

# Avaliação de Algoritmos de Localização Indoor baseados em Mapa de Assinatura de WLANs

Antonio Regilane L. Paiva, F. Rodrigo P. Cavalcanti e Hitalo J.B. Nascimento

**Resumo**— Há um crescente número de trabalhos sobre WIPSS (WLAN-based Indoor Positioning System), sistema de posicionamento local sem infraestrutura dedicada, baseado em redes locais sem fios (WLANs). Algoritmos de localização para esses sistemas são constantemente propostos, requerendo sua implementação para validação e configuração. O presente trabalho apresenta um simulador interativo para análise de algoritmos de localização 2D baseados em Mapas de Assinatura de RSSI (indicador de intensidade do sinal recebido). O simulador foi implementado em Matlab. Por fim, um WIPSS foi implementado em uma residência para a validação do simulador.

**Palavras-Chave**— Sistema de posicionamento local, localização 2D, WLAN, simulador.

**Abstract**— There are a growing number of studies on WIPSS (WLAN-based Indoor Positioning System), that is, local positioning system without dedicated infrastructure based on wireless local area networks (WLANs). Several location algorithms for these systems have been proposed recently, requiring their implementation for validation and configuration. This paper presents an interactive simulator for performance analysis of 2D location algorithms based on signal strength maps. The simulator was implemented in Matlab. A WIPSS has also been implemented in practice for validating the simulator.

**Keywords**— Indoor positioning system, 2D localization, WLAN, simulator.

## I. INTRODUÇÃO

Sistemas de posicionamento local (Indoor Positioning System - IPS) são soluções para localização, rastreamento e pose de pessoas e objetos dentro de ambientes fechados através de informações sensoriais coletadas por dispositivos dedicados ou por dispositivos móveis comerciais, como notebooks, smartphones, tablets e PDAs, e aparelhos de multimídia com conectividade sem fio. Utilizam um sistema de coordenadas relativas não padronizado. São tão eficazes em ambientes fechados como o GPS é em ambientes abertos.

O IPS e o GPS juntos forneceram aos SBL (Serviços Baseados em Localização) cobertura com poucas limitações geográficas. Além de orientar o usuário na sua localização interna e na procura de objetos, a informação do posicionamento local é um recurso fundamental para otimização de processos de alocação de recursos dentro das instalações de uma empresa. Os SBL atuando internamente, podem oferecer serviços como:

- Museus, shoppings e lojas (*m-commerce*): informações, promoções e localização de produtos e serviços para usuários próximos dos mesmos;

Antonio Regilane L. Paiva, F. Rodrigo P. Cavalcanti e Hitalo J.B. Nascimento, Grupo de Pesquisa em Telecomunicações sem Fio (GTEL), Universidade Federal do Ceará (UFC), Fortaleza-CE, Brasil, Emails: lane.p.l@hotmail.com, rodrigo@gtel.ufc.br, hitalo@gtel.ufc.br.

- Gestão Hospitalar: rastreamento de profissionais da saúde, pacientes e equipamentos médicos em situações de emergência;
- Logística: localização e rastreamento de pessoas, ativos, mercadorias, cargas, transportes etc.;
- AAL (Ambient Assistant Living): sistemas que prestam assistência para idosos e pessoas em recuperação em suas casas através de monitoração de sinais vitais, detecção de situações de emergência e acidentes (*fall detect*) [1];
- Robótica: robôs autônomos e atividades cooperativas entre eles.

Os IPSs com infraestrutura dedicada utilizam equipamentos específicos e por isso em geral garatem melhor precisão, mas possuem alto custo devido à compra, instalação e manutenção da infraestrutura dedicada. Os IPSs sem infraestrutura dedicada podem ser baseados em: WLAN [2], VLC [3], Bluetooth [4] e ZigBee [5]. Esses IPSs utilizam as infraestruturas dessas redes e os dispositivos móveis comerciais para fornecerem um serviço de autolocalização para o usuário. Os algoritmos de localização utilizados nesses tipos de IPSs são normalmente baseados na intensidade do sinal recebido (Received Signal Strength - RSS). Exemplos de métodos de localização por RSS são: modelo de predição de distância de propagação por RSS com multilateração e Mapa de Assinatura de RSS (mapeamento de RSS por coordenadas) com reconhecimento de padrões [2]. Trabalhos recentes sobre IPSs são apresentados anualmente na Microsoft Indoor Localization Competition (IPSN). Na IPSN 2014, em um ambiente de 300m<sup>2</sup> dividido em duas salas e equipado com 10 APs, os melhores trabalhos tiveram precisão de 0.72 metros (infraestrutura dedicada) e 1.56 metros (sem infraestrutura dedicada) [6].

Um WIPSS é um IPS baseado em tecnologia de acesso da família de padrões IEEE 802.11, também conhecida por WiFi. Sua ideia central é aproveitar a infraestrutura existente de WLAN a fim de projetar um IPS de baixíssimo custo, atribuindo uma função adicional as WLANs: a função de localização de NICs (adaptador de rede sem fio) presentes em aparelhos com tecnologia WiFi. Nesse tipo de IPS, os APs (pontos de acesso) fazem o papel de *beacons* ativos, emitindo periodicamente quadrados de gerenciamento do tipo Beacon que podem ser captados por um dispositivo móvel, que então faz a medição de RSS para cada Beacon. Pode-se utilizar modelos de perda por propagação e/ou Mapas de Assinatura de RSS (MA) para localização do usuário [2], porém o WIPSS com algoritmo de localização baseado em MA possuem os melhores resultados [2] e tem sido o método mais estudado. Esse tipo de método estima a localização do usuário a partir

do reconhecimento de padrões entre um vetor de RSS medido em tempo real (fase on-line) pelo dispositivo móvel e um MA levantado na fase off-line [2]. Os métodos de classificação normalmente usados são: WKNN [7], método probabilístico [7], RNA [8], [9] e SVM [10]. Esses métodos apresentam uma pequena queda de desempenho quando utilizados para rastreamento (usuário em movimento) [2]. Para melhorar o desempenho do rastreamento é adicionado ao algoritmo um filtro Bayesiano, implementado com um filtro de Kalman [7] ou um filtro de Partículas [11]. Além disso, os dados dos sensores inerciais presentes nos *smartphones* podem ser utilizados para melhorar o desempenho dos WIPSS [12].

Diante de tantas soluções para WIPS, neste trabalho é apresentado um simulador de algoritmos de localização baseado em Mapas de Assinatura de RSS para análise de desempenho desses sistemas. O simulador foi desenvolvido em Matlab com os seguintes recursos:

- Simulador de perfil de RSS baseado na planta baixa do ambiente;
- Interface gráfica;
- Criação de grid e coordenadas;
- Montagem interativa de grafos;
- Criação interativa e automática do Mapa de Assinatura de RSS e da trajetória de teste;
- Algoritmos de localização baseados em Mapas de Assinatura de RSS: WKNN, método probabilístico e RNA.
- Filtros de rastreamento: filtro de Kalman e filtro de Partículas;
- Simulação de Monte Carlo e resultados na forma de tabelas e gráficos estatísticos;
- Monitor configurável de RSSI para plataforma Windows, compatível com Matlab;
- Pedestrian Dead Reckoning (PDR): técnica de rastreamento de pedestres baseada no reconhecimento dos padrões de passos por dados de um acelerômetro [11];
- Filtro de Partículas para fusão entre dados de: grafos, algoritmos de localização e PDR.

Para validar o simulador, foi projetado e implementado um WIPS em uma residência.

## II. SIMULAÇÃO DO PERFIL DE RSS

O perfil de RSS são matrizes de RSS estimadas para cada AP. A estimativa é realizada no simulador a partir do fornecimento de uma imagem da planta baixa do ambiente, que deve apresentar somente as componentes de vedação (paredes) completamente preenchidas com preto e *background* branco.

O usuário define as posições dos APs interativamente e configura os seguintes parâmetros de simulação: relação de escala distância-pixel (quantos centímetros reais corresponde um pixel da imagem), tamanho do elemento do grid, potência de cada AP, expoente de perda de percurso  $n$ , distância de referência e parâmetros de sombreamento.

O perfil de RSS foi estimado pelo o método de traçado de raios, traçados da localização do AP até os limites da imagem, e pelo modelo de predição de RSS de Motley-Keenan [13]:

$$P_R = P_T - PL_0 - 10.n.\log(d/d_0) - \sum_{j=1}^{N_w} k_{w,j}.W_j(\varphi) \quad (1)$$

onde:

- $P_R$ : potência recebida (dBm);
- $P_T$ : potência transmitida (dBm);
- $PL_0$ : perda por propagação a uma distância de referência  $d_0$  (dB);
- $d$ : distância de propagação (m);
- $n$ : expoente de perda de percurso;
- $k_{w,j}$ : número de paredes do tipo  $j$ ;
- $W_j$ : perda de propagação empírica devido às paredes do tipo  $j$  (dB);
- $N_w$ : número de paredes entre o transmissor e o receptor.

O simulador considera que as antenas dos APs são omnidirecionais. Primeiramente, são calculadas todas as distâncias do AP aos pontos da imagem, que são então usadas na equação (1), reduzida aos três primeiros termos, para a obtenção das potências recebidas por ponto. A segunda etapa considera as perdas por atenuações devido às paredes do ambiente. O termo de atenuação por paredes na equação (1) é estimado por pesquisa de pixels de tom preto no raio  $i$  de ângulo  $\theta$  traçado do AP aos limites da imagem, de forma que a atenuação é proporcional ao número de pixels pretos encontrados e a uma constante de atenuação definida pelo usuário. Dessa forma, a atenuação se torna proporcional à espessura da parede e ao ângulo de incidência. Todo processo da segunda etapa é repetido para os demais raios, variando  $\theta$  de 0 a  $2\pi$ . Esses dois processos geram as matrizes de RSS.

Depois que as matrizes de RSS por AP são construídas, os efeitos de sombreamento causados por obstáculos são simulados da seguinte forma:

- Cria-se uma matriz R de zeros com as mesmas dimensões de RSS;
- Gera-se  $n$  valores aleatórios do modelo  $10(\sigma_S.\text{randn}/10)$ , onde *randn* é um gerador de valores pseudoaleatórios de uma distribuição normal e  $\sigma_S$  é a variação de RSS devido ao sombreamento;
- Aloca-se os  $n$  valores gerados em posições aleatórias de R;
- Cria-se uma máscara M de filtro espacial passa-baixa de dimensões configuráveis [14];
- Atualiza-se os valores de RSS pela relação:  $\text{RSS} = \text{RSS} - R * M$ , onde (\*) é o operador de convolução.

Exemplo de perfil de RSS estimado pelo simulador é mostrado na Figura 1.

O tempo médio de execução do simulador de perfil de RSS para 10 APs, determinado com as instruções *tic* e *toc* do Matlab, foi de 11.5 segundos. O processador utilizado foi um Intel Core i5-3330S (4 núcleos, 2.7 GHz). A memória RAM foi de 4.0 GB. A versão do Matlab foi R2009b 64-bits.

## III. SIMULAÇÃO DE ALGORITMOS DE LOCALIZAÇÃO

Para essa simulação são necessárias as matrizes de RSS construídas na simulação do perfil de RSS. Primeiramente, o

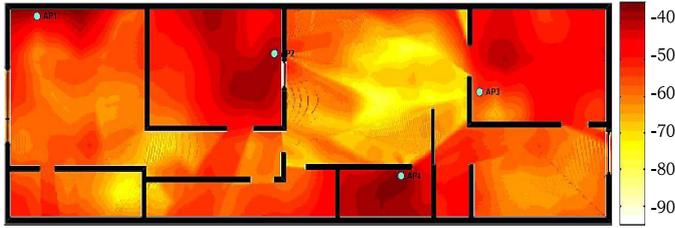


Fig. 1. Perfil de RSS representativo, estimado para quatro APs. Valores em dBm.

usuário define interativamente os pontos de coleta de dados para levantamento do MA e a trajetória do dispositivo móvel (DM) para a fase de teste online, conforme a Figura 2. As amostras em cada ponto de coleta são construídas a partir do valor de RSS correspondente a suas coordenadas, com variações gaussianas proporcionais à atenuação no ponto. O vetor de RSS coletado em um dos pontos da trajetória do DM é construído com os valores de RSS dos APs correspondentes ao ponto, adicionando valores pseudoaleatórios de um modelo log-normal de desvio padrão  $\sigma_R$ , configurado pelo usuário [15]. O parâmetro  $\sigma_R$  representa a variação nos valores de RSS devido aos efeitos de sombreamento. Com posse das matrizes do MA e dos vetores de RSS da trajetória do DM, os algoritmos de localização podem ser simulados. O diagrama de blocos do simulador está ilustrado na Figura 3.

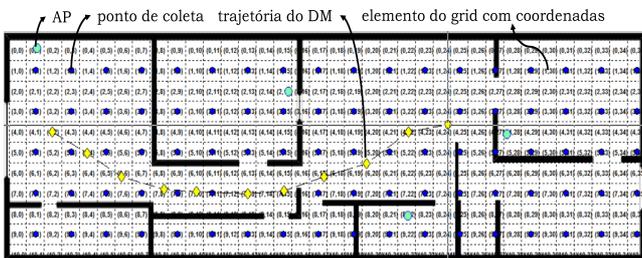


Fig. 2. Posições dos APs, grid, pontos de coleta e trajetória do DM indicados pelo usuário.

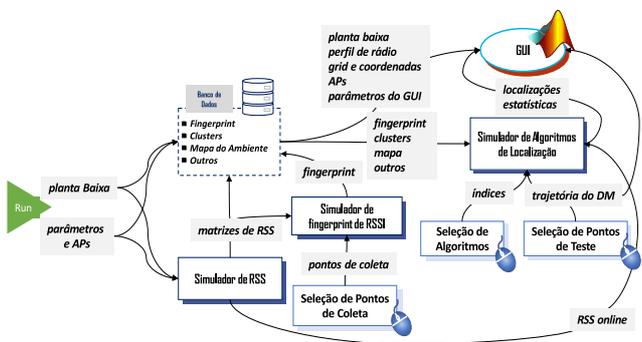


Fig. 3. Diagrama de blocos do simulador.

#### IV. PROTÓTIPO DO WIPS

Para validação do simulador, foi criado um protótipo de WIPS em Matlab, utilizando um notebook Dell com NIC

Dell Wireless DW1702 802.11b/g/n, chipset Atheros com resolução de RSSI de -15 a -95 (dBm). O levantamento do Mapa de Assinatura de RSS foi realizado com um aplicativo desenvolvido em C++ para plataforma Windows, utilizando o API Native WiFi da Microsoft. O aplicativo de aquisição de RSS online foi feito com o mesmo procedimento. Os aplicativos realizam *scanners* passivos no modo monitor a procura dos Beacons com os SSIDs registrados, e então fazem as leituras de RSSI para cada Beacon e os salvam em um arquivo. Os códigos foram compilados com o ADI Visual Studio 2015. Os arquivos dos aplicativos são gerados de forma a facilitar o uso da função *importdata* do Matlab.

O teste foi realizado numa residência de 108  $m^2$  (mesma planta baixa apresentada nas Figuras 1 e 2). Foi adotado um grid de alta resolução, com 396 células de 50cm x 50cm. Foram utilizados 4 APs de fornecedores diferentes e em canais diferentes.

O MA foi levantado em 41 pontos, com 100 amostras por ponto e por AP, sendo que 25 dessas amostras foram coletadas com a antena do notebook direcionada para o norte, 25 para o sul, 25 para o leste e 25 para o oeste. O tempo de *scanner* utilizado foi de 3 segundos por RSSI. As dimensões da matriz do MA foram, portanto:

$$n_{pontos} \times n_{AP} \times n_{amostras}/4 \times 4 = 41 \times 4 \times 25 \times 4.$$

O tempo total de aquisição do MA foi de aproximadamente 15 horas.

Na fase online, o aplicativo desenvolvido faz a leitura de um único vetor de RSSI, demorando entre 2 a 3 segundos. O vetor de RSSI é passado para o Matlab, onde são executados os algoritmos de localização e a interface gráfica. Um filtro de média móvel com tamanho da janela de 5 amostras é utilizado para suavizar quedas bruscas de RSSI [16]. O vetor RSSI suavizado é utilizado no classificador, que determina os  $K$  pontos do MA que possuem o melhor casamento de RSSI com os valores online. A localização é então estimada pelo centroide dos  $K$  pontos. A RNA foi treinada para todo o MA, incluindo 20 amostras por localização, tendo somente uma saída: a localização do MA correspondente ao melhor casamento entre os vetores de RSSI. O erro de rastreamento é minimizado com um filtro de Kalman, que leva em consideração um modelo de movimento linear e incertezas gaussianas. O filtro de Kalman utiliza a localização passada, a localização presente estimada pelo centroide (variável observada) e o modelo de movimento para prever, como menor erro, a localização presente [11].

No teste, foram utilizados três classificadores: WKNN, Probabilístico Exponencial e RNA-RBN. Cada um representa uma classe de algoritmos. As configurações dos algoritmos são mostradas na Tabela I, onde  $K$  é o número de vizinhos mais próximos,  $\sigma_{RSS}^2$  é a variação de RSS no DM e  $\sigma_{acc}^2$  é a variação dos dados do acelerômetro. Detalhes sobre os algoritmos podem ser vistos em [7], [8], [9] e [11].

#### V. RESULTADOS E VALIDAÇÃO

Para validação do simulador, foram realizados experimentos utilizando o protótipo descrito na seção anterior. O experimento foi realizado para o usuário parado e em movimento, com velocidade média de 0.70 m/s. Para o usuário parado,

TABELA I  
CONFIGURAÇÕES DOS ALGORITMOS.

Algoritmos	Parâmetros
WKNN	$K = 4$
Exponencial	$K = 4$ Número de bins(conjunto de dados) = 32
RBN	Camada de Entrada 900 neurônios função: base radial Camada de Saída 90 neurônios função: linear
Filtro de Kalman	$\sigma_{RSS}^2 = 1$ $\sigma_{acc}^2 = 1$ $\Delta t = 0.8$

foram realizadas 100 estimativas de localização no mesmo ponto. O resultado desse experimento está resumido na Tabela II. Com base no erro médio, para o usuário parado, o WKNN mostrou o melhor resultado, com erro médio de 39 cm. O método Probabilístico Exponencial (PE) apresentou erro médio de 54 cm e a RBN apresentou erro médio de 72 cm. Para o usuário em movimento, a localização foi feita por ponto da trajetória. Os resultados desse experimento estão resumidos na Tabela III e na Figura 5. O WKNN e o PE sem filtro de Kalman tiveram praticamente os mesmos resultados, com erros médios de 120 cm e medianas de 140 cm. A RBN obteve erro médio de 200 cm e mediana de 204 cm. Todos os resultados nesse experimento foram inferiores ao experimento com o usuário parado. O WKNN por exemplo, teve erro máximo de 72 cm para o caso parado e erro máximo de 255 cm para o caso em movimento. O Filtro de Kalman (FK) visa suavizar a trajetória do usuário estimada pelo sistema, tornando-a mais coerente com a trajetória real. Dessa forma, a aplicação do FK deve reduzir o erro, principalmente quando o algoritmo tiver um baixo desempenho. Com o FK, o algoritmo PE obteve erro médio 109 cm, uma melhoria de 9.2%, tornando-o um pouco melhor que WKNN, que obteve 6.8% de melhoria. A RBN com FK não liderou os resultados, mas teve uma melhoria de 15% no erro médio, que foi bem superior aos demais. Isso mostra que trajetória gerada pelo algoritmo RBN puro fogia muito de uma trajetória suave. A soberania disputada do PE utilizando FK e a melhoria da RBN podem ser melhor visualizadas através das distribuições cumulativas de probabilidade da Figura 5.

Para alcançar resultados parecidos com os empíricos, o simulador foi executado com a seguinte configuração: planta baixa fiel ao ambiente, relação distância-pixel de 3 cm/pixel, potência de transmissão do AP de 100 mW, expoente de atenuação  $n$  de 2.4,  $PL_0$  de 40.2 dB, atenuação de 1 dB/pixel,  $\sigma_S$  de 3 dB,  $\sigma_R$  de 3 dB, 100 repetições para o usuário parado, pontos de coleta, localização, trajetória reais utilizados no experimento e os parâmetros da Tabela I. Os melhores resultados estão resumidos na Tabela IV, Tabela V e Figura 4. Para o usuário parado, os resultados do simulador foram fiéis aos resultados empíricos, onde o WKNN obteve também o melhor desempenho. Para o usuário em movimento, WKNN e PE apresentaram resultados parecidos, com PE um pouco melhor. O FK trouxe uma melhoria de 7.0%, 3.3% e 6.2% para WKNN, PE e RBN, respectivamente. O ponto importante foi

que, apesar dos valores simulados não serem exatamente iguais devido à natureza aleatória do problema, a margem de erro simulada foi consistente com os resultados empíricos desse e dos demais trabalhos sobre tema [6], [17], que garantem erro médio inferior a 3m. Um outro ponto importante foi que o simulador demonstrou a soberania de um algoritmo sobre o outro, ou seja, se no protótipo o desempenho de um algoritmo foi bem superior ao outro, no simulador também foi.

Com o simulador, foi também possível verificar a influência do número de APs no desempenho do algoritmo. O número de APs define a dimensão do vetor de dados de onde é estimada a informação de localização. Os APs funcionam como pontos de referência, de forma que o aumento no número de APs implica no aumento da discrepância dos dados vinculados às possíveis localizações. Poucos APs, digamos 2, geram pouca informação sobre uma localização, aumentando a probabilidade de haver localizações com os mesmos vetores de dados vinculados, causando, assim, ambiguidade na estimativa da localização. Mais APs geram mais informação, reduzindo então essa probabilidade. Com base nesse ponto de vista, deve haver um ponto onde o aumento do número de APs, consequentemente da informação, seja desnecessário, pois os dados já possuiriam discrepância suficiente para uma mineração de dados eficaz. A Figura 6 apresenta o resultado da simulação da influência do número de APs sobre o desempenho de algoritmos de localização, onde é ilustrado o raciocínio acima, mostrando que o erro médio não é função linear do número de APs e que o aumento do número APs a partir de um certo ponto, pouco influencia no desempenho dos algoritmos. A mesma conclusão foi alcançada experimentalmente em [7].

TABELA II  
ESTATÍSTICAS DOS RESULTADOS DO EXPERIMENTO COM O PROTÓTIPO  
PARA O USUÁRIO PARADO.

	EM <sup>2</sup> (m)	DP(m)	50th(m)	Min(m)	Max(m)	< 2m(%)
WKNN	0.40	0.22	0.51	0.00	0.72	100
Expn	0.55	0.20	0.51	0.00	1.02	100
RBN	0.89	0.35	0.72	0.72	1.61	100

\* EM: erro médio; DP: desvio padrão; 50th: mediana; Min: mínimo; Max: máximo;  
< 2m: proporção de resultados abaixo de 2 m.

TABELA III  
ESTATÍSTICAS DOS RESULTADOS DO EXPERIMENTO COM O PROTÓTIPO  
PARA O USUÁRIO EM MOVIMENTO.

	EM(m)	DP(m)	50th(m)	Min(m)	Max(m)	< 2m(%)
WKNN	1.28	0.51	1.14	0.00	2.55	90
Expn	1.20	0.56	1.14	0.00	2.55	90
RBN	2.00	1.22	2.04	0.00	4.60	46
FK-WKNN	1.20	0.52	1.21	0.00	2.60	98
FK-Expn	1.10	0.54	1.05	0.00	2.62	95
FK-RBN	1.70	1.20	1.60	0.00	4.24	65

## VI. CONCLUSÃO

Nesse trabalho foi descrito, implementado e validado um simulador de algoritmos de localização 2D para WIPS. Os resultados das simulações foram bem plausíveis. O simulador fornece um ambiente de análise para esses tipos de algoritmos

TABELA IV

ESTATÍSTICAS DOS RESULTADOS DO SIMULADOR PARA O USUÁRIO PARADO.

	EM(m)	DP(m)	50th(m)	Min(m)	Max(m)	< 2m(%)
WKNN	0.39	0.30	0.51	0.00	1.61	100
Expn	0.55	0.26	0.51	0.00	1.44	100
RBN	0.73	0.39	0.51	0.51	2.55	99

TABELA V

ESTATÍSTICAS DOS RESULTADOS DO SIMULADOR PARA O USUÁRIO EM MOVIMENTO.

	EM(m)	DP(m)	50th(m)	Min(m)	Max(m)	< 2m(%)
WKNN	1.41	1.10	1.14	0.51	4.61	85
Expn	1.19	1.38	0.72	0.00	5.63	80
RBN	2.37	2.88	1.48	0.00	9.81	60
FK-WKNN	1.31	1.14	0.93	0.00	4.62	85
FK-Expn	1.15	1.37	0.72	0.00	5.63	80
FK-RBN	2.23	2.84	1.33	0.00	9.30	65

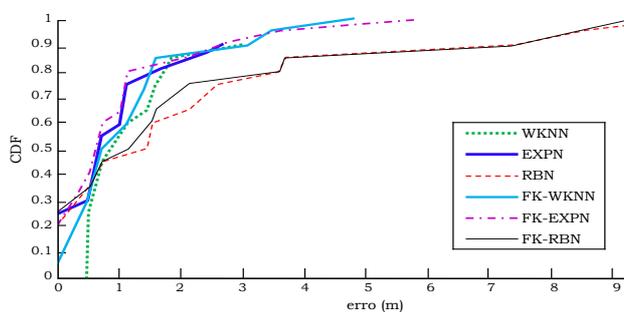


Fig. 4. CDFs (Cumulative Distribution Functions) dos resultados do simulador para o usuário em movimento.

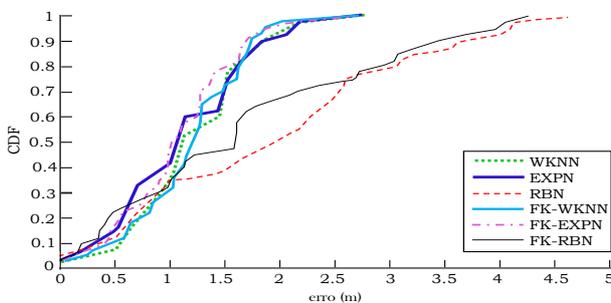


Fig. 5. CDFs (Cumulative Distribution Functions) dos resultados do experimento com o protótipo para o usuário em movimento.

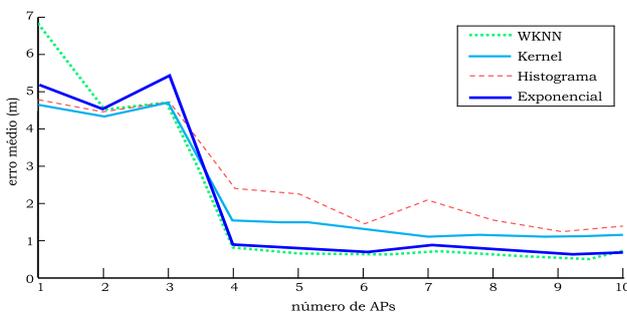


Fig. 6. Simulação da influência do número de APs no desempenho de algoritmos de localização. Simulação baseada no experimento de Honkavirta [7].

que facilita o planejamento, o entendimento, a configuração e os projetos de WIPS. A flexibilidade de configuração de diferentes cenários de teste no simulador permite testar os limites dos algoritmos de localização sem a complexidade que seria necessária em ambientes reais. Com simulador, é possível fazer análise da precisão e exatidão de WIPSS para diversos ambientes e canais de RF. As estatísticas geradas pelo simulador não serão exatamente as mesmas de um WIPS real, porém, as discrepâncias entre as estatísticas de um algoritmo e entre algoritmos foram consistentes nos resultados das simulações.

## REFERÊNCIAS

- [1] P. Barsocchi, F. Potorti, and P. Nepa, "Device-free indoor localization for aal applications," in *Wireless Mobile Communication and Healthcare*. Springer, 2012, pp. 361–368.
- [2] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. Ieee, 2000, pp. 775–784.
- [3] W. Chunyue, W. Lang, C. Xuefen, L. Shuangxing, S. Wenxiao, and D. Jing, "The research of indoor positioning based on visible light communication," *Communications, China*, vol. 12, no. 8, pp. 85–92, 2015.
- [4] L. Aalto, N. Göthlin, J. Korhonen, and T. Ojala, "Bluetooth and wap push based location-aware mobile advertising system," in *Proceedings of the 2nd international conference on Mobile systems, applications, and services*. ACM, 2004, pp. 49–58.
- [5] M. Sugano, T. Kawazoe, Y. Ohta, and M. Murata, "Indoor localization system using rssi measurement of wireless sensor network based on zigbee standard," *Target*, vol. 538, p. 050, 2006.
- [6] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, and S. Sen, "A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned," in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*. ACM, 2015, pp. 178–189.
- [7] V. Honkavirta, T. Perälä, S. Ali-Löytty, and R. Piché, "A comparative survey of wlan location fingerprinting methods," in *Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on*. IEEE, 2009, pp. 243–251.
- [8] L. Gogolak, S. Pletl, and D. Kukolj, "Neural network-based indoor localization in wsn environments," *Acta Polytechnica Hungarica*, vol. 10, no. 6, pp. 221–235, 2013.
- [9] C. Laoudias, P. Kemppi, and C. G. Panayiotou, "Localization using radial basis function networks and signal strength fingerprints in wlan," in *Global telecommunications conference, 2009. GLOBECOM 2009. IEEE*. IEEE, 2009, pp. 1–6.
- [10] M. Brunato and R. Battiti, "Statistical learning theory for location fingerprinting in wireless lans," *Computer Networks*, vol. 47, no. 6, pp. 825–845, 2005.
- [11] H. Nurminen, A. Ristimäki, S. Ali-Löytty, and R. Piché, "Particle filter and smoother for indoor localization," in *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*. IEEE, 2013, pp. 1–10.
- [12] H. Wang, H. Lenz, A. Szabo, J. Bamberger, and U. D. Hanebeck, "Wlan-based pedestrian tracking using particle filters and low-cost mems sensors," in *Positioning, Navigation and Communication, 2007. WPNC'07. 4th Workshop on*. IEEE, 2007, pp. 1–7.
- [13] A. Motley and J. Keenan, "Personal communication radio coverage in buildings at 900 mhz and 1700 mhz," *Electronics Letters*, vol. 24, no. 12, pp. 763–764, 1988.
- [14] R. GONZALEZ and R. WOODS, "Processamento de imagens digitais, 2000," Editora: Edgard Blücher LTDA.
- [15] R. G. Akl, D. Tummala, and X. Li, "Indoor propagation modeling at 2.4 ghz for ieee 802.11 networks," 2006.
- [16] G. Ren, Z. Yongxiang, and T. Long, "The research on rss filter for indoor positioning system based on wireless lans," *Advances in Information Sciences and Service Sciences*, vol. 3, no. 4, 2011.
- [17] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 6, pp. 1067–1080, 2007.