Sensor Network Design for Smart Highways

Shyamakshi Ghosh, Shrisha Rao, *Member, IEEE*, and Balkrishnan Venkiteswaran

***Abstract*—A unique and efficient system model is proposed for sensor networks to create sensor-smart highways. The approach first considers various cell geometries encountered with two-dimensional roads, and then extends the same to three-dimensional roads. Mathematical relations describing each type of cell are derived, and the analyses indicate the numbers and positions of sensor nodes needed in each cell of a certain type. We then propose algorithms for computing the deployment positions of sensor nodes on roads by describing them as consisting of cells of different geometries. We also validate the approach by computing a sensor deployment using real geospatial data.**

***Index Terms*—sensor-smart road, cell architecture, cell geometry, two-dimensional cells, three-dimensional cells, leaf node, head node**

I. Introduction

SENSOR networks can be efficiently utilized to reduce human intervention and management. These networks use wireless embedded sensors that combine sensing, computation, and communication in a single, tiny, resource-constrained device that can be termed as a node. They have many applications, and one of the key applications of sensor networks is surveillance of a physical environment by monitoring critical conditions. The need for better techniques to handle traffic congestion and inculcate safer driving by collecting more and more data on the move has led to the conceptualization of smart highways equipped with sensor networks [2].

This paper deals with one such kind of sensor network that may be deployed on roads. In this paper, we mainly consider the system architecture, i.e., the placement of sensor nodes, a critical issue that affects both the cost and the detection capability of a wireless sensor network. We divide the road into cells of various geometric shapes, and give the appropriate cell layout for each shape. We also suggest algorithms for dividing roads into cells, and validate the approach using urban geospatial data.

We focus on static placement of nodes, rather than stochastic deployment methods such as sprinkling of nodes which are known to be ineffective in uneven terrains [3], [4]. In this work, we establish the relation between system architecture and sensor characteristics. We understand that hilly terrain, represented by a three-dimensional model, requires much more sophisticated sensor characteristics while using similar system architecture as in two-dimensional roads. This may not be a feasible solution due to constraints on energy and other resources. Therefore we find out how the same sensor characteristics as in the two-dimensional model can be applied to the system architecture of a three-dimensional model.

In order to deploy the sensors on the roadways, we need to model roads and also choose a network topology for the nodes. Nodes in the network are arranged in a hierarchical tree-like fashion as suggested by GaoTao and MingHong [5]. Node placements are determined using a polyline representation of the road, which is a usual practice in civil engineering and GIS [6], [7], [8].

Karpiński *et al.* [2] suggest some possible applications for such sensor-driven smart highways, like vehicle tracking, periodic traffic volume snapshots, road surface condition monitoring, incident detection, pedestrian on-road notification, etc. Some other possible applications include providing automated driving or driving assist signals [9], [10], monitoring vehicle information such as speed, driving lane, etc., and detection of road blocks caused by rain, snow, etc. Such applications are especially significant in hilly terrains that are subject to frequent blockages due to landslides or other reasons, and on busy highways where stoppages are not tolerable.

The concept of sensor networks on roads has been proposed by Nekovee [11], but that work is limited to identification of the various applications and the networking and communication challenges involved. Karpiński *et al.* [2] have focused on the software architecture, but no one has yet proposed an effective system architecture for both the two-dimensional and three-dimensional road scenarios. Shorey and Choon [4] observe that random-deployment methods do not guarantee full coverage because of their stochastic nature. We aim at full coverage by deterministically dividing the road into cells, and then tiling the cells with sensor nodes.

Static placement of nodes is influenced by various factors, such as terrain properties and sensor characteristics—coverage, connectivity, and power budget. For hilly terrains, the major challenge is connectivity, as terrain obstacles and sharp turns are very common. In such cases, the issue is an optimized node placement for efficient connectivity without loss of coverage and much of power efficiency. Dhillon and Chakrabarty [3] state that knowledge of terrain is just for surveillance, and does not directly solve the sensor placement problem. In contrast, we establish the impact of terrain properties on the system architecture and resolve the issue of connectivity.

Manuscript received August 31, 2010; revised February 14, 2011, July 25, 2011; accepted September 17, 2011. The work of S. Rao was supported in part by the Centre for Artificial Intelligence and Robotics, Defence Research and Development Organization, under contract CAIR/CARS-06/CAIR-23/0910187/09-10/170.

A previous version of this work [1] was presented by the first two authors under the same title at the 5th Annual IEEE Conference on Automation Science and Engineering, Bangalore, India, in August 2009.

S. Ghosh did her work while a student at the International Institute of Information Technology - Bangalore, Bangalore 560 100, India. She is presently with Samsung India Software Operations, Bangalore.

S. Rao is with the International Institute of Information Technology - Bangalore, Bangalore 560100, India (e-mail: shrao@ieee.org).

B. Venkiteswaran did his work while a student at the International Institute of Information Technology - Bangalore, Bangalore 560 100, India. He is presently with GT Nexus, Bangalore.

Digital Object Identifier 10.1109/TSMCA.2012.2187185

The system architecture chosen has a great impact on the sensor characteristics as well. According to Dhillon and Chakrabarty [3], in distributed sensor networks the sensor placement directly influences resource management, the type of processing done based on its back-end processing, and the exploitation of the sensed data.

In this paper, Section II refers to related work. Section III describes the system model, first by describing the node hierarchy and cell geometry for roads in general (Section III-A), and then (Section III-B) by unitizing hilly roads into *cells* and categorizing each into six categories: *plain cell*, *monotonic cell*, n*-tonic cell*, *plain curved cell*, *monotonic curved cell* and n*-tonic curved cell*. For each category, we compute the number of sensors and the cell area for a given coverage and transmission range. Section IV describes the algorithm for deployment, and Section V presents simulation results on real urban geospatial data from Bangalore, India. Section VI concludes the paper.

II. Related Work

Applications based on sensor networks on smart roads are discussed in many papers, e.g., [2], [9], [12], [13], [14], [15]. Strategies for sensor deployment in contexts other than smart roads are also considered in several papers. The deployment models for smart roads are mostly considered to be random, with some having sensor-equipped vehicles, though there are general attempts (not specific to this domain) to achieve optimal deployment [16], autonomous placement and localization [17], or to come up with theoretical models [18], [19]. In road networks, static deployment, such as with Kwon *et al.* [12], is often proposed. Kwon *et al.* propose several algorithms to predict collisions among vehicles at intersections via sensor networks located at intersecting regions. They assume sensors to be uniformly installed on a road. These sensors detect the locations and speeds of vehicles and send data to a base station located at the center of an intersection. Coleri *et al.* [13] provide an algorithm for traffic surveillance. Their system is comprised of a wireless sensor network that generates traffic information such as the number of cars, speed of cars, etc., and sends the data to an access point (similar to the base station in [12]). However, in both these papers, the manner of sensor deployment is not covered in detail. Algorithms such as those provided by Kwon *et al.* [12] and Coleri *et al.* [13] that look for static deployment can be easily implemented on the system model proposed here, which can also easily apply with other applications, i.e., not only with roads but also in other similar domains like airport runways, bridges, railroad tracks, etc.

Ganesan *et al.* [20] consider the joint optimization of sensor placement and transmission structure. They look at the simplified problem of optimal placement in the one-dimensional case, and then extend it to the two-dimensional case. Their algorithm provides a two- to three-fold reduction in total power consumption and a one to two orders of magnitude reduction in bottleneck power consumption, for the two-dimensional case. We find that with the same power consumption, i.e., the same sensor characteristics, their system architecture changes for three-dimensional sensor networks on hilly roads.

Sensor placement on two- and three-dimensional grids has been formulated as a combinatorial optimization problem and solved using integer linear programming, by Chakrabarty *et al.* [21], albeit at high computational cost. (By contrast, in the present paper the computational complexity remains the same for the two- and three-dimensional cells.)

Mangharam *et al.* [22] introduce a simulator called GrooveSim that models intra-vehicular communication within a street-map based topography, to deliver vital information on safety alerts, traffic congestion, etc. GrooveSim however is based on the Global Positioning System (GPS) which is not effective in deep foliage or tunnels. There are various proposals for sensor networks on roads, with some also requiring sensors placed on vehicles. One such is by Zoysa *et al.* [23], which proposes a public-transport based sensor network to monitor and report a road surface. While monitoring the road surface, it is assumed that vehicles are driven over rough patches as well, which may not be true as vehicles tend to avoid them. Ravelomanana [24] investigates characteristics such as topology, diameter, and degree of reachability, of a three-dimensional randomly-deployed sensor network. Alam and Haas [25] examine various polyhedra for an efficient coverage and connectivity within a 3D random sensor network and conclude that a truncated octahedral placement strategy is best. Clouqueur *et al.* [26] propose a sensor deployment strategy using the minimum exposure as a measure of the goodness of deployment. Younis and Akkaya [27] report on the recent state of the research on optimized node placement in wireless sensor networks, along with the open problems in this area. Ghosh and Das [28] provide a mathematical framework and formalize the coverage-connectivity problem, and also compare various sensor deployment strategies and algorithms based on their goals, assumptions, complexities, and usefulness in practical scenarios. Park *et al.* [29] propose a placement algorithm that takes into account geographical features as well as the desired level of sensor redundancy to produce a set of useful sensor locations.

The power consumption of a sensor can vary based on traffic loads and frequency of data communication. Other issues that have been explored include: network coverage given low duty-cycled sensors; the energy conservation problem; reduced network reconfiguration time while prolonging the network lifetime by scheduling the active and sleeping periods of the sensors; and other issues. Prior work in these areas of research can be found in [30], [31], [32], [33]. Corke *et al.* [34] have discussed design principles for a solar-powered sensor node.

III. System Model

In this model, nodes are laid on both sides of the road. These nodes interact among themselves based on the hierarchy mentioned in Section III-A, to gather and process information. Based on the height, slope and aspect of the road in the cell, we define the various categories in which the cells can be classified. For each cell category, we derive formulae to compute the area of the cell, the number of nodes in the cell, the inter-node spacings within the cell, etc. Further, these are used by the algorithms to create the sensor deployment

strategies as defined in Section IV; these algorithms take polyline representations of the three-dimensional roads as inputs.

A. Cell Architecture

The proposed architecture consists of nodes, made from low cost and readily available hardware, arranged in a three-level tree-like hierarchy as shown in Figure 1. The *leaf nodes*, being at the lowest level, have sensors that collect data and transmit it to the next higher level *head nodes*. Head nodes are simple embedded devices that act as receivers of the data sent by leaf nodes.

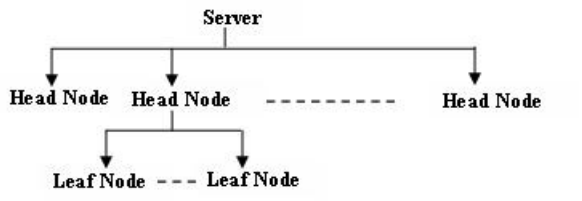

Fig. 1. Three-Level Tree-Like Hierarchy

Leaf nodes are *anisotropic*, and could be a set of Berkeley motes comprising low-power 8-bit, 128Kb memory processors, RF communication devices, and sensors. A head node could be a small computer that requires minimal processing but large memory because it accumulates data sent by the leaf nodes. The head node memory requirement may be estimated using an M/M/1 queueing analysis [35], [36].

All head nodes send their data to one or more servers which form the highest level. These servers could be networked mainframes which process data received from the head nodes. Users of the system log onto the servers. We limit the functionality of a head node to just collecting data and sending to the server, and do not require information processing because of these reasons:

- Processing would require sophisticated hardware and thereby increase the cost of head nodes, and thus of the overall system.
- Maintenance and upgrade of head nodes would be difficult.
- Each head node would have limited data, making processing incomplete.

On the other hand, we cannot also entrust the leaf node with communicating with a server directly, for similar reasons—it would make such nodes quite expensive (in contrast with the MEMS motes that we suggest using), as a central server for a highway could be some distance away, and sensing hardware that can communicate with it would probably not come cheap. For these reasons, it is essential that such a sensor network system have a three-tiered structure (as may also be seen in case of vehicular networks; e.g., see the system layouts in Dikaiakos *et al.* [37] and Huang and Lin [38]).

Communication among leaf nodes, head nodes and the servers can be in either of the following two ways: pull mode, where the server sends a query to the head node, which then asks the leaf nodes for data, triggering the sensors that capture the data; and push mode, where the server, head nodes, and leaf nodes operate on a schedule, with leaf nodes pumping in data at regular intervals, and head nodes forwarding collected data to the server.

We approximate the road into equispaced segments called *cells*, in line with current models in road traffic research [39]. A cell is a measured stretch of road. The cell dimensions depend on the node characteristics, such as its sensor's transmission and coverage ranges. The cells together cover the entire road. A head node and some leaf nodes are deployed over a cell. As the leaf nodes are anisotropic and have an angular coverage range θ, the nodes can tile up the cell as shown in Figure 2, which gives complete coverage of the road segment within the cell.

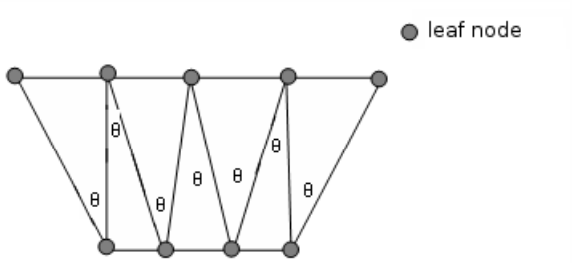

Fig. 2. Leaf Nodes Tile the Cell

A cell can be a combination of curving and non-curving road segments. In such cases, we divide the cell into *sub-cells* such that a sub-cell has exactly one horizontal curve, as discussed later.

The deployment geometry is based on the transmission ranges and angular coverage capacities of leaf nodes. The intersection of the road boundary with the semicircular coverage area centered at the head node generates a trapezoidal cell shape on the road stretch as shown in Figure 3, where the trapezium vertices are the locations of the leaf nodes (placed at the limit of the head node to leaf node transmission range). The trapezoidal shape is optimal because it results in maximum utilization of the transmission power of leaf nodes. In Figure 3, if x and y are the power levels required by the corner leaf nodes A and D to transmit data to the head node, then $x = y$ in a trapezoidal cell.

In general, we can say that the shape of the cell depends on the position of the head node. At a junction, the remaining road area can be considered as a cell of either rectangular, circular, or trapezoidal shape, as may suit that junction best.

B. Sensor Layout

We categorize each road segment by a cell based on the properties that describe a road:

- *Height*: Elevation, i.e., the value at the Z axis.
- *Slope*: Angle it makes with XY plane.
- *Aspect*: Direction in which the road faces (angle it makes on the X axis).

(1) *Cell Geometry:*

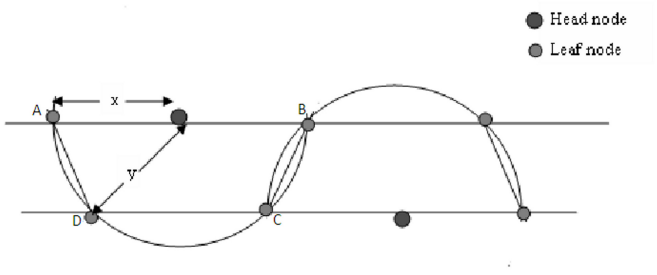

Fig. 3. Trapezium Cell

Based on the above-mentioned properties, we categorize any road segment as being one of these six types.

- *Plain Cells*: Plain cells are trapezium-shaped flat cells, as discussed in Section III-A and depicted as $ABCD$ in Figure 3, that have zero slope, a constant height, and a constant aspect. Considering the centerline of a plain cell, we get a line segment parallel to the XY plane.
- *Monotonic Cells*: Monotonic cells have, as shown in Figure 4, a constant slope, a constant aspect, and an increasing or decreasing height. In Figure 4, $ABHG$ is a plane, hence has a constant slope as any point on $ABHG$ makes the same angle with the X axis, and also has a constant aspect as edges of the plane AB and GH are straight line segments.

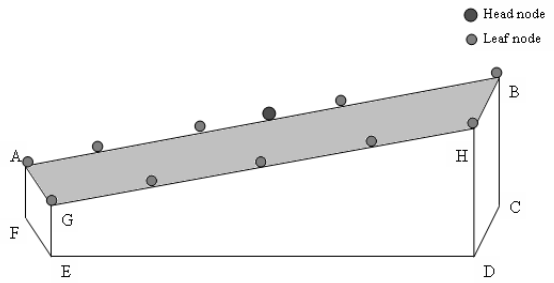

Fig. 4. Monotonic Cell

- *n-Tonic Cells*: n-tonic cells have a varying height and thus a varying slope, but with a constant aspect. Figure 5 shows an n-tonic cell with varying slope (randomly-picked points on the road surface $ABHG$ make different angles with the X axis), and a constant aspect (edges GH and AB when projected on the XY plane are straight line segments).
- *Plain Curved Cells*: Plain curved cells can be represented by Figure 12. Plain curved cells have a constant height, zero slope, and a changing aspect due to turn. When considering the centerline we get a polyline rather than a straight line segment.
- *Monotonic Curved Cells*: Monotonic curved cells, as shown in Figure 6, have an increasing or decreasing height and a constant slope (any point on plane $ABHG$ makes the same angle with the X axis), and have a changing aspect (edges AB and GH are polylines rather than line segments).

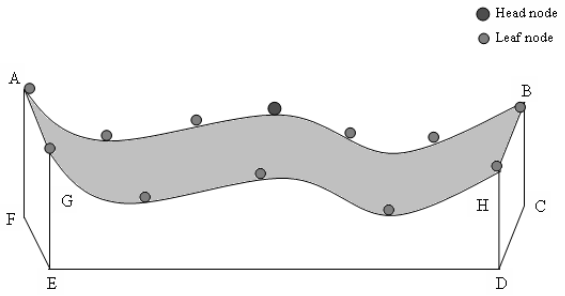

Fig. 5. n-Tonic Cell

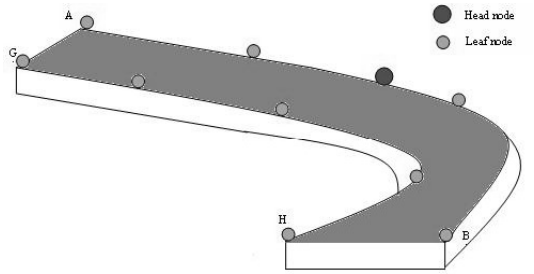

Fig. 6. Monotonic Curved Cell

- *n-Tonic Curved Cells*: n-tonic curved cells have, as shown in Figure 7, a varying height and thus a varying slope, and a changing aspect. The projections of edges AB and GH on the XY plane are polylines rather than line segments.

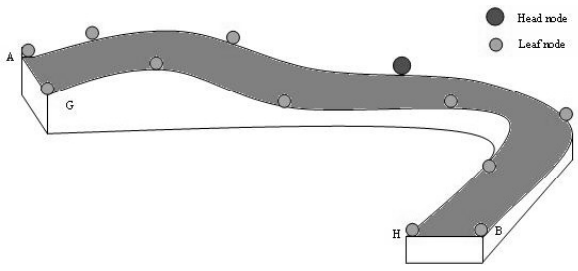

Fig. 7. n-Tonic Curved Cell

(2) *Cell Properties*:
Now we consider the cell properties, such as area and the node density, for each cell type. We follow the notational conventions shown in Table I.

Case 1: Plain Cells

Consider a plain cell with one head node and n leaf nodes. In Figure 8 (compare with Figure 2), the head node is at R, and four of the n leaf nodes (which are at the corners of the trapezium) communicate with it at a maximum range of a. VPS is the coverage range of leaf node (θ) at point P. Using classical geometry, we can compute the area of a trapezoidal cell:

$$wa(1 + \sqrt{1 - (w/a)^2}) \quad (1)$$

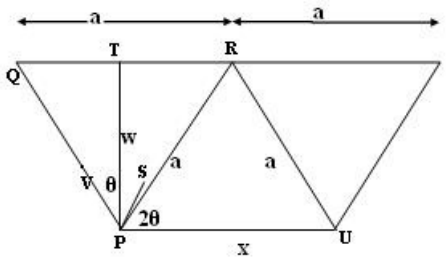

Fig. 8. Area of a Trapezoidal Cell

As the leaf nodes tile the cell, the number of leaf nodes can be determined from their angular coverage θ. From Figure 8, it is clear that the spacing between leaf nodes is:

$$2w \tan\left(\frac{\theta}{2}\right) \tag{2}$$

Each leaf node covers the following area of the road:

$$A = w^2 \tan\left(\frac{\theta}{2}\right) \tag{3}$$

The number of nodes is lower-bounded by the ratio of the total area of the cell and the coverage per sensor. Hence, from (2) and (3) the number of nodes in a cell is:

$$N = \frac{wa\left(1 + \sqrt{(1 - w^2/a^2)}\right)}{w^2 \tan(\theta/2)} \tag{4}$$

Based on the above equations, Table II cites some examples that show that depending on the width of the road and the transmission range, the spacing between leaf nodes can range from 10m to 20m for an angular range of 45°.

Case 2: Monotonic Cells

Monotonic cells have an average constant slope, a constant aspect and an increasing or decreasing height. When we transform a monotonic cell into a normal cell we see an extension in the transmission range, which signifies an increase in the area and nodes of the cell as in Figure 9. We term this extended transmission range as *pseudo-transmission range*, because the real transmission range of leaf nodes still remains a, but the cell properties change. From Figure 10, we know that:

$$a' = \frac{a}{\cos\phi}, \tag{5}$$

where ϕ is the slope.

So we modify (1) to find the area of 3D cells as follows:

$$A = wa'(1 + \sqrt{1 - (w/a')^2}) \tag{6}$$

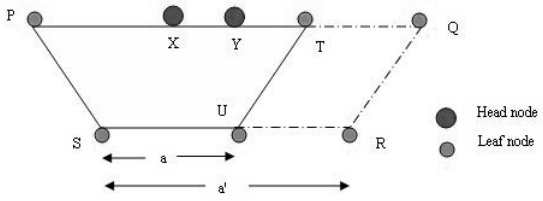

Fig. 9. Transforming Into a Normal Cell

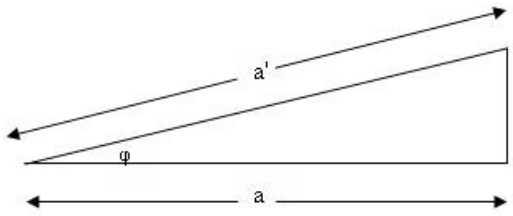

Fig. 10. Monotonic Cell

The area covered by the leaf node remains the same, as shown in (3), but N, the number of nodes in cell is now modified as given below, compared to (4):

$$\frac{wa'\left(1 + \sqrt{1 - (w/a')^2}\right)}{w^2 \tan(\theta/2)} \tag{7}$$

TABLE I
NOTATION USED

Symbol	Description
w	width of the road
θ	angular range of a leaf sensor
a	transmission range or the distance between corner leaf node and head node within a cell
a'	pseudo-transmission range
ϕ	slope of a cell
N	number of leaf nodes in a cell
A	area of a cell
S	arc length of function to represent road surface in cell
K	curvature
R	radius of curvature

TABLE II
CALCULATIONS (IN METERS)

w	a	A (m^2)	N	Spacing
12	15	288	5	10
12	25	563	10	10
24	30	1152	5	20
24	50	2253	10	20

TABLE III
COMPARISON BETWEEN VARIOUS CATEGORIES

Cell Category	Cell Properties	Area of Cell	Leaf Nodes
Plain	Slope = 0	390	15
Monotonic	Slope = 30	455	17
n-tonic	$S = 500$	3999	150
Plain curved	$R = 200$	22978	867
n-tonic curved	$R = 200, S = 4600$	36918	1392

See Table III for the area of a monotonic cell computed assuming the width of the road w = 8m; transmission range of leaf a = 25m; and sensor coverage range $\theta = 45°$. So the area of the cell is 455m^2 and the number of leaf nodes are 17, a 17% increase in the cell area and 13% increase in the number of nodes as compared to the

plain cell. This means the data processing and storage capabilities of head nodes must be 17% more in case of monotonic cells under these conditions.

Case 3: n-Tonic Cells

n-tonic cells have fluctuating height, a varying slope, and a constant aspect. We consider the road surface along which vehicles are driven to be a function $y = f(x)$. We compute the distance the vehicle travels along the road in a cell from $x = a$ to $x = b$. The horizontal distance covered is $2a$, but the pseudo-transmission range a' is half the arc length of $f(x)$, S.

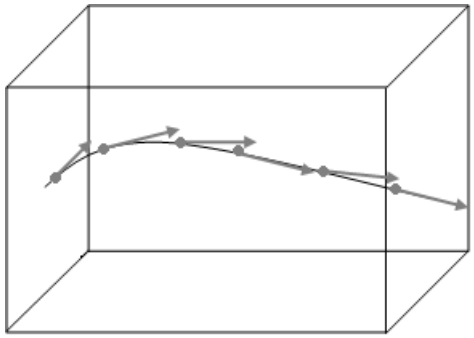

Fig. 11. Finding the Arc Length of a Road Surface

To find the arc length of a sample function $y = f(x)$ in the interval $[a, b]$, we approximate the function with a sequence of straight lines and add their lengths, as shown in Figure 11. We break the interval $[a, b]$ into smaller segments along the X-axis of width dx. When dx is infinitesimal, the segment is approximated as a straight line. The length of the hypotenuse ds is approximately the length along the curve for the little segment. From Figure 11 we have:

$$ds = \sqrt{dx^2 + dy^2} = \sqrt{1 + \left(\frac{dy}{dx}\right)^2} \cdot dx \tag{8}$$

$\frac{dy}{dx}$ is the slope of the segment. Also, $\frac{dy}{dx} = f'(x)$. To find the total length along the curve S, we integrate over the lengths of all the segments:

$$S = \int ds = \int_b^a \sqrt{1 + (f'(x))^2} \tag{9}$$

For n-tonic cells, the range is from $[0, 2a]$. Hence from (9), we get the arc length to be:

$$S = \int_0^{2a} \sqrt{1 + (f'(x))^2} \tag{10}$$

$$a' = \frac{S}{2} \tag{11}$$

The area and number of leaf nodes in an n-tonic cell can be derived using (6) and (7) respectively. Table III shows that for w=8m, a=25m, $\theta = 45^\circ$ and arc length S=500m, the area of the cell is 3999m^2 and the number of leaf nodes is 150, more than ten times that of a plain cell.

Case 4: Plain Curved Cells

Consider a car driving along a curvy road. The tighter the curve, the more difficult the driving is apt to be. The *curvature* describes this *tightness*. If the curvature is zero then the curve looks like a line near this point. However, if the curvature value is non-zero, with no change in slope, such a curve is known as a *horizontal curve*. A simple horizontal curve is a circular arc. Horizontal curves can be of composite nature, like spirals which are composed of circular arcs, as discussed in [40].

Plain curved cells have a zero slope, a constant height, and a changing aspect. If its polyline representation is a function $y = f(x)$ on the XY plane, the curvature of a function on the XY plane is given by (12), the derivation of which can be seen in [41].

$$K(x) = \frac{\mid f''(x) \mid}{(1 + (f'(x))^2)^{3/2}} \tag{12}$$

The radius of curvature R is considered at the midpoint of the horizontal curve and is represented as:

$$R = K(a) \tag{13}$$

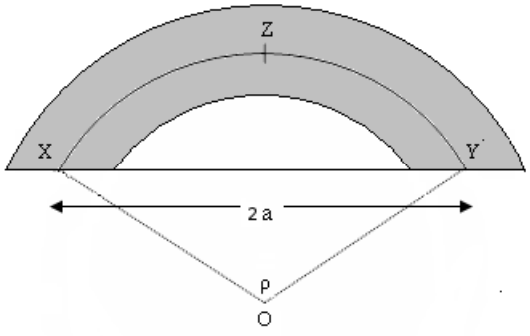

Fig. 12. Curved Cell

As shown in Figure 12, the length of the segment XY is $2a$. If R be the radius of curvature, ρ the angle XY makes with the center can be calculated as:

$$\sin\left(\frac{\rho}{2}\right) = \frac{2a}{2R} \tag{14}$$

$$\rho = 2\sin^{-1}\left(\frac{a}{R}\right) \tag{15}$$

So the area A in case of a plain curved cell is:

$$\frac{\rho\pi((R + \frac{w}{2})^2 - (R - \frac{w}{2})^2)}{2\pi} = 2wR\sin^{-1}\left(\frac{a}{R}\right). \tag{16}$$

The number N of leaf nodes in the cell therefore is:

$$\frac{2R\sin^{-1}\left(\frac{a}{R}\right)}{w\tan\left(\frac{\theta}{2}\right)} \tag{17}$$

The area and the number of nodes in a plain curved cell follow equations (16) and (17) respectively. Table III shows that the area and the number of nodes in a plain

curved cell are about 58 times those of a plain cell, when w=8m, a=25m, $\theta = 45^\circ$ and R=200m.

A cell can be a combination of one or more horizontal curves and non-curving road segments. In such a case, we divide the cell into sub-cells such that each sub-cell has exactly one horizontal curve. Hence, twice the transmission range $2a$ in equation (14) should be substituted by the segment length, l of the horizontal curve in the sub-cell. So substituting $2a$ by l in (16) and (17), we have the area A of the sub-cell as follows.

$$2wR\sin^{-1}\left(\frac{l}{2R}\right) \tag{18}$$

The number N of nodes in the sub-cells is:

$$\frac{2R\sin^{-1}\left(l/2R\right)}{w\tan\left(\theta/2\right)} \tag{19}$$

Multiple sub-cells within a cell can share a head node. If there is an obstruction between two consecutive sub-cells, each sub-cell can have a different head node in order to maintain connectivity with leaf nodes.

Case 5: Monotonic and n-Tonic Curved Cells

The road surface in a monotonic or n-tonic curved cell can be represented by functions in parametric form such that $x = f(t)$; $y = g(t)$; $z = h(t)$. For plain curved cells, the height represented by $z = h(t)$ is a constant.

Figure 13 shows the transformation of a projected monotonic curved cell on the XY plane into plain curved cell. Though R remains the same, the transmission range of the leaf nodes, i.e., $2a$ represented by LM, is extended to a pseudo-transmission range of $2a'$ represented by PQ.

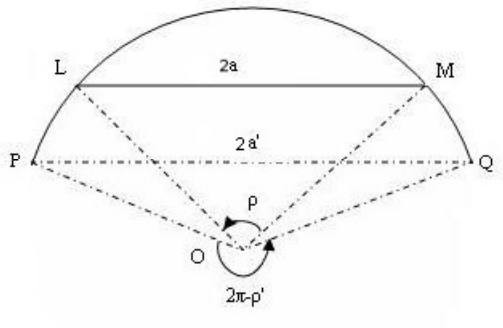

Fig. 13. Transformation of a Monotonic Curved Cell Into a Plain Curved Cell

To derive the radius of curvature R, we consider the projected horizontal curve $y = f(x)$ on the XY plane.

As we consider a simple horizontal curve, we need to transform it into the equation of a circle:

$$(x-h)^2 + (y-k)^2 = R^2,$$

where R is the radius of curvature and (h, k) is the center of the circle.

The arc length S is given by:

$$\int \sqrt{(f'(t))^2 + (g'(t))^2 + (h'(t))^2} \tag{20}$$

Figure 13 shows the projected transformation of an n-tonic curved cell (with the length of a segment of the horizontal curve being a) into a plain-curved cell (of pseudo-transmission range a'), with the radius of curvature R remaining the same. a' can be found as:

$$S = R\rho = 2R\sin^{-1}\left(\frac{a'}{R}\right)$$

$$a' = R\sin\left(\frac{S}{2R}\right) \tag{21}$$

From (21), we substitute the value of a in (16) and (17) as:

$$A' = 2wR\sin^{-1}\left(\frac{a'}{R}\right) \tag{22}$$

The number N of nodes in the cell from (17) comes to:

$$\frac{2R\sin^{-1}(a'/R)}{w\tan(\theta/2)} \tag{23}$$

From the above equations, we calculate the area and the number of nodes in the n-tonic curved cell. As shown in Table III, when w =8m, a =25m and $\theta = 45^\circ$, it is clear that there is an increase of about 60% in the area of the cell and the number of leaf nodes compared with a corresponding plain curved cell.

If a cell consists of monotonic or n-tonic curved sub-cells with segments to the horizontal curve of length l, then the area of the cell and number of nodes in the cell can be given as:

$$l' = 2R\sin\left(\frac{S}{2R}\right) \tag{24}$$

$$A' = 4wR\sin^{-1}\left(\frac{l'}{2R}\right) \tag{25}$$

$$N' = \frac{4R\sin^{-1}(l'/2R)}{w\tan(\theta/2)} \tag{26}$$

IV. Sensor Placement Strategies

This section introduces algorithms that form the strategies for deploying the leaf and head nodes on roads. As an input to the algorithms, the road is represented in one of its common representations, as a polyline, by considering its centerline. To begin with, we illustrate that there is an easy transition of a cell geometry to its polyline representation, and *vice versa*.

A. Polyline Representation Of Cells

In this section we show that a polyline is an appropriate representation of a cell. It can also be used to find the actual cell shape as shown in Figure 14.

We consider the road centerline to be represented by a polyline. The centerline extraction of roads can be done as by Toth and Grejner-Brzezinska [42]. We assume the road to be flat. The Highway Design manual of the California Department of Transportation [43] states that for new construction, the slope should be 4:1 or flatter. This implies that at every point on the road surface, when a cross section along the width of the road is taken, the height, aspect and slope are constant. So a polyline which is the centerline of the road can be used to represent the fluctuations in the height, slope and aspect.

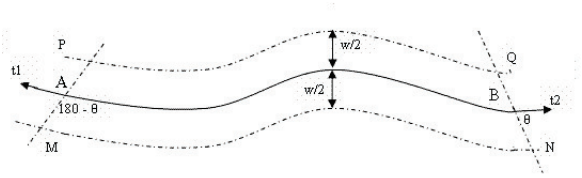

Fig. 14. Finding Cell Shape From Polyline

Let polyline $AB = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_n, y_n, z_n)\}$ be the set of coordinates used to represent a cell on a road.

With known values of w and θ we can find the cell shape and plot two polylines along the sides of the cell.
Polyline PQ = $\{(x_1, y_1 - \frac{w}{2}, z_1), (x_2, y_2 - \frac{w}{2}, z_2), \ldots\}$
Polyline MN = $\{(x_1, y_1 + \frac{w}{2}, z_1), (x_2, y_2 + \frac{w}{2}, z_2), \ldots\}$

From a tangent $t1$ at point A, we draw a line cutting the polyline PQ and making an $\angle\theta$, and cutting the polyline MN and making an $\angle(180^\circ - \theta)$.

At point B, we draw a line cutting polyline PQ making $\angle\theta$ with the tangent $t2$ at point B and $\angle(180^\circ - \theta)$ with the polyline MN. The resultant is the cell shape derived from the polyline.

From the polyline representation of a road, we may find the arc length, the pseudo-transmission range a', and cell characteristics as mentioned in Section III-B.

B. Deployment

In this section we introduce the algorithms for deploying the leaf and head nodes on a road.

The first step in deploying the nodes is to divide the projected polyline into segments of equal length. Such a polyline segment is a centerline representation of a cell. We classify a segment as curved or non-curved—if a road segment has a constant aspect then it is non-curving, and if it has a changing aspect it is curving. Using the function representation of the cell, we compute the area of the cell and the number of nodes within the cell. If the cell is not a plain cell, then we need a more powerful head node for that cell. As an alternative, we can divide the cell into sub-cells equal in number to the ratio of the area of the cell and the area of a corresponding plain cell.

Let D be a polyline representing the centerline of a road; D is an ordered set of co-ordinate points or vertices. The edges are straight line segments defined by consecutive pairs of points:

D = $\{(x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_n, y_n, z_n)\}$

The centerline of a road can also be represented by a set of polyline segments $T = \{C_1, C_2, \ldots, C_m\}$, where C_i is the i^{th} polyline segment of the road, given by an ordered set of co-ordinate points, such as:

$C_1 = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_r, y_r, z_r)\}$
$C_2 = \{(x_{r+1}, y_{r+1}, z_{r+1}), \ldots, (x_s, y_s, z_s)\}$
$\vdots$
$C_m = \{(x_t, y_t, z_t), \ldots, (x_n, y_n, z_n)\}$
where $1 < r < n$, $1 < s < n$, and $1 < t < n$.

The X, Y, and Z coordinates for each of the cells are different. The first vertex of the cell is considered to be the origin. The last vertex of the cell is the point on the polyline which intersects the X axis at a distance $2a$.

The functions used by the algorithms are:

- *clip(startIndex,$\mathcal{Y}$,m)* returns a point on the polyline represented by set of co-ordinate points $\mathcal{Y}$, such that the difference in X co-ordinate values of $startIndex$ and the point returned is of value m.
- *subset(startIndex,endIndex,$\mathcal{Y}$)* returns a subset of set $\mathcal{Y}$ with elements starting from the $startIndex$ ordinal value of $\mathcal{Y}$ till the element at $endIndex$ ordinal value of $\mathcal{Y}$.
- *belongs(x,$\mathcal{Y}$)* returns TRUE only if $x \in \mathcal{Y}$.
- *low($\mathcal{Y}$)* returns the lowest ordinal value (the first element) of a set $\mathcal{Y}$.
- *high($\mathcal{Y}$)* returns the highest ordinal value (the last element) of a set $\mathcal{Y}$.
- *getValue($i, \mathcal{Y}$)* returns the i^{th} ordinal value of a set $\mathcal{Y}$.
- *setValue($i, \mathcal{Y}, x$)* sets x as the i^{th} ordinal value of a set $\mathcal{Y}$.
- *cardinality($\mathcal{Y}$)* returns $|\mathcal{Y}|$, the cardinality of a set $\mathcal{Y}$.
- *equals(a,b)*: if $a \leftarrow (x_a, y_a, z_a)$ and $b \leftarrow (x_b, y_b, z_b)$, then the function returns TRUE if $x_a = x_b$ and $y_a = y_b$ and $z_a = z_b$.
- *index($x, \mathcal{Y}$)*: if $x \in \mathcal{Y}$, this returns the ordinal value, else it returns -1.
- *isNull($\mathcal{Y}$)* returns TRUE if a set $\mathcal{Y}$ is the empty set.
- *findRadiusCurvature(startIndex, endIndex,C_i)* returns the radius of curvature of the circular arc located from coordinates at $startIndex$ to $endIndex$ in cell C_i.
- *findArcLength(startIndex, endIndex, C_i)* returns the radius of curvature of the polyline located from coordinates at $startIndex$ to $endIndex$ in cell C_i. Let $x = f(t)$, $y = g(t)$ and $z = h(t)$ be the parametric form of equations that represent part of the polyline segment of C_i. Then, $S \leftarrow \int \sqrt{(f'(t))^2 + (g'(t))^2 + (h'(t))^2}$.

Algorithm IV.1, *FRAGMENT_ROAD* takes in D, the polyline representation of a road, as an input and fragments it into a set T of polyline segments. It is based on the fact that any polyline representation of cell C_i is a subset of D, i.e., $C_i \subset D$. In Algorithm IV.1 we generate subsets of D where each of the subsets represents a single cell.

Algorithm IV.1: FRAGMENT_ROAD(D)

$i \leftarrow 1$
$j \leftarrow cardinality(D)$
$k \leftarrow 1$
while $(i \leq j)$
do
$start \leftarrow getValue(i, D)$
$end \leftarrow clip(start, D, 2a)$
$C_k \leftarrow subset(start, end, D)$
$k \leftarrow k + 1$
$i \leftarrow index(end, D)$
$T = \{C_1, C_2, \ldots, C_m\}$
return (T)

In Algorithm IV.1, we clip the road polyline of length $2a$ on D starting from the first vertex of the road polyline, to get

the end index. With the start and end indices we represent the first cell by allotting the coordinates of D from the start to the end indices to the cell. In the same way, other subsequent cells are represented. Thus from the Algorithm IV.1, we have m polyline segments of the road.

Let G be an ordered set of endpoints of the circular horizontal curves on a road. If $(a, b) \in G$, then a is the starting vertex and b is the end vertex of the circular horizontal curve. Also, $G \subset D \times D$.

We need to determine the circular horizontal curves particular to a cell. If a curve extends to multiple cells, we need to split the curve and allot it to these respective cells. Let H be an ordered set of endpoints of curves split according to cells. If $(a, b) \in H$, then a is the starting vertex, b is the end vertex of the circular horizontal curve, and both a and b are points in the same cell. Also, $H \subset D \times D$. Algorithm IV.2 returns H, which is the set of cell-wise circular curve endpoints.

Algorithm IV.2: CELLWISE_CURVE_ENDPOINTS($G, C_1, C_2, \ldots, C_n$)

$i \leftarrow 1$
$j \leftarrow cardinality(G)$
$l \leftarrow 1$
for each $i \in (1, 2, \ldots, j)$
do
- $(a, b) \leftarrow getValue(i, G)$
- **for each** $k \in (1, 2, \ldots, m)$
  do
  - **if** belongs(a, C_k) **and** belongs(b, C_k)
    then
    - $setValue(l, H, (a, b))$
    - $l \leftarrow l + 1$
  - **else if** belongs(a, C_k)
    then
    - $setValue(l, H, (a, high(C_K)))$
    - $l \leftarrow l + 1$
    - **for each** $p \in (k + 1, \ldots, m)$
      do
      - **if** belongs(b, C_p)
        then
        - $setValue(l, H, (low(C_p, b)))$
        - $l \leftarrow l + 1$
      - **else**
        - $setValue(l, H, low(C_p, high(C_p)))$
        - $l \leftarrow l + 1$

return (H)

In Algorithm IV.2, starting with (a, b) the first element in G and the first cell C_1, if a and b are in the same cell, we add (a, b) to H. If not, we consider the next cell. If b lies in the next cell C_2 we split (a, b) into two elements $(a, high(C_1))$ and $(low(C_2), b)$, and add them to H. If b lies in the next cell C_3, we split (a, b) into three elements $(a, high(C_1))$, $(low(C_2), high(C_2))$, and $(low(C_3), b)$. Similarly the other elements in G are examined to populate H.

To find if a road segment is curved, we use Algorithm IV.3. For a given cell, we check if elements of H are part of the cell coordinates. Algorithm IV.3 takes in C_i, the i^{th} polyline segment of the road, as an input, and returns h, an ordered set of curve endpoints present in the cell, where $h \subset H$. If the polyline segment is non-curving, it returns a null set. Algorithm IV.3 is called as a routine by the main Algorithm IV.5.

In Algorithm IV.3 we create h, the ordered set of curve endpoints in the given cell C_i. For each set of coordinates in H, we compare it with the cell coordinates, and populate the set h in case the curve endpoints lie within the coordinates of the cell. If the cell is non-curving, the set h is an empty set.

Algorithm IV.3: IS_ROAD_SEGMENT_CURVED(C_i, H)

$k \leftarrow 1$
$j \leftarrow cardinality(H)$
$l \leftarrow 1$
$h \leftarrow \emptyset$
for each $k \in (1, 2, \ldots, j)$
do
- $(a, b) \leftarrow getValue(k, H)$
- **if** belongs(a, C_i) **and** belongs(b, C_i)
  then
  - $setValue(l, h, (a, b))$
  - $l \leftarrow l + 1$
- $k \leftarrow k + 1$

return (h)

Algorithm IV.4 is used to find the area of a curved cell. It is called as a routine by the main Algorithm IV.5. Algorithm IV.4 takes in a curved cell C_i, the i^{th} polyline segment, and returns the cumulative area of the sub-cells. Each of these could be curving or non-curving sub-cells, such that each curved sub-cell is comprised of a circular horizontal curve.

Algorithm IV.4: AREA_OF_CURVED_CELL(h, C_i)

$k \leftarrow 1$
$j \leftarrow cardinality(C_i)$
$n \leftarrow 1$
$m \leftarrow 1$
$A' \leftarrow 0$
while $n \leq j$
do
- $(a, b) \leftarrow getValue(k, h)$
- **if** equals(a,getValue(n,C_i))
  then
  - /* Curving sub-cell */
  - $m \leftarrow index(b, C_i)$
  - $R \leftarrow findRadiusCurvature(n, m, C_i)$
  - $S \leftarrow findArcLength(n, m, C_i)$
  - $l' \leftarrow 2R \sin \frac{S}{2R}$
  - $A' \leftarrow A' + 4wR \sin^{-1} \frac{l'}{2R}$
  - $n = m + 1$
- **else**
  then
  - /* Non-curving sub-cell */
  - $m \leftarrow index(a, C_i)$
  - $S \leftarrow findArcLength(n, m - 1, C_i)$
  - $a' \leftarrow \frac{S}{2}$
  - $A' \leftarrow A' + wa'(1 + \sqrt{1 - (w/a')^2})$
  - $n \leftarrow m + 1$
- $k \leftarrow k + 1$

return (h)

Algorithm IV.4 takes C_i cell coordinates and h ordered sets of curve endpoints of C_i as input. Considering the first curve endpoints present in the cell, we check if the starting vertex of the curve is the first vertex of the cell. If it is the first vertex of the cell, then we have a curved sub-cell and calculate the area for the sub-cell, and if not, we consider it a non-curving sub-cell and calculate the area of the same. Similarly, we consider other curving or non-curving sub-cells of a curved cell, and get the cumulative area of the cell.

Algorithm IV.5 suggests a deployment scheme for each of the segments, as described in Section III-B. It uses Algorithm IV.3 and Algorithm IV.4. In Algorithm IV.5, for each of

Algorithm IV.5: DEPLOYMENT_SCHEME(T, w, a, θ)

/* Calculate area of cell, A and number of nodes, N */
$A \leftarrow wa(1 + \sqrt{1 - (w/a)^2})$
$N \leftarrow \frac{wa(1+\sqrt{(1-w^2/a^2)})}{w^2 \tan(\frac{\theta}{2})}$

for each $C_i \in (C_1, \ldots, C_m)$

do
- /* Calculate area of cell, A' and number of nodes, N' */
- /* from Algorithm IV.3*/
- $h \leftarrow IS\_ROAD\_SEGMENT\_CURVED(C_i, H)$
- **if** $isNull(h)$
  - **then**
    - /* Non-curving cell */
    - /* (parametric form) */
    - $x \leftarrow f(t)$, $y \leftarrow g(t)$ and $z \leftarrow h(t)$
    - /* Calculate S, arc length of segment */
    - $S \leftarrow \int \sqrt{(f'(t))^2 + (g'(t))^2 + (h'(t))^2}$
    - $a' \leftarrow \frac{S}{2}$
    - $A' \leftarrow wa'(1 + \sqrt{1 - (w/a')^2})$
    - $N' \leftarrow \frac{wa'(1+\sqrt{1-(w/a')^2})}{w^2 tan\theta/2}$
- **else**
  - **then**
    - /* Curving cell */
    - /* (from Algorithm IV.4) */
    - $A' \leftarrow AREA\_OF\_CURVED\_CELL(h, C_i)$
    - $N' \leftarrow \frac{A'}{w^2 tan\theta/2}$
- Let $q \leftarrow \frac{N'}{N}$.
- /* Divide the cell into q subcells. */
- /* Place N leaf nodes at a gap of $2w\tan(\frac{\theta}{2})$. */
- /* q head nodes at a gap of $\frac{2a}{q}$. */
- /* The first head node to be placed at $\frac{a}{q}$ from the start. */

the cells we calculate the area of the cell and the number of leaf nodes in the cell. Then we find the ratio of the computed number of leaf nodes with the number of leaf nodes in a plain cell. This ratio determines the need of dividing the cell into sub-cells or the need of a more powerful head node.

Algorithm IV.5 returns the value q, the ratio of the greater power of the required head node as compared with the head node for a corresponding plain cell. An alternative would be to place one head node at a distance of a. In both cases, we need to place N leaf nodes at a gap of $2w\tan(\frac{\theta}{2})$.

C. Algorithm Performance

The algorithms have a computation complexity of the order of length of the road, i.e., $\mathcal{O}(n)$. Algorithm IV.5, which provides the sensor layout of each cell, needs inputs from Algorithm IV.1, which segregates the road into cells. Algorithms IV.1 and IV.2 are prerequisites for Algorithm IV.5, but once the cell coordinates are known, Algorithm IV.5 can be parallelized.

V. URBAN GEOSPATIAL DATA

A geospatial digital vector storage format named a "shapefile," specified by the Environmental Systems Research Institute (ESRI) [44], is a (mostly) open specification used to describe various spatial geometries such as points, polylines, and polygons. For the purpose of simulation, we obtained spatial information for roads in the city of Bangalore, India, in shapefile format. We classified roads into two categories, "Arterial" and "All" (this roughly agrees with the nomenclature used by urban planners in India). "Arterial" roads are the roads within the city that are used the most, and "All" roads include Arterial roads as well as other miscellaneous roads.

The estimation of resources required to build the system can be recorded as shown in Table IV, which shows consolidated information about the number of head nodes and leaf nodes to be deployed based on the sensor placement strategy described here, for different roads (i.e., "Arterial" roads and "All" roads in Bangalore). The overall cost to be incurred for this deployment can be estimated, which in turn can be used to compare the costs incurred when the sensors are deployed randomly and when the sensors are placed intelligently using the proposed sensor placement strategy.

We have assumed the amortized costs of a head node and a leaf node to be $2000 and $200 respectively. This estimate has been made considering the approximate present-day costs of all the sensor mote components, the costs of labor to put together these components, and also deployment costs. In other words, the costs of other equipment including the servers and other support infrastructure are considered to be amortized over the deployed head and leaf nodes only. Since the deployment costs involved can be known *a priori*, the road segments that are heavily used (i.e., "Arterial" roads) or those that are accident-prone can be given higher priority for wireless sensor deployment.

From the simulation results recorded in Table IV, it can be observed that the ratio of number of leaf nodes to head nodes for "Arterial" roads is much lower than that for other minor roads. The reason for this is that the "Arterial" roads are generally well-maintained and are more likely to require plain cells for the most part, while the other minor roads have many sharp turns and obstructions and are more likely to require curved cells. Based on the cell architecture described in Section III-A and the results recorded in Table III, we know that when a curved cell is transformed into a plain cell, due to an increase in the transmission range the curved cell area covered by a leaf node is greater than the plain cell area covered by the same leaf node, and there is a need for more leaf nodes to be deployed for a curved cell.

TABLE IV
RESOURCE ESTIMATION

Roads Mapped	Head Nodes	Leaf Nodes	Cost ($)
Bangalore (Arterial)	6985	13970	16,764,000
Bangalore (All)	26223	94546	71,355,200

The final deployment plan that can be used to place sensors on roads is excerpted in Table V. It shows a sample of the list of all the sub-cells along with the information on the number of leaf nodes, the area covered, and the distance between any two leaf nodes placed within a sub-cell. The coordinates of each road segment with the details of the number of sensors deployed in each of them can also be recorded.

TABLE V
DEPLOYMENT PLAN : A SAMPLE OF A COMPLETE SET OF RECORDS

ID	*StartX* (*long.*)	*StartY* (*lat.*)	*EndX* (*long.*)	*EndY* (*lat.*)	Leafs	Area (m^2)	Length (m)	Leaf Gap (m)
1	77.7132	12.97154	77.71305	12.971806	13	152.59	33.79	2.69
2	77.71305	12.971806	77.71289	12.972071	13	154.57	34.23	2.69
3	77.71289	12.972071	77.71274	12.972337	13	152.59	33.79	2.69
4	77.71274	12.972337	77.712585	12.972602	13	153.35	33.95	2.69
5	77.712585	12.972602	77.71243	12.972869	13	154.22	34.15	2.69
6	77.71243	12.972869	77.7123	12.973148	13	154.04	34.11	2.69
7	77.7123	12.973148	77.71222	12.973328	8	98.81	21.84	2.69
8	77.71222	12.973328	77.71219	12.973392	3	35.8	7.83	2.69

VI. CONCLUSIONS

The motivation for this work arose from the fact that it is beyond question that sensor network technologies will find uses in highway infrastructures in the foreseeable future. However, the deployment strategy and architecture for such a sensor network system have not hitherto been explored in depth. Placement of sensor nodes using random sprinkling is known not to be effective. This work thus considers a cell model similar to ones used in studies of traffic flows, and also takes into account the different types of cells, based on the three-dimensional characteristics of road surfaces. Using basic trigonometric and geometric reasoning, we arrive at expressions for the number of nodes required in each cell. This in turn gives rise to algorithmic strategies that can be used for sensor placement. We validate this entire approach by computing a deployment plan given real urban geospatial data, which in turn leads to a simple cost estimate. This approach can be followed fairly easily by urban planners, highway engineers, and sensor network designers anywhere.

ACKNOWLEDGEMENTS

The authors would like to thank Suvarna Bhatt and Ted Herman for a detailed reading of, and corrections to, an earlier draft of this paper. The authors would also like to thank S. Rajagopalan for providing shapefiles and other simulation data, and Balakrishnan K. and K. Vanamali for help with the simulation. Careful and detailed comments by a reviewer helped clarify several key points, and highlighted the primary features of the work in a much better way, which the authors gratefully acknowledge.

REFERENCES

[1] S. Ghosh and S. Rao, "Sensor network design for smart highways," in *5th Annual IEEE Conference on Automation Science and Engineering (IEEE CASE 2009)*, Bangalore, India, Aug. 2009, pp. 353–360, doi:10.1109/COASE.2009.5234136.

[2] M. Karpiński, A. Senart, and V. Cahill, "Sensor networks for smart roads," in *PERCOMW '06: Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 306–310.

[3] S. S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference, 2003 (WCNC 2003)*, vol. 3, Mar. 2003, pp. 1609–1614.

[4] R. Shorey and C. M. Choon, *Mobile, Wireless and Sensor Networks: Technology, Applications and Future Directions*. Wiley-Interscience, 2005.

[5] S. GaoTao and L. MingHong, "A hierarchical architecture for energy-efficient information dissemination in sensor networks," in *Proceedings of the Second International Conference on Embedded Software and Systems (ICESS '05)*. Los Alamitos, CA, USA: IEEE Computer Society, 2005, pp. 316–321.

[6] S. Filin and Y. Doytsher, "A linear mapping approach to map conflation: Matching of polylines," *Surveying and Land Information Systems*, vol. 59, no. 2, pp. 107–114, 1999.

[7] Q. Zhang and I. Couloigner, "Automatic road change detection and GIS updating from high spatial remotely-sensed imagery," *Geo-Spatial Information Science*, vol. 7, no. 2, pp. 89–95, Jun. 2004.

[8] ——, "A framework for road change detection and map updating," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 34, pp. 720–734, 2004.

[9] L. D. Postema, "Automated Highway Systems: Incremental deployment as a solution for the future of transportation," *Washington Internships for Students of Engineering*, Aug. 1998. [Online]. Available: http://www.wise-intern.org/journal/1998/POSTEMA.PDF

[10] P. Varaiya, "Smart cars on smart roads: Problems of control," *IEEE Trans. Autom. Control*, vol. 38, no. 2, pp. 195–207, Feb. 1993.

[11] M. Nekovee, "Sensor networks on the roads: the promises and challenges of vehicular adhoc networks and grids," 2005. [Online]. Available: http://www.semanticgrid.org/ubinesc/ubi_v1.pdf

[12] O. Kwon, S.-H. Lee, J.-S. Kim, M.-S. Kim, and K.-J. Lee, "Collision prediction at intersection in sensor network environment," *Journal of Surveying Engineering*, pp. 982–987, Sep. 2006.

[13] S. Coleri, S. Y. Cheung, and P. Varaiya, "Sensor networks for monitoring traffic," in *Forty-Second Annual Allerton Conference on Communication, Control, and Computing*. University of Illinois, Aug. 2005.

[14] J. Santa and A. F. Gomez-Skarmeta, "Sharing context-aware road and safety information," *IEEE Pervasive Comput.*, vol. 8, no. 3, pp. 58–65, Jul. 2009, doi:10.1109/MPRV.2009.56.

[15] V. Terziyan, O. Kaykova, and D. Zhovtobryukh, "UbiRoad: Semantic Middleware for Context-Aware Smart Road Environments," in *Fifth International Conference on Internet and Web Applications and Services (ICIW 2010)*, May 2010, pp. 295–302, doi:10.1109/ICIW.2010.50.

[16] A. Ababnah and B. Natarajan, "Optimal control-based strategy for sensor deployment," *IEEE Trans. Syst., Man, Cybern. A*, vol. 41, no. 1, pp. 97–104, Jan. 2011, doi:10.1109/TSMCA.2010.2049992.

[17] R. V. Kulkarni and G. K. Venayagamoorthy, "Bio-inspired algorithms for autonomous deployment and localization of sensor nodes," *IEEE Trans. Syst., Man, Cybern. C*, vol. 40, no. 6, pp. 663–675, Nov. 2010, doi:10.1109/TSMCC.2010.2049649.

[18] J. M. Kay and J. Frolik, "An expedient wireless sensor automaton with system scalability and efficiency benefits," *IEEE Trans. Syst., Man, Cybern. A*, vol. 38, no. 6, pp. 1198–1209, Nov. 2008, doi:10.1109/TSMCA.2008.2001078.

[19] M. Krysander and E. Frisk, "Sensor placement for fault diagnosis," *IEEE Trans. Syst., Man, Cybern. A*, vol. 38, no. 6, pp. 1398–1410, Nov. 2008, doi:10.1109/TSMCA.2008.2003968.

[20] D. Ganesan, R. C. Rzvan, and B. Beferull-Lozano, "Power-efficient sensor placement and transmission structure for data gathering under distortion constraints," *ACM Trans. Sen. Netw.*, vol. 2, no. 2, pp. 155–181, 2006.

[21] K. Chakrabarty, S. Iyengar, H. Qi, and E. Cho, "Grid coverage of surveillance and target location in distributed sensor networks," *IEEE Trans. Comput.*, vol. 51, pp. 1448–1453, May 2002.

[22] R. Mangharam, D. S. Weller, D. D. Stancil, R. Rajkumar, and J. S. Parikh, "Groovesim: A topography-accurate simulator for geographic routing in vehicular networks," *Proceedings of The Second ACM International Workshop on Vehicular Ad-Hoc Networks (VANET '05)*, pp. 59–68, 2005.

[23] K. D. Zoysa, C. Keppitiyagama, G. P. Seneviratne, and W. W. A. T. Shihan, "A public transport system based sensor network for road surface condition monitoring," in *NSDR '07: Proceedings of the 2007 workshop on Networked systems for developing regions*. New York, NY, USA: ACM, 2007, pp. 1–6.

[24] V. Ravelomanana, "Extremal properties of three-dimensional sensor networks with applications," *IEEE Trans. Mobile Comput.*, vol. 3, no. 3, pp. 246–257, July–September 2004.

[25] S. M. N. Alam and Z. J. Haas, "Coverage and connectivity in three-dimensional networks," *Proceedings of The Twelfth Annual International Conference on Mobile Computing and Networking (MobiCom 2006)*, pp. 346–357, Sep. 2006.

[26] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja, "Sensor deployment strategy for target detection," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. New York, NY, USA: ACM, 2002, pp. 42–48.

[27] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," *Ad Hoc Netw.*, vol. 6, no. 4, pp. 621–655, 2008.

[28] A. Ghosh and S. K. Das, "Review: Coverage and connectivity issues in wireless sensor networks: A survey," *Pervasive Mob. Comput.*, vol. 4, no. 3, pp. 303–334, 2008.
[29] J. H. Park, G. Friedman, and M. Jones, "Geographical feature sensitive sensor placement," *Journal of Parallel and Distributed Computing*, vol. 64, no. 7, pp. 815–825, Jul. 2004.
[30] C. fan Hsin and M. Liu, "Network coverage using low duty-cycled sensors: random & coordinated sleep algorithms," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*. New York, NY, USA: ACM, 2004, pp. 433–442.
[31] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2003, pp. 28–39.
[32] G. Xing, C. Lu, R. Pless, and J. A. O'Sullivan, "Co-grid: an efficient coverage maintenance protocol for distributed sensor networks," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*. New York, NY, USA: ACM, 2004, pp. 414–423.
[33] C.-F. Huang, L.-C. Lo, Y.-C. Tseng, and W.-T. Chen, "Decentralized energy-conserving and coverage-preserving protocols for wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 2, no. 2, pp. 182–187, 2006.
[34] P. Corke, P. Valencia, P. Sikka, T. Wark, and L. Overs, "Long-duration solar-powered wireless sensor networks," in *EmNets '07: Proceedings of the 4th workshop on Embedded networked sensors*. New York, NY, USA: ACM, 2007, pp. 33–37.
[35] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th ed. New York: McGraw-Hill, 2002.
[36] L. Kleinrock, *Queueing Systems*. New York: Wiley Interscience, 1975.
[37] M. D. Dikaiakos, A. Florides, T. Nadeem, and L. Iftode, "Location-Aware Services over Vehicular Ad-Hoc Networks using Car-to-Car Communication," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1590–1602, Oct. 2007, doi:10.1109/JSAC.2007.071008.
[38] C.-M. Huang and M.-S. Lin, "RG-SCTP: Using the relay gateway approach for applying SCTP in vehicular networks," in *IEEE Symposium on Computers and Communications (ISCC 2010)*. IEEE, Jun. 2010, pp. 139–144, doi:10.1109/ISCC.2010.5546743.
[39] C. F. Daganzo, "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transportation Research Part B: Methodological*, vol. 28, no. 4, pp. 269–287, Aug. 1994. [Online]. Available: http://ideas.repec.org/a/eee/transb/v28y1994i4p269-287.html
[40] H. Dong, S. M. Easa, and J. Li, "Approximate extraction of spiralled horizontal curves from satellite imagery," *Journal of Surveying Engineering*, vol. 133, no. 1, pp. 36–40, Feb. 2007.
[41] J. Stewart, *Calculus*. Pacific Grove, CA, USA: Brooks/Cole Publishing Co., 1999.
[42] C. K. Toth and D. A. Grejner-Brzezinska, "Near real-time road centerline extraction," in *ISPRS Commission III, Symposium 2002 on Photogrammetric Computer Vision (PCV02)*, 2002.
[43] "Chapter 300: Geometric Cross Section," in *CalTrans Highway Design Manual*. California Department of Transportation, 2006. [Online]. Available: http://www.dot.ca.gov/hq/oppd/hdm/hdmtoc.htm
[44] ESRI, "ESRI Shapefile Technical Description," Jul. 1998. [Online]. Available: http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf