# Constructing Accurate, Space-Efficient, Wireless Coverage Maps for Vehicular Contexts

David N. Cottingham, Robert K. Harle, Andy Hopper
Computer Laboratory, University of Cambridge, Cambridge CB3 0FD, United Kingdom
{david.cottingham, robert.harle, andy.hopper}@cl.cam.ac.uk

## ABSTRACT

Wireless connectivity for vehicles is a fast-growing market, with a plethora of different network technologies already in use. Surveys of the numbers of IEEE 802.11b/g access points in cities point to hundreds to thousands of networks within each square kilometre, with coverage areas that are not easily predicted due to the complexities of the urban environment. In order to take advantage of the diversity in wireless networks available, we need data concerning their coverage. Methods of generating such coverage maps that are accurate, space-efficient and easy to query are not a well addressed area. In this paper, we present and evaluate, using a large corpus of real-world data, novel algorithms for processing large quantities of signal strength values into coverage maps that satisfy such requirements.

## Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless*

## General Terms

Algorithms, Measurement

## Keywords

Wireless, vehicles, coverage maps

## 1. INTRODUCTION

Providing Internet access to vehicles on the move is a growing market and area of research [19]. There are many applications, from simple ones such as web browsing and e-mail to more demanding tasks such as voice-over-IP conversations. In addition, we are likely to see vehicles being used as mobile sensor platforms that upload their sensor data for further processing [14], such as in the generation of pollution maps or for congestion-aware traffic routing. Many applications will require a guaranteed minimum quality of service (QoS), particularly two-way, real-time applications such as video calling. Others, such as the background downloading of updated city maps will be more tolerant of disconnections, but will require as high a throughput as possible.

Today, many different wireless network technologies exist that could be used together to provide near-ubiquitous connectivity to vehicles. For the majority (including IEEE 802.11x, UMTS cellular, and WiMax), the throughput achievable is dependent on the signal strength experienced by the mobile terminal. As a consequence, the coverage areas of the highest throughput networks are becoming ever smaller (e.g. UMTS HSPA cell coverage is less than that of GSM GPRS, whilst 802.11g has a lower range than 802.11b). There is vast diversity in the networks available: the CarTel project [3] recorded over 32,000 distinct WiFi networks in Cambridge, USA, whilst other work found some city APs whose coverage overlapped with that of up to 85 others [1].

The main problem using multiple heterogeneous wireless networks over time is selecting when, and to what network, to perform a handover to. Handover schemes may be *reactive*, where the target network is selected on instantaneous measures such as signal strength, or *proactive*, where extrinsic information concerning the networks is used. In particular, knowledge of the coverage areas of the many networks available can enable mobile clients to increase their QoS significantly. In this work, we look at how to construct such coverage maps.

Coverage maps enable clients that are location-aware to prepare for a network handover before the first beacon for such a network is encountered, e.g. by beginning to decrease their TCP advertised window size towards zero. Clients are also free to set their stabilisation threshold to be a single beacon, as the coverage map indicates how long the network will be available (and hence if it is worth connecting to). Finally, because devices are aware of the sizes of the regions of radio shadow (i.e. how long a network is *not* available for) they are able to decide whether a handover to another overlapping network should be performed, or whether the disruption caused by the temporary radio shadow is less than that which would be caused by the handover.

We focus on the constrained problem domain of vehicles, rather than on unconstrained pedestrian mobility. The rationale behind this choice is that most long distance or high speed mobility takes place on vehicles, and hence it is here that optimising handovers will be most challenging. The vast majority of vehicles move on well-defined routes, such as the road network, railway lines, or air corridors

## 2. RELATED WORK

Various authors have proposed proactive handover algorithms that assume coverage maps are available. One approach compiled a database of signal strengths to predict handovers [24], whilst another used knowledge of the motion of a vehicle to reserve band-

width on networks that would come into range, implicitly assuming such coverage was known [13]. Most recently, the Mobisteer project [18] used a steerable-beam antenna to record, for each road segment, the wireless network with the lowest packet drop rate. On subsequent journeys, this network was automatically connected to once more. The approach we propose involves mapping the coverage of *all* wireless networks, and then allowing a mobile node to select which to connect to, based on its current needs (e.g. highest throughput, or fewest handovers), rather than choosing only one network as being the "best" for each road.

A variety of techniques have been suggested for generating coverage maps that are more detailed than those produced by the wardriving community. In most cases, the RSS and/or throughput is surveyed at a number of locations, and these are then used to predict the coverage at nearby locations which were not included in the survey. Kamakaris and Nickerson [10] exhibit human-readable contour maps of a university campus generated by linear interpolation. However, they do not go on to evaluate this method's accuracy, or whether it would scale to thousands of input points. Other work on contour map generation and contour simplification [15] has been carried out by simulation rather than measurement: a wireless access point is considered to have a boundary within which it provides service, and outside which it is unusable. Such approaches neglect the fact that the RSS value is perturbed by noise, implicitly assuming that a single measurement is representative.

Another approach used grids of cells, each with its own associated RSS value, as coverage maps. These cells could be of fixed or variable size, and their values were updated whenever a new reading was obtained. Radio propagation path loss models were used to estimate from a reading in one cell what those of its neighbourings would be [12]. However, these approaches assume that we have all necessary geographical topology information to provide to the path loss models; this is unlikely to be the case in cities, where wireless propagation is complex.

Several projects have collected large quantities of RSS or throughput data. The CarTel project carried out a large-scale survey of the performance of the wireless access points (APs) found in a city [3]. The aim was to ascertain what connection quality such APs could provide, rather than map them. Similarly, measurements carried out by Microsoft Research investigated how well beacons from APs were received by vehicles, and detailed how the locations of areas of good or bad coverage were fixed [16], but did not set out to process large amounts of raw input data into coverage maps that could be easily queried.

Wireless positioning systems are another domain for which large quantities of RSS data is collected, RADAR [2] and Intel's Place Lab [11] being two well known examples. The principal difference between our work and that in wireless positioning systems is that we seek to create highly *space-efficient* representations of the RSS data recorded during the collection phase, rather than store all of the database on a user's device. We also seek to make this space-efficient representation in such a way that it is useful for proactive handover algorithms, i.e. to be able to efficiently answer questions such as "what is the coverage area of network $x$?". Clearly, the data sets used in Place Lab (and similar systems) could be processed to answer such queries: the crucial question is *how such processing is done*, and how efficient this is. In this paper, we aim to answer these questions.

## 2.1  Contributions

In this paper we focus solely on the techniques used to process the raw RSS data into coverage maps, and do not go into detail about the *mechanisms* behind the uses of coverage maps.

Our contributions are therefore: outlining the needs and requirements for coverage maps; adapting existing algorithms from other fields to the problem of smoothing large quantities of irregularly-spaced, noisy received signal strength (RSS) data for both UMTS and 802.11b/g; using synthetic data to optimise our proposed algorithms; and, evaluating the algorithms in terms of their accuracy and compactness of result, using a corpus of real-world data collected over 3 years.

## 3.  DATA COLLECTION

As part of the Sentient Vehicles project [5] at the University of Cambridge Computer Laboratory we have collected a large quantity of sensor data from a vehicle which is driven by many members of our research group for their day-to-day activities. Sensors include vehicle parameters such as speed, coolant temperature, and engine revs, environmental aspects such as humidity and carbon dioxide concentration, and received signal strength (RSS) for different wireless networks.

We utilise an Orinoco PCMCIA IEEE 802.11b/g wireless card in conjunction with the wireless-tools utilities under Linux to scan for wireless networks as the vehicle is driven. Similarly, an Option GlobeTrotter 3G PCMCIA card connected to Vodafone's cellular network is regularly probed to ascertain UMTS signal strength. We deliberately do not attempt to record the base station being used, as cellular networks do not allow clients to choose which to use. Some areas will exist where a client obtains service from different base stations at different times: we have not found this to pose a problem in our work. Both cards utilise external antennas. RSS readings are obtained on average every 1.5 seconds from the 802.11 card and every 4 seconds from the UMTS card. These are logged fully autonomously whilst the vehicle is driven. Location information is obtained from an onboard GPS receiver (based on the highly accurate SiRFstar III chip) every two seconds. This is then linearly interpolated in order to obtain the position of each wireless RSS reading.

### 3.1  Hardware Specificity

Both the UMTS and 802.11b/g cards give hardware-specific RSS measurements, i.e. these would be different for a different hardware configuration. Previous work by Haeberlen *et al.* has shown that the relationship between the RSS values reported by different 802.11b/g cards is linear and moreover is simple to determine [9], whilst the 3GPP TS 27.007 standard provides a conversion from unitless UMTS RSS to values of signal power in units of dBm. RSS readings from different HSDPA modems should therefore be approximately consistent. Hence, in our scheme, all values in a coverage map that was collected with a hardware configuration that was different to that with which the map was being used could be easily adjusted with a one-time operation.

By collecting RSS data (as opposed to recording the instantaneous throughputs), we do not limit the utility of the coverage maps to a particular hardware configuration that is capable of a particular throughput for a given RSS value. In addition, a throughput (rather than RSS) map would be specific to a particular protocol (in particular TCP or UDP), packet size, and forward error correction rate, and hence would not be as generally applicable. In contrast, RSS data is only subject to the physical effects on the radio channel, such as attenuation and interference, which will be present no matter which higher-layer protocol utilises the channel.

### 3.2  Our Dataset

Over the course of the last 3 years we have collected in excess of 820,000 UMTS and 9.5 million IEEE 802.11 RSS data points in

and around the city of Cambridge, UK, a subset of which is shown in Figure 1. This has allowed us to build up a comprehensive data set that can be used to examine how wireless coverage varies in a real urban environment over a long period of time.



**Figure 1: Map of UMTS cellular RSS (darker implies higher RSS) and base station sites (green circles) around the city of Cambridge, UK.**

## 4. SIGNAL STRENGTH VARIABILITY

Coverage maps implicitly assume that RSS readings are stable (or vary deterministically) over time. In order to validate this assumption we recorded large numbers of RSS values in two locations, collecting 1.34 million UMTS RSS readings over 2.5 months (continuously sampling every 4 seconds), and 1.27 million 802.11b/g RSS readings over 7 months (not continuous, but including day and night periods). We found no correlation with time of day, or temperature, absolute humidity, pressure and wind speed, as recorded by our weather station. Other research such as Intel's Place Lab project has also reached similar conclusions concerning the stability of RSS readings for a given location for IEEE 802.11b/g [4, 16].

Our results showed that both UMTS and 802.11b/g RSS values for a given location can be approximated by normal distributions, having standard deviations of 3 dBm and 3.5 dBm respectively (hence 90% of values will be within 6 dBm and 7 dBm of the mean, respectively). Therefore, we make the assumption that at a given location, RSS has a single "true" value, which is perturbed by noise taken from a distribution with zero mean and with the relevant of the above standard deviations.

It is important to note that RSS can be affected by environmental factors such as cell breathing. In an urban environment, moving objects such as vehicles will also cause RSS to vary [8]. We hypothesise that the variation in the RSS values we have recorded is due to these environmental effects.

### 4.1 Relationship of Throughput to RSS

Finally, another assumption that must be validated is the relationship between throughput and RSS. This link has been experimentally shown for IEEE 802.11b/g [17]. For cellular networks this is not generally evident with robust modulation schemes, as used for GSM/GPRS. However, third generation cellular networks using HSPA or EDGE do show such dependence [6], and we have carried out *qualitative* experiments to ascertain the approximate TCP throughputs achievable on a live cellular network. Our results are given in Table 1. The relationship is relatively intuitive, since higher order modulation and coding schemes (which allow higher

throughputs) can only be used when the signal to noise ratio is high enough. We omit a similar table for 802.11g for space reasons.

| UMTS RSS (dBm) | TCP Throughput (Mbit/s) |
|---|---|
| -63 | 1.28 |
| -75 | 1.20 |
| -93 | 1.08 |
| -111 | 0.32 |
| <-111 | 0 |

**Table 1: Measured values of UMTS RSS & TCP throughput.**

## 5. COVERAGE MAPPING ALGORITHMS

The input values to a coverage mapping algorithm are in general not equally spaced, and are subject to random noise. Therefore, an algorithm should cope with such data, whilst producing maps that allow RSS to be predicted with low error, and that are space-efficient.

Linked to the second requirement is the idea that the raw sensor data from vehicles will be uploaded to a central authority, and combined by that entity using the algorithms we describe in this paper. The resulting coverage maps would then be distributed back to the vehicles. Hence, vehicles are *not* required to have significant computing resources.

In contrast to other work on coverage mapping, we focus specifically on mapping RSS on roads. This constraint allows us to space efficiently represent these maps by representing the coverage along each road as a line, rather than a surface. By reducing the problem to one of line simplification we can produce coverage *extents*. These consist of a tuple $(v_{start}, v_{end}, l_{start}, l_{end}, t)$ which consists of a start value (RSS), an end value, a start co-ordinate, an end co-ordinate (measured as the proportion along the total length of the road), and the timestamp of the most recent data point used to create the extent. An extent signifies that between the start and end points the value of the sensor concerned (RSS in the case of coverage maps) varies in a linear fashion from the start value to the end value. Hence, the output of any coverage mapping algorithm should be a small number of contiguous extents spanning the road's length.

In order to satisfy the above criteria, we have adapted four algorithms which have not been traditionally applied to this problem domain. Each algorithm is briefly described in turn, followed by our adaptations to it. We also compare our results with an established algorithm in the field.

### 5.1 Nearest Neighbour Interpolation

The simplest (but most processing intensive) technique for constructing coverage maps is to pick sample points along the road in question, estimate the value of the quantity under investigation at the sample point, and then generate extents from those sample points. We carry this out using nearest neighbour inverse-distance weighted interpolation, as proposed by Shepard [23]. This is one of the few interpolation algorithms that is able to utilise irregularly spaced input data, and does not snap it to a regular grid prior to interpolation. The technique has been used before for coverage mapping [10], and hence we include it here for comparison purposes. Our adaptations are only to make it suitable for generating extents.

#### 5.1.1 Original Algorithm

Firstly, we pick sample points separated by a particular interval, $\lambda$, along the road's length, such as every 100 metres. Regardless of the length of the road a sample point is picked at $l = 0$ and another

at $l = \text{roadLength}$. For each sample point we query the database to find the set of data points, $S$ that are within a certain maximum distance $\alpha$ (currently approximately 10 metres), above which they are considered too far away to be correlated with this sample point. The value, $v_j$, of the $j$th sample point (ordered by length along the road) is then calculated as:

$$v_j = \begin{cases} \sum_{i=0}^{|S|} \frac{s_i}{d_i^2} & \text{if } d_i > \epsilon \\ \frac{\sum_{i=0}^{|E|} s_i}{|E|} & E = \{s_i \in S | d_i \leq \epsilon\} \text{ and } |E| \neq 0 \end{cases}$$

where $d_i$ is the distance from data point $i$ to the sample point under consideration, and $s_i$ is the value at data point $i$. The second condition assumes that all data points at a distance less than or equal to $\epsilon$ (set to approximately 1 metre) are considered to be at the location of the sample point, and are hence averaged in preference to weighting the values of nearby neighbours. It should be noted that this has the possible drawback that the value of a sample point could be set to that of a nearby data point that was an outlier. However, it can be argued that this data point is at (or is very near to) the sample point, and hence should be regarded as the authoritative value.

### 5.1.2   Adaptations

Having calculated values using nearest neighbour interpolation for all the sample points along a road, we then amalgamate them into extents on the basis of how different their values are. Initially, the first extent represents only the first sample point, $v_0$. To amalgamate further sample points into it, we take the mean of the sample points currently represented by the extent, $\overline{v_{0..j}}$ (where in this case $j = 0$), and compare this to the next sample point to be amalgamated, $v_{j+1}$ (in this case $v_1$). If $\overline{v_{0..j}}\gamma \geq |v_{j+1} - \overline{v_{0..j}}|$, where $\gamma$ is in the range [0,1], then $v_{j+1}$ is amalgamated into the current extent. The higher the value of $\gamma$ the greater the allowed difference between the current mean value of the extent and the next sample point that may be averaged together. In this implementation $\gamma$ is set to 0.2. The resulting extent's start point is $l_{\text{start}} = \max(l_j - \frac{\lambda}{2}, 0)$, where $l_j$ is the position of the $j$th sample point as a fraction of the road length. Its end point is $l_{\text{end}} = \min(l_{j+1} + \frac{\lambda}{2}, \text{roadLength})$. The amalgamation process continues until there are no further sample points or $\gamma$ is exceeded, in which case a new extent is begun, and amalgamation restarts from that sample point. A special case occurs if $\text{roadLength} < \lambda$, in which case the value of a sample point at the mid-point of the road is obtained, and then the mean of the start, end and mid-point sample values is deemed to be the value for an extent spanning the entire road.

The resulting extents are pairs of start and end lengths along the road, with a single associated sensor value. Hence, when stored in the database, the extent is $(v, v, l_{\text{start}}, l_{\text{end}}, t)$.

There are two problems with this approach; firstly, there may be a large number of points within distance $\alpha$ of a sample point that make the above process very time consuming. Secondly, picking sample points at a regular distance $\lambda$ risks smoothing out features that may be significant that exist between the sample points. Whilst $\lambda$ could be dynamically varied, this would require a knowledge of the surface to be sampled, which is in essence what we are attempting to achieve with this algorithm.

## 5.2   Dominant Point Detection

Corner, or dominant point, detection was originally developed in order to derive simplified representations of two-dimensional closed curves. We examined whether these techniques could be applied instead to find the "dominant" points of a graph of RSS values over the length of a road.

### 5.2.1   Original Algorithm

We define an open digital curve $S$ as an ordered sequence of points $S = \{p_1, \cdots, p_n\}$ where each $p_i = (x_i, y_i)$, and the $x_i$s are monotonically increasing. In the case of a closed curve, $p_1$ is a neighbour of $p_n$, as the start and end point of the curve must be identical (and hence the $x_i$s are not monotonically increasing). When analysing the curve to find its "corners", the aim is to find the local curvature maxima, i.e. those points at which the rate of change of gradient with length is greatest. These are known as the *dominant points*.

The first stage is to calculate the Freeman Chain codes for all the points, and eliminate those that are collinear, as described in [7].

Next, we calculate the *region of support*, $k_i$ for each $p_i$. The larger the region of support the greater the number of input points support the hypothesis that $p_i$ is a dominant point. We briefly overview Teh and Chin's method [25].

We define $l_{ik} = |\overline{p_{i-k}p_{i+k}}|$ to be the length of a chord between two points, and $d_{ik}$ to be the perpendicular distance of $p_i$ from $\overline{p_{i-k}p_{i+k}}$ We initially set $k = 1$, and increase it until the condition

$$\frac{d_{ik}}{l_{ik}} \geq \frac{d_{ik+1}}{l_{ik+1}} \text{ for } d_{ik} > 0$$
$$\frac{d_{ik}}{l_{ik}} \leq \frac{d_{ik+1}}{l_{ik+1}} \text{ for } d_{ik} < 0$$
$$\text{false for } d_{ik} = 0$$

yields true. The final value of $k$ is stored in $k_i$, indicating that the points in the region of support for $p_i$ are $D_i = \{p_{i-k}, \cdots, p_i, \cdots, p_{i+k}\}$.

We now calculate the $k$-cosine curvature, $c_i$, the angle that the curve turns through as we traverse each $D_i$, as $c_i = \frac{\mathbf{a}_{ik} \cdot \mathbf{b}_{ik}}{|\mathbf{a}_{ik}||\mathbf{b}_{ik}|}$ Where $\mathbf{a}_{ik} = \overline{p_i p_{i+k}}$ and $\mathbf{b}_{ik} = \overline{p_i p_{i-k}}$. This implies that $c_i$ will be nearer to 1 if the angle turned through by $D_i$ is small, and tend to -1 as the angle approaches $\pi$ radians. We then perform three further elimination steps:

$k$**-Cosine discard threshold:** For each $p_i$, if $c_i > \mu$, eliminate $p_i$ from consideration. This eliminates points at the centers of very broad angles, which are unlikely to be dominant points ([26], step 4).

**Suppress small regions of support that are overlapped by neighbours:** For each $p_i$, if $k_i < k_{i+1}$ or $k_i < k_{i-1}$, eliminate $p_i$ from consideration ([26], step 5).

**Discard large angled points if adjacent to a small angled point:** For each $p_i$ that has not yet been eliminated, if $k_i = 1$ and

$p_{i+1}$ has not been eliminated $\wedge c_i \leq c_{i+1} \vee$
$p_{i-1}$ has not been eliminated $\wedge c_i \leq c_{i-1}$

then eliminate $p_i$ from consideration ([25], step 3c).

As a development to the Teh-Chin method of calculating the region of support, Wu proposed a dynamic method for determining the value of $k$ [26] that involves assuming that $k_i$ is close in magnitude to $k_{i-1}$. In this method, we start with $k = k_{i-1}$, and on the $j$th iteration try a value of $k$ that is $j$ more than $k_{i-1}$ and another that is $j$ less than $k_{i-1}$. We therefore also evaluated this method for generating coverage maps.

### 5.2.2   Adaptations

The original dominant point detection algorithms were only intended for the simplification of closed curves (polygons). In order to apply them to non-closed curves such as our graphs of RSS, we trialled two approaches. Initially the algorithm's iteration through different values of $k$ was constrained in order that $i + k \leq n$ and $i - k \geq 0$. This meant that at the beginning and end of the curve incorrect decisions were made over whether points should be discarded. To correct this we reflected the curve in the y-axis at both ends, such that if the value of $k$ exceeded the first bound given above, $d_{ik}$ was calculated between $p_i$ and the chord $\overline{p_{i-k}p_{n-(i+k-n)}}$, and similarly for the case of $i - k < 0$.

A further observation we made was that the Teh-Chin algorithm performs better on sparse data (i.e. few points per metre of road) than does Wu's algorithm, and vice-versa for dense data. We therefore implemented a dynamic algorithm that segmented input data into regions of high and low density, and applied the Teh-Chin and Wu algorithms to the relevant sections. We term this the *Density-Dependent* algorithm.

When using Wu's algorithm with dense data, a greater degree of smoothing was needed. Consequently, a second $k$-cosine discard threshold was added, $\mu_2$. Hence, the first step after calculating cosine curvatures is modified to eliminate $p_i$ if $c_i > \mu$ or $c_i < \mu_2$.

Taking advantage of the knowledge of the region of support of each point, we smoothed the output of the dominant point detection algorithms by discarding any points with a $k_i < \kappa$, where $\kappa > 1$, and is chosen by experimentation. This is because points with few others "supporting" have less raw data to support the hypothesis that this corner in the curve is due to real data rather than a few outliers. However, with sparse data, regions of support will evidently be smaller (due to fewer points per unit length of road) than those of high density. Hence, in our dynamic algorithm we also vary $\kappa$ depending on the density of the input data, using $\kappa = 2$ for high densities, and $\kappa = 0$ for low densities.

Finally, we further smooth the output by removing extents that are very short. For all remaining points $p_i$ we compare the length of $\overline{p_i p_{i+1}}$ to a threshold $\zeta$. If it is smaller than $\zeta$ we remove $p_i$ from consideration. We loop over the set of candidate points until there are no extents below the threshold. This ensures that extents that concern very small distances are ignored. Hence, $\zeta$ must be set to reflect the minimum distance over which a vehicle travelling at a plausible speed would have time to adapt its network connections in order to take advantage/cope with the change in network performance. We empirically determined $\zeta$ to be 10 metres for high densities, and 9 metres for low densities.

## 5.3 Savitzky-Golay Smoothing

Although the dominant point detection algorithms described in the previous section work well on their own, for large quantities of noisy data they are still prone to outputting either a large number of dominant points, or, if $\kappa$ and $\mu$ are too high or too low respectively, too few to be representative of the inputs. Therefore, we investigated the use of a filtering step prior to executing a dynamic dominant points algorithm.

### 5.3.1 Original Algorithm

Savitzky-Golay smoothing [22] is a windowed low-pass filter originally used for analysing chemical spectroscopy data. For each input point $p_i$ a high-order polynomial is fitted to the data within the window centred on $p_i$ using the least squares method. The value corresponding to $p_i$ that is output is the value of the fitted polynomial at the $x$ co-ordinate of $p_i$. The window is then moved to $p_{i+1}$, and an entirely separate least squares procedure is executed. The use of a polynomial fit preserves local maxima and minima better than other window-based smoothing filters [20].

We have therefore implemented the algorithm as given by Press *et al.* [20], which involves straightforward matrix operations. The algorithm assumes equally spaced data, which we in general do not have. However, with dense data, this constraint can be relaxed to one where provided that the change in the $y$ value of the input data over the window length is small (i.e. the majority of the points have similar values), the algorithm can be used. Hence, the algorithm is only of use where we have dense input data. In addition, we use a window size of 101 points (empirically determined to be large enough to achieve the necessary degree of smoothing), hence the

input must have at least this many data points in order for the algorithm to be run over it, thus excluding roads that have not been driven along multiple times.

### 5.3.2 Adaptations

The number of output points of the Savitzky-Golay smoothing step is equal to the number of input points, but the output graph now has a much smoother profile. To reduce the number of points, we pass the data into the density-dependent dominant points algorithm described in Section 5.2.2 to obtain a more space-efficient representation. Hence, whilst our implementation of the Savitzky-Golay algorithm is not innovative, coupling it with the dominant point detection algorithm is, to the best of our knowledge, a technique that has not been previously used.

## 6. SIMULATION RESULTS

Apart from the nearest neighbour interpolation algorithm, none the algorithms proposed in this paper have previously been used for processing RSS data. Therefore, synthetic data were generated, processed by the algorithms, and the results compared to the known values used to generate the synthetic data. This proof of concept stage also allowed us to optimise the algorithms' parameters in preparation for their use on real data, as described in Section 7.

## 6.1 Synthetic Data

To generate synthetic data, we examined traces recorded by our vehicle, and created two traces each for UMTS and 802.11b/g that provided a single value for any length along the road. The data used for the lines mimicked the pattern qualitatively inferred from data for real roads as recorded by our vehicle. We term these the *source curves*.

Points were generated from the source curves by picking locations along the length of the (synthetic) road at random. We term the set of such locations $X$. The number of points in $X$ was varied to simulate different densities of source data. The value of the source curve (denoted $s$) at each of these locations was calculated, i.e. $s(x), x \in X$, and then perturbed by adding noise, $n_x$ sampled from a Normal distribution with zero mean and a standard deviation of the appropriate value (3 dBm for UMTS, 3.5 dBm for 802.11b/g, see Section 4), giving $p(x) = s(x) + n_x$ as the perturbed value. In this way, synthetic curves were produced that were similar to those curves seen on real test drives, but for which the true values were known. For each of the four source curves, 40 different synthetic data sets were generated, 10 at each of four different point densities (102, 250, 500 and 1000 points per 100 metres), and each having its own unique set of (random) perturbations to the source curve.

## 6.2 Evaluation Criteria

Each of the 160 synthetic data sets were processed using each of the proposed algorithms from Section 5 (bar the Nearest Neighbour Interpolation algorithm, as this has been used previously by others for generating maps for wireless positioning algorithms), and the dominant points recorded (resulting in a function we term $d$). The true value at each point in $X$, $s(x)$, was then compared to the value at that location on the dominant points curve output by the algorithms, $d(x)$. The mean square error (MSE) of all the points in $X$ was calculated, i.e. $\frac{\sum_X (s(x) - d(x))^2}{|X|}$. The mean of the MSEs for each algorithm over all the synthetic data derived from each source curve was then calculated.

In a similar fashion, the compression ratio (CR) was also evaluated for each algorithm. This metric is commonly used to evaluate

dominant point algorithms' ability to approximate an input shape with as few points as possible. It is calculated by dividing the number of dominant points outputted by each algorithm by the number of synthetic data input points (and hence should be < 1). This provides an indication of what compression has been achieved in the representation. To a certain extent, there will exist a trade-off between how compact the result is and how accurate the predictions that can be made using it are.

## 6.3 Simulation Results

The results of using this synthetic data are shown in Figures 2 and 3. Several conclusions can be drawn from them.

**On synthetic data, the MSE is acceptably low.** Given that the standard deviation of the noise added to the source curve was 3 dBm for UMTS and 3.5 dB 802.11b/g, MSEs of less than 4 dBm (UMTS) or 5 dBm (802.11b/g) suggest that the approximation algorithms perform well.

**Compression can be more than a factor 50.** The compression ratios for Wu's and the Savitzky-Golay smoothing algorithms are very low, suggesting a very compact result. Compression ratios decrease with increasing input point density, showing that the algorithms perform well on large quantities of data.

**Wu's algorithm has poor MSE at low point densities.** This is most probably due the fact that at low densities the regions of support of neighbouring points are unlikely to be correlated, which is an assumption made by the algorithm. However, at high densities, this algorithm's MSE is comparable to that of the others, and the standard deviation of its MSE is also much reduced.

**Savitzky-Golay smoothing has consistently low MSE and CR.** The results suggest that (on synthetic data at least), this algorithm performs consistently well.

**The Teh-Chin and Density-Dependent algorithms have consistently poor CRs.** At low point densities this is offset by their better MSE than Wu's algorithm. However, at high densities, such high CRs mean Wu's algorithm is (overall) more suitable.

**The Density-Dependent algorithm combines the best of Teh-Chin and Wu.** At densities lower than 1000 points per 100 metres, the Density-Dependent algorithm has an MSE comparable to the Teh-Chin algorithm, (and similarly high CRs). At higher densities it retains its very good MSE, whilst achieving CRs lower than those of the Teh-Chin algorithm.

## 6.4 Parameter Optimisation

In addition to the above, experiments were also conducted to ascertain the best value of the $k$-cosine threshold that should be used. The distribution of $k$-cosines over the input data is surprisingly non-uniform. Instead, most values are either very close to 1 (implying an angle of close to zero), or close to 0 (implying a right angle). This distribution (particularly at high densities) is due to to the input points being relatively close together, and hence with noisy data the angles will be very sharp.

Because of this quite bimodal distribution of cosine curvatures, it was found that the MSE and CR performance of the dominant point algorithms as the cosine curvature discard threshold was varied between $-1 < \mu < 1$ was a step function, the discontinuity occurring at zero, i.e. when the angle is 90 degrees. At $\mu = -1$ or $\mu = 1$, both MSE and CR were very high, reflecting that at these discard thresholds, nearly all and none, respectively, of the input points would be discarded. A value of $\mu = -0.9$ was chosen for regions of low point density, in order that only those points with very large angles would be discarded, as they are unlikely to be important. Points with smaller angles are retained. In contrast, at high point densities, $\mu = -0.1$, and the high discard threshold, $\mu_2$,

is chosen to be 0.1, to provide a degree of smoothing of the input data, given that there are large amounts.

## 7. EXPERIMENTAL EVALUATION

Having shown that our proposed algorithms perform well on synthetic data, (where their results could be compared against a known value) and optimised their parameters, we proceeded to test their performance on real data collected by our vehicle. In order to evaluate the algorithms on a real-life data set, we executed each different algorithm on the corpus of data collected by our vehicle to generate a coverage map. We then used sensor records from several randomly selected journeys that were *not* in the input corpus in order to evaluate how accurate the predictions made by the coverage map were when compared to the real RSS values experienced on the sample journeys. We also analysed the space-efficiency of the resulting extents.

For each sample journey, each input tuple of 2-D position and sensor value $(l_i, v_i)$ was snapped to the closest point on the relevant road's centre line, becoming $(x_i, v_i)$. The database was then queried for the coverage map's stored value $s_i$ at length $x_i$ along the road. For all the input points, $d_i = v_i - s_i$ was calculated, as well as the mean and standard deviation of those differences. Each algorithm's extents were evaluated using each sample journey, in order to compare their accuracy.

The two metrics that are important in evaluating the performance of coverage mapping algorithms are the difference between predicted and actual values, and the space-efficiency of the extents. We now consider each in turn.

### 7.1 Prediction Error

In order to be successful, a coverage map's prediction for a given location must be similar enough to a value subsequently recorded at that location to be useful. The difference between these two values will in part be due to the natural variation in RSS values, as explained in Section 4.

Tables 2 and 3 show the different algorithms compare for UMTS and 802.11b/g, whilst Figures 5(a) and 5(c) are the corresponding CDFs. These show that for UMTS prediction, the Density-Dependent and Wu's algorithms perform best, with the Savitzky-Golay smoothing algorithm also having a low prediction error (90% confidence interval of 12 dBm). For 802.11b/g, the Savitzky-Golay algorithm is by far the most accurate, with a confidence interval of only 10.40 dBm. Figures 4(a) and 4(b) show the spread of prediction errors. Significantly, the Savitzky-Golay algorithm has only one outlier (circles on the graph) for UMTS prediction errors, showing that its predictions are consistently good. The Density-Dependent and Wu's algorithms do have outliers, suggesting that their performance can sometimes be very poor.

An important question is whether these values are significant, e.g. does a 90% confidence interval of 12.00 dBm impact a user far more than one of 14.00 dBm? Table 1 shows that for UMTS the relationship between TCP throughput and RSS does not appear to be linear: errors in RSS prediction will be more significant in areas of poor coverage. We posit that that this because in areas of poor coverage packet losses will be more frequent, each of which will cause TCP's congestion window to fall to near zero. Hence, the window size will never be allowed to increase to large values that would allow high throughputs. Given this, we estimate that for RSS values below -90 dBm, an error of 1 dBm is approximately equivalent to 40 Kb/s, whilst the same error at an RSS above -90 dBm would be far less (7 Kb/s). Hence, in areas of poor coverage, a 90% confidence interval of 12.00 in RSS translates into a TCP estimate that is approximately 80 Kb/s more accurate than one of 14.00

**(a) UMTS (1)**    **(b) UMTS (2)**    **(c) 802.11b/g (1)**    **(d) 802.11b/g (2)**
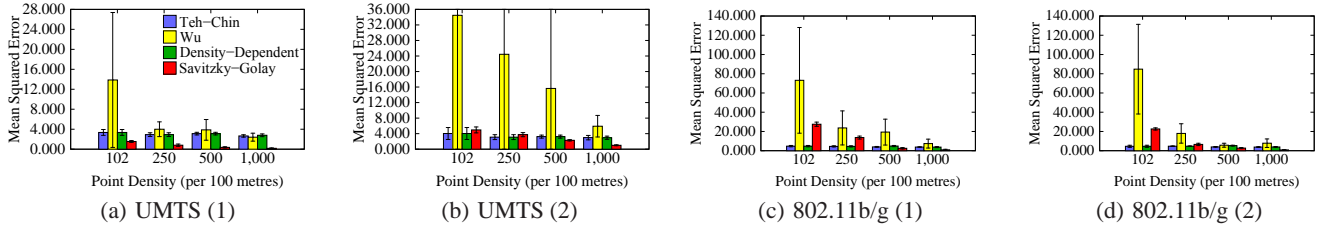
**Figure 2: Comparison of Mean Squared Errors (lower is better) in representation achieved at different point densities, by algorithm, of synthetic data generated from two different source curves for each of UMTS and 802.11b/g.**



**(a) UMTS (1)**    **(b) UMTS (2)**    **(c) 802.11b/g (1)**    **(d) 802.11b/g (2)**
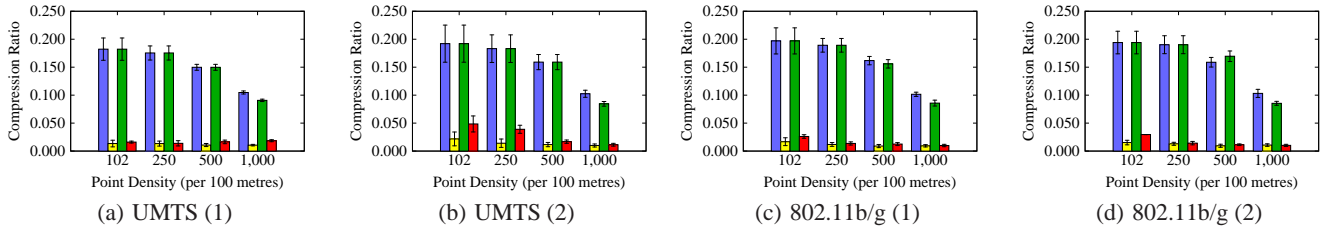
**Figure 3: Comparison of Compression Ratios (lower is better) achieved at different point densities, by algorithm, of synthetic data generated from two different source curves for each of UMTS and 802.11b/g.**

dBm. Hence, the difference in algorithm prediction performance is significant for an end user.

Overall, the algorithms' performance in the very worst case would be an error of 480 Kb/s (poor coverage), or 84 Kb/s (good coverage). These are acceptably low compared to the maximum throughputs achievable, and hence show the utility of the coverage maps generated by our algorithms.

Similarly, we estimate that a worst case error of 10.00 dBm for 802.11g would correspond to a throughput difference of 4-5 Mb/s. Whilst this is a large value, it should be born in mind that the maximum TCP throughput achievable with 802.11g is 20 Mb/s. Hence, a user will still derive utility from a prediction that is subject to such error.

Overall, we conclude that Savitzky-Golay smoothing followed by the an application of the density-dependent dominant points algorithm, performs best, as it combines a low 90% confidence interval in prediction errors for both UMTS and 802.11b/g with few severe prediction errors (outliers).

| Algorithm | $\overline{d}$ | $\sigma_d$ | 90% C.I. | $|d|$ | Tests |
|---|---|---|---|---|---|
| Nearest Neighbour | -9.64 | 4.40 | 14.64 | 9.74 | 748 |
| Teh-Chin | -7.94 | 4.34 | 13.30 | 8.20 | 748 |
| Wu | -7.72 | 3.84 | 12.00 | 7.92 | 748 |
| Density-Dependent | -6.86 | 4.40 | 12.00 | 7.08 | 748 |
| Savitzky-Golay | -7.90 | 3.56 | 12.26 | 8.02 | 748 |

**Table 2: Prediction errors for UMTS ($d$), all in dBm.**

## 7.2 Extent Density

Ideally, our algorithms should produce as few extents as possible per unit length of road (i.e. a low extent density, ED), in order that the coverage database distributed to a vehicle be small and can be efficiently queried. Tables 4 and 5 show how the Nearest Neighbour and Wu's algorithms perform well, with few extents produced per metre of road for both UMTS and 802.11b/g. Meanwhile, the Density-Dependent and Teh-Chin's algorithm perform poorly (a

| Algorithm | $\overline{d}$ | $\sigma_d$ | 90% C.I. | $|d|$ | Tests |
|---|---|---|---|---|---|
| Nearest Neighbour | -7.16 | 4.86 | 13.00 | 7.39 | 83 |
| Teh-Chin | -5.87 | 6.41 | 13.61 | 7.34 | 79 |
| Wu | -4.65 | 6.61 | 13.57 | 6.44 | 65 |
| Density-Dependent | -7.71 | 6.32 | 14.00 | 8.91 | 59 |
| Savitzky-Golay | -5.07 | 4.39 | 10.40 | 5.71 | 72 |

**Table 3: Prediction errors for 802.1b/g ($d$), all in dBm.**

low value of $ED^{-1}$ shows how few metres each extent covers), with the Savitzky-Golay algorithm being between these two groups, as shown in the CDFs in Figures 5(b) and 5(d). Figures 4(c) and 4(d) show the distributions of extent densities. Significantly, for UMTS the Nearest Neighbour algorithm has several outliers which are indicative occasional very poor performance (many extents generated per metre). In contrast, Wu's and the Savitzky-Golay algorithm do not have significant outliers. These three algorithms have similar distributions for 802.11b/g.

Given the above, we can conclude that Wu's algorithm, and Savitzky-Golay smoothing followed by an application of the dynamic density-dependent algorithm, perform well as regards the number of extents generated per metre, and hence are space-efficient. As an example, a typical WiFi hotspot covering 200 metres of road would require only 6 extents in order to represent its coverage,

Overall, we conclude that the Savitzky-Golay smoothing followed by an application of the dynamic density-dependent algorithm performs best out of the algorithms we propose, given its low prediction error and good space efficiency. In addition, we note its benefits over the Nearest Neighbour interpolation algorithm that has been used for coverage mapping efforts in the past.

## 8. SCALABILITY

The requirements for computing resources required to produce coverage maps are not onerous. We tested our algorithms on a Pentium IV 3.2 GHz processor with 1 GB of RAM. The system

(a) UMTS prediction errors    (b) 802.11b/g prediction errors    (c) UMTS extent densities    (d) 802.11b/g extent densities
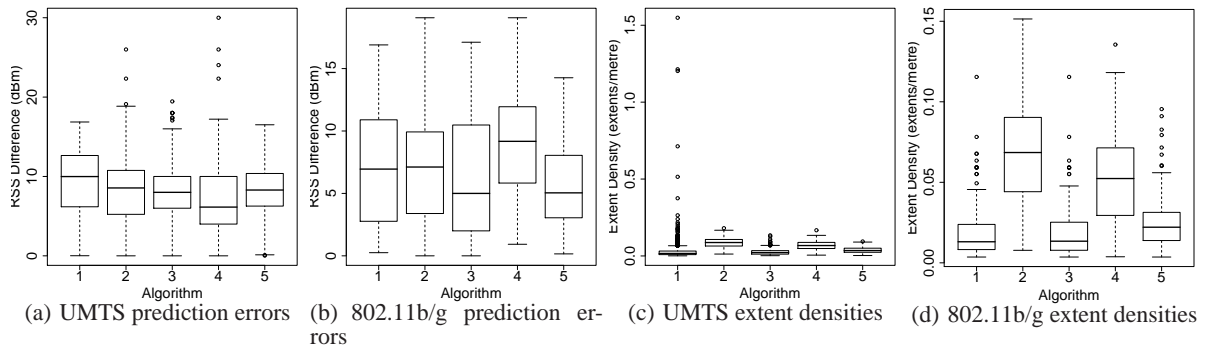
**Figure 4: Box plots of prediction errors and extent densities by algorithm (Key: 1 Nearest Neighbour, 2 Teh-Chin, 3 Wu, 4 Density-dependent, 5 Savitzky-Golay).**

| Algorithm | ED | ED$^{-1}$ | Num. of Roads |
|---|---|---|---|
| Nearest Neighbour | 0.030 | 33.409 | 1380 |
| Teh-Chin | 0.086 | 11.607 | 234 |
| Wu | 0.028 | 36.362 | 234 |
| Density-Dependent | 0.068 | 14.615 | 234 |
| Savitzky-Golay | 0.039 | 25.756 | 107 |

**Table 4: Mean Extent Density (ED) (extents/m) for UMTS.**

| Algorithm | ED | ED$^{-1}$ | Num. of Roads |
|---|---|---|---|
| Nearest Neighbour | 0.019 | 53.383 | 178 |
| Teh-Chin | 0.068 | 14.610 | 178 |
| Wu | 0.019 | 52.245 | 161 |
| Density-Dependent | 0.052 | 19.357 | 170 |
| Savitzky-Golay | 0.027 | 36.577 | 83 |

**Table 5: Mean Extent Density (ED) (extents/m) for 802.11b/g.**

analysed 2,444 roads in the Cambridge area, finding 1,380 roads that had a enough RSS data points to construct a coverage map, and 115 with one or more 802.11b/g networks with the minimum number of points necessary.

The five algorithms were each run on each candidate road, processing a total of over 765,000 UMTS and over 1.2 million 802.11b/g data points. 5,879 UMTS extents and 2,396 802.11b/g extents were generated and added to the database. The entire process took 4,714 seconds. The running time includes the printing of a significant amount of debug output, and therefore could be further decreased. In addition, we note that this figure involves processing each road up to 5 times, whereas in a real deployment only one algorithm would be used.

As we and others have seen, the number of 802.11b/g APs in a city can be thousands [1, 3]. Whilst a coverage map could be made to include coverage information for each of these APs, i.e. the many of them will not have sufficient coverage to make their usage by a vehicle worthwhile, or will not permit such usage. Hence, a coverage map only need include those APs that could be useful. This might include all the hotspots for a particular provider that a user has a subscription to, or all those belonging to a community WiFi scheme such as Fon (http://www.fon.com/).

Finally, we note the distinction between the (very large) corpus of RSS data that is collected and the (compact) coverage map. The raw readings are uploaded to a server, which then uses our algorithms to generate a coverage map, and distributes this to the vehicles. Hence, vehicles need not have vast computing resources.
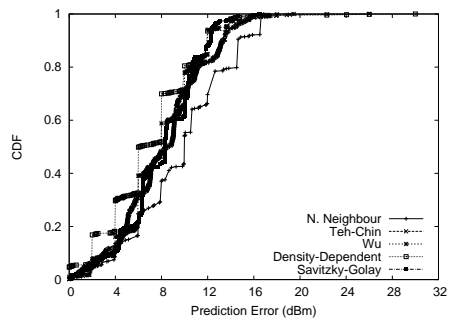
## 9. SENSITIVITY TO CHANGE

A key question concerns how frequently a coverage map needs to be updated. Our experience in Cambridge has shown that cellular network deployments appear to be relatively static over long periods of time (see Figure 1), as might be expected, given the cost of installing new base stations. In contrast, user-managed wireless LANs are likely to (dis)appear much more frequently. We suggest that the more investment required in a network deployment the less likely it is to frequently change in coverage. Fortuitously, it appears that users are more likely to use such networks due to their greater ubiquity, rather than connecting to wireless LANs belonging to private dwellings, in part because of the billing and security infrastructure required for public access to a network. Hence, we believe that the principal networks that will be present in a coverage maps will not require very frequent updates. Based on our data collected over 3 years, it appears that a survey would be needed at least yearly. This is likely to be due to the large number of students living in our city, and hence there is a high degree of population churn; other areas are likely to differ.
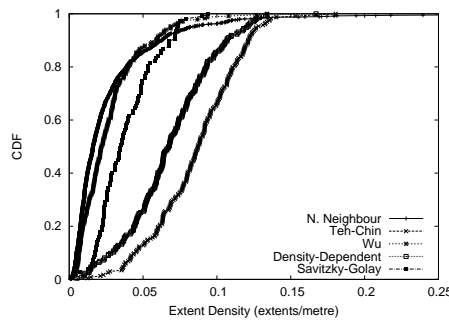
## 10. CONCLUSION

Wireless network access for vehicles is increasingly in demand, though its deployment is not ubiquitous and hence is unpredictable. Proactive handover algorithms have been proposed that depend on the existence of coverage maps to perform optimal network selection. This paper has presented and evaluated four novel algorithms for generating coverage maps, and compared them to the Nearest Neighbour interpolation algorithm used in other work. We have found that using the Savitzky-Golay smoothing algorithm, followed by an application of our density-dependent dominant points algorithm, has the lowest prediction errors for both UMTS and IEEE 802.11b/g, whilst maintaining adequate space-efficiency.
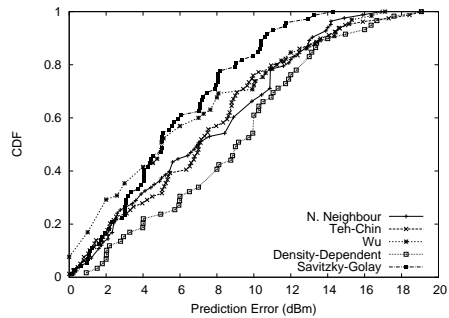
## 11. REFERENCES

[1] A. Akella, G. Judd, S. Seshan, and P. Steenkiste. Self-management in chaotic wireless deployments. *Wireless Networks*, 13(6):737–755, Dec. 2007.

[2] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *IEEE INFOCOM*, volume 2, pages 775–784, 2000.

[3] V. Bychkovsky et al. A measurement study of vehicular internet access using in situ Wi-Fi networks. In *Proc. ACM MobiCom*, Sep. 2006.

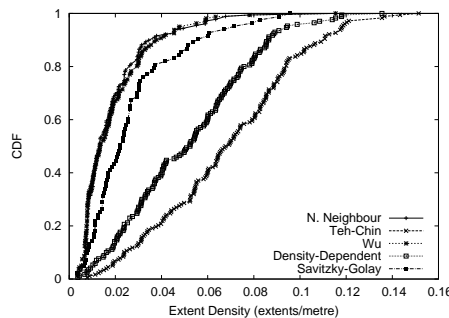[4] Y. Cheng et al. Accuracy characterization for metropolitan-scale Wi-Fi localization. In *Proc. ACM*

(a) UMTS prediction errors



(b) UMTS extent densities



(c) 802.11b/g prediction errors



(d) 802.11b/g extent densities

**Figure 5: CDFs of prediction error and extent density. For clarity, the UMTS extent densities x-axis has been truncated.**

*MobiSys*, pages 233–245, 2005.

[5] J. J. Davies, D. N. Cottingham, and B. D. Jones. A sensor platform for sentient transportation research. In *Proc. EuroSSC*, LNCS 4272, pages 226–229, Oct. 2006.

[6] J. Derksen et al. HSDPA performance and evolution. *Ericsson Review*, 84(3), 2006.

[7] H. Freeman. Computer processing of line-drawing images. *ACM Comput. Surv.*, 6(1):57–97, 1974.

[8] G. Gaertner and V. Cahill. Understanding link quality in 802.11 mobile ad hoc networks. *IEEE Internet Computing*, 8(1):55–60, Jan. 2004.

[9] A. Haeberlen et al. Practical robust localization over large-scale 802.11 wireless networks. In *Proc. ACM MobiCom*, pages 70–84, 2004.

[10] T. Kamakaris and J. V. Nickerson. Connectivity maps: Measurements and applications. In *Proc. HICSS*, pages 307–311, January 2005.

[11] A. LaMarca et al. Place Lab: Device positioning using radio beacons in the wild. In *Proc. PerCom*, volume 3468 of *LNCS*, pages 116–133, 2005.

[12] H. Larkin. Wireless signal strength topology maps in mobile adhoc networks. In *Proc. Embedded and Ubiquitous Computing Conference*, volume LNCS 3207, pages 538–547, 2004.

[13] D.-S. Lee and Y.-H. Hsueh. Bandwidth-reservation scheme based on road information for next-generation cellular networks. *IEEE Trans. Vehicular Technology*, 53(1):243–252, Jan. 2004.

[14] U. Lee, E. Magistretti, M. Gerla, P. Bellavista, and A. Corradi. Dissemination and harvesting of urban data using vehicular sensor platforms. *Submitted to IEEE Trans. Vehicular Technology*, 2008.

[15] S. Lück, C. M. Mueller, M. Scharf, and R. Fetscher. Algorithms for hotspot coverage estimation based on field strength measurements. In *Proc. IEEE VTC*, pages 1086–1090, Apr. 2007.

[16] R. Mahajan, J. Zahorjan, and B. Zill. Understanding WiFi-based connectivity from moving vehicles. In *Proc. ACM IMC*, pages 321–326, Oct. 2007.

[17] C. Na, J. K. Chen, and T. S. Rappaport. Measured traffic statistics and throughput of IEEE 802.11b public WLAN hotspots with three different applications. *IEEE Trans. Wireless Communications*, 5(11):3296–3305, Nov. 2006.

[18] V. Navda et al. MobiSteer: using steerable beam directional antenna for vehicular network access. In *Proc. ACM MobiSys*, Jun. 2007.

[19] J. Ott and D. Kutscher. A disconnection-tolerant transport for drive-thru internet environments. In *Proc. IEEE INFOCOM*, Mar. 2005.

[20] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C++*, chapter 14, pages 655–660. Cambridge University Press, 2002.

[21] A. Rosenfeld and E. Johnston. Angle detection on digital curves. *IEEE Trans. Computers*, C-22:875–878, Sep. 1973.

[22] A. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36:1627–1639, Jul. 1964.

[23] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proc. 23rd ACM national conference*, pages 517–524, 1968.

[24] M. Siebert, M. Lott, M. Schinnenburg, and S. Göbbels. Hybrid information system. In *Proc. IEEE VTC*, May 2004.

[25] C.-H. Teh and R. T. Chin. On the detection of dominant points on digital curves. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(8):859–872, Aug. 1989.

[26] W.-Y. Wu. A dynamic method for dominant point detection. *Graphical Models*, 64:304–315, 2003.