# ERROR-TOLERANT MULTI-MODAL SENSOR FUSION (SHORT PAPER)

*Farinaz Koushanfar[*], Sasha Slijepcevic[†], Miodrag Potkonjak[†], Alberto Sangiovanni-Vincentelli[*]*

[*]EECS Dept. UC Berkeley, Berkeley, CA, USA

[†]CS Dept. UCLA, Los Angeles, CA, USA

{farinaz,alberto}@eecs.berkeley.edu, {sascha,miodrag}@cs.ucla.edu

## ABSTRACT

Embedded sensor networks (ESNs) are one of the prime candidates for widely used ubiquitous computing systems that will bridge the gap between computing and physical worlds. One of the most important generic ESN tasks is multi-modal sensor fusion, where data from sensors of different modalities are combined in order to obtain better information mapping of the physical world. One of the key prerequisites for all ESN applications, including multi-modal sensor fusion, is to ensure that all of the techniques and tools are error- and fault-tolerant while maintaining low cost and low energy consumption.

We address the problem of multi-modal sensor fusion (MSF) by developing two generic schemes that are sufficient to solve the MSF problem for a majority of common types of sensors. The first scheme assumes binary sensors; the second considers multilevel sensors. For binary sensors, we have developed a heterogeneous back-up scheme, where one type of resources is substituted with another. For multi-level sensor fusion, we consider a system of sensor readings, where the sensors are of different types. The sensor readings are not completely independent in the sense that the computational part of the system already has a relation model that defines the correlations between different sensor measurements. The multi-level sensor fusion then exploits the correlations between the faulty measurements, and finds the measurement points that minimize the overall error of the model. For each technique, we present efficient algorithms and demonstrate their effectiveness on a set of benchmark examples.

## 1. INTRODUCTION

Embedded sensor network (ESN) is a large-scale distributed embedded network that consists of numerous wirelessly connected nodes. Each node is equipped with a certain amount of sensing, actuating, computation, communication, and storage resources. Following vision of a ubiquitous computing environment [9], ESNs provide inexpensive and pervasive bridge between physical and computational worlds. The bridge is built through the multi-dimensional sensing capabilities of the nodes in the system (e.g., light sensing, acoustic sensing, seismic sensing) and collaborative computing among the nodes. At the same time, ESN has a number of important system design considerations including low power, low cost, real-time and in particular, reliability and fault-tolerance. Sensor-based networks will often operate in potentially hostile, or at least unconditioned environments. They will have continuous mode of operation, higher structural complexity, and components such as sensor and actuators, which have significantly higher fault rates than the traditional semiconductor integrated circuits-based systems.

We emphasize the importance of heterogeneous fault-tolerance techniques, where a single type of resource backs up different types of resources. The key idea is to adapt application algorithms to match the available hardware and the applications needs. We envision that each of five primary types of resources: computing, storage, communication, sensing and actuating can replace each other with suitable change in application software. For example, if communication bandwidth is reduced and all of the computation power is available, the system can compress data using more computationally intensive compression schemes. We focus our attention on how to back-up one type of sensor with another. There are two main reasons for this decision. The first is that technology trends indicate that sensing has by far the highest fault rates. The second is that there is a wide consensus that multi-modal sensor fusion is the key for successful and widespread use of embedded sensor networks.

## 2. FAULT-TOLERANT MULTIMODAL SENSOR FUSION FOR BINARY SENSORS

The problem of fault-tolerant multi-modal sensor fusion for digital binary sensors is informally illustrated using the example from Fig. 1. We have a set of objects $G$={A, B, C, D, E}. Each object has two different attributes, $X$ and $Y$, which are observed by sensors of types $x$ and $y$, respectively. Each object is unique, i.e. no two objects have identical values for both, $X$ and $Y$. The goal is to use the attributes $X$ and $Y$ to uniquely identify each object, using the least cost. For the sake of simplicity, we assume that both $x$ and $y$ sensors have the same cost.

We map the objects from $G$ into a 2-dimensional space as shown in Fig. 1. The sensors that observe the attribute $X$ are assigned values $x_i$, and the sensors observing $Y$ are assigned values $y_i$. The output of a sensor set to $x_i$ is 0, if the

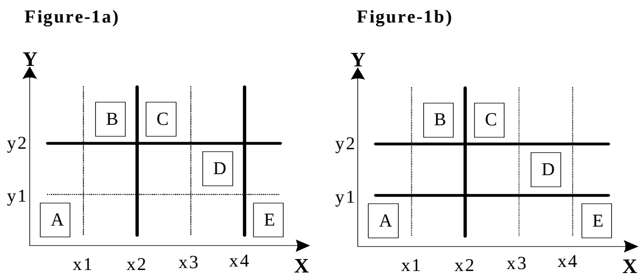



**Figure 1 - An example of the MSF for binary sensors**

objects attribute is less than $x_i$ (or $y_i$) in *X* (or *Y*) dimension, and is 1 otherwise. The fault model for sensor assumes a sensor is either functional and reports the correct value, or is indefinitely stuck at one value. The sensors set to values $x_i$ and $y_i$ are shown by the lines $X=x_i$ and $Y=y_i$. The MSF problem of uniquely identifying each object in the 2-dimensional space using minimum number of sensors maps to the problem of selecting the minimum number of lines $X=x_i$ and $Y=y_i$ such that each object is within a unique grid unit formed by the lines. Fig. 1a and 1b show two such solutions. In 1a, the solution uses two *X* sensors $\{x_2, x_4\}$ and one *Y* sensor, $y_2$. However, in 1b, the solution uses one *X* sensor $x_2$ and two *Y* sensors $y_1$ and $y_2$. Thus, in Fig. 1b, the sensor $y_1$ replaces the sensor $x_4$ from 1a. This suggests a heterogeneous back up scheme for the binary multimodal sensor fusion, where sensors of different types replace each other and cover each other faults.

In the rest of the section, we formally formulate the multi-modal sensor assignment problem and propose an ILP formulation and an efficient heuristic to solve it.

The MULTI-MODAL SENSOR ALLOCATION PROBLEM can be formulated in the following way:

INSTANCE: Set *A* of points $p_i$ $(x_{i1},..,x_{im})$, in *m*-dimensional space where $1\leq i \leq n$, a positive integer *J*, set *H* that consists of *m(n-1)* [*m-1*]-dimensional hyperplanes, each of which is perpendicular to one of the *m* axes. Each hyperplane is separating two points $p_i$ and $p_j$ that have the closest coordinates along the particular axis to which the hyperplane is perpendicular.

QUESTION: Find a subset of hyperplanes *H*, such that any two points $p_i$ and $p_j$ are separated by at least one of the selected hyperplanes and the cardinality of *H* is at most *J*.

The proof of the NP-completeness of the MMSA problem is outlined in [6].

We have developed two different techniques to solve the allocation problem: ILP-based and simulated annealing based. ILP solvers are attractive since they guarantee optimal solution. In addition, many smaller instances of practical importance can be solved using this approach. In the cases when ILP is not applicable, we can use simulated annealing as an alternative optimization mechanism. The ILP formulation for the problem can be stated as follows: INPUTS: set of N *m*-dimensional points $p_i(x_{i1},..., x_{im})$, $1\leq i\leq n$. Set of all possible tests *T*, with elements $t_k$, $1\leq t_k\leq m(n-1)$. The tests $t_k=(l(n-1)+1)..(l+1)(n-1)$ test the values in the dimension *l*, $1\leq l\leq m$, each separating two closest point in that dimension. The cost of each test $t_k$ is $c_k$.

We define the variable $X_k$ as follows:

$X_k=1$ if test $t_k$ is selected

$X_k=0$ otherwise.

The objective function is to minimize the total cost of all of the selected tests:

$$\text{OF:}\quad \sum_{k=1}^{m(n-1)} X_k \cdot c_k$$

For each pair of points, $p_i$ and $p_j$, there should be at least one test that has a different outcome when applied to these two points. We define an auxiliary matrix $A[n\times k(m-1)]$ with constant elements $a_{ik}$,

$a_{ik}=1$ if the test $t_k$ produces 1 on point $p_i$

$a_{ik}=0$ otherwise.

We need a linear expression that produces 0, if a test produces identical results on the two points $p_i$ and $p_j$, and 1 otherwise. One such expression that has the required property is $X_k\times(a_{ik}+a_{jk}) \times (1-a_{ik}\times a_{jk})$. Therefore, to have a different test result on each set of two points $p_i$ and $p_j$, we write the following constraints:

For each pair of points $p_i$ and $p_j$,

$$\sum_{k=1}^{m(n-1)} x_k \cdot (a_{ik} + a_{jk}).(1 - a_{ik}.a_{jk}) \geq 2$$

The simulated annealing (SA) solution uses the standard SA code. The four components of simulated annealing (i.e., moves - neighborhood structure, objective function, cooling schedule, and stopping criteria) are defined for the allocation problem. A move is the replacement of one sensor with another sensor of the same type. The goal is to maximize an objective function. We use the standard geometric cooling schedule. Finally, as a stopping criteria, we use the user specified number of steps in which the improvement did not occur. We first propose the number of sensors that is lower bound on the potential solution, as our initial solution. We calculate this bound assuming that all dimensions have the same number of sensors and each *n*-dimensional compartment will eventually contain one point. After that, we run the SA algorithm. During this running process, we modify the move so that one type of sensor can be replaced with another type of sensor. We accumulate statistics about which type of sensor helps the most to improve objective function after each move, and use this information to decide which type of sensor to add or remove.

## 3. MSF FOR MULTI-LEVEL SENSORS

There is a wide class of sensors that have multi-level output value. The fault model for binary output sensors represents a sensor as being stuck at one value. For multi-level sensors, however, a faulty sensor can have an erroneous value that is varying and at a different level, compared to the actual value. Thus, for multi-level sensor fusion, we consider a system of erroneous sensor readings, where the sensors are of different types. The key observation is that a known model defines the correlations between different sensor measurements. This model can be either statistical or analytical. In the statistical modeling approach, we conduct a statistical learning process to find the correlation between the sensor readings and construct a meaningful model. In analytical modeling, we use the equations as the correlating constraints between the sensor outputs.

Perhaps the best way to introduce our MSF approach for multilevel sensors is to take a closer look at a small example. The example uses analytical models as the correlating system between the measurements. We have an object *O* that moves along its trajectory that includes points $p_i$ $\{1 \leq i \leq k\}$ in an embedded sensor network that consists of a number of nodes. We assume that we have four types of sensors: RSSI-based or acoustic signal-based distance discovery, speedometer, accelerometer, and compass, which measure the angle in a 2D physical space. Three distance measurements can be used to locate the object *O* in any particular moment. Euclidian space, Newton mechanics, and trigonometry laws can be used to establish relationships between measurements of different modalities. In Fig. 2, the moving object is a person, while the black dots represent sensors. The light shaded person shows the sampling points of the moving object within the field, while the present location of the object is shown in a black shade.

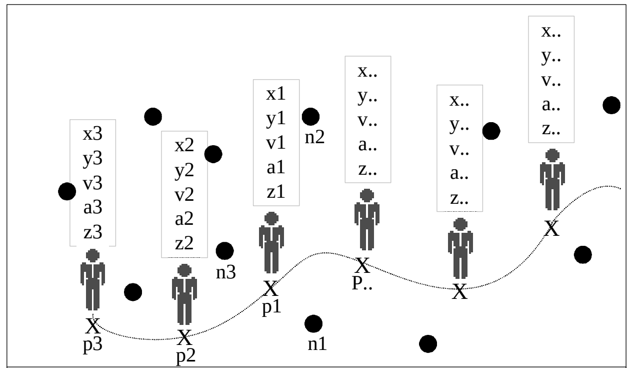
At each point, the person could communicate to the sensors that are within its communication range and form a multilateration equation with them. The multilateration equations 1-3 are just shown here as a sample for the point $p_1(x_1,y_1)$ from which *O* can communicate to three nodes $n_1(s_1,t_1)$, $n_2(s_2,t_2)$, and $n_3(s_3,t_3)$ in its vicinity. $R_1$, $R_2$ and $R_3$ are the measured distances from the point $p_1$ to the nodes $n_1$, $n_2$ and $n_3$, respectively.

$$(x_1 - s_1)^2 + (y_1 - t_1)^2 = R_1^{\,2} \qquad \textbf{Eq 1}$$

$$(x_1 - s_2)^2 + (y_1 - t_2)^2 = R_2^{\,2} \qquad \textbf{Eq 2}$$

$$(x_1 - s_3)^2 + (y_1 - t_3)^2 = R_3^{\,2} \qquad \textbf{Eq 3}$$

Assuming that at each point the person needs to do multilateration with at least three nodes to get its location information, we would have a minimum of nine equations for multilateration of the three points $p_1$, $p_2$, and $p_3$ shown in Fig. 2. On the other hand, Newton Equations of Motions




(xi,yi) - location coordinates, vi - speed, ai - acceleration, zi - compass.

**Figure 2 - Multi-level, multi-modal sensor fusion used in tracking**

yield relationship between the measurements from the accelerations (denoted by *a*) and velocity (denoted by *v*), and the sampling time between two consecutive points. For the sake of simplicity we assume that the time interval between two consecutive samples is fixed and is equal to $\Delta t$. Equations 4-7 reveal the relationship between the velocity and acceleration, and distance between the pairs of points $p_1$-$p_2$ and $p_2$-$p_3$ respectively.

$$\sqrt{(x_1 - x_2)^2 - (y_1 - y_2)^2} = \frac{1}{2} \cdot a_1(\Delta t)^2 + v_1(\Delta t) \qquad \textbf{Eq 4}$$

$$\sqrt{(x_3 - x_2)^2 - (y_3 - y_2)^2} = \frac{1}{2} \cdot a_2(\Delta t)^2 + v_2(\Delta t) \qquad \textbf{Eq 5}$$

$$a_1.\Delta t = v_1 - v_0 \qquad \textbf{Eq 6}$$

$$a_2.\Delta t = v_2 - v_1 \qquad \textbf{Eq 7}$$

In addition, compass output (denoted by *z*) is also related to the location coordinates of the points, as shown in equations 8 and 9 below.

$$z_1 = \tan^{-1}\left( \frac{y_2 - y_1}{x_2 - x_1} \right) \qquad \textbf{Eq 8}$$

$$z_2 = \tan^{-1}\left( \frac{y_3 - y_2}{x_3 - x_2} \right) \qquad \textbf{Eq 9}$$

The key observation is that we have more equations (15 equations consisting of nine equations from multilaterations, plus another six from Newton's equations and compass) than variables (12) that may have errors. So, if one of sensor is not functioning, we can calculate it from the established system of equations. Also, for each variable, we can find how much it has to be altered in order to make the whole system of equations maximally consistent. The variables that have to be altered the most are most likely measured by faulty sensors. Therefore, one way to identify and correct sensor measurements is to try all scenarios where exactly one type of sensor measurements is not taken into account and compare the maximal error in the system.

## 4. EXPERIMENTAL RESULTS

The evaluation of the new approach and algorithms is a challenging task: the MSF problem is a NP-complete problem there are no established benchmarks and previously published results for the addressed problem. Nevertheless, it is still possible to evaluate the proposed algorithms in sound and convincing way.

During the evaluation process, we evaluate the algorithms for sensor assignment and allocation. First, we generate an instance of the problem, for which the optimal solution is known, in the following way. We assume that the cost of all *m* types of sensors equal. We first construct a solution. The solution consists of the equal number of sensors in each direction. Next, we place exactly one object in each of the *m*-dimensional hypercube defined by the selected sensors. Each object is placed in random location within the hypercube. It is easy to see that the selected sensors are optimal. We can additionally obscure solution by not placing objects in a small number of the hypercubes or by not using exactly the same number of sensors in each dimension. Note that we can also combine smaller arbitrary instances solved by our ILP approach to create large new instances of the problem with known solution.

The evaluation of the simulated annealing-based algorithm is shown in Table 1. The first two columns indicate the number of objects and the number of dimensions. The next three columns indicate the size of solution generated by the simulated annealing program in 2 minutes on 1 GHz Pentium processors. The final column indicates the size of the optimal solution. Each row represents results from 10 different instance of the problem with the same characteristics.

| Number of points | Dimension | SA-solution | | | Optimal |
|---|---|---|---|---|---|
| | | worst | median | best | |
| 100 | 2 | 22 | 20 | 19 | 18 |
| 100 | 3 | 15 | 13 | 12 | 12 |
| 200 | 2 | 33 | 30 | 28 | 28 |
| 300 | 3 | 25 | 20 | 18 | 18 |
| 500 | 4 | 23 | 19 | 16 | 16 |
| 800 | 4 | 26 | 22 | 19 | 18 |
| 1000 | 5 | 25 | 20 | 17 | 15 |

**Table 1 – experimental results for the Simulated Annealing (SA)-based algorithm**

## 5. RELATED WORK

ESN have recently emerged as a premier research topic. A number of high profile applications for wireless sensor networks have been envisioned [9][3]. Fault tolerance in measurements by a group of sensors, was first studied by Marzullo [7]. Marzullo proposed a flexible control process program that tolerates individual sensor failures. Issues addressed include modifying specifications in order to accommodate uncertainty in sensor values and averaging sensor values in a fault-tolerant way. In [5], an algorithm is presented that guarantees reliable and fairly accurate output from a number of different types of sensors when at most *k* out of *n* sensors are faulty. The results of the scheme are applicable only to certain individual sensor faults and traditional networks. However, they do not address the reliability issues that are induced by the ad-hoc nature of the wireless sensor networks.

Multi-sensor data fusion is a problem that recently has attracted a great deal of attention in a number of scientific and engineering communities [1][8][4]. Majority of these works are restricted to sensor fusion of sensors of the same modality. Constraints, in addition to statistical models and analytical equations, are one of main building blocks for our approach. Constraint-based sensor fusion for vision has been advocated in [2].

## 6. CONCLUSION

We have developed a new approach to multimodal fusion that is a very important task in embedded sensor networks. The key idea is to use one type of sensor to back-up sensors of different types by exploiting flexibility during multi-modal sensor data fusion. We formulated the problem for two different types of sensors (binary and multi-level), established computational complexity of associated problems, and have developed algorithms to solve them. Finally, we have demonstrated the effectiveness of our approach and algorithms on a set of illustrative examples.

## 7. REFERENCES


[1] R. R. Brooks and S. S. Iyengar, "Multi-Sensor Fusion: Fundamentals and Applications With Software," Prentice-Hall, 1997.

[2] J. J. Clark and A. L. Yuille, Data Fusion for Sensory Information Processing Systems, Kluwer, 1990.

[3] D. Estrin, R. Govindan, and J. Heidemann. "Embedding the Internet: Introduction," Communications of the ACM, vol. 43, no. 5, pp. 38-42, 2000.

[4] G.D. Hager, "Task-Directed Sensor Fusion and Planning - A Computational Approach," Kluwer, 1990.

[5] D.N. Jayasimha, "Fault tolerance in multi-sensor networks," IEEE Transactions on Reliability, vol. 45, no.2, pp. 308-15, June 1996.

[6] F. Koushanfar, M. Potkonjak, A. Sangiovanni-Vincentelli. "Fault tolerance techniques in wireless ad-hoc sensor networks", UC Berkeley technical reports, 2002.

[7] K. Marzullo, "Tolerating failures of continuous-valued sensors," ACM Transactions on Computer Systems, vol. 8, no.4, pp. 284-304, November 1990.

[8] P. K. Varshney, "Distributed Detection and Data Fusion," N.Y.: Springer-Verlag, 1997.

[9] M. Weiser, "The Computer of the Twenty-First Century," Scientific American, pp. 94-100, September 1999.