayesian, and Dempster-Shafer theories are the most commonly used methods in multi-sensor data fusion. In this paper, we provide a brief description of all four methods. However, our approach will focus on Dempster-Shafer model for integrated information processing using data from multiple, heterogeneous sensors. The main reasons for this election were the appropriateness of the input and output types in Dempster-Shafer model and its wide-spread use for similar problems in the literature. We plan to expand our work into the alternative fusion techniques as part of our ongoing research.

Bayesian Theory: The basic principle of Bayes' theory is that all the unknowns are treated as random variables and that the knowledge of these quantities can be represented by a probability distribution. In addition, Bayesian methodology claims that the probability of a certain event represents the degree of belief that such an event will happen. The degree of belief is associated with a probability measure that can be updated by additional observed data. All the new observations are added to update the prior probability and therefore obtain a posterior probability distribution [3]. The Bayesian statistics have numerous advantages since it is the only known coherent system for quantifying objective and subjective uncertainties. Additionally, it provides principled methods for the model estimation and comparison and the classification of new observations. Also, it provides principle methods for dealing with missing information.

Dempster-Shafer theory is considered to be a generalization of the Bayesian theory of subjective probability. Dempster-Shafer allows us to "base degrees of belief for one question on probabilities for a related question" [6]. In fact, the Dempster-Shafer theory is based on two ideas: the degrees of belief for a question are obtained from subjective probabilities associated with a related question, and the degrees of belief are combined using Dempster's rule "when they are based on independent items of evidence". One of the most important advantages of the Dempster-Shafer theory is that it does not associate probabilities to questions of interest as Bayesian methods do. Instead, the belief for one question is based on probabilities for a related question; therefore, the Dempster-Shafer theory can effectively model uncertainty. Additionally, the Dempster-Shafer theory doesn't require or demand the use of probabilities whenever possible [1]. Furthermore, Dempster-Shafer allows the computation of additional support and plausibility, as opposed to the Bayesian theory [2]. We plan to incorporate Dempster-Shafer method into our smart environment model as part of our ongoing work.
Dempster-Shafer model of combination evidence integrates data, independently, from r correlated sensors' inputs in the following pattern:

$$m^{1,2,\ldots,r}(C) = \frac{\sum_{c1 \cap c2 \cap \cdots \cap cr = \emptyset} m_1(c_1), m_2(c_2), \ldots, m_r(c_r)}{1 - \sum_{c1 \cap c2 \cap \cdots \cap cr = \emptyset} m_1(c_1), m_2(c_2), \ldots, m_r(c_r)}$$

where C is the proposition, so we have in this case C = {$c_1, c_2, \dots, c_r$} (universal set $\rightarrow$ C).
$c_1, c_2, \dots, c_r$ are independent decisions by sensors regarding proposition C.
m_1, m_2, …, m_r are independent belief or mass functions. Also, $\sum_{c_1 \cap c_2 \cap \dots \cap c_r = \emptyset} m_1(c_1), m_2(c_2), \dots, m_r(c_r)$ accounts for conflicts in the belief distributions from the sensors and assures that the combined belief is normalized to the unit interval.

DEMPSTER-SHAFER DATA FUSION

The fusion engine illustrated in Figure 1in this project is the model we use to integrate information from sensors. The engine receives inputs from both visual and non-visual sensors and provides a set of relevant decisions (outputs).

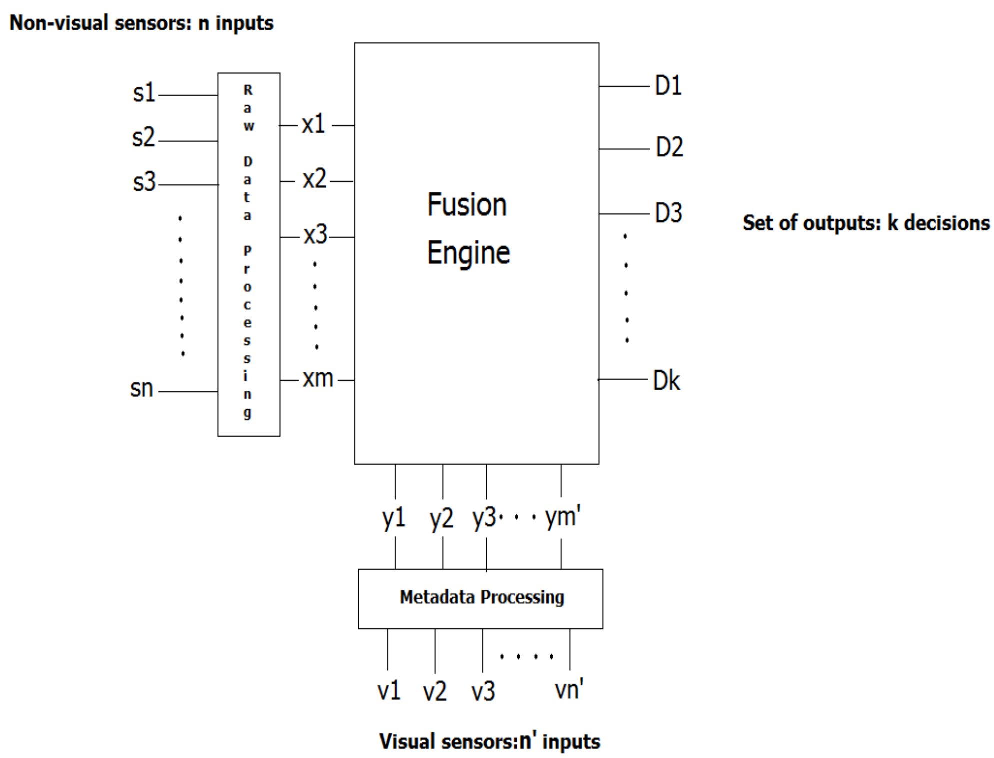

Figure 1. Fusion Engine Components

As the diagram in Figure 1 shows, $s_1, s_2, s_3 \dots s_n$ are inputs from different non-visual sensors. These inputs first go through a correlation model (raw data processing in Figure 1) that determinates the correlations among the sensor inputs and transmits *m* independent outputs to the fusion engine as inputs. These outputs (fusion engine inputs) are labeled as $x_1, x_2, x_3 \dots x_m$.

The fusion engine inputs x_1, x_2, $x_3 \dots x_m$ correspond to $m_1(c_1), m_2(c_2), \dots, m_r(c_r)$ (refer to the algorithm section), which represent the mass functions of correlated sensors. However, this matching does not necessitate matching x_1 to $m_1(c_1)$, x_2 to $m_2(c_2)$ etc. as the data fusion model we use consider integrating information from both non-visual and visual sensors. As it is explained below, data from visual sensors is pre-processed before it can be used by the fusion engine. This pre-processing results in a proper format of information to be passed to the fusion engine that employs Dempster-Shafer technique.

We first provide a brief description of visual sensors and the extraction of visual information that comes from these sensors, followed by the description of the non-visual sensor information processing.

For visual sensors, we use optical and infrared cameras to record raw videos. The acquired videos are then processed to extract meta-data information to be used in the fusion algorithm described above. The processing of images from such visual sensors requires a preliminary processing where some intermediary image features such as moving objects and their boundaries are extracted for further processing [7]. The final extracted visual information forms a metadata that can be fed to the designed fusion engine that integrates it with other sensor data from other heterogeneous sensors.

The extraction of visual information can be a real challenge because of “the lack of proper low-level algorithms for robust feature extraction” [7]. Problems such as camouflage effects, light changes, foreground aperture, and ghosting can relatively affect the extraction of visual information. Here, we use a motion detection algorithm to extract relevant visual information about the moving objects in the recorded video. The algorithm chosen for this purpose has its implementation in OpenCV, which is an open-source computer vision library, originally developed by Intel. A modification at the level of the sample program inputs results in a movement detection represented in a red box that surrounds every moving component (the size of the box is relative to the size of the moving component). The information extracted from the videos is further processed to form a metadata from which relevant information can be extracted and can easily be integrated with other non-visual sensor information. The metadata in this context includes a list of information such as the number of moving objects, the nature of movement, the type of the moving objects (human or animal), the actions performed by the moving objects, and the area they occupy.

In the fusion engine model depicted in Figure 1, v_1, v_2, v_3 …, $v_{n'}$ represent the information collected (metadata) from every visual sensor. In our case, this information includes the number of moving pixels in every frame from the recorded video. These inputs are processed to create appropriate input format. In order to come up with mass functions to be used by the fusion engine, we first find the spatial area of the box that surrounds every moving component as shown in Figure 2 and divide it by the area of the frame as follows:

$$\text{Size Factor} = \frac{a_t}{C} * 100$$

where a_t is the area of the bounding box that surrounds the moving component at a given time t and C is a constant and is equivalent to 307200 pixels, which is the size or the area of the frame.

Computing Size Factor values allow us to come up with one single value or mass function that takes into consideration the persistence of the moving object (the period of time the moving object takes).

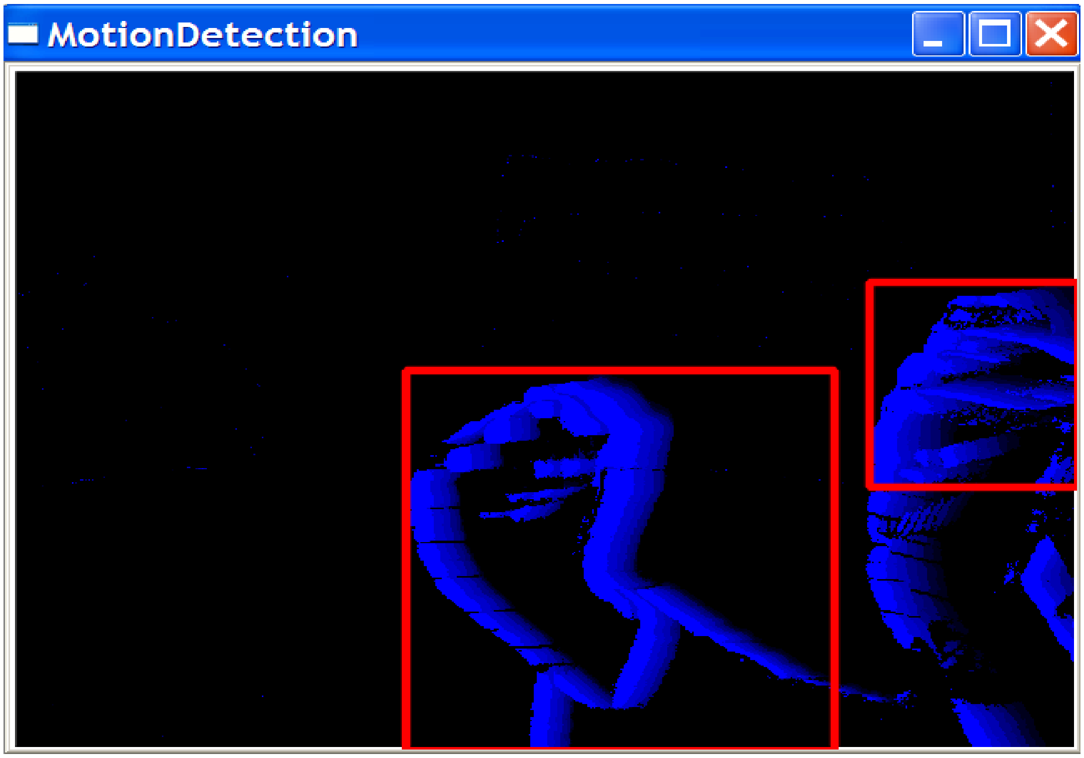

Figure 2. Bounding boxes surrounding two moving objects

As is the case with non-visual sensor inputs, pre-processed visual sensor inputs, y_1, y_2, y_3 …, $y_{m'}$, also form some of the mass functions represented as $m_1(c_1), m_2(c_2),$ $m_3(c_3), \ldots, m_p(c_p)$ in the algorithms section. The initial inputs, v_1, v_2, v_3 …, $v_{n'}$, are in the form of metadata, which are the information we extract from the recorded videos. After tracking moving objects on a given video, more work is done on detecting the different features of these moving objects. Features such as the number of moving objects, the nature of the moving objects (human, animal…), and the nature of movements (fast, slow…) the objects perform are examples of information that can be used by the fusion engine. After extracting such important information (metadata), we perform another processing on the metadata to come up with an input format, belief function, compatible with the data fusion model we are using (Dempster-Shafer model).

In data fusion context, the outputs of such a model are in the form of decisions that should be performed as part of the “smart environment” where different types of sensors collect information. As Figure 1 shows, the set of decisions D1, D2,…, Dk are the independent fusion engine outputs (or decisions). The decisions generated are independent in the sense that each group of correlated sensors leads to a different decision D1, D2, …, or Dk. These decisions can help save energy, improve security, guide rescue operations etc. depending on the particular nature or the need of the environment.

SIMULATION TOOL

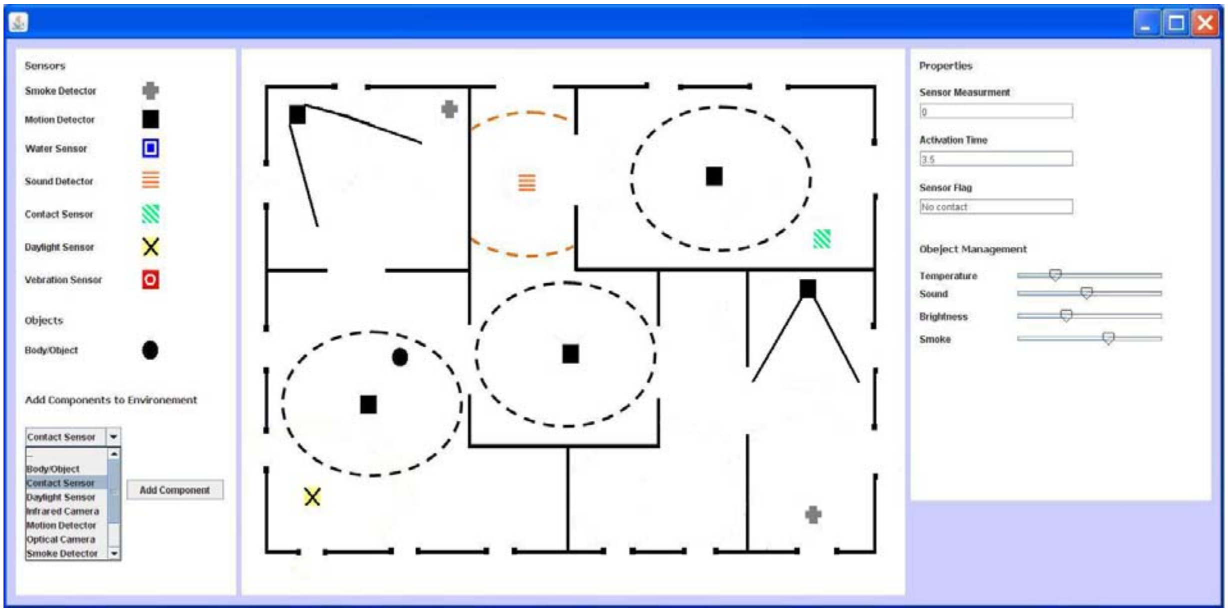

Figure 4. Simulation tool interface

In our study of multi-sensor data fusion, we implement a simulation tool that helps us construct a virtual smart environment, as shown in Figure 4. The smart environment has basically different types of sensors such as: motion detector, smoke detector, daylight sensor, and other types of sensors. In addition to sensors, there are objects that can be moving around to generate case scenarios where motion is a factor to be considered. Emergency cases such as fire or flood can be studied using the implemented simulation tool. This tool is implemented using JAVA and it facilitates the study of multiple scenarios because the user can chose any type of sensors implemented in the tool as well as manage the environment's state such as increasing the temperature (fire case) or adding moving objects or water (flood scenario). A specific set of attributes must be defined for each sensor. These may include range, angle, sensitivity, and direction. Every sensor has a detection area and detection occurs when the coverage area and attributes of a given object overlap with the detection range and sensitivities of a given sensor. The simulation tool is our main data generator where sensors' flags and data are fed to the fusion engine where decision making process takes place.

CASE DISCUSSION & CONCLUSIONS

The data fusion field has provided means for better decision making in recent years. In many emergency or rescue cases, data fusion can effectively contribute in making better decisions more efficiently. In this paper, we investigate the use of information fusion in a real-time decision making process where the source of information is a heterogeneous wireless sensor network of visual and non-visual sensors. Here, we discuss a case scenario in order to illustrate how our information fusion technique and the simulation environment can help with a challenging, real-time decision making problem

In an emergency case, temperature sensors, smoke detectors, and cameras detect the presence of the fire. In this sense, information collected from a temperature sensor and a smoke detector can be combined together to detect any apparent changes in the covered area. If the information from the sensors shows a change in temperature and an existence of smoke, the camera in the concerned area can be triggered to give a visual idea about what really is happening there. Alarms can also be triggered to allow the system controller to take the appropriate actions to evacuate the building for example. The way every sensor collects its data depends on the type of the sensor. For example, the temperature sensor detects an increase in the temperature of the room in case of a starting fire. The increase in temperature might not necessarily imply the existence of a fire, but it can also be associated to the over-heating of the computers in case the room includes computers. Therefore, the camera can be triggered at this level before the smoke detector detects smoke particles in the room. On the other hand, the system can also be programmed to wait for the smoke detector to detect first the smoke particles in the room before the camera can be triggered to investigate the issue.

In this project, we have shown that Dempster-Shafer data fusion technique can be effectively used in a smart environment with a heterogeneous, inter-dependent set of sensors. We have been able to generate statistically independent input for the Dempster-Shafer fusion model and demonstrate the effect through a simulation tool. As a next step, we plan to further test our quality methods on different types of sensors in different environments. Agricultural environments make a good candidate where there is a variety of sensors being used and where data quality is very important.

REFERENCES

[1] Challa, S., Koks, D., An Introduction to Bayesian and Dempster-Shafer Data Fusion, http://robotics.caltech.edu/~jerma/research_papers/BayesChapmanKolmogorov.pdf, retrieved October 28, 2009.

[2] Dempster-Shafer Theory, http://en.wikipedia.org/wiki/Dempster-Shafer_theory , retrieved October 28, 2009.

[3] Fisher C., Eitel L., Richard W., and Shobha C., *Introduction to Information Quality*. Cambridge : MITIQ, 2006.

[4] H.B, Mitchell, *Multi-Sensor Data Fusion*. Berlin: Springer, 2007.

[5] Ramsden, J., *Bioinformatics: an Introduction*. Dordrecht: Kluwer Academic Publishers, 2004.

[6] Shafer, G., Dempster-Shafer Theory, http://www.glennshafer.com/assets/downloads/articles/article48.pdf, retrieved October 28, 2009.

[7] Snidaro, L., Claudio, P., Christian, M.,Gian, F., Visual Sensor Technology for Advanced Surveillance Systems: Historical View, Technological Aspects and Research Activities in Italy, http://www.mdpi.com/1424-8220/9/4/2252/pdf, retrieved November 12, 2009.

[8] Waltz, Edward, James, L. ,*Multisensor Data Fusion*. Norwood: Artech House, Inc, 1990.

[9] Guo, Shuo, Zhong, Ziguo, He, Tian. "FIND: Faulty Node Detection for Wireless Sensor Networks". University of Minnesota, Twin Cities. May 13th, 2010 <http://www-users.cs.umn.edu/~sguo/paper/sensys09.pdf>

[10] Mukhopadhyay, Shoubhik, Panigrahi, Debashis, Dey, Sujit. "Data aware, Low cost Error correction for Wireless Sensor Networks". University of California, San Diego. May 13th, 2010

[11] "Cricket v2 User Manual". MIT Computer Science and Artificial Intelligence Lab. June 30th, 2010 <http://cricket.csail.mit.edu/v2man.pdf>.

[12] Desai, Rattan. "Indoor Localization and Surveillance using Wireless Sensor Network and Pan/Tilt Camera". September 3rd, 2010 <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5426659>