A New Mobility Model Based on Maps

Jens Kammann, Michael Angermann

German Aerospace Center (DLR)
Institute of Communications and Navigation
P.O. Box 1116, D-82230 Wessling, Germany
{*Jens.Kammann, Michael.Angermann*}@dlr.de

Bruno Lami

Système Électronique et Inform. Industrielle
École polytechnique de l'université de Nantes
P.O. Box 50609, F-44306 Nantes, France

Abstract— **In this paper we present a mobility model for simulations of wireless access scenarios. This new model has been developed to reduce the shortcomings of previous models in capturing higher-order statistical dependencies in user behavior. The proposed model strikes a balance between the abstraction level of some purely stochastic path generation models that lack certain important properties of the real world and completely deterministic models that fail to produce answers to inherently stochastic problems such as call admission control, hand-off/handover or prefetching of data. The proposed model uses freely definable building maps and radio cell coverage information and generates sensible paths between randomly selected way-points. The paths are generated by a combined diffusion/steepest gradient algorithm. Various models for user speeds can be incorporated. The resulting patterns of user mobility and temporal radio coverage are well applicable in driving simulations with the purpose of addressing key questions in mobile wireless scenarios.**

I. INTRODUCTION

Current research in mobile wireless network access is facing highly interesting and demanding challenges due to increased performance requirements, and more flexible but also more complex options in access technologies. Mobile devices no longer rely only a single wireless interface (e.g. GSM/GPRS, UMTS, IS95, cdma2000, W-CDMA) for data and voice traffic but can concurrently employ these classical interfaces in conjunction with short-range communication technologies such as Wireless LAN IEEE 802.11 and Bluetooth as well as broadcast technologies e.g. data broadcast over DVB. We found that in order to understand and evaluate the networking issues of scenarios with this kind of devices and the corresponding infrastructure we need both traffic models and user mobility models. These should capture higher-order statistics of user behavior than are available today in order to answer pending questions in many diverse topics, such as:

- dynamic adaptive routing (both in network layers and application layers due to the availability of multiple parallel TCP/IP stacks on the devices) via the different interfaces
- call admission control
- hand-off/handover of voice and data calls
- prefetching/caching/hoarding during low-cost/high-bandwidth coverage

The proposed algorithm in this paper provides a model that is well-applicable for capturing the aspects of user mobility in simulations for the described topics. It is important to keep in mind that our proposed model is not intended to be incorporated into the algorithms that perform and optimize these tasks, since the knowledge (building layout, radio coverage, typical way-points) is usually not assumed to be available for these algorithms. Instead these algorithms use and adapt or learn much simpler models. Our model is typically used for performance evaluation in simulations where it takes the role of the real world model that creates the true behavior which generates the input for these algorithms. A thorough overview of previous mobility models ranging from the simple Brownian motion model via Markovian models that govern user acceleration, speed and direction to location-trace based models can be found in [1].

The paper is organized as follows: The following chapter describes the map mobility model and presents two path finding algorithms. Chapter III reviews speed models which will be used by the simulation tool presented in chapter IV. Finally we demonstrate an application of both algorithms and a software tool before we finish with conclusions and an outlook.

II. DESCRIPTION

The motion of a mobile node is confined to a rectangular area which is spatially discretized into a *layout map matrix* **L** that defines the accessible and inaccessible areas for users with

$$l_{i,j} = \begin{cases} 1 & \text{if} \quad l_{i,j} \quad \text{is} \quad \text{accessible} \\ 0 & \text{if} \quad l_{i,j} \quad \text{is} \quad \text{not accessible} \end{cases} \quad (1)$$

A set $\mathcal{W}$ of N_w *waypoints* $\{(x_1, y_1) \dots (x_{N_w}, y_{N_w})\}$ has to be specified. The user motion will take place on
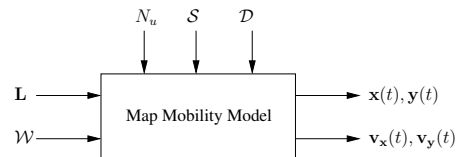

Fig. 1.   Inputs/Outputs

"sensible" paths between these waypoints. The definition and computation of sensible paths will be given in section 2. *Speed models* are used in order to define movement on these paths. A number of speed models are well known, ranging from the simplest constant speed model to more advanced models with deterministic or stochastic acceleration. The tuple $\mathcal{S}$ summarizes the parameters of the chosen speed model. Related to the speed model is the model for the *dwell time*, which is defined as the duration a mobile node spends after reaching a waypoint before starting towards the next waypoint. These dwell times are determined by a dwell time model with parameter tuple $\mathcal{D}$. Currently we lack substantial experimental evidence on how dwell times are distributed in reality. We consider the derivation of good models for the dwell times to be still an open issue. Some speed models generate time intervals with zero velocity. These models can be used as a substitute for a separate dwell time model. Nevertheless we prefer to formulate separate models for dwell times as soon as experimental data becomes available. Depending on the chosen models for speed and dwell times the type and number of elements of $\mathcal{S}$ and $\mathcal{D}$ vary.

Both models have to be chosen carefully to properly represent the dominant real world conditions and statistic properties of the investigated application scenario.

Fig. 2. Path finding

(a) Lee

2	1	2	3	4	5	6
1	W₁	1	2	3	4	5
2	1	2			5	4
3	2			7	6	5
4	5		9	8	7	6
5	6			W₂	8	7
6	7				9	8

(b) Diffusion

0.05	0.1	0.2	0.3	0.4	0.3	0.2
0.1	W₁	0.4	0.5	0.6	0.5	0.3
0.2	0.4	0.5	0.6	0.7	0.6	0.4
0.3	0.5	0.6	0.7	0.8	0.7	0.5
0.4	0.6	0.7	0.8	0.9	0.8	0.6
0.5	0.7	0.8	0.9	W₂	0.9	0.7
0.4	0.5	0.6	0.7	0.9	0.7	0.5

Fig. 3. Contour plot

A. Path finding

In oder to simulate the user behavior we assume a given number of possible waypoints. A path finding algorithm is needed to find possible paths between any two waypoints, avoiding obstacles such as walls or building corners. In the following we review two possible algorithms: The Lee algorithm and the Diffusion Model.

1) Lee Algorithm: The Lee algorithm [2] was developed for automatically wiring printed circuit boards. It determines the shortest rectangular path from a start to an end point. Fig. 2(a) shows a sample trace which avoids an obstacle: From the destination waypoint (W_2) all direct neighbors (left, right, up, down) are incremented by one. This is repeated until waypoint (W_1) is reached. Obstacles are preset with infinity. The path from W_1 to W_2 is found by backtracking from W_2 towards lower values until W_1 is reached. The path may be not unique, as different paths with same distance can be found.

2) Diffusion Model: This model is derived from the gas diffusion in space studied in thermodynamics and used to get applied for path finding of robots [3]: The idea is to have a source continuously effusing gas that disperses in free space and which gets absorbed by walls and other obstacles. We compute for all
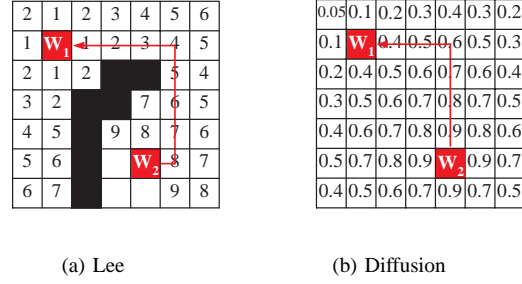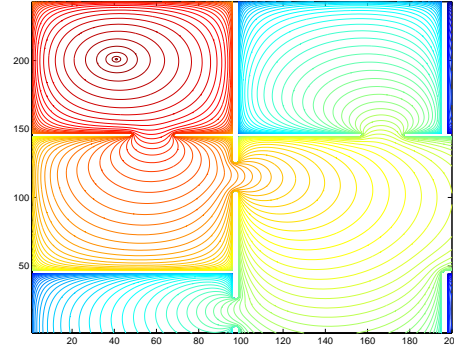
$$W_r(x_r, y_r) \quad \in \quad \mathcal{W}$$

the corresponding diffusion matrix $\mathbf{D_r}$ Therefore, we assume a filter matrix $\mathbf{F}$ of size $n \times n$ with

$$f_{i,j} = 1/n^2 \quad \forall\, i,j: \quad i,j = 0, 1, \ldots, n \qquad (2)$$

The diffusion is expressed by a convolution of the diffusion matrix $\mathbf{D_r}$ with the filter matrix $\mathbf{F}$ element-wise multiplied by the layout map matrix $\mathbf{L}$

$$d_{i,j}(k+1) = l_{i,j} \cdot \sum_{p=1}^{n} \sum_{q=1}^{n} d_{i+p-1, j+q-1}(k) \cdot f_{p,q} \qquad (3)$$

Constantly refreshing the source is represented by forcing

$$d_{x,y} := 1 \qquad (4)$$

Eq. (3) is evaluated repeatedly until

$$d_{i,j} > 0 \quad \forall\, i,j: \quad i = 0, \ldots, N_x, j = 0, \ldots, N_y \qquad (5)$$

Backtracking to waypoint W_1 is done the same way than in the previous example: Start moving at W_2 in direction of falling numbers within the matrix until W_1 is reached (see fig. 2(b)).

Fig. 3 shows a contour plot of the matrix for illustration purposes. The path always follows at right angle to the contour lines.
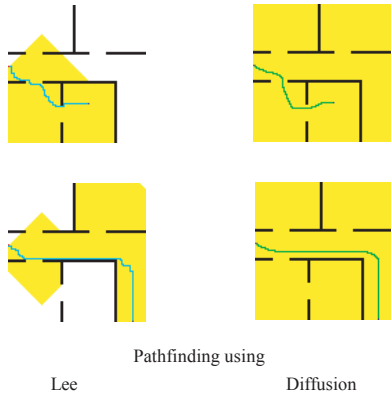
Pathfinding using

Lee   Diffusion

Fig. 4. Comparison of path finding results

3) Comparison: A comparison of two paths calculated both using Lee and Diffusion Algorithm is shown in Fig. 4. It can be seen that the diffusion provides a more realistic path than the one generated by using Lee's algorithm, since the path keeps a distance from obstacles. However, the computational effort of the diffusion algorithm is higher than the other, because the diffusion algorithm requires always the calculation of the entire matrix, whereas using Lee's algorithm partial calculation will let to the same result (denoted by yellow area in the maps). Both algorithms have in common that the computation needs to be performed only once for a given waypoint. Then a path from any other way-point point can be found by following the steepest gradient. The border behavior is the same for both models: We do not consider any wrap-around nor any border reflection.

B. Sample Layouts

The model was tested with the following sample layouts shown in fig. 5.

- *Office:* A typical cubicles-style office layout is given in fig. 5(a)
- *Airport:* Fig. 5(b) shows a layout typically found at airports: Two interconnected airport terminals with gates in niches ("fingers").
- *City:* Fig. 5(c) shows a map from a city center taken from an aerial view

Fig. 6 shows the corresponding results after applying the diffusion algorithm.

III. SPEED MODELS

After the path between two waypoints has been found, a mobility model is needed for simulating user mobility between these points. An overview of mobility models can be found in [1], [4]. Speed models have significant effects on the statistical properties of the simulated motion. For our purpose a scalar value has to be generated at discrete time instants $t = k \cdot T_s, k = 0 \ldots N$ with sampling time

T_s. Models that generate only one constant velocity for all mobile nodes may result in unwanted artifacts in certain statistics of interest, e.g. the finite set $\mathcal{W}$ of N_W waypoints in conjunction with the deterministic path generation algorithm results in $N(N-1)$ distinct paths (if both directions are counted). Fig. 7(b) shows an example how this results in a probability density function $f_{T_c}(T_c)$ for the duration of contact T_c of a given cell with mobile nodes. The discrete peaks might be exploited e.g. by an adaptive handover/hand-off algorithm to achieve better results than possible in the real world, thus rendering these models useless.

In order to avoid these sometimes unwanted effects the velocity v is generated by a random process. Nevertheless it usually is not desirable to obtain totally uncorrelated values of velocity. To achieve a better imitation of real movements the velocity $v(t + T_s)$ is calculated from its previous value $v(t)$ and an acceleration value $a(t)$, which is itself a random variable drawn from a probability density function $f_a(a)$.

$$v(t + T_s) = v(t) + a(t) \cdot T_s \qquad (6)$$

This constitutes a typical random walk process [5]. If eq. (6) is used to compute the velocities it is a necessary condition that the expected value of the acceleration $E\{a(t)\} = 0$ (cond. I), if the velocities are not to rise to infinity. It is also reasonable to limit the acceleration between a_{min} and a_{max}, $a_{min} \leq 0 \frac{m}{s^2} \leq a_{max}$ with regard to the physical properties of the mobile node. If this random process is used the resulting expected value of the velocity also becomes zero, i.e. the mobile node moves back and forth on its path centered about its starting point. The values of the velocities are ensured not to rise to infinity but can still rise to arbitrarily large values with finite probabilities. Though a random process that follows eq. (6) has very benign properties for statistical analysis, we prefer a modified form that incorporates a nonlinear clipping of the velocities to the interval $[v_{min}, v_{max}]$, $0 \frac{m}{s} \leq v_{min} \leq v_{max}$.

$$v(t + T_s) = min(max(v(t) + a(t) \cdot T_s, v_{min}), v_{max}) \quad (7)$$

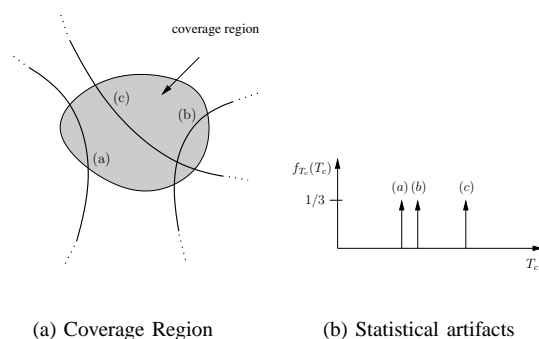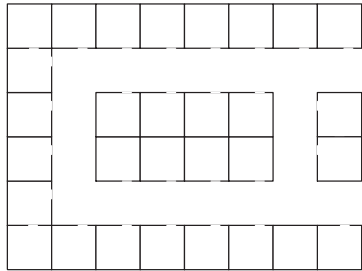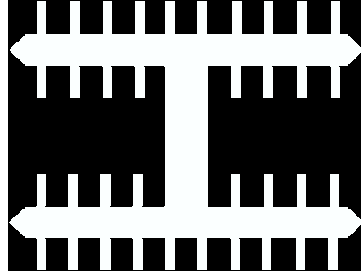(a) Coverage Region   (b) Statistical artifacts
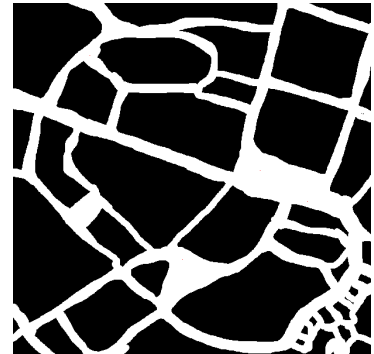
Fig. 7. Repercussions of constant speed and deterministic path generation
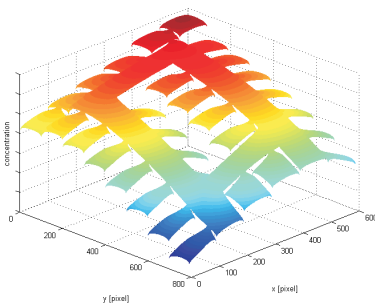
(a) Office   (b) Airport   (c) City
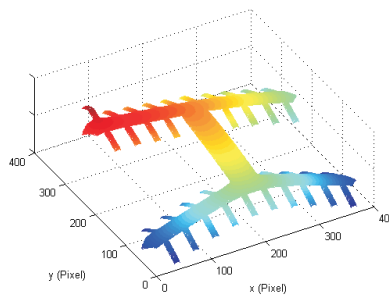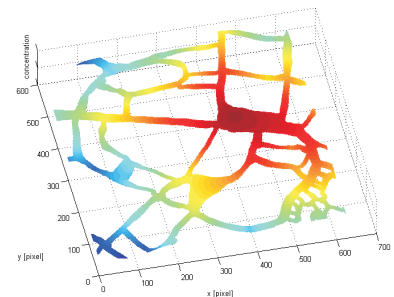
Fig. 5.   Sample Maps

(a) Office   (b) Airport   (c) City

Fig. 6.   Diffusion Maps

We found that choosing the acceleration to be symmetrically and uniformly distributed on the interval $[-a_{max}, a_{max}]$ results in reasonable results for our applications. Though cond. I holds in this case, it is not necessary if eq. (7) is used.

The speed model presented in [1] also works well in conjunction with our path generation algorithm. This model employs one random process for generating speed change events and one random process for choosing the next target speed. It generates a target speed of $0 \frac{m}{s}$ with probability $p(v = 0)$ (e.g. 0.3) for a duration that is drawn from an exponential distribution with a mean duration of e.g. 25 seconds. We use these durations that occur on the path as substitutes for dwell times that are supposed to occur at waypoints.

IV. SIMULATION TOOLS

Based on the algorithms introduced in this paper, a simulation tool was written in JAVA and tested on several platforms. The software reads one layout with corresponding coverage maps for each access point. These files are stored in PPM-Format which can be generated by an image processing software of choice. The following color coding applies: Black denotes blocked areas such as walls and other obstacles, white means free space. Waypoints are specified using red points. For coverage maps, areas in green denote coverage whereas white surfaces mean no coverage by the given access point. The tool both provides a graphical user interface (see fig. 8) and a simple multi threaded TELNET-Interface for remote control. More information can be found in [6].
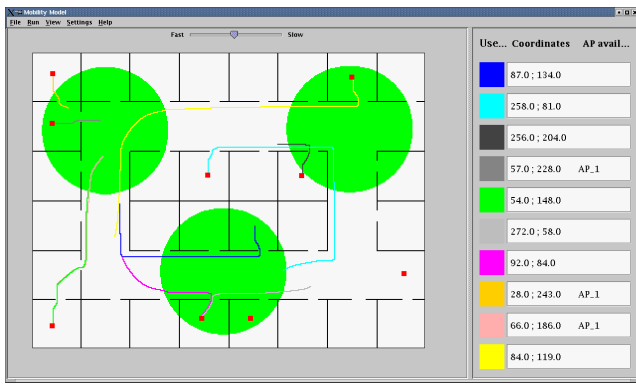
Fig. 8.   Simulation Tool

V. APPLICATION

Fig. 9 shows a sample application for the path based mobility model: Given is a hotspot scenario where mobile users access data from a wireless access point. The mobility model is executed on a server which has connectivity to the mobile users and access point by other means than the short range connection under test (e.g. using a packet based public mobile network). Each mobile user/device subscribes to its mobility information, each access point subscribes to all user mobility information and accepts or denies short range network access according to this information.

This application allows for wireless network analysis without the need of actual movement of the mobile devices.

VI. CONCLUSIONS

The presented mobility model is straightforward to implement. Despite its simplicity the diffusion algorithm generates paths that seem to imitate real world paths of mobile users quite well. The model can be used to investigate a
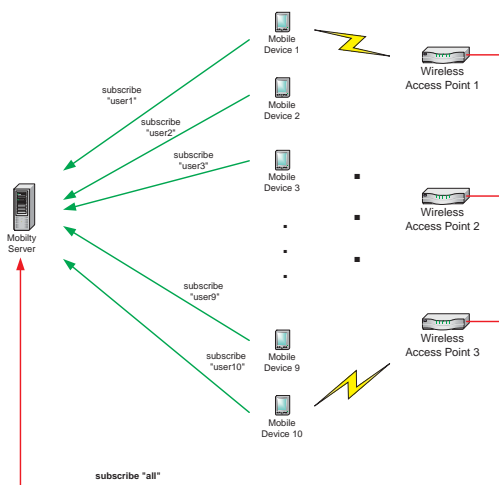
multitude of mobile wireless scenarios by using appropriate layout maps and adjusting the parameters of the speed and dwell time models. The generated position and velocity data has become a valuable input to our own research. Scenarios can be defined with very little effort. Among its main advantages is the ability to visually inspect the plausibility of the generated movements. Nevertheless we still lack verification and/or calibration against real world measured data of user mobility. We regard the availability of measured data as one of the most urgent issues in order to facilitate and justify future research.

Algorithms for handover, call admission control, prefetching etc. show a clear trend towards exploiting higher order statistics of the underlying statistical processes. In order to simulate the performance of these algorithms the models used for simulation have to keep up with this trend and become statistically "richer".

When observing the real world application of wireless services we see a strong correlation between the mobility and the *data* traffic. Generally speaking the higher the instantaneous mobility (velocity) is, the lower the amount of requested content becomes. In order to simulate this effect we consider it advantageous to couple traffic and mobility models for certain scenarios.

The presented work is still in progress and improved versions of the model will be included into the software tool and made available for researchers at [6].

REFERENCES

[1] C. Bettstetter, "Smooth is better than sharp: A random mobility model for simulation of wireless networks," in *Proceedings 4th ACM International Workshop on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM'01)*, (Rome, Italy), pp. 19–27, July 2001.
[2] C. Lee, "An algorithm for path connections and its applications," in *IRE Transactions on Electronic Computing EC-10*, pp. 346–365, 1961.
[3] G. K. Schmidt and K. Azarm, "Mobile robot path planning and execution based on a diffusion equation strategy," in *Advanced Robotics, Vol. 7, No. 5*, pp. 479–490, 1993.
[4] M. M. Zonoozi and P. Dassanayake, "User mobility modeling and characterization of mobility patterns." IEEE Journal on Selected Areas in Communications, 15(7):1239-1252, September 1997.
[5] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. New York: McGraw Hill Book Co., 1984.
[6] "Mobility model simulation tool." http://www.kn-s.dlr.de/kammann /mobility/.

Fig. 9.   Sample Application