Orthogonal Organized Finite State Machine Application to Sensor Acquired Information

Brian J. d'Auriol, John Kim, Sungyoung Lee, and Young-Koo Lee

Department of Computer Engineering, Kyung Hee University, Korea
dauriol@acm.org, johnkim_korea@yahoo.ca, sylee@oslab.khu.ac.kr,
yklee@khu.ac.kr

Abstract. The application of the Orthogonal Organized Finite State Machine (OOFSM) to the representation of data acquired by sensor networks is proposed. The OOFSM was proposed in earlier work; it is succinctly reviewed here. The approach and representation of the OOFSM to sensor acquired data is formalized. The usefulness of this OOFSM application is illustrated by several case studies, specifically, gradients, contouring and discrete trajectory path determination. In addition, this paper informally discusses the OOFSM as a Cellular Automata.

1 Introduction

Finite State Machines (FSMs) have a long history of theoretical and practical developments. In brief simplicity, an FSM is characterized by a set of states and a set of transitions between these states, often together with definitions of the set of start and terminal states as well as perhaps with other attributes. The majority of FSMs in the literature do not consider the spatial relationships between states. In [1], an orthogonal arrangement of states is considered: this is termed an *Orthogonal Organized Finite State Machine (OOFSM)*. There are several advantages of such an organization including the definition of indexing and selection functions to select regions of interest as well as the state space discretization of continuous complex dynamic systems [1].

The intent of the OOFSM as developed in [1] is to realize a discretized representation of a continuous complex dynamic system. In particular, trajectories in the continuous system are represented by a sequence of labeled transitions between states in the OOFSM, these labels are in fact based on the index (metric) space. The goals of the original work include the understanding of the behavior of the regions that trajectories pass through. In the OOFSM representation, these regions can be identified by the indexing/selection functions. Ultimately, one of the aims is to predict trajectory evolutions towards cascading failure states.

Sensor networks are especially designed for data acquisition. Sensor networks can be wired or wireless, static or mobile, and may have other properties such as autonomic, low-power budgets and small physical size [2]. Sensors may be placed in a physical environment that is modeled by a dynamic system. In such cases, the sensors provide observations of the partial or full state space of the dynamic system.

V. Malyshkin (Ed.): PaCT 2007, LNCS 4671, pp. 111–118, 2007.

The earlier work is extended in a new direction in this paper. First, we consider the problem of observable data given an OOFSM representation. In particular, we consider sensor network acquired data and treat these data points as observations. In this work, we generate the acquired data via simulation, although, empirically obtained data could also be used. Our objectives are: a) representing the observable data as a discrete labeled trajectory in an OOFSM, b) describing the discrete regions of interest and/or behavior associated with these discrete trajectories. We do not consider the problem of relating the observable data in the discrete space back to a continuous system in this paper.

This paper is organized as follows. The next section, Section 2, reviews the definition of OOFSM based on [3] and is provided here as the succinct formal definition of the OOFSM abstraction. Section 3 describes the approach and methodology used in this paper. Section 4 describes several applications of the OOFSM applied to sensor acquired data. The relationship with Cellular Automata is discussed in Section 5. Technological aspects are discussed in Section 6. Conclusions are given in Section 7.

2 Review [3]

Orthogonal Organized Finite State Machines (OOFSM) [1] represent a lattice partitioned, and therefore a discretized, state space of a dynamic system. Formally, it is defined by $M = (Y, \mathcal{L}, \overline{V}_Y)$. A lattice partitioning $\mathcal{L}$ applied to an n dimension state space $X = \{x_1, x_2, \ldots, x_n\}, x_i \in \mathbb{R}$ leads to a set of discretized states $\mathcal{L} : X \rightarrow Y$ where $Y = \{y_0, y_1, \ldots, y_{o-1}\}$ for some finite o. In general, $\mathcal{L}$ defines a set of partition boundaries $P = \{p_{i_j} | 1 \leq i \leq n, 0 \leq j \leq o-1\}$ with $p_{i_j} \in \mathbb{R}^{n-1} = (b_l, b_u)_{i_j}, b_u > b_l$. Each p_{i_j} is aligned normal with the corresponding ith state variable; $\iota(b)$ denotes this value. A discrete direction vector field $\overline{v}_j = (\ldots, a_i, \ldots)$ where $a_i = \bigcup_k v_{j_{k_i}}$ is the union of a set of discrete direction vectors $\{v_{j_k} \mid k \geq 1\}$ in state y_j of Y; $a_i \in \{-1, 0, 1\}$. The intersection of a trajectory $e \in E$ with $p_j \in P$ for a fixed j derives v_j; the intersections of all $e \in E$ with $p_j \in P$ derives $\overline{v}_j$ for a fixed j. Lastly, the set $\overline{V}_R = \{\overline{v}_j \mid j \in R\}$ for region R defines a region field; $\overline{V}_Y$ denotes some general region field. A uniform region field has the same region field for each $y_j \in R$. For convenience, elements of Y may be interchangeably expressed in terms of the dimension of the system. Figure 1 illustrates an OOFSM for: $n = 2$, uniform unit $\mathcal{L}$ so that $o = 16$ and $P = \{p_{1_0}, p_{2_0}, p_{1_1}, p_{2_1}, \ldots p_{1_j}, p_{2_j}, \ldots p_{1_{15}}, p_{2_{15}}\}$ such that $p_{1_0} = (b_{l_{1\,(0,0)}}, b_{u_{1\,(0,0)}})$ where $\iota(b_{l_{1\,(0,0)}}) = 0$ and $\iota(b_{u_{1\,(0,0)}}) = 1$ (i.e., the values on the x_1 axis corresponding with the lower and upper boundaries of the 'vertical' partition pair comprising the 'left' and 'right' sides of state $y_{0,0}$) and so forth with $X = \{x_1, x_2\}$, and $\overline{V}_Y = \overline{V}_{R_1} \cup \overline{V}_{R_2}$ where $\overline{v}_{0,3} = (\{0\}, \{0\})$ defines the uniform region field $\overline{V}_{R_1}$ for $R_1 = \{y_{k,3} | 0 \leq k \leq 3\}$ and $\overline{v}_{0,0} = (\{0\}, \{1\})$ defines the uniform region field $\overline{V}_{R_2}$ for $R_2 = \{y_{k,l} | 0 \leq k \leq 3, 0 \leq l \leq 2\}$ (i.e., there are two uniform region fields with the first being null (terminal states) associated with the 'top row' and the second being 'up-wards only' associated with the remaining states).

3 Approach and Methodology

The initial inputs are obtained from a sensor network. Such a sensor network typically has many spatially distributed sensors that acquire data at specific times. In general, the data has two fundamental properties: *structure* and *value*. Its structure is derived from two possibilities, either the physical placement of the sensors determines physical coordinates (e.g. x,y,z coordinates, GPS coordinates, etc.) or the data itself has some *apriori* defined structure (e.g. vectors, tensors, etc.) Its value refers to the semantics of the actual measurement. Values have ranges (e.g. an interval in $\mathbb{R}$). These properties have been noted elsewhere in the literature, for example, in data visualization [4, 5].

Let the set $D^* = \{D_1, D_2, \ldots\}$ denote a collection of temporal organized data values where D_i denotes all the sensor data at some ith time. And, $D = (d_1^s, d_2^s, \ldots, d_{m_1}^s, d_1^v, d_2^v, \ldots, d_{m_2}^v)$ where d_i^s, d_i^v denotes, respectively, structure and value components and each d_i^s, d_j^v such that $1 \leq i, j \leq m_1, m_2$ is in the, respectively, maximal measurement range of the associated sensor's data organization and data value.

If a dynamic system is known, than X and $\overline{V}_Y$ are also given. Given X, then $d_i \mapsto x_j$ for $1 \leq i \leq m$ and $j \in [1..n]$, that is, there are m observable states in an n dimensional state space, $m \leq n$.

For the case where there is no dynamic system or it is unknown, both X and $\overline{V}_Y$ need be determined from the sensor data. Let $\bar{X}$ denote the determined space (the shift of notation provides the semantics that no underlying dynamic system is involved). Consider the two cases:

1. $\bar{X}$ is determined from the data's structure: $d_i^s \mapsto x_i$ for $1 \leq i \leq m_1$, $n = m_1$. Here, the structure states are observable and the state space directly represents the organization of the data. $\mathcal{L}$ reflects the OOFSM structuring imposed on the organization of the sensors. Transitions through this space reflect ordered selections of the value elements. Let some arbitrary bijective function $f(D^*) \mapsto \overline{V}_R$, that is f applied to the sensor acquired data results in a set of state transitions. The choice for f is motivated by seeking *logical* orderings of the sensor data subject to the nearest neighbor connections mandated by the OOFSM.
2. $\bar{X}$ is determined from the data's value: $d_i^v \mapsto x_i$ for $1 \leq i \leq m_2$, $n = m_2$, that is, all states are considered to be observable. $\mathcal{L}$ reflects the discretization over the sensor measurements: in this paper we assume that the discretization results in well-behaved transitions, for example, to ensure nearest-neighbor state changes. In [1], a partial region field was discussed as a model for transitions determined by a finite sub-set of all possible trajectories in the underlying dynamic system. Similar here, we can say that $D \mapsto y_k$ for some kth state and $D^* \mapsto \overline{V}_R$ where the region field is a partial field. The larger D^*, the more state transitions may be defined and the more complete the region field becomes.

Now, Y has been determined. Each state in Y is labeled; a simple practical method is to select b_l from all P in $\mathbb{N}^m$ (e.g. $y_{0,0}, y_{1,0}$, etc. in Figure 1).

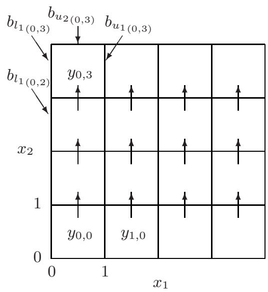

Fig. 1. Illustration of state space definitions, $n = 2$, $o = 16$ and uniform unit partition $\mathcal{L}$

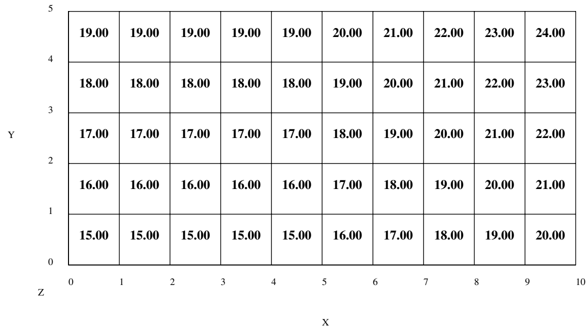

Fig. 2. Temperature distribution in the corresponding OOFSM

4 Applications to Sensor Acquired Data

This section consists of simulated examples as case studies. The first two case studies are based on a simulation of 50 two-dimensional lattice-arranged temperature sensors constructed with each sensor's location placed such that the location represents the center of the state determined by $\mathcal{L} = \{$boundaries intersecting the axes at ordinal values$\}$. Hence, the data's structure consists of x,y coordinates and its value is a scalar in $\mathbb{R}$ (we ignore the operating ranges of sensors here). The third case study eliminates the structure and instead, considers the state system to be composed of discretized ranges over each sensor's value. This more closely represents the view-point adopted by observable states associated with a dynamic system.

4.1 Gradient

A temperature gradient is considered in this case study. The temperature values are distributed according to simple (linear) assumptions (since we are not interested here in simulation accuracy with thermo-models). Figure 2 shows the raw data temperature distribution in the corresponding OOFSM while Figure 3 shows a typical visualization of the temperature distribution in the OOFSM. These figures are generated by AVS/Express visualization software. Neither the raw data nor the visualization provide sufficient clarity regarding the possible bifurcation in the system; as shown dramatically in Figure 4. In this figure, the uniform vector fields corresponding to the transitions from low-values to high-values are plotted; hence two regions of behavior are identified.

4.2 Contouring

A temperature contour is considered in this case study. The distribution, shown in Figure 5, is somewhat modified from that used earlier (the change better clarifies the results). Figures 6 and 7 show typical visualizations of the data, the first

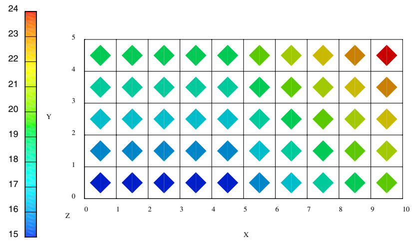

Fig. 3. Temperature visualization

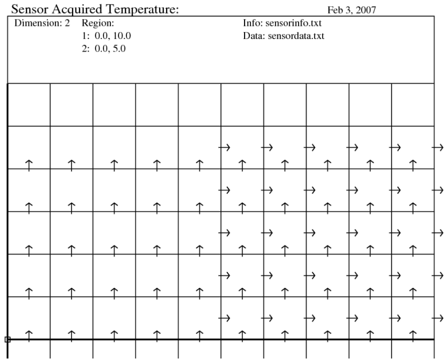

Fig. 4. Uniform vector fields in the OOFSM

uses a standard scatter-to-uniform 2ed-order interpolator to fill-in data values in-between the sensor points, and the second graphs the contours based on the interpolated values. As before, AVS/Express software is used. The corresponding OOFSM in which the contours are represented by state transitions only to neighboring states of the same value is shown in Figure 8. The discontinuity in the trajectory path between states 6,2 and 7,1 containing the data value of 17 occurs due to the non-neighboring transitions, exactly in this case, corner-wise. A refined lattice partitioning would usually take care of this situation. Furthermore, we could allow the corner-wise transition to pass by the corresponding neighboring states (e.g. via state 6,1 or 7,2) Note that there were many such corner-wise transitions in the previous case study.

4.3 Temperature System

Let the lattice partitioning impose a discretization over the ranges of sensor acquired temperatures. Since each sensor uniquely monitors its environment, each sensor provides an independent temperature measurement. For each such measurement, the discretization reflects a single dimension of the overall state space; hence, the number of temperature dimensions equals the number of sensors. Such high dimensional state systems are very common in dynamic systems.

For this discussion, we assume two sensors, hence a two-dimensional state space. Let us choose a partitioning such that each state is unit temperature as illustrated in Figure 9. This figure shows a hypothetical trajectory as might be determined by a sequence of temperature measures over time.

5 Cellular Automata Discussion

There is a close relationship between the OOFSM described in this paper and cellular automata (CA). In [6], four features are identified to characterize a CA: geometry, cell neighborhoods, cell states and local transition rules for cell

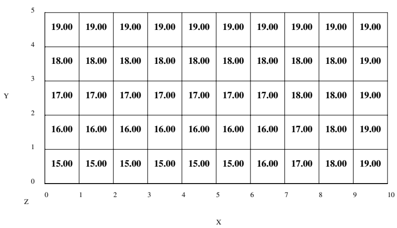

Fig. 5. Temperature distribution in the corresponding OOFSM

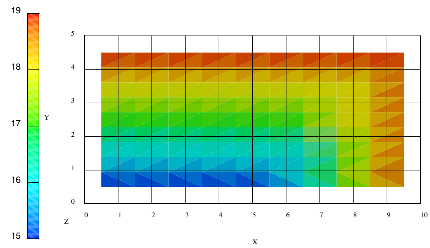

Fig. 6. Temperature visualization, standard 2ed-order interpolation from the scatter data

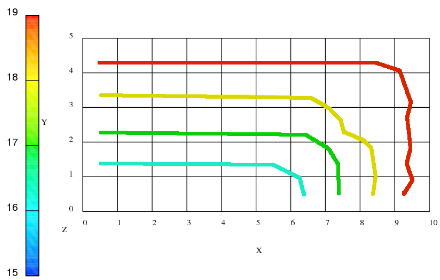

Fig. 7. Temperature visualization showing contouring

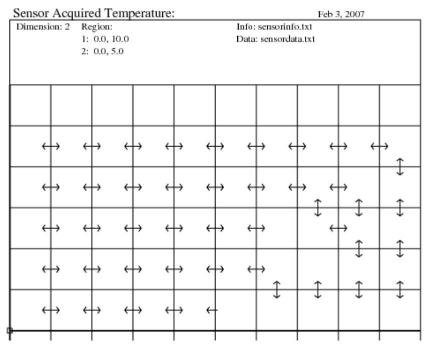

Fig. 8. Uniform vector fields in the OOFSM

state changes. The orthogonal structure of the OOFSM, i.e. as given by $\mathcal{L}$, corresponds with an n-dimension CA of the orthogonal neighborhood type. The neighborhood is defined by all the transitions into a given cell state, that is, all the states in the OOFSM with at least one discrete direction vector defining a transition from that cell state to the current cell state. For example, in Figure 1, the neighborhood of $y_{i,j}$ is $y_{i,j-1}$ for $0 \leq i \leq 3, 1 \leq j \leq 3$. Cell states and the local transition rules are contextually defined by the applications. In this paper, the cell states reflect properties of the sensor acquired data. The local transition rule is a cell-centric interpretation of the *factors* that determines the discrete direction vectors defining the neighborhood. This refers to $f(D^*)$ for Case 1 and D^* for Case 2 of Section 3. This completes the informal description of the OOFSM as a CA.

An example for the Gradient application discussed in Section 4.1 is given. Recall, the sensor data is D^* with temperature d_1^v and that f defines transitions from low to high temperature values. Let $t_k = d_1^v$ for the kth cell state (a matter of convenience). Then, the local rule may be defined as $\max_{y_k \in \overline{Y}}(t_k) < t_j$ where $\overline{Y}$ denotes the set of states of the neighborhood. For the particular temperature

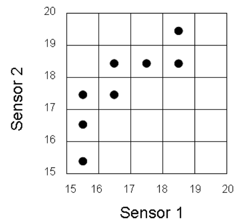

Fig. 9. Two temperature sensor state space, a hypothetical trajectory is shown

values given in Figure 2, a specific local rule could be: select any neighbor and add one to its state. The interpretation of local rule here suggests that the local rule is *representation-driven* and not compute-driven. The state values are already known, but the local rule is not. The process is to infer the local rule from the known parameters.

6 Technological Aspects

Some brief comments about the concurrency inherent in the application of the OOFSM to sensor acquired data are made in this section. A full treatment of the concurrency inherent in the OOFSM, the related CA and associated processes is beyond the scope of this paper.

Consider the computations needed for to determine $\overline{Y}$. For Case 1 of Section 3, without loss of generality, $\overline{Y}$ is computable by considering two states which share a surface. Each such pairing is independent of another (assuming that concurrent updates are handled in concept by appropriate semaphore locks). Hence, there is a high degree of inherent fine-grained parallelism. For example, in the Gradient application, Section 4.1, the pairings are: $(y_{i,j}, y_{i+1,j})$, $(y_{i,j}, y_{i-1,j})$, $(y_{i,j}, y_{i,j+1})$ and $(y_{i,j}, y_{i,j-1})$. For Case 2 of Section 3, $\overline{Y}$ is computable by considering the time sequences in D^*. When D^* is known (as for example when the data is stored at a centralized database), then there is inherent fine-grained parallelism between D_i and D_{i+1}. However, when D^* is available as a real-time stream, then the process itself is inherently sequential due to the streaming. Each $\overline{v}_k$ is local to the state k and may be stored locally in a distributed-memory multicomputer. The issues of partitioning and mapping fine-grained parallelism onto multicomputers have been well investigated (see for example [7]); past experience suggests that further performance analysis is needed.

7 Conclusion

The Orthogonal Organized Finite State Machine (OOFSM)was proposed in earlier work as a mathematical model that supported representation and visualization of dynamic systems. In this paper, its use is broadened by considering the OOFSM representation of data acquired by sensors. The usefulness of this

OOFSM application is illustrated by several case studies. Specifically, gradients, contouring and discrete trajectory path determination were studied. In addition, this paper informally discusses the OOFSM as a Cellular Automata.

This paper concentrated on the ideas behind these novel application areas of the OOFSM. Clearly, enhanced simulations and experimental results are needed to provide realistic data sets which in turn would be used in realistic evaluations of our approach. This constitutes the bulk of our intended future work.

Acknowledgements

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITFSIP (IT Foreign Specialist Inviting Program) supervised by the IITA (Institute of Information Technology Advancement).

References

1. d'Auriol, B.J.: A finite state machine model to support the visualization of cmplex dynamic systems. In: Proceedings of The, International Conference on Modeling, Simulation and Visualization Methods (MSV'06), pp. 304–310. CSREA Press, Las Vegas (2006)
2. Chong, C.-Y., Kumar, S.P.: Sensor networks: Evolution, opportunities, and challenges 9(8), 1247–1256 (2003)
3. d'Auriol, B.J., Carswell, P., Gecsi, K.: A transdimension visualization model for complex dynamic system visualizations. In: Proceedings of The 2006 International Conference on Modeling, Simulation and Visualization Methods (MSV'06), pp. 318–324. CSREA Press, Las Vegas (2006)
4. Brodlie, K., Carpenter, L.A., Earnshaw, R.A., Gallop, R., Hubbolt, R., Mumford, A.M., Osland, C.D., Quarendon, P. (eds.): Scientific Visualization: Techniques and Applications. Springer, Heidelberg (1992)
5. Ware, C.: Information Visualization Perception for Design, 2nd edn. Morgan Kaufmann, San Francisco (2004)
6. Sarkar, P.: A brief history of cellular automata. ACM Computing Surveys 32(1), 80–107 (2000)
7. d'Auriol, B.J., Bhavsar, V.C.: Generic program concurrent modules for systolic computations. In: Arabnia, H. (ed.) Proc. of the 1999 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'99), pp. 2012–2018. Las Vegas, Nevada, USA (1999)