Optimised Wireless Channel Characterisation in Large Complex Environments by Hybrid Ray Launching-Collaborative Filtering Approach

Fran Casino, *Student Member, IEEE,* Leyre Azpilicueta, *Member, IEEE*, Peio Lopez-Iturri, *Student Member, IEEE*, Erik Aguirre, Francisco Falcone, *Senior Member, IEEE* and Agusti Solanas, *Senior Member, IEEE*

***Abstract*— Simulation techniques based on deterministic methods such as Ray Tracing and Ray Launching, are widely used to perform radioplanning tasks. However, the quality of the simulations depends on the number of rays and the angular resolution. The computational cost of these simulations in High Definition prevents their use in complex environments and their Low Definition counterparts are used instead. In this article we propose a technique based on collaborative filtering to lessen the poor quality problems of Low Definition simulations. We show that our approach obtains results very similar to those of High Definition in much less time. Also, we compare our approach with other well-known techniques and we show that it performs better in terms of accuracy and precision. The use of combined deterministic/collaborative filtering techniques allows the estimation of radioplanning tasks in large, complex scenarios with a potentially large amount of transceivers.**

***Index Terms*—3D-Ray Launching, Collaborative Filtering.**

I. Introduction

The use of wireless systems has increased in the last decade, given the popularity of mobile networks, wireless LAN and wireless sensor networks. The advent of Context-Aware environments, mainly driven by the trend in Smart City/Smart Region development, will increase further more the deployment of 4G mobile networks, Internet of Things and the overall evolution towards high capacity and capillarity of 5G systems. This will have a decisive impact on many areas from Smart Transportation and sustainability to e-participation and Smart Healthcare [1]. In order to achieve this goal, multiple wireless systems will operate in complex environments, thanks to HetNet architectures.

One of the main considerations is to control interference precisely, in order to increase coverage/capacity ratios. In this sense, given the wide variety of wireless systems under consideration and the inherent complexity of large, dense urban scenarios, radioplanning tasks are compulsory in order to account for useful server signals as well as intra-system and inter-system interference sources. Several techniques can be employed, from semi-empirical regressive methods, which exhibit large errors and measurement dependent models, to deterministic based techniques such as full wave electromagnetic simulation. As a midpoint between precision and computational cost, Ray Launching (RL) methods offer a good trade-off between precision and computational cost. However, when large, complex scenarios in which many potential transceivers can be located, RL exhibits high computational cost and convergence constraints [2,3]. In order to minimise computational cost for certain scenarios, in this article we propose the combination of in-house 3D RL code with Collaborative Filtering (CF), a set of techniques used by Recommender Systems (RS). RS [4] evolve from the field of knowledge discovery in databases 'KDD' [5]. Systems for KDD are used to mine understandable patterns within large collections of data. CF [6] is a kind of recommender system that comprises a large family of recommendation methods. The aim of CF is to make suggestions on a set of items I (e.g. books or films) based on the preferences of a set of users U that have already acquired and/or rated some of those items. Recommendations provided by CF methods are based on the premise that similar users are interested in similar items (i.e. they share similar patterns). Hence, items well rated by user u_a might be recommended to user u_b, if u_a and u_b are similar. In order to predict whether an item would interest a given user, most CF methods rely on matrices M of *n* users (rows) and *p* items (columns), where each matrix cell $m_{i,j}$ stores the rate of user *i* on item *j* [7]. The main idea is to use the ability of CF methods to predict rates and infer the values of empty cells in matrices obtained in Low Definition (LD) simulations, by implementing a knowledge database employing HD simulation results. The proposed methodology has been applied to received power levels, within the complete simulation volume represented in matrix form, although it can be extended to other parameters if required.

II. Hybrid CF and 3D ray Launching

The proposed method, which comprises two steps: (i) database creation and (ii) values prediction, has been validated on scenarios representing rooms with different sizes and a variety of obstacles and materials.

A. Database creation:

A knowledge database, that will be used to predict missing values in LD simulations, is created. Each scenario, modelled by a matrix $M_{n\times p}$, is managed differently depending on the approach used, (i.e., 1D- or 2D-approach):

1) Database creation: 1D-Approach

The matrix $M_{n\times p}$ is serialised into a vector $V = (v_1, v_2, \dots\dots, v_{Lv})$, i.e. a point in $\mathbb{R}^{Lv}$, where $L_v = n \times p$. From this vector we create a set of vectors $SV = \{sv_1, sv_2, \dots\dots, sv_{L_v - L_{sv}+1}\}$. Each vector in SV has length L_{SV} and can be represented as a point in a subspace $\mathbb{R}^{L_{SV}}$. Note that each vector $sv_i = (v_i, v_{i+1}, \dots\dots, v_{i+L_{SV}-1})$, $\forall i \in [1, L_V - L_{SV} + 1]$. Figure 1 shows an example of the creation of the knowledge database consisting of vectors in a subspace $\mathbb{R}^3$, i.e. with $L_{SV} = 3$.

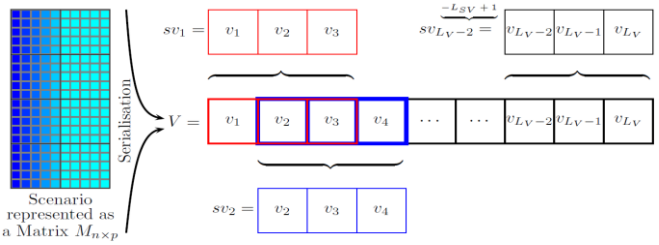

Fig. 1. Database creation example. $L_{SV} = 3$.

Databases might contain information from multiple scenarios. However, note that each knowledge database contains vectors $sv_i \in \mathbb{R}^{L_{SV}}$ only (i.e. vectors from the same subspace). For different subspaces, different databases have to be created. Moreover, it is important to emphasise that databases are created in pairs: DB_{LD} and DB_{HD}. DB_{LD} is the database created with scenarios simulated in *LD*, whilst DB_{HD} is the database created with the same scenarios simulated in *HD*. This way, it is possible to keep a relation between patterns in *LD* and *HD*.

2) Database creation: 2D-Approach

In the two-dimensional approach, the scenario represented by the matrix

$$M_{n\times p} = \begin{pmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,p} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & m_{n,p} \end{pmatrix}$$

is divided into a set SM of sub-matrices $SM = \{sm_{1,1}, sm_{1,2}, \dots, sm_{1,(p-q+1)}, \dots, sm_{(n-q+1),1}, sm_{(n-q+1),2}, \dots, sm_{(n-q+1),(p-q+1)}\}$, where each sub-matrix $sm_{i,j}$ is a squared matrix of size $q \times q$, so that,

$$sm_{i,j} = \begin{pmatrix} m_{i,j} & \cdots & m_{i,(j+q-1)} \\ \vdots & \ddots & \vdots \\ m_{(i+q-1),j} & \cdots & m_{(i+q-1),(j+q-1)} \end{pmatrix}, \forall i \in [1, n-q], \forall j \in [1, p-q]$$

Figure 2 shows an example of the creation of the knowledge database with 3×3 sub-matrices. Note that 2D databases contain less entries than 1D databases. However, such entries will contain more values than one-dimensional vectors. Like in the 1D-approach, knowledge databases are created in pairs.

B. Values prediction:

Once the knowledge database is created, given an *LD* simulation *S* with missing values (i.e. empty cells resulting from low angular resolution) our aim is to predict them so that the resulting values are as similar as possible to those obtained in *HD* simulations.

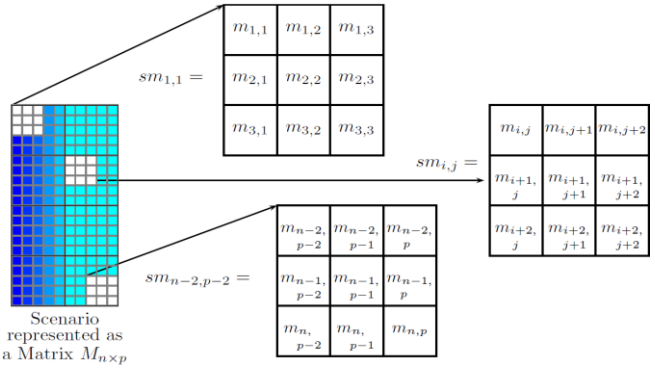

Fig. 2. Database creation example with 3×3 sub-matrices.

1) Values prediction: 1D-Approach

For the one-dimensional approach, we assume that the values in S are normalised (i.e. to be comparable with those in the knowledge database) and that a pair of knowledge databases DB_{LD} and DB_{HD} with *LD* and *HD* patterns in a given subspace $\mathbb{R}^{L_{SV}}$ have been selected. Then, for every vector $v_j \in \mathbb{R}^{L_{SV}}$ from S containing missing values, the k closest patterns $CP^{LD} = \{cp_1, cp_2, \dots, cp_k\}$ in DB_{LD} are found. Next, their corresponding *HD* patterns $CP^{HD} = \{cp_1', cp_2', \dots, cp_k'\}$ in DB_{HD} are determined and its average is computed as $\overline{cp'} = \frac{1}{k}\sum_{i=1}^{k} cp_i'$. The values of the average pattern $\overline{cp'}$ are used as the prediction of the missing values. Without loss of generality, the closest patterns are determined using the Euclidean distance over non-missing values of the vectors. In order to optimise the search process, we group patterns depending on the value of the sum of its elements. Therefore, given a vector sv_i with length L_{SV}, we sum its values and compute distances only with patterns that have a similar sum, according to a threshold, as depicted in Figure 3. In order to increase the quality of the predictions, we only consider vectors having, at most, one missing value. This procedure is iteratively applied until all missing values are determined and, if necessary, knowledge databases with patterns in lower dimensionality subspaces are used.

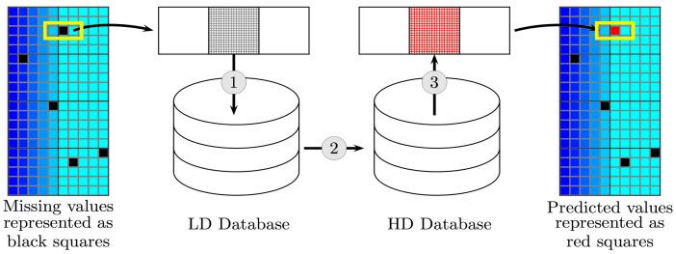

Fig. 3. Missing values prediction: 1) The most similar patterns are found in the LD Database, 2) The corresponding HD patterns are determined, 3) The average of those values is computed and used to replace the missing value.

2) Values prediction: 2D-Approach

For the two-dimensional approach, we assume that the values in *S* are normalised and that a pair of knowledge databases DB_{LD} and DB_{HD} with *LD* and *HD* patterns represented by matrices of size $q \times q$ have been selected. Then, similarly to the 1D-approach, for every sub-matrix $sm_{i,j}$ of size $q \times q$

from S containing missing values, the k closest sub-matrices $CSM^{LD} = \{csm_1, csm_2, \dots, csm_k\}$ in DB_{LD} are found. Next, their corresponding HD sub-matrices $CSM^{HD} = \{csm_1', csm_2', \dots, csm_k'\}$ in DB_{HD} are determined and its average is computed as $\overline{csm'} = \frac{1}{k}\sum_{i=1}^{k} csm_i'$. The values of the average sub-matrix $\overline{csm'}$ are used as the prediction of the missing values. Like in the 1D-approach, the closest sub-matrices are determined using the Euclidean distance over non-missing values and the search process is also optimised. This procedure is similar to the one showed in Figure 3, this time using sub-matrices instead of one-dimensional vectors.

The proposed CF techniques are coupled to a 3D Ray Launching code implemented at UPNA. The algorithm is based on Geometric Optics + Uniform Theory of Diffraction, with rays launched by predefined sources within a solid angle definition, interacting with the simulation scenario as a function of obstacle geometry and dispersive material characterization. The values of angular resolution and number of reflections have been applied following previous convergence analysis studies [2,3].

III. Simulation Examples and Verification of the Algorithm

To assess our solution we have simulated 12 diverse scenarios with 30 to 40 layers, each containing a variety of features (i.e. corridors, columns, walls, doors and furniture). Each scenario has been simulated in LD and HD. With these simulations we have created two sets of databases. First, three 1D LD knowledge databases with vector lengths $L_{SV} = 3, 5,$ and 7 and their three HD counterparts. Second, three two-dimensional LD databases with $q = 2, 3$ and 4 and the corresponding HD ones. Each knowledge database contains more than half a million patterns/vectors in their corresponding subspaces. We have applied our solution on 4 LD simulations (not included in the knowledge database) with diverse dimensions, density (i.e. percentage of occupied space) and sparseness (i.e. percentage of missing values), as depicted in Table I. In each one of the scenarios the transmitter is placed in different locations, in order to obtain generalizable results independent of antenna location. The specific location for each scenario is given in column Row/Column/Layer in Table I.

With the aim to analyse the versatility, accuracy and performance of our approach, we have tested two different types of prediction strategies, shown in Table 2. The first family (i.e. Strategies from 1D.1 to 1D.3) corresponds to the 1D-approach. The second, which includes strategies from 2D.1 to 2D.3, corresponds to the two-dimensional method. For instance, in Strategy 1D.1 we first predict the missing values with the LD knowledge database of vectors with $L_{SV} = 7$. If all missing values are not filled, we next apply the LD knowledge database with $L_{SV} = 5$, and if there remain missing values we apply the LD knowledge database with $L_{SV} = 3$. In Strategy 2D.1, we use sub-matrices with $q = 2$ to compute predictions. Obviously, each family of strategies uses their corresponding 1D or 2D knowledge databases. In all cases we consider an aggregator value (i.e., number of closest patterns that will be used to compute the prediction) k=100. Thus, for each missing value, we find the most similar hundred vectors in the corresponding subspace and compute their average.

TABLE I
Test Simulations Features.

	Rows	Cols	Layers	Source position (R,C,L)	Time (s) HD	Time (s) LD	Density %	Spars. %
Sim-1	130	70	42	95,45,6	110137	2087	2.60	13.93
Sim-2	126	182	38	82,56,35	25360	2981	3.72	43.14
Sim-3	124	273	32	70,212,23	37236	3765	3.06	43.71
Sim-4	58	62	35	18,31,11	81509	1390	3.78	0.77
Sim-5	36	60	38	35,5,30	30637	1112	4.74	1.34
Sim-6	30	30	35	3,20,11	55711	637	4.02	0
Sim-7	30	30	35	10,3,10	36112	712	8.63	0
Sim-8	50	50	35	40,20,13	43628	1088	2.98	0.24
Sim-9	40	30	35	38,15,11	90288	954	3.45	0.02
Sim-10	40	30	35	6,5,16	77479	1008	2.29	0.01
Sim-11	36	60	38	21,50,9	112405	1406	4.74	0.84
Sim-12	32	64	32	10,30,9	47578	918	0.52	0
Sim-A	70	50	30	10,42,21	64139	1040	4.98	2.203
Sim-B	90	60	30	10,42,13	80004	1228	6.10	7.092
Sim-C	175	80	40	46,50,14	52583	1765	6.69	17.051
Sim-D	196	136	38	20,80,5	48910	2037	1.04	37.636

TABLE II
Summary of Prediction Strategies

Strategy	Prediction strategy
1D.1	$L_{SV} = 7 \rightarrow L_{SV} = 5 \rightarrow L_{SV} = 3, (k = 100)$
1D.2	$L_{SV} = 5 \rightarrow L_{SV} = 3, (k = 100)$
1D.3	$L_{SV} = 3, (k = 100)$
2D.1	$q = 2, (k = 100)$
2D.2	$q = 3, (k = 100)$
2D.3	$q = 4, (k = 100)$

We compare the prediction quality of our collaborative filtering prediction strategies with other four well-known methods, namely linear interpolation, average simulation value per layer, average simulation value per row, and average simulation value per column. We compute the error between the real values of the HD simulation and the predicted values determined by the above methods. To compute this error we apply the mean absolute error 'MAE', defined as follows:

$$MAE = \frac{\sum_{i=1}^{n} |p_i - r_i|}{n} \quad (1)$$

where n is the number of missing values predicted, p_i is the predicted value for missing element i, and r_i is the real value of i in the HD simulation. Note that the HD simulation is only used to compute the error but it is not involved in the prediction process. In Table 3 and Table 4 we show the MAE results and times, respectively, of the aforementioned methods for each simulation. Since our method could be applied to each layer independently, the overall procedure could be easily parallelised. Hence, Table 4 shows the worst layer prediction time, which is the actual total cost of our method. As an example, one of the implemented indoor test scenarios, as well as simulation results for the estimation of received power at all locations is depicted in Figure 4. The different results correspond to different simulation techniques, from High Definition, to Low Definition + CF and finally only LD. Our LD+CF solution offers qualitatively an adequate result in terms of received power level estimation.

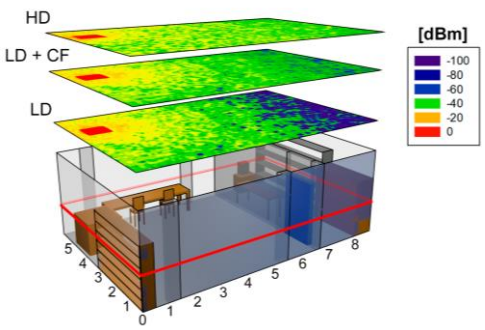

Fig. 4. Received power level estimation when employing HD (top), LD+CF (middle) and LD (bottom). The scenario corresponds to Sim B.

TABLE 3: MAE RESULTS (IN DB).

	Sim-A		Sim-B		Sim-C		Sim-D		All	
Method	MAE	σ	MAE	σ	MAE	σ	MAE	σ	MAE	σ
Strategy 1D.1	6.47	0.75	9.91	0.79	6.99	0.52	10.70	1.40	9.25	1.05
Strategy 1D.2	6.48	0.79	10.14	0.92	7.06	0.60	11.14	1.31	9.53	1.04
Strategy 1D.3	6.21	0.86	8.16	1.02	7.16	0.56	13.38	1.09	10.62	0.90
Strategy 2D.1	5.88	1.58	6.85	0.48	8.42	0.96	15.11	1.06	11.82	1.01
Strategy 2D.2	5.69	1.56	7.01	0.55	7.19	0.56	11.02	0.59	9.19	0.84
Strategy 2D.3	5.82	1.89	7.05	0.59	6.84	0.49	9.78	0.58	8.52	0.79
Linear Interpolation	23.03	4.36	22.88	3.14	24.08	2.83	24.91	2.21	23.98	2.60
Layer mean value	8.84	4.11	5.31	0.46	9.31	1.46	16.37	1.27	12.76	1.42
Column mean value	7.86	3.48	5.11	0.52	9.83	1.62	16.58	1.20	13.02	1.39
Row mean value	7.59	0.93	7.95	0.99	8.65	1.55	14.05	1.28	11.51	1.32
Raw values	56.26	2.09	45.67	1.08	62.07	1.59	73.19	1.15	66.42	1.33

TABLE 4: TIMES (IN SECONDS) TO PREDICT ALL MISSING VALUES

	Time (in seconds)					
Simulation	**1D.1**	**1D.2**	**1D.3**	**2D.1**	**2D.2**	**2D.3**
Sim-A	17	19	17	15	16	16
Sim-B	54	54	35	42	49	56
Sim-C	152	167	179	132	141	143
Sim-D	473	479	531	361	396	416

The results in Table 3 show that our strategies achieve lower/better MAE values than the rest of methods for Sim-A, Sim-C and Sim-D and that Strategy 2D.3 is the overall best method. However, it is worth emphasising that the minimum error is not always obtained by the same strategy. Therefore, if more accurate results were to be obtained, parameters such as *k*, vector subspace dimensionality, sub-matrices size and its corresponding database might be tuned depending on each simulation's features. In opposition to the results of Sim-A, Sim-C and Sim-D, the MAE obtained by mean-based methods in Sim-B is lower/better than that achieved by our proposed strategies. This is due to the fact that Sim-B is very homogeneous and the range of measurements is very short, thus benefiting mean-based methods. The outcomes obtained by 2D strategies (i.e. Strategy 2D.3) outperform those obtained by 1D strategies for Sim-A, Sim-B, Sim-C and Sim-D (cf Table 3). Therefore, we may conclude that 2D strategies capture more accurately wave propagation behavior. Moreover, as observed in Table 4, 2D strategies require less time to perform predictions. This occurs because the optimization performed in the search process has greater impact in 2D databases since they contain less patterns than 1D databases. It is worth noting that database creation time is not included, as it corresponds to a unique process within the simulation process.

This result supports our main claim that collaborative filtering could be used to improve simulations accuracy (even in its simplest approaches). However, it is important to notice that pure mean-based methods do not capture the real neighbourhood conditions of the data and might only be used in very homogeneous scenarios. Also, we observe that sparseness has an adverse effect on the overall prediction accuracy of all methods. This is especially apparent in Sim-D, which is very sparse. This result is not surprising since with less information is more difficult to make good decisions. Overall, as it can be seen in Table 3 (column: "All"), Strategy 2D.3 outperforms the rest of strategies, and obtains the best prediction accuracy, which is about 25% better than the best method not proposed by us.

IV. CONCLUSION

We have proposed the use of collaborative filtering to improve simulations accuracy by predicting missing values. We have shown that our method outperforms other well-known approaches such as linear interpolation and mean-based methods. The main contributions of this article can be summarised as follows: (i) we show that 1D/2D collaborative filtering methods can be applied to improve the accuracy of LD simulations. (ii) We have proposed several prediction strategies that provide better results than classic collaborative filtering methods, (iii) we have shown that our proposal is fast and could help to reduce HD simulation costs.

ACKNOWLEDGMENT

The authors wish to acknowledge the financial support of Project TEC2013-45585-C2-1-R, funded by the Spanish Ministry of Economy and Competiveness.

REFERENCES

[1] A. Solanas, C. Patsakis, M. Conti, I. S. Vlachos, V. Ramos, F. Falcone, O. Postolache, P. A. Pérez-Martínez, R. Di Pietro, D. N. Perrea, A. Martínez-Ballesté, "Smart health: A context-aware health paradigm within smart cities", IEEE Communications Magazine, Vol. 52, N.8, pp. 74-81, 2014.

[2] L. Azpilicueta, M. Rawat, K. Rawat, F. Ghannouchi and F. Falcone, "A Ray Launching-Neural Network Approach for Radio Wave Propagation Analysis in Complex Indoor Environments", IEEE Transactions on Antennas and Propagation, Vol. 62, N. 5, pp. 2777-2785, May 2014.

[3] L. Azpilicueta, M. Rawat, K. Rawat, F. Ghannouchi and F. Falcone, "Convergence Analysis in Deterministic 3D Ray Launching Radio Channel Estimation in Complex Environments", Applied Computational Electromagnetic Society Journal, Vol. 29, No. 4, pp. 256-271, April 2014

[4] P. Resnick and H. Varian, 'Recommender systems,' *Communications of the ACM*, 1997, vol. 40, no. 3, pp. 56–58.

[5] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, 'Application of dimensionality reduction in recommender system-a case study,' *ACM WebKDD 2000 Web Mining for ECommerce Workshop*, 2000

[6] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, 'Using collaborative filtering to weave an information tapestry,' *Communications of the ACM*, 1992, vol. 35, no. 12, pp. 61–70.

[7] Y. Shi, M. Larson, A. Hanjalic, 'Collaborative Filtering beyond the user-item matrix: A survey of the state of the art and future challenges,'*ACM Computing Surveys* (CSUR), 2014, vol. 47, no. 1, pp. 1-45.