Research Article

A Design Approach for Controlled Self-Organization-Based Sensor Networks Focused on Control Timescale

Daichi Kominami and Masayuki Murata

Graduate School of Information Science and Technology, Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan

Correspondence should be addressed to Daichi Kominami; d-kominami@ist.osaka-u.ac.jp

Received 12 December 2012; Accepted 26 April 2013

Academic Editor: J. Barbancho

Copyright © 2013 D. Kominami and M. Murata. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many researches on network control with a design principle of self-organization have been studied for large-scale networks. Since self-organized control is based on local interactions between system elements, it has high scalability, adaptability, and robustness; however, the management of the whole system is very difficult. In order to solve this problem, a controlled self-organization scheme has been proposed, which aims for desired system behavior by controlling a part of self-organized nodes. Although there are many practical proposals on the scheme, no design approach for it has ever been investigated. In this paper, we propose and evaluate a design approach for the network based on controlled self-organization, paying attention to the control timescale. Through computer simulations, we show the adaptability and stability of the proposed design approach.

1. Introduction

Future wireless sensor networks will have massive numbers of elements and should have highly scalable and adaptable properties. However, controlling such a large-scale network is very difficult challenges. For this purpose, self-organization has attracted an increasing attention due to its nature of scalability, adaptability, and robustness [1–3]. Each element in self-organization makes a decision on the basis of local interactions and local rules, which leads to the emergence of global behavior. However, this pure self-organization has some problems because of its bottom-up design [4], such as difficulty of managing the whole network and slow convergence speed after perturbations.

Practical realization of a self-organized network requires complicated emergent behavior to be manageable. However, decision-making based on local interactions in large systems results in emergent behavior, and precise management or control of such behavior is unrealistic. To solve these problems, [5] proposes controlled self-organization. The authors of that paper suggest the use of an observer/controller architecture, where an observer and a controller are responsible for correcting system-level behavior. In controlled self-organization, an external observer and controller are responsible for “external control,” guaranteeing that system behavior remains within constraints set by the system manager. The main task of the observer is to monitor system behavior by sampling information from a subset of system elements. The controller evaluates the system behavior reported by the observer and performs control actions that influence the system toward a given objective function. This observation/control loop is performed periodically to satisfy system goals. The observer/controller architecture is responsible for ensuring the desired behavior of the system, for guaranteeing high system performance, and for encouraging convergence of the system state, thus making the self-organized system manageable by controlling some of the self-organization components. Various applications of controlled self-organization are found in [6–8].

Although controlled self-organization is important for the realization of large-scale wireless sensor networks, the potential for unexpected situations due to simultaneous external and self-organized control remains poorly understood. Robustness to network topology change is also important for wireless sensor networks, where changes due to wireless channel conditions, node positions, and the number of nodes are commonplace. If communications protocols are not sufficiently flexible regarding environmental perturbations, various types of performance degradation may occur, such as data collection failures, data delivery delays, and increased energy consumption.
These perturbations and controls in each layer in the wireless sensor network architecture operate on widely different timescales. MAC layer protocols support one-hop communication, where data transmission takes a few milliseconds in most IEEE 802.15.4 sensor networks [9]. Energy-efficient MAC protocols with sleep scheduling for prolonging network lifetime are often assumed in wireless sensor networks, which raises the lower limit of one-hop communication timescale due to the sleep cycles of tens of milliseconds to seconds [10–12]. Routing layer protocols have to deal with topological changes to realize source-to-destination communications. In [13–15], static sensor nodes manage the network topology by using periodic Hello messages every several tens of seconds. The timescale of the external control in controlled self-organization should be longer than that of the routing layer because global behavior of a self-organized network arises as a result of that routing process. Thus, because these control timescales substantially differ, it is insufficient to discuss robustness within only one layer.

In this paper, we propose a design approach for a scalable and robust network based on controlled self-organization, paying attention to the control timescale. We show that a design for robustness in only one layer cannot improve various types of perturbations that cause topological changes. As a solution to these problems, we propose a controlled self-organization-based routing protocol. We apply the controlled self-organization scheme to a potential-based routing and thereby propose controlled potential-based routing (CPBR). Our study considers periodic environmental monitoring systems where sensor nodes deliver monitored data to multiple static sink nodes with CPBR. Then, we discuss how the timescale of control in the MAC, routing, and external control layers should be designed and investigate this through computer simulation.

The rest of this paper is organized as follows. In Section 2, we briefly present each layer’s control, and in Section 3, we give descriptions of perturbation models. Then, in Section 4, we explain how to design them. We present the simulation results in Section 5. Finally, we conclude our paper in Section 6.

2. Overview of the Each Layer’s Control

In this section, we give overviews of CPBR [17], and especially we discuss the control timescale in a MAC layer, routing layer, and external control.

2.1. Sleep Control in MAC Layer. One-hop communication is performed in the MAC layer, which takes several milliseconds in the most sensor network scenarios. Therefore, it is difficult to deal with perturbations that cause the topology changes with cycle of a few milliseconds or less. Moreover, in many MAC protocols in the sensor network, the sleep control is assumed, where power-saving operation is expected. For example, B-MAC [10], which is a widely known MAC protocol with the sleep control, allows nodes to sleep every tens of milliseconds to several seconds. Since each node can communicate with its neighbor nodes only when it is awake, the cycle of this sleep control means the minimum unit time of one-hop data transmissions. Hence, such frequent environmental perturbations (~1 ms) are dealt with retransmissions in the MAC layer. We use the intermittent receiver-driven data transmission protocol [16] as a MAC protocol. This protocol is one of the receiver-driven or receiver-initiated MAC protocols where nodes periodically sleep and transmit a beacon, containing an identifier (ID) of the nodes, to inform their neighbors that they are ready to receive data as shown in Figure 1. When a sender node receives a message containing an ID, it returns a send-request message (SREQ) so as to communicate with the sender of the ID.

2.2. Route Management in Routing Layer. CPBR is a kind of potential-based routing protocols, and it utilizes the proactive route management. In a potential-based routing, all nodes have a scalar value “potential.” This potential of a node is lower as the hop count from the nearby sink node is smaller. Therefore, a node only forwards data to the neighbor with lower potential than its own for delivering data toward a sink node.

In CPBR, a potential of node $n$ at time $t$, denoted by $\phi(n, t)$, is given by (1). $Z(n)$ is a set of neighbors of node $n$ and $|Z(n)|$ is the size of it. For the calculation of potentials, each node has to manage its neighbors’ potential. In order to do that, each node informs its potential to its neighbors periodically. When a node receives a neighbor’s potential, it registers the potential of the neighbor, and when it cannot receive any potential from a neighbor during a certain time period, it clears the memory of the neighbor’s potential received previously:

$$\phi(n, t + 1) = \phi(n, t) + \frac{1}{|Z(n)|} \sum_{k \in Z(n)} [\phi(k, t) - \phi(n, t)].$$  (1)

2.3. External Control. CPBR presumes a multisink sensor network and it performs global control of a potential field in order to balance the traffic loads of sink nodes, which is difficult to manage only by local interactions and rules. In CPBR, a control node, which is able to communicate with all sink nodes, is responsible for observing and controlling of potentials of all sink nodes. The control node controls potential of sink node $d$ at time $t$, denoted by $\Phi(d, t)$, via (2). $m$ is a metric for the control given by the network manager. Then, $m(d, t)$ is collected from sink node $d$ periodically (every $T_m$), and $\overline{m(t)}$ is the average of the metric at time $t$. Potentials...
3. Perturbation Model

We assume four types of perturbations that cause topological changes in the network.

3.1. Varying Wireless Channel Condition. The burst packet errors occur in various timescales as mentioned in [18]. Then, we assume that burst packet errors happen due to varying wireless channel condition according to the Gilbert-Elliot model [19]. In this model, wireless channel is described with two-state Markov chain; that is, each link has two conditions “good” and “bad,” respectively and alternates the conditions stochastically. In this paper, when a condition of a link is “good,” no bit error occurs in the link and when “bad,” bit error and packet loss always happen. The probabilistic transition of the channel condition occurs at fixed cycles $T_c$.

3.2. Node Mobility. The mobility of individual sensor node (except for sink nodes) is based on the random waypoint model [20]. A node determines a destination and moves there with constant speed. After arriving at the destination, it pauses for a definite period of time and moves for a new destination again. This destination and speed are randomly chosen. This mobility brings about quasicyclic changes to the network topology.

3.3. Node Addition/Failure. We assume a random addition and failure of a number of sensor nodes. Since such node addition is carried out in a planned manner, we refer to it as periodical perturbations. Mean time to failure of devices in the network also means that node failure is of a cyclical nature. This node addition occurs at the same time in the simulation, and the same is true for node failures.

4. Design Approaches of the Control Timescale

In this section, we present design approaches for a controlled self-organization-based network particularly focused on control timescales in a MAC layer, a routing layer, and an external control. An overview of their timescales is illustrated in Figure 2 along with those of perturbations.

4.1. MAC Layer Design. As to changes of wireless channel conditions that arise with the cycle of 1 ms to 1,000 ms, retransmission in a MAC layer is important. In a MAC layer, a node obtains more opportunities to detect a next-hop node when the cycle of sleep control is shorter, in case the node holds a data for a certain period of time (denoted by $T_d$) until it finishes forwarding the data to the next-hop node. However, as to the changes with a cycle shorter than this, a MAC layer cannot handle them fundamentally, and we need to choose a robust modulation method against severe changes of radio in the physical layer.

4.2. Routing Layer Design. When movement, additions, and failures of nodes occur, latest route information is necessary for data delivery. Therefore, correct selection of a next-hop node is attained as the updating cycle gets shorter. Similar to possibly supposed scenarios on wireless sensor networks, our research supports static and comparatively slow mobility of nodes, taking account of monitoring application of human health, animal behavior, and so forth. Then, every tens of seconds of periodical messages are used for neighbor detection, and message exchanges to maintain route information.

4.3. External Control Design. Comparatively, long-term perturbations such as movement, additions, and failures of sensor nodes may cause global topological changes, which cannot be dealt with by self-organized routing protocols based on only local information. Thus, since these perturbations degrade the performance of such routing protocols, the control and observation mechanism is required for normal operation.

Similar to the principle of routing layer design, the shorter control and observation cycle seem to be better. However, this cycle is closely bound together with the cycle of self-organized route construction in the routing layer, and therefore, the external control process and self-organized routing process can interfere mutually. In addition, convergence speed of self-organized methods is generally slow, and when the external control is conducted before routes do converge, a system does not satisfy the desired performance.

In order to examine the convergence speed of self-organized potential calculation, first we show analytical solution...
of the 2-dimensional diffusion equation: $\frac{\partial \phi(x, y, t)}{\partial t} = D \nabla^2 \phi(x, y, t)$. Here, we change the Cartesian coordinates $(x, y)$ to polar coordinates $(r, \theta)$ in order to reduce one of variables ($r_{\min} < r < r_{\max}$ and $-\pi < \theta < \pi$). Since we consider symmetric diffusion of potential from the origin, the solution of the equation is independent of angular coordinate $\theta$. Then, the diffusion equation is converted into polar coordinates as follows:

$$\frac{\partial}{\partial t} \phi(r, t) = D \left( \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial \phi(r, t)}{\partial r} \right) \right). \quad (3)$$

Various boundary conditions can be found in natural world, and we assume two simple Dirichlet boundary conditions: $\phi(r_{\min}, t) = \phi_{\min}$ and $\phi(r_{\max}, t) = \phi_{\max}$ ($\phi_{\min} < \phi_{\max}$). The solution of (3) under the conditions is represented by (4), which is a sum of exponential functions:

$$\phi(r, t) = \sum_{n=0}^{\infty} A_n e^{-\gamma_n^2 D \tau} R(r, n) + C(r). \quad (4)$$

In the solution, $q_n$ and $A_n$ are functions of constant number $n$. Here, $q_n (n = 0, 1, 2, \ldots)$ is the real root of the following equation and satisfies the condition $q_k < q_{k+1}$ for any nonnegative integer number $k$:

$$J_0 (\phi_{\min} q_n) = -Y_0 (\phi_{\max} q_n) J_0 (\phi_{\max} q_n) = 0, \quad (5)$$

where $J_0(x)$ and $Y_0(x)$ are the zero-order Bessel function of the first kind and the zero-order Bessel function of the second kind, respectively. $A_n$ depends on an initial condition and given an initial condition $\phi(r, 0) = 0$, $A_n$ is calculated according to the following equation:

$$A_n = \frac{\pi^2 q_n^2}{2} \left( \frac{Y_0 (q_n r_{\max})}{J_0 (q_n r_{\max})} \right)^2 \int_{r_{\min}}^{r_{\max}} r \left( a \log(r) + b \right) \left( \frac{J_0 (q_n r)}{Y_0 (q_n r)} \right) dr. \quad (6)$$

$R(r, n)$ is a function only dependent on $n$ and radial coordinate $r$ as represented in the following equation:

$$R(r, n) = J_0 (q_n r) - \frac{J_0 (q_n r_{\min})}{Y_0 (q_n r_{\min})} Y_0 (q_n r). \quad (7)$$

$C(r)$ is represented by a basic logarithm function, $a \log(r) + b$, where $a$ and $b$ are constant number and calculated as follows: $a = (\phi_{\max} - \phi_{\min}) / (\log(r_{\max}) - \log(r_{\min}))$, $b = \phi_{\min} - ((\phi_{\max} - \phi_{\min}) / (\log(r_{\max}) - \log(r_{\min}))) \log(r_{\min})$.

From (4), it can be found that the potential $\phi(r, t)$ exponentially converges without relying on the distance from the potential source, but relying on time. In [21], the authors point that the solution of the discrete diffusion equation also exponentially converges. From the above discussion, we could obtain an approximate solution of the diffusion equation. If the solution is represented by a basic exponential function, $f(x) = a e^{-x/r} + v$, convergence of the potential can be estimated using time constant $r$. It is worth noting that calculation of $r$ requires the value of $\phi$ after convergence. Therefore, in order to understand the convergence behavior of the system, computer simulation is one of the means.

For an example of the potential convergence, in Figure 3, the simulation results of the potential convergence in two grid networks ($99 \times 99$ and $49 \times 49$) are shown. The center node is a potential source (potential is 100) and the outer circumferential nodes have potential of zero. Each node performs potential exchanges with its nearby four or eight nodes and updates its own potential according to (1) every time step. In the figure, the horizontal axis means the time step and the vertical axis is potential of a neighbor node of the center node. The symbols (circle, square, and triangle) mean 90% and 99% convergence from an initial value of zero in each result. From the results, convergence speed becomes quicker as a network size becomes small and as communication range increases. Figure 4 shows the potential convergence in a random network with 100 sensors. Due to the random deployment of sensor nodes, the convergence speed in this case is slower than that in grid cases.

5. Simulation Results

In this section, we evaluate the packet delivery ratio under the periodical environmental perturbations. We use an event-driven simulator written by C++ for evaluation. For a network model, we deploy 100 sensor nodes at random places over the square region 500 m on a side and install a sink node in a corner of the domain. Each sensor node generates one data every 500 s, and it is delivered to the sink node in a multihop manner. For a communication model, we utilize the disk model, and communication between two nodes within communication range is successful unless a message collision occurs or wireless channel condition between the nodes is
5.1. Transitions of Channel Conditions. When the cycle of the sleep control in the MAC layer is set to 0.5 s, 1.0 s, and 2.0 s, respectively, the data delivery ratio against the periodic transition of the channel condition is shown in Figure 5. When the transition of the channel condition arises with the cycle of 10 ms and 100 ms, it turns out that shorter sleep control cycles are required for a high data delivery ratio. Since the MAC layer quickly responds to change in the channel conditions and the opportunity of the retransmission in the MAC layer increases as a sleep control cycle is shorter, even if there is no support in an underlying layer, perturbations with shorter cycle are absorbed. On the other hand, when perturbations occur with the cycle more than 1,000 ms, the delivery performance deteriorates greatly, and above the cycle, the MAC layer cannot handle perturbations. Therefore, it is essential to cope with such perturbations in a higher layer.

Table 1: Setting of parameters.

| Parameters                              | Value     |
|-----------------------------------------|-----------|
| Communication range                     | 100 m     |
| Time to live (TTL)                      | 32 hops   |
| $T_d$                                   | 5 s       |
| Channel-condition transition probability| 30%       |
| (good to bad)                           |           |
| Channel-condition transition probability| 70%       |
| (bad to good)                           |           |
| Node speed                              | 4–6 km/h  |
| Pause time                              | 250–350 s |
| Memory span for neighbor potential      | 250 s     |
| Update interval of potential            | 50 s      |

5.2. Node Mobility. Here, we set the same value to the cycles of potential advertisement and update. In order to eliminate the influence of perturbations other than the cycles, the number of the maximum relay has a sufficiently large value so that there may be no excess. Figure 6 shows that the delivery ratio is decreasing as the update cycle of potential becomes large. It is because as longer the update cycle is, the higher the possibility that a potential of a node is incorrect, and as a result, the node transmits data away from the sink node, or discards the data since there have already been no appropriate next-hop neighbor nodes.

Given the update cycle of potential is $T_i$ (s), when a data is generated at a node at a certain time, the elapsed time from the last update is $(1/2)T_i$ on average. Since we assume that nodes move at 4–6 kilometers per hour, the displacements of nodes from the last update are presumed to be $0.55T_i$–$0.83T_i$ (m). In our network model, where 100 sensor nodes and square domain with a 500 m side are assumed, the average distance with the nearest node is about 50 m, and therefore, connectivity with the nearest node can change with the cycle of a 10 s order. It is obvious that connectivity between other neighbor nodes can change with much shorter cycle. Therefore, in this network model, it is desirable that the value of the update cycle is at least shorter than 10 s, and when it is
5.3. Cross-Layer Interaction. Here, we set two sink nodes at the center (sink 1) and a corner (sink 2) of the network. In Figures 8 and 9, we show the potential of two sink nodes (Figures 8(a) and 9(a)) and the number of received data by the sink nodes at every control interval (Figures 8(b) and 9(b)). In those results, 50 sensor nodes are added at a random position at time 20,000 s, and random 50 sensor nodes fail at 40,000 s. The potential update cycle is set to 50 s as shown in Table 1, and from the preliminarily experiment, it is found that simulation time of 500 s (100 s) can obtain the 99% (90%) convergence of the potential at each node.

The potential of two sink nodes is controlled by (2) so that the number of received data mutually becomes equal for every fixed cycle. At the beginning of the simulation, more data arrive at sink 1. Then, the control node makes potential of sink 1 up in order to reduce the number of received data by sink 1. Equalization of the received data by the sink nodes is attained at 12,000 s as shown in Figure 8(b) and their potential is also converged. Furthermore, equalization of the received data is attained right from the beginning as shown in Figure 9(b).

Meanwhile, some changes take place to the number of the received data immediately after 20,000 s when addition of 50 nodes occurs. In addition, in this case, convergence finishes within about 10,000 s (or immediately after the perturbation). Shortly after the failures of sensor nodes at 40,000 s, the number of received data decreases. It is because a convergence commences after nodes erase the potential of failed nodes. The time for erasing depends on the potential memory span shown in Table 1. It turns out that after failure, as well as the addition, potential converges.

6. Conclusion

In this paper, we discussed an approach for network design based on controlled self-organization. This approach is for future large-scale and complex networks. As an example of networks based on controlled self-organization, we focus on a wireless sensor network where a self-organized routing protocol and an external control mechanism are applied. In particular, our concern is on cyclic nature of the environmental perturbation. In order to obtain robustness of a system against environmental perturbations, multiple network layers should not handle them separately, but should cope with in a coordinated fashion. We show that our approach can deal with various perturbations by appropriately defining the control timescale of each layer. Further investigation on other perturbations and networks and validation with real experiments are our future work.
Acknowledgment

This research was supported in part by “Grant-in-Aid for JPSP Fellows (24738)” of the Japan Society for the Promotion of Science (JSPS) in Japan.

References

[1] C. Prehofer and C. Bettstetter, “Self-organization in communication networks: principles and design paradigms,” *IEEE Communications Magazine*, vol. 43, no. 7, pp. 78–85, 2005.

[2] A. Banerjee, R. Agarwal, V. Gauthier, C. K. Yeo, H. Affifi, and F. Lee, “A self-organization framework for wireless ad hoc networks as small worlds,” *IEEE Transactions on Vehicular Technology*, vol. 61, no. 6, pp. 2659–2673, 2012.

[3] H. Zhang, J. Llorca, C. C. Davis, and S. D. Milner, “Nature-inspired self-organization, control, and optimization in heterogeneous wireless networks,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 7, pp. 1207–1222, 2012.

[4] C. Müller-Schloer, H. Schmeck, and T. Ungerer, *Organic Computing—A Paradigm Shift for Complex Systems*, Autonomic Systems, Springer, 2011.

[5] J. Branke, M. Mnif, C. Müller-Schloer et al., “Organic Computing—a addressing complexity by controlled self-organization,” in *Proceedings of the 2nd International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA ’06)*, pp. 185–191, November 2006.

[6] F. Nafz, J. P. Steghöfer, H. Seebach, and W. Reif, “Formal modeling and verification of self-* systems based on observer/controller-architectures,” in *Assurances for Self-Adaptive Systems*, pp. 80–111, Springer, 2013.

[7] F. Allerding, M. Premm, P. K. Shukla, and H. Schmeck, “Electrical load management in smart homes using evolutionary algorithms,” in *Evolutionary Computation in Combinatorial Optimization*, pp. 99–110, Springer, 2012.

[8] B. M. Ali and Z. Nacereddine, “Web service-based emergence control in cooperative information systems,” in *Proceedings of International Conference on Information Technology and e-Services (ICITeS ’12)*, pp. 1–5, March 2012.

[9] J. Zheng and M. J. Lee, “A comprehensive performance study of IEEE 802.15.4,” *Sensor Network Operations*, pp. 218–237, 2004.

[10] J. Polastre, J. Hill, and D. Culler, “Versatile low power media access for wireless sensor networks,” in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys ’04)*, pp. 95–107, November 2004.

[11] M. Buettner, G. V. Yee, E. Anderson, and R. Han, “X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks,” in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys ’06)*, pp. 307–320, November 2006.

[12] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, “PW-MAC: an energy-efficient predictive-wakeup MAC protocol for wireless sensor networks,” in *Proceedings of the 30th IEEE International Conference on Computer Communications (INFOCOM ’11)*, pp. 1305–1313, April 2011.

[13] K. Zeng, K. Ren, W. Lou, and P. J. Moran, “Energy aware efficient geographic routing in lossy wireless sensor networks with environmental energy supply,” *Wireless Networks*, vol. 15, no. 1, pp. 39–51, 2009.

[14] J. Heo, J. Hong, and Y. Cho, “EARQ: energy aware routing for real-time and reliable communication in wireless industrial sensor networks,” *IEEE Transactions on Industrial Informatics*, vol. 5, no. 1, pp. 3–11, 2009.

[15] Y. M. Lu and V. W. S. Wong, “An energy-efficient multipath routing protocol for wireless sensor networks,” *International Journal of Communication Systems*, vol. 20, no. 7, pp. 747–766, 2006.

[16] D. Kominami, M. Sugano, M. Murata, and T. Hatauchi, “Energy-efficient receiver-driven wireless mesh sensor networks,” *Sensors*, vol. 11, no. 1, pp. 111–137, 2011.

[17] D. Kominami, M. Sugano, M. Murata, and T. Hatauchi, “Controlled potential-based routing for large-scale wireless sensor networks,” in *Proceedings of The 14th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM ’11)*, pp. 187–195, October 2011.

[18] T. Rusak and P. Levis, “Burstiness and scaling in the structure of low-power wireless links,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 13, no. 1, pp. 60–64, 2009.

[19] E. N. Gilbert, “Capacity of a burst-noise channel,” *Bell System Technical Journal*, vol. 39, no. 9, pp. 1253–1265, 1960.
[20] D. B. Johnson and D. A. Maltz, “Dynamic source routing in Ad
Hoc wireless networks,” in Mobile Computing, pp. 153–181, 1996.

[21] A. Cunha, R. Teixeira, and L. Velho, “Discrete scale spaces via
heat equation,” in Proceedings of the 14th Brazilian Symposium
on Computer Graphics and Image Processing, pp. 68–75, IEEE,
October 2001.