Universidade Federal de Pernambuco

Centro de Informática

Graduação de Ciência da Computação

Um Estudo Sistemático Sobre Técnicas de *Dead Reckoning* para Localização *Indoor*

Vinícius de Moraes Rêgo Cousseau

Trabalho de Graduação

Recife

Dezembro de 2016

Universidade Federal de Pernambuco

Centro de Informática

Vinícius de Moraes Rêgo Cousseau

Um Estudo Sistemático Sobre Técnicas de *Dead Reckoning* para Localização *Indoor*

Trabalho apresentado ao Programa de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Silvio de Barros Melo

Recife

Dezembro de 2016

Agradecimentos

Agradeço a todo o corpo de funcionários que compõe o Centro de Informática da Universidade Federal de Pernambuco. Me preparo para receber o grau de Bacharel e despedir-me do centro, ainda que parcialmente, com a certeza de que sem o ambiente seguro e agradável proporcionado pelas pessoas envolvidas, talvez me faltasse estímulo a dar continuidade aos meus estudos. Sou grato por poder ter estudado em um centro de excelência, e espero que futuros alunos possam encontra-lo da mesma maneira.

A todos os professores envolvidos na minha formação acadêmica e pessoal. Em especial, ao professor Dr. Pedro Machado Manhães de Castro, por ter me estimulado a seguir interessado em certas áreas da computação e ter me ajudado a entender um pouco mais o cenário acadêmico destas, à professora Dra. Judith Kelner, por ter me guiado durante quase dois anos de iniciação científica e pesquisa, e ao professor Dr. Silvio de Barros Melo, que aceitou me orientar no desenvolvimento deste trabalho e se mostrou disposto a tirar minhas dúvidas quando necessário.

A todos que compõe a In Loco Media, empresa que me fez conhecer e ter interesse na área de localização *indoor* e em todo o potencial que ela apresenta. Agradeço em especial à equipe de localização pela ótima convivência diária e pela motivação gerada, e a Pedro Augusto Bello, cujas valiosas orientações e conselhos guiaram algumas das minhas decisões, somando-se a uma ajuda indispensável durante o desenvolvimento deste projeto.

Ao grupo de amigos da minha turma da faculdade, que me acompanharam desde o início dessa jornada e a tornaram excepcional, compartilhando momentos de dificuldade, desespero e, principalmente, alegria. Em especial, agradeço a Duhan Caraciolo, Guilherme Peixoto, Larissa Passos, Lucas Lima e Mateus Moury, amigos com os quais tive o prazer de realizar a maior parte dos meus projetos durante o curso. Agradeço novamente a Larissa Passos e a Mateus Moury pelas incontáveis caronas a mim oferecidas. Sem elas, não tenho dúvidas de que meu desempenho em diversas matérias seria muito reduzido.

A todos os amigos que vim a conhecer no Colégio Equipe, durante meu intercâmbio pelo programa Ciência sem Fronteiras, na universidade, ou pelos caminhos da vida. É uma honra poder estar cercado de pessoas tão especiais e poder contar com a companhia das mesmas no dia a dia, fisicamente e virtualmente, e em cada nova aventura, seja ela uma nova fase pessoal na minha vida ou simplesmente um novo estágio em um jogo de tabuleiro.

A toda a minha família e especialmente à minha mãe, Grasiela, e ao meu pai, Jairo, pelo esforço realizado por eles para proporcionar as condições necessárias para que eu chegasse a esse ponto. Ainda à minha mãe, Grasiela, que é a pessoa mais resiliente e perseverante que já conheci. Ela, de forma quase heroica, cuidou e cuida de mim durante os momentos mais difíceis da minha vida, me aconselhando e me resguardando, muitas vezes colocando seu próprio bem-estar em risco, e este é o agradecimento mais importante que tenho a fazer.

Resumo

A extração de dados para auxiliar diversos tipos de aplicações e negócios aliada ao uso em massa de *smartphones* que, carregados de sensores, fornecem uma rica coleção de dados sobre seus usuários, tem gerado crescente interesse nos últimos anos. Dentre os possíveis dados a se extrair, destaca-se a localização do usuário. Esta é tradicionalmente obtida a partir do GPS, entretanto, este sistema sofre uma grande queda de precisão em ambientes *indoor*. Nesse contexto, é comum confiar nas medições dos sensores inerciais dos *smartphones* para inferir a posição do usuário, e a este processo de navegação inercial dá-se o nome de *dead reckoning*. Este trabalho tem como objetivo a realização de um estudo sistemático sobre técnicas de *dead reckoning* para localização *indoor*. Será discutido o processo de localização em interiores de acordo com uma perspectiva histórica do assunto. Em seguida, serão explorados os conceitos gerais de *dead reckoning* em tal contexto, problemáticas e técnicas desenvolvidas. Em particular, será mostrado como os dados oriundos dos sensores de *smartphone* podem ser transformados em informação relevante. Por fim, o trabalho apresenta a implementação de um módulo de confiabilidade para *dead reckoning*.

Palavras-chave: *dead reckoning*, localização *indoor*, processamento de sinais, sensores

Abstract

Extracting data to support several types of applications and businesses alongside the mass adoption of smartphones which, being loaded with sensors, provide a rich collection of data about their users, has been a growing trend during the last few years. Among the possible pieces of data to extract, the users' location stands out. This information is traditionally obtained through GPS, however, the system suffers a great precision loss in indoor environments. In this context, it is common to trust in the smartphone's inertial sensor readings to infer its user's position, in an inertial navigation process named dead reckoning. This dissertation intends to perform a systematic study about dead reckoning techniques for indoor localization. The work discusses the localization process in interiors according to a historical perspective of the subject. Subsequently, it explores general concepts about dead reckoning in the aforementioned context, as well as problematics and previously developed techniques. In particular, the dissertation shows how data gathered from the sensors of a smartphone can be transformed into relevant information. Finally, this work presents the implementation of a reliability module for dead reckoning.

Keywords: dead reckoning, indoor localization, signal processing, sensors

Sumário

1. Introdução

Nos últimos anos, a facilidade de acesso a novas tecnologias de baixo custo e a ubiquidade da computação causaram um grande aumento na quantidade de dados produzidos pelas pessoas. Com esse aumento, criou-se a necessidade de estudar formas de extrair e interpretar esses dados e, adicionalmente, muitas oportunidades para aqueles dispostos a utiliza-los. Um dos principais fatores que contribuiu para a geração de novos dados foi a popularização de *smartphones*. Considerando apenas o ano 2014, foram vendidos mais de 1,2 bilhão de *smartphones* no mundo (FRIEDMAN, 2015).

Por serem carregados de sensores e utilizados diariamente em proximidade a seus usuários, os *smartphones* fornecem uma rica coleção de dados. Dentre os possíveis dados a se extrair dos dispositivos, destaca-se a localização do usuário. A partir desta é possível tornar aplicações computacionais sensíveis ao contexto no qual o usuário está inserido, sendo localização um dos dois principais problemas a serem superados pela computação ubíqua (WEISER, 1999), juntamente com escalabilidade.

Diversas aplicações sensíveis ao contexto tiveram sua criação possibilitada pela obtenção da localização do usuário. Existem aplicações, por exemplo, que controlam elevadores de acordo com a localização de pessoas nas proximidades, otimizando o algoritmo de movimentação deste, e outras aplicações que oferecem publicidade direcionada ao usuário de acordo com a localização dele (IN LOCO MEDIA, 2016). Algumas das técnicas, inclusive, alcançam precisões a nível de prateleiras em um supermercado (LYMBEROPOULOS et al., 2015).

Localizar uma pessoa por meio de dispositivos digitais não é um desafio recente: Há cerca de 40 anos foi lançado o *Global Positioning System* (GPS), sistema que localiza usuários de receptores em qualquer lugar na terra que possua contato com seus satélites. O GPS é mantido pelo governo dos Estados Unidos (NRC, 2014) e possui garantias de precisão de acordo com o ambiente onde o usuário do receptor se encontra.

A informação do GPS, porém, sofre uma grande perda de acurácia quando o usuário está localizado em ambientes fechados (*indoor*), não possuindo bom contato com os satélites GPS. Em contrapartida, para ser possível localizar usuários em ambientes *indoor* é necessária uma boa precisão e alta granularidade nas técnicas, vez que tais ambientes costumam ser menores e mais densos que espaços abertos.

Somando-se os fatos supracitados a estudos que revelam uma tendência crescente dos seres humanos gastarem mais tempo diariamente em ambientes fechados, sendo 87% em 2001 segundo Klepeis et al. (2001), é possível concluir que a localização precisa em espaços *indoor* é crucial. Consequentemente, tornou-se propício o desenvolvimento da área de localização de usuários em ambientes *indoor* como um campo de estudo a parte.

Estudos iniciais no campo de localização *indoor* investigaram o uso de dispositivos e modificações extras para obter a posição do usuário (BAHL; PADMANABHAN, 2000). Essas técnicas precursoras para a área possuíam diversas limitações e não ofereciam uma precisão muito satisfatória quando comparada com a oferecida pelo GPS em ambientes abertos.

Adicionalmente, as soluções propostas não atuavam de forma generalizada para qualquer local e sem informação prévia.

Pesquisas na área renovaram-se com as novas possibilidades geradas pelos *smartphones*, visto que os dados oriundos dos diversos sensores dos aparelhos poderiam ser usados em detrimento de dispositivos adicionais ou extensas adaptações físicas em dispositivos existentes. Além disso, criar soluções mais generalizadas se tornou uma tarefa muito mais tangível devido à facilidade de acesso.

Nas técnicas utilizadas nesse contexto é comum confiar nas medições de alguns sensores específicos dos *smartphones*, tais como giroscópio e acelerômetro, para inferir a velocidade e orientação do usuário a partir de uma posição inicial e, com o decorrer do tempo, localiza-lo. A esse processo dá-se o nome de *deduced reckoning*, comumente abreviado para *dead reckoning*. Combinado com outras técnicas, o *dead reckoning* é usado nos mais avançados sistemas de localização *indoor* da atualidade, como mostram os estudos realizados por Lymberopoulos et al. (2015).

Pesquisas sobre o uso de *dead reckoning* para a localização em tempo real de usuários em ambientes urbanos foram conduzidas independentemente de pesquisas gerais sobre localização *indoor* (LADETTO, 2002). Diversos sensores inerciais eram acoplados a algum equipamento, e os dados oferecidos por estes eram então utilizados para localizar o usuário. Grandes vantagens apresentadas por esses métodos eram as possibilidades de se utilizarem sensores de maior qualidade e posicioná-los em partes ótimas do corpo, resultando em uma maior precisão e confiabilidade.

O uso de dispositivos extras, porém, torna pouco viável a aplicação em massa da técnica, vez que seria necessário um investimento adicional por parte dos usuários. Além disso, alguns equipamentos não são práticos o suficiente para o uso diário. Essa praticidade, em contrapartida, é alcançada pelos *smartphones*.

Apesar da praticidade, há uma dependência inerente na precisão dos sensores dos *smartphones* que causa problemas para o *dead reckoning*, visto que cada tipo de sensor pode apresentar uma forma diferente de erro de medição, erro cumulativo ou de viés inicial. Esses problemas são amplificados quando se leva em conta que os sensores em *smartphones* são geralmente de menor qualidade.

Extrair e processar os dados necessários para *dead reckoning* em um smartphone, portanto, não é uma tarefa trivial. Torna-se relevante, consequentemente, um estudo sobre diferentes abordagens para tal processo, como elas se encaixam no contexto de localização *indoor* e em que pontos elas podem ser melhoradas.

1.1 Objetivos

O principal objetivo deste trabalho é realizar um estudo sistemático sobre técnicas de *dead reckoning* no contexto de localização em ambientes *indoor*. Especificamente, objetiva-se explicar o processo básico de *dead reckoning* por uma visão matemática e computacional, discorrer sobre diferentes abordagens para realiza-lo, comparar tais abordagens, mostrar as problemáticas envolvidas, e explicitar como o processo se encaixa dentro de um sistema completo para localização *indoor*.

Adicionalmente, é realizada uma análise de técnicas de localização *indoor* sob uma perspectiva histórica, com o intuito de contextualizar o trabalho. Pretende-se também prover uma implementação de um módulo de confiabilidade para *dead reckoning*, com o intuito de providenciar uma maneira simples e eficiente para qualquer sistema de localização *indoor* obter tal informação.

1.2 Estrutura do Trabalho

O trabalho é composto por 5 capítulos principais, incluindo este capítulo introdutório. O Capítulo 2 discorrerá sobre conceitos básicos de processamento de sinais e sensores necessários para o entendimento das técnicas a serem descritas. Apenas algumas técnicas e alguns sensores são abordados, de acordo com a relevância destes para a área e para o trabalho em si.

No capítulo seguinte é realizada uma análise em perspectiva histórica de técnicas de localização *indoor*, das técnicas seminais até o estado da arte. O intuito do capítulo é prover conhecimento sobre esse campo no qual as técnicas de *dead reckoning*, foco do trabalho, são utilizadas e, mais especificamente, como elas colaboram com o processo.

O Capítulo 4 é o principal capítulo deste trabalho, sendo equivalente a um capítulo de desenvolvimento. Nele, são mostradas abordagens para o processo de *dead reckoning*, suas vantagens e desvantagens, problemas inerentes, e usos. Nele também é fornecida uma implementação do módulo de confiabilidade para *dead reckoning*.

No Capítulo 5, por fim, são expostas as conclusões do trabalho. Um breve resumo do trabalho realizado e das contribuições dele é fornecido. Além disso, são mostrados os desafios encontrados, considerações sobre o módulo implementado e sugestões de trabalhos futuros para pesquisadores interessados.

2. Conceitos Fundamentais

As técnicas de localização *indoor* e de *dead reckoning* utilizam diversos conceitos de álgebra linear, processamento de sinais e sensores em *smartphones*. Portanto, antes de explorar tais técnicas e suas aplicações, é necessário um entendimento sobre os conceitos fundamentais envolvidos nelas. Esta seção procura explicitar os sensores comumente presentes nos *smartphones* que são relevantes para o contexto de localização, suas características e utilidades, e clarificar conceitos de processamento de sinais, especialmente filtros. Conceitos básicos de álgebra linear são omitidos desta seção.

2.1 Processamento de Sinais

No âmbito da localização em ambientes fechados, os sensores contidos nos *smartphones* recebem valores contínuos oriundos de ações do usuário e/ou mudanças no ambiente. Para ser possível operar sobre esses valores em código, é necessário que estes sejam amostrados a uma certa taxa.

O conceito de amostragem refere-se à conversão de um sinal contínuo para um sinal discreto, ou seja, a coleta de sinais percebidos pelos sensores e a transformação destes em valores discretos sobre os quais é possível operar em um código, pelos mesmos sensores. A taxa de amostragem, portanto, refere-se ao número médio de amostras obtidas por segundo por um sensor, representada em *Hertz* (Hz). O período de amostragem, medido em segundos, é inverso à taxa de amostragem.

O processo de amostragem de um sinal é sujeito a diversas formas de distorção, tais como *Jitter* e ruído. O conceito de ruído em processamento de sinais está relacionado a flutuações aleatórias em um sinal elétrico, tais como flutuações causadas pela agitação térmica das cargas em um condutor elétrico ou causadas por uma corrente direta (VAN EXTER, 2003).

Técnicas de *dead reckoning* requerem sinais tão livres de distorções quanto possível, portanto, aplicam filtros nos sinais recebidos dos sensores dos *smartphones*. Dentre os filtros comumente utilizados nesse contexto, destacam-se o filtro passa-baixa, o filtro passa-alta e o filtro complementar. Esse trabalho discorre sobre cada um desses três filtros nas subseções seguintes.

É importante adicionar que as interpretações dos filtros dadas nas subseções seguintes são de um ponto de vista discreto e, mais especificamente, computacional. Os filtros podem ser mais formalmente definidos a partir de suas fórmulas no domínio de frequência. Os requisitos das aplicações de localização *indoor*, porém, impossibilitam o uso de fórmulas fechadas para os filtros, optando-se por aproximações.

2.1.1 Filtro Passa-Baixa

O filtro passa-baixa aceita sinais com uma frequência menor que um certo limiar, enquanto elimina sinais de mais alta frequência. O limiar de frequência pode ser definido para cada aplicação deste. Em código, um filtro passa-baixa pode ser escrito como uma interpolação linear

entre um valor atual e o novo sinal recebido, sendo a constante de interpolação definida pelo desenvolvedor da aplicação. Um exemplo simples de filtro passa-baixa em Java para Android, aplicado ao acelerômetro, pode ser visto no Código 1.

Código 1 - Filtro passa-baixa com fator de inércia igual a 0.96

```
1. public static final float ALPHA = 0.96f;
2. public float[] gravity = { 0.0f, 0.0f, 0.0f };
3.
4. public void lowPassFilter(SensorEvent event) {
5.   gravity[0] = ALPHA * gravity[0] + (1 - ALPHA) * event.values[0];
6.   gravity[1] = ALPHA * gravity[1] + (1 - ALPHA) * event.values[1];
7.   gravity[2] = ALPHA * gravity[2] + (1 - ALPHA) * event.values[2];
8. }
```

Para calcular o limiar de corte de um filtro passa-baixa, utiliza-se o conceito da constante de tempo do filtro. A constante de tempo de um filtro pode ser entendida como a duração relativa de um sinal sobre o qual o filtro vai operar. Dado o período de amostragem (dT) e a constante de tempo (τ) do sensor, o coeficiente de interpolação do filtro pode ser encontrado pela Equação 1.

$$\alpha = \frac{\tau}{\tau + \mathrm{dT}} \tag{1}$$

O coeficiente de interpolação pode ser interpretado como a inércia do sistema. O valor no Código 1 foi aproximado assumindo-se uma constante de tempo de 0.5s e um período de amostragem de 0.02s, equivalente a uma taxa de amostragem de 50 Hz. A Tabela 1 mostra os resultados de sucessivas chamadas à função do Código 1 em termos do valor aproximado de cada eixo do vetor de gravidade armazenado, dado que em cinco eventos consecutivos foi medido um vetor $\vec{v} = (2, 5, 10)$.

Tabela 1 - Resultados da execução da função *lowPassFilter* por 5 iterações, aproximados para duas casas decimais.

#Evento	0	1	2	3	4	5
X	0.0	0.08	0.15	0.23	0.30	0.37
Y	0.0	0.20	0.39	0.58	0.75	0.92
Z	0.0	0.40	0.78	1.15	1.50	1.84

Fonte - Elaborada pelo autor.

Analisando os resultados da Tabela 1, é possível notar que o vetor armazenado leva algumas iterações para convergir ao valor recebido em cada medição. Por causa disso, mudanças bruscas e singulares seriam atenuadas por subsequentes mudanças. Quanto mais inércia o filtro possui, mais lentamente o sistema responde a mudanças nos valores de entrada. Esse comportamento é desejado para se lidar com ruídos.

2.1.2 Filtro Passa-Alta

O filtro passa-alta atua de forma oposta ao filtro passa-baixa, isto é, apenas sinais de alta frequência são aceitos enquanto os sinais de baixa frequência são filtrados. O fator α para um filtro passa-alta pode ser calculado de forma similar à mostrada na Equação 1, porém este representa aqui tanto a taxa de contribuição de mudanças na entrada como a taxa de decaimento dos valores processados.

O decaimento em filtros passa-alta refere-se ao fato de que os valores armazenados por eles tendem a zero caso a entrada permaneça constante, independentemente do valor armazenado. O valor de α governa o quão rápido acontece esse decaimento, sendo valores altos representantes de baixas taxas de decaimento, e vice-versa. Como o valor de α governa as duas características do filtro, diferentemente do passa-baixa, geralmente é mais complexo encontrar um valor único que forneça um bom *trade-off* entre elas.

Uma aproximação do filtro passa-alta pode ser derivada considerando-se que o resultado esperado é obtido a partir do sinal original subtraído do sinal processado por um filtro passa-baixa. Essa derivação é apresentada no Apêndice A, ao fim deste trabalho. O Código 2 expressa uma implementação de um filtro passa-alta seguindo a fórmula apresentada.

Código 2 - Filtro passa-alta simples

```
1. public static final float ALPHA = 0.96f;
2. private static double lastInput = 0.0;
3. private static double output = 0.0;
4. private static boolean initialized = false;
5.
6. public double highpassFilter(double input) {
7.   if (!initialized) {
8.     output = input;
9.     lastInput = input;
10.    initialized = true;
11.    return output;
12.  }
13.  output = ALPHA * (output + input – lastInput);
14.  lastInput = input;
15.  return output;
16. }
```

Dentre suas possíveis aplicações, o filtro passa-alta destaca-se como uma ferramenta para a obtenção de mudanças drásticas em um curto tempo percebidas por um sensor. Em técnicas de localização *indoor*, um exemplo de tais mudanças drásticas seria um giro abrupto do aparelho detectado pelo giroscópio (WANG et al., 2012).

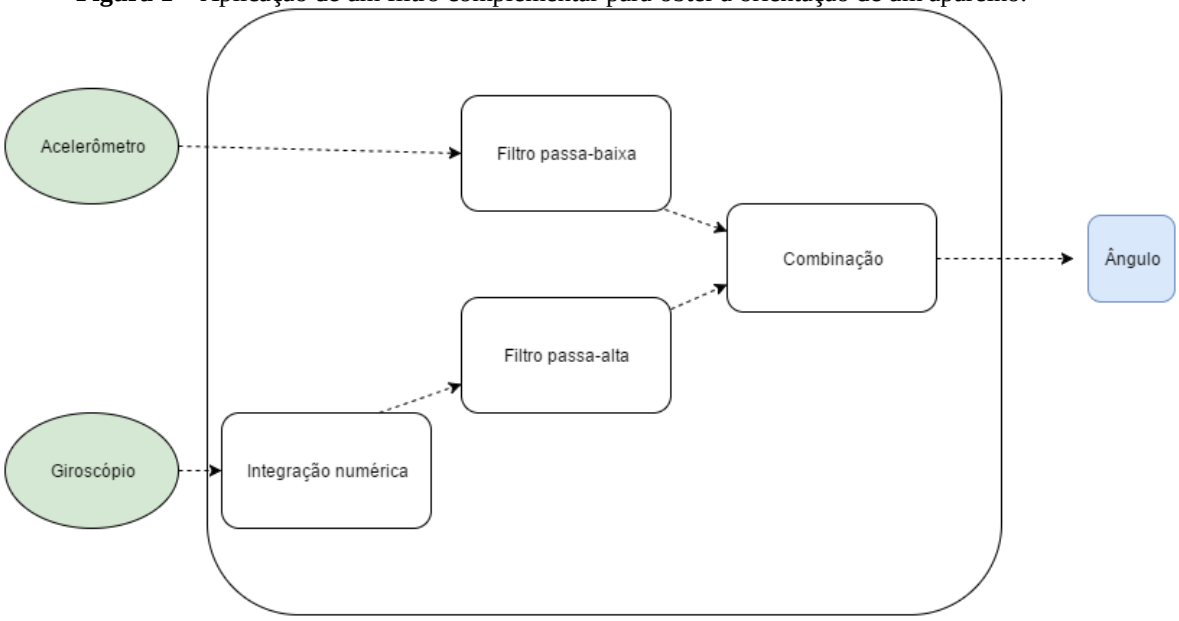
2.1.3 Filtro Complementar

Um filtro complementar nada mais é do que a aplicação simultânea de um filtro passa-baixa nas medições de um sensor e de um filtro passa-alta nas medições de outro sensor. Ele se diferencia de um filtro de banda pelo fato de receber entradas de dois sensores simultaneamente,

e não do mesmo sensor. Portanto, o filtro complementar oferece a possibilidade de se extrair as melhores informações de cada sensor e combiná-las em um dado mais puro, tendo sido extensivamente utilizado em robótica e aviação (HIGGINS JR, 1975).

Um caso de uso moderno do filtro complementar é na obtenção da orientação de um aparelho. Enquanto um resultado baseado puramente no acelerômetro ou no giroscópio poderia ser suficiente, distorções de alta frequência no acelerômetro e de baixa frequência no giroscópio tornam os resultados de ambos imprecisos. A Figura 1 mostra o uso de um filtro complementar neste caso.

Figura 1 – Aplicação de um filtro complementar para obter a orientação de um aparelho.



Fonte - Elaborada pelo autor.

É possível observar que se aplicando um filtro passa-alta nas medições do giroscópio, após estas serem integradas numericamente para se obter o ângulo de rotação, e um filtro passa-baixa nas medições do acelerômetro, eliminam-se as características indesejadas de cada sensor e então sobra apenas um sinal mais puro. Uma explicação em detalhes de cada sensor é dada na seção 2.2.

2.2 Sensores em Smartphones

Aparelhos de telefonia celular modernos são carregados de sensores que fornecem dados puros e geralmente precisos sobre características ambientais, de movimentação e posição. Os dados fornecidos por esses sensores costumam ser acessíveis a desenvolvedores a partir das *Software Development Kits* (SDKs) dos respectivos sistemas operacionais do aparelho. O desenvolvedor pode, então, utilizar esses dados para criar aplicações. Poder definir uma taxa de amostragem para cada sensor é uma característica comum nas SDKs do mercado.

O uso da maioria dos sensores não causa um grande impacto na bateria do *smartphone*. Ainda é possível que o desenvolvedor controle o estado de alguns sensores, enquanto outros

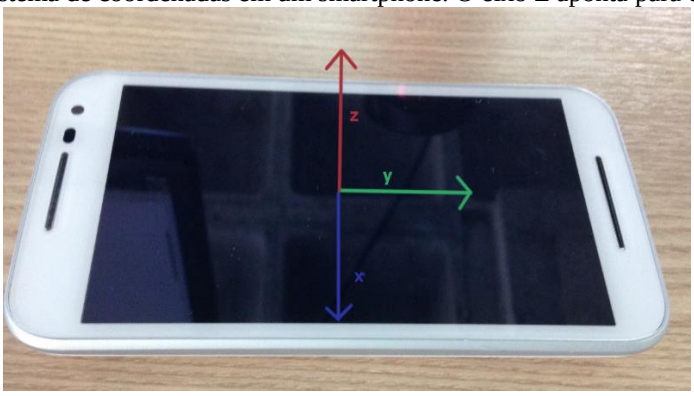
sempre estarão ligados por padrão. O sensor de GPS é uma exceção em respeito ao consumo de bateria, visto que ele provoca um intenso uso da mesma. Técnicas de localização indoor, porém, não dependem do sensor de GPS e usufruem de uma melhor eficiência energética.

Os sensores mais relevantes atualmente no contexto de *dead reckoning* são os relacionados à movimentação e posição do *smartphone*: O acelerômetro, o giroscópio e o magnetômetro. Estes três sensores, portanto, são descritos em detalhes a seguir. Sensores ambientais como termômetros e fotômetros, que medem, respectivamente, a temperatura e luminosidade do ambiente, não são comumente utilizados em técnicas de localização *indoor*.

2.2.1 Acelerômetro

O acelerômetro é um sensor que mede a aceleração própria sendo aplicada ao smartphone. Existem acelerômetros com variados números de eixos, sendo os mais comuns em *smartphones* modernos os acelerômetros de três eixos. A Figura 2 mostra o sistema de coordenadas local do aparelho que é utilizado pelo acelerômetro. Em *smartphones* Android, o acelerômetro consome cerca de dez vezes menos energia do que os demais sensores de movimentação (ANDROID, 2016).

Figura 2 – Sistema de coordenadas em um smartphone. O eixo Z aponta para cima.



Fonte - Elaborada pelo autor.

Por medir a aceleração própria, dispositivos em repouso em uma superfície irão gerar medições de aproximadamente $9.8\frac{\mathrm{m}}{s^2}$ no eixo Z do acelerômetro. Consequentemente, o acelerômetro de um *smartphone* em queda livre apresentaria um valor próximo a $0\frac{\mathrm{m}}{s^2}$ no eixo Z. O cálculo da aceleração própria no dispositivo (A_d) realizado pelo acelerômetro é baseado nas forças aplicadas ao próprio sensor (F_i) levando em conta a força da gravidade (G) e a massa do aparelho (m), a partir da Equação 2.

$$A_d = -G - \sum_i \left(\frac{F_i}{m}\right) \tag{2}$$

Para isolarmos a aceleração do dispositivo da força de gravidade, uma possível abordagem é utilizar um filtro passa-alta sobre as amostras. Similarmente, o vetor da gravidade pode ser

isolado utilizando-se um filtro passa-baixa. Essa abordagem, porém, não oferece um valor muito preciso devido a ruídos no sensor. Uma abordagem que oferece resultados mais precisos é utilizar um filtro complementar com o acelerômetro (passa-baixa) e o giroscópio (passa-alta).

2.2.2 Giroscópio

O giroscópio é o sensor responsável por mensurar a taxa de rotação, em $\frac{rad}{s}$, em volta dos eixos X, Y e Z do aparelho. O sinal detectado pelo giroscópio em geral é consideravelmente ruidoso, e suas medições também podem ser afetadas por um viés inicial, chamado de *drift*. Esse *drift* diz respeito aos valores fornecidos pelo sensor mesmo quando o aparelho não está sofrendo rotação. O *drift* pode variar de acordo com o tempo e também fatores externos como a temperatura do ambiente e do próprio giroscópio (MIRANDA et al., 2006).

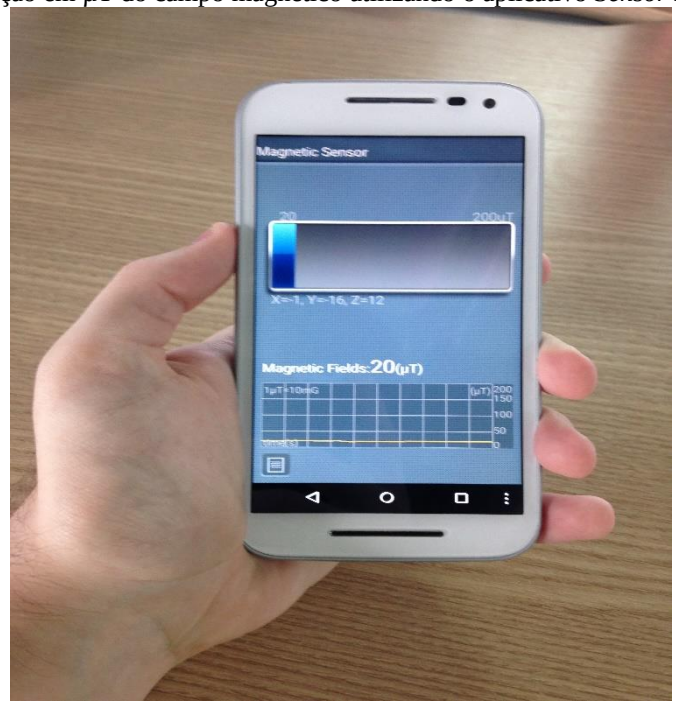
A instabilidade do giroscópio o torna não muito confiável a longo prazo, portanto, é comum que se usem filtros passa-alta para obter apenas as variações de orientação pontuais no contexto de localização *indoor*. Essas variações podem ser usadas, por exemplo, para calibrar a localização estimada de um usuário (WANG et al., 2012) ou reposicionar o sistema de coordenadas local de um *smartphone* (ROY et al., 2014). Tentativas de detecção e correção do viés inicial envolvem o uso de outros sensores.

Apesar do giroscópio ser um sensor de relativamente fácil acesso, muitos *smartphones* modernos considerados *low-end* e *mid-range* estão sendo fabricados sem ele. Esse padrão é particularmente prejudicial para a área de localização *indoor*, visto que muitas técnicas confiam nos valores fornecidos pelo giroscópio para localizar os usuários de *smartphones* (BELLO, 2015).

2.2.3 Magnetômetro

Também denominado bússola, o magnetômetro mede o campo magnético terrestre e mudanças neste, em cada um dos três eixos (X, Y e Z) em µT. O magnetômetro é muito sensível a distorções causadas por objetos ferromagnéticos no ambiente, e por isso é comum que o sensor seja calibrado de acordo com algum modelo para lidar com esse tipo de interferência, evitando a obtenção de resultados enviesados (WANG et al., 2012). Um exemplo de dados fornecidos por um magnetômetro pode ser visto na Figura 3, e uma demonstração de interferências pode ser vista na Figura 4.

Figura 3 – Medição em µT do campo magnético utilizando o aplicativo *Sensor Box for Android*[1].



Fonte – Elaborada pelo autor.

Figura 4 – Medições da bússola no mesmo smartphone da Figura 3, próximo a um Mac Mini[2].

Fonte – Elaborada pelo autor.

[1] https://play.google.com/store/apps/details?id=imoblife.androidsensorbox&hl=pt_BR
[2] http://www.apple.com/br/mac-mini/

Especificamente, na Figura 3 pode-se observar um medidor que mostra a norma do vetor medido pelo magnetômetro em µT, e o valor para cada eixo logo abaixo. Na parte inferior da tela, vê-se que o valor do campo magnético não variou muito nos últimos segundos. Já na Figura 4, quando o *smartphone* é colocado próximo a um Mac Mini, o magnetômetro lê um valor muito maior do que no caso anterior (163 µT contra 20 µT), devido às interferências causadas pelos componentes do computador. Esse caso é exagerado, visto que em situações comuns de uso um *smartphone* não estaria tão próximo a um causador de interferência como é mostrado na figura.

Além das interferências causadas por objetos ferromagnéticos, que podem ser divididas nos tipos *hard iron* e *soft iron*, magnetômetros também podem sofrer interferências da temperatura, oriundas de fabricação ou até mesmo oferecer resultados indesejáveis devido à própria sensibilidade do sensor. Há um ramo de pesquisa dedicado à idealização de técnicas para a calibração automática de magnetômetro, e uma das ramificações deste é a calibração automática do magnetômetro para pedestres em ambientes urbanos ou veiculares (WAHDAN, 2015).

Para técnicas de *dead reckoning* e localização *indoor* em geral, ter conhecimento do verdadeiro vetor do norte magnético terrestre é de extrema importância. A partir dele, é possível transferir cálculos de um sistema de coordenadas local do aparelho para um sistema de coordenadas global (BELLO, 2015). O problema de interferências, porém, é muito mais intenso em ambientes fechados vez que há uma maior presença de elementos ferromagnéticos nas proximidades do aparelho.

2.3 Considerações Finais

O processamento dos dados de sensores nos *smartphones* é uma etapa crucial em técnicas de *dead reckoning* para localização em ambientes internos. Foi mostrado que a escolha do sensor e dos filtros a se utilizar, além da parametrização destes, deve ser feita cuidadosamente de acordo com o dado que se deseja obter e o contexto da aplicação. Para localização *indoor*, os casos mais comuns são utilizar um filtro passa-baixa nos dados do acelerômetro e um filtro complementar nos dados do giroscópio e acelerômetro ou magnetômetro.

As interferências sofridas por um magnetômetro, exemplificadas na subseção 2.2.3, permanecem como um problema a ser enfrentado por técnicas de localização que procuram utilizar o sensor: Não há uma maneira geralmente aceita para se lidar com tais interferências. O giroscópio sofre de problemas similares. Dados estes conceitos fundamentais e uma definição básica de *dead reckoning*, o capítulo a seguir aborda técnicas de localização *indoor* através de um ponto de vista histórico.

3. Localização *Indoor*

A localização de pessoas em ambientes *indoor* é um desafio a parte na área de localização, tendo sido pesquisado constantemente na última década, dado que a humanidade passa cada vez mais tempo em interiores. Diversos fatores contribuem para esse desafio, dentre eles: Limitações espaciais, impossibilidade de acesso a sinais de satélite ou redes mais amplas e grande presença de obstáculos físicos (tais como paredes e portas). Além disso, espaços fechados costumam ter pequenas dimensões quando comparados aos exteriores, logo, uma precisão que é aceitável para localização *outdoor* pode não ser aceitável para localização *indoor*.

Um possível caso de uso de uma técnica de localização *indoor* pode ser, por exemplo, detectar em qual fileira de um supermercado está um cliente, com o intuito de oferecer informações sobre os produtos nela contidos. Vê-se que neste caso, uma diferença de um metro ou menos na localização pode ser crucial para o correto funcionamento da aplicação. Seguindo esse pressuposto, certas técnicas modernas de localização *indoor* visam fornecer uma precisão a nível de centímetros (LYMBEROPOULOS et al., 2015).

O acoplamento de *dead reckoning* em técnicas de localização *indoor*, foco deste trabalho, é mencionado anos após o início das pesquisas. Adicionalmente, localização *indoor* em si é um tópico de pesquisa vasto e diversificado, podendo ser abordado a fundo em um trabalho separado. Não obstante, é de considerável importância para o trabalho oferecer conhecimento sobre algumas das principais técnicas de localização *indoor*, sendo este o objetivo desse capítulo.

3.1 Dependência em Infraestrutura

Técnicas de localização *indoor* podem ser classificadas em dois principais grupos de acordo com a dependência das mesmas em infraestrutura adicional, isto é: Técnicas dependentes em infraestrutura extra e técnicas independentes de infraestrutura extra. Ao adicionar-se tal dependência adicionam-se também custos de investimento e de manutenção da infraestrutura, porém, geralmente é possível obter uma melhor precisão na tarefa.

Nos primórdios da área foram desenvolvidas diversas técnicas de localização *indoor* baseadas em infraestrutura, tais como técnicas por infravermelho (WANT et al., 1992), som (PRIYANTHA; BALAKRISHNAN, 2000) e *Angle on Arrival* (TEKINAY, 1998), cada uma tendo problemas particulares e compartilhando de um problema principal: má escalabilidade. A possibilidade de oferecer mais precisão, entretanto, incentiva a pesquisa de técnicas relacionadas até hoje.

Adicionalmente, essa vantagem em relação a técnicas independentes de infraestrutura pode justificar o investimento em *hardware* adicional para técnicas dependentes em certos casos. Um exemplo atual de *hardware* são os iBeacons[3] oferecidos pela Apple, que se baseiam em redes Bluetooth[4] *Low Energy* (BLE) para localizar usuários de *smartphones*. Diversos outros *beacons* usando a rede BLE já foram desenvolvidos[5].

[3] https://developer.apple.com/ibeacon/
[4] https://www.bluetooth.com/
[5] https://indoo.rs/

Pesquisas são conduzidas atualmente para otimizar o espalhamento de *beacons* em um certo ambiente com o intuito de maximizar a cobertura do sinal BLE (RAJAGOPAL et al., 2016). São conduzidas também investigações sobre ligamento dinâmico de *beacons* para poupar energia e, logo, estender a vida de bateria destes para reduzir a frequência de manutenções necessárias na infraestrutura (DAI et al., 2015).

A má escalabilidade de tais técnicas, porém, é muito limitante quando se leva em conta um outro cenário atual do campo, que enfrenta o desafio de localizar milhões de usuários de *smartphones* em tempo real sem obriga-los a ter um investimento extra para isso. Em comparação, técnicas livres de infraestrutura adicional não possuem limites físicos para sua escalabilidade.

Algumas técnicas livres de infraestrutura adicional utilizam puramente o campo magnético terrestre, a partir de leituras de magnetômetro, para localizar as pessoas (SONG et al., 2016). A grande vantagem dessas técnicas reside na gratuidade e ausência de limites para o uso da sua fonte principal, porém, o tratamento da grande quantidade de interferências geradas em ambientes *indoor* é complexo e pode limitar a técnica em termos de escalabilidade e precisão.

A maioria das técnicas livres de infraestrutura, por sua vez, aproveita o fato de que *smartphones* possuem acesso a redes locais sem fio (WLANs) e que o sinal destas, dado que esteja em conformidade com o padrão IEEE 802.11, oferece informação sobre a potência do sinal e a identificação do ponto de acesso, sendo praticamente onipresente nas áreas mais populosas do mundo. A utilização dessa fonte, juntamente com os sensores embarcados, elimina a necessidade de equipamentos adicionais e pode abranger uma grande parcela de pessoas, dada a disseminação de *smartphones* nos últimos anos (FRIEDMAN, 2015).

Partindo desse ponto, também é usual que se misturem fontes adicionais de informação com o sinal WiFi para localizar o usuário. Uma dessas possíveis fontes é a interpretação dos dados de sensores a partir de *dead reckoning*. Consequentemente, o histórico de técnicas apresentado neste capítulo abrange apenas as principais técnicas de localização livres de infraestrutura adicional predominantes no campo ao longo dos anos. Uma descrição geral e extremamente detalhada sobre técnicas de localização *indoor* é elaborada por Mautz (2012).

3.2 RADAR

O RADAR (BAHL; PADMANABHAN, 2000) pode ser considerado o sistema precursor de localização *indoor* baseada em redes WLAN, desenvolvido em meados do ano 2000. Na época, uma boa quantidade de pesquisa já era conduzida no campo de sistemas e serviços sensíveis à localização, porém, pouca atenção era dada à tarefa de localizar pessoas, especialmente dentro de estabelecimentos. O estado da arte consistia, majoritariamente, de técnicas baseadas em redes de sensores em infravermelho.

As técnicas de localização dependentes em dispositivos infravermelho possuíam um alcance limitado e sofriam quedas de precisão em ambientes muito afetados por luz solar. Além disso, eram totalmente desacopladas de qualquer outra rede de dados e implicavam na instalação de dispositivos no ambiente cujos objetos deviam ser localizados, como *Active Badges* (WANT

et al., 1992). Outras técnicas menos proeminentes na época procuravam utilizar redes de telefonia (TEKINAY, 1998) ou o próprio GPS (ENGE; MISRA, 1999).

Para evitar as limitações das outras abordagens, o RADAR procura utilizar redes WLAN, presentes em boa parte dos ambientes fechados, para localizar pessoas baseando-se na potência do sinal de cada ponto de acesso WiFi recebido em um dispositivo móvel. A partir dessa estratégia, o RADAR também complementa redes WLAN com informações de localização dos seus usuários, enriquecendo-as.

A potência do sinal emitido por um ponto de acesso é mensurável em uma dada localização dentro de um estabelecimento a partir da placa de rede do dispositivo que recebe o sinal, sendo medida geralmente em decibéis miliwatt (dBm). Ao se afastar do ponto de acesso que emite o sinal, é esperado que o valor mensurado diminua, e que ele aumente na situação contrária. Essa premissa é provada empiricamente pelos autores do artigo.

Para localizar um usuário, a estratégia básica do RADAR consiste em avaliar as potências de sinal recebidas por ele, juntamente com dados prévios que associam localizações do ambiente a um conjunto de potências de sinal, para detectar a localização cujas leituras mais se assemelham com as do usuário. O algoritmo utilizado para detectar os melhores pareamentos de localizações é chamado de *Nearest Neighbor(s) in Signal Space* (NNSS).

Sendo $\vec{S} = \langle ss_1, ss_2, \dots, ss_k \rangle$ o vetor de potências de sinal detectado pelo usuário em um certo momento, $\vec{S'} = \langle ss'_1, ss'_2, \dots, ss'_k \rangle$ um vetor de potências pré-computadas em uma dada localização e K o número de pontos de acesso no ambiente, o algoritmo NNSS calcula a distância em espaço de sinal do vetor $\vec{S}$ com os vetores de medições prévias e retorna o vetor $\vec{S'}$ que minimiza tal distância. A função de distância usada pelos autores é a Euclidiana, porém outras funções como a de Manhattan ou Mahalanobis[6] podem ser utilizadas.

O NNSS pode ser aprimorado, também, calculando-se o centroide de um número N de vizinhos mais próximos ao invés de um único vizinho. Os experimentos realizados pelos autores para diversos valores de N, entretanto, mostraram que as melhorias trazidas por essa abordagem não foram significantes.

São propostas duas possíveis formas de obter-se o conjunto de dados pré-computado que é necessário para a execução do NNSS: A partir de um método empírico ou a partir de uma modelagem da propagação dos sinais no ambiente. A abordagem empírica para o problema depende de uma fase *off-line* de medições no estabelecimento no qual deseja-se implementar o sistema, chamada de *fingerprinting*, porém oferece uma precisão maior na localização dos usuários. Por outro lado, utilizar um modelo de propagação elimina a limitação de uma etapa *off-line* a troco de menores precisões.

Na construção do conjunto de dados na etapa de *fingerprinting*, coletam-se amostras em pontos aleatórios do local, cada uma consistindo de um determinado número de leituras das potências dos sinais emitidos por cada ponto de acesso. O número de pontos de medição e o número de amostras por ponto fica a cargo do usuário. Ao invés de armazenar-se os dados de todas as amostras, calculam-se a média e o desvio padrão amostral dos sinais em cada ponto.

Juntamente com as potências de sinais e a localização onde foi realizada a medição, a orientação do usuário (Norte, Sul, Leste ou Oeste) também é armazenada. A razão disto é o fato

[6] https://www.mathworks.com/help/stats/mahal.html

do corpo humano atuar como um atenuador dos sinais recebidos, sendo esta atenuação mais proeminente quando o corpo se posiciona totalmente entre o dispositivo de medição e o ponto de acesso. O RADAR, portanto, não somente estima a localização dos usuários, mas também a orientação deles, ainda que de forma rústica.

Para eliminar o trabalho de *fingerprinting*, que demanda horas de esforço humano e precisa ser refeito toda vez que um ponto de acesso é relocado ou removido do local, um modelo matemático de propagação pode ser utilizado para simula-lo. Isto é, funções podem ser utilizadas para criar um conjunto de dados fictício em um dado local. É sobre essa hipótese que os autores propõem a versão do RADAR baseada em um modelo de propagação.

Um dos principais desafios de criar um modelo de propagação de sinais dentro de um dado estabelecimento é a grande quantidade de fenômenos que precisam ser levados em conta nas equações, para que a qualidade da localização seja mantida em relação ao sistema operando sobre uma fase empírica. Exemplos desses fenômenos são reflexões, difrações e propagação por múltiplos caminhos, descritos em detalhes por Mitra (2009). Além disso, o tipo de material de construção usado e o número de objetos em um local também influenciam o modelo.

O modelo proposto pelos autores leva em conta uma taxa que representa o decaimento da potência do sinal de acordo com a distância do dispositivo de medição para o ponto de acesso, além de um fator de atenuação que decrementa o sinal de acordo com o número de paredes que ele atravessou. O modelo é chamado de *Wall Attenuation Factor*, e é representado pela Equação 3.

$$P(d) = P(d_0) - 10n\log\left(\frac{d}{d_0}\right) - \begin{cases} nW * \text{WAF}, & nW < C \\ C * \text{WAF}, & nW \geq C \end{cases} \quad (3)$$

Na Equação 3, n é a razão de distância, $P(d_0)$ é a potência do sinal em uma referência d_0 e d é a distância dos dispositivos. Adicionalmente, C é o número máximo de paredes a serem atravessadas pelo sinal, nW é o número de paredes efetivamente atravessadas e F é um fator de atenuação. Os valores de $P(d_0)$, n, C e F são escolhidos empiricamente baseando-se nas características do local modelado. Para estimar esses valores, é necessário um mapa do estabelecimento.

Após determinados os valores necessários para o modelo matemático, são amostrados pontos aleatórios em um *grid* sobreposto ao mapa do local a ser modelado e, para cada um deles, o vetor de potências de sinais de cada ponto de acesso é estimado. As medidas geradas por esse processo são, consequentemente, utilizadas como o conjunto de dados pré-computados que seriam outrora obtidos a partir de *fingerprinting*.

Ambas as abordagens foram testadas no mesmo local, contendo diversas salas, portas e paredes. Para os testes, foram utilizados 3 pontos de acesso. O método empírico contou com 70 pontos de coleta distintos, sendo as medições em cada um deles realizada para as quatro orientações possíveis (Norte, Sul, Leste e Oeste) e com 20 leituras seguidas.

Enquanto o sistema baseado no método empírico atingiu uma precisão de 2.94m na mediana e 1.92m no primeiro quartil, o método baseado na modelagem de propagação atingiu 4.3m na mediana e 1.86m no primeiro quartil. É notória a diferença de precisão entre os dois,

porém a necessidade da eliminação de uma etapa manual pode ser crítica o suficiente para justificar o uso do modelo de propagação.

Posteriormente, o problema atacado pelo RADAR é estendido para o caso de rastreamento de usuários caminhando com velocidade constante. Para tal, o sistema utiliza uma janela de tamanho W das últimas 10 amostragens de sinais obtidas pelo usuário e calcula a média destes, alimentando o sistema básico de localização. Portanto, o problema é reduzido a uma sequência de etapas de localização de usuários estacionários, atingindo uma acurácia 19% abaixo da obtida durante localizações estáticas.

Analisando a abordagem para rastreamento de usuários proposta, identifica-se um possível caso onde o uso de *dead reckoning* seria um incremento para o sistema. Entretanto, essa fusão de informações não é mencionada na época. Após a divulgação do RADAR, diversos trabalhos foram desenvolvidos com o intuito de incrementar as ideias nele apresentadas. Dentre esses trabalhos, um dos mais notórios foi o HORUS, que é explicado na seção seguinte.

3.3 HORUS

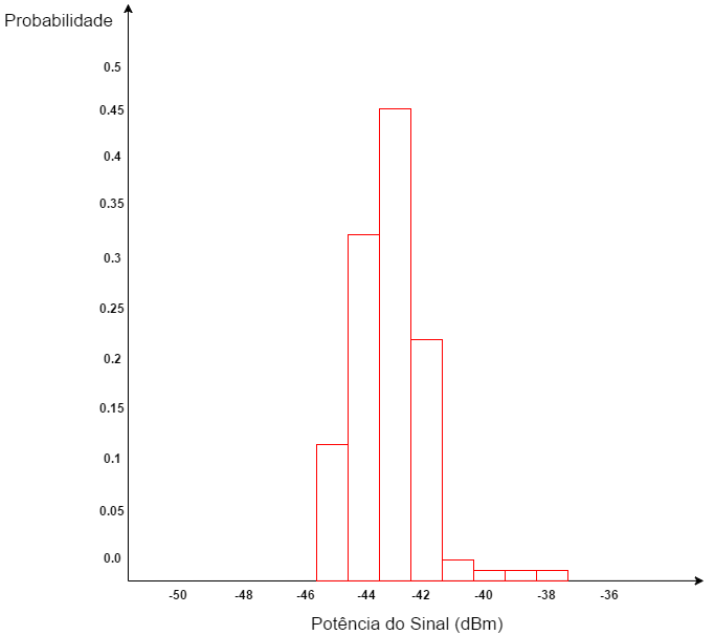
O sistema HORUS de localização *indoor* (YOUSSEF, 2004) foi apresentado alguns anos após o RADAR, e sua proposta consiste em um sistema modularizado, independente de infraestrutura extra, baseado em redes WLAN e separado em duas fases, assim como o RADAR. A primeira fase do sistema consiste no mapeamento *off-line* das potências de sinais (*fingerprinting*) e o pré-processamento destas em um determinado estabelecimento. A segunda etapa, por sua vez, consiste na estimativa *online* da localização dos usuários baseada no mapa previamente construído.

O mapa criado pelo HORUS na sua fase de *fingerprinting* armazena a distribuição das potências de sinais de cada ponto de acesso em cada localização amostrada. Essa distribuição pode ser paramétrica ou não-paramétrica, e o sistema apresenta módulos para ambas as abordagens. A principal diferença teórica do HORUS para o RADAR reside na abordagem probabilística que ele emprega para a localização dos usuários, enquanto o RADAR é puramente determinístico.

A abordagem probabilística do sistema consiste em, dado um vetor de potências de k sinais $\vec{s}$, encontrar a localização x que maximize a probabilidade condicional $P(x|s)$, ou seja, a localização cuja probabilidade é máxima dado que as potências de sinais observadas foram $s_1 \dots s_k$. É importante notar que o espaço X de localizações é considerado discreto, sendo essa condição relaxada posteriormente por um módulo separado.

Modelando as distribuições das potências de sinais de uma forma não-paramétrica, a etapa *off-line* se resume a armazenar em um mapa a probabilidade conjunta de todos os pontos de acesso WLAN (APs) para cada localização. A probabilidade em cada AP pode ser estimada a partir de um histograma normalizado representando as faixas de potência de sinal, exemplificado pela Figura 5. Considerando que $\vec{s}$ é constante para cada x, e que todas as localizações em X são equiprováveis, a fase *online* deve maximizar a Equação 4, onde cada $P(s_i|x)$ é trivialmente consultado no mapa. Em seguida, $P(x|s)$ é estimada a partir do teorema de Bayes.

Figura 5 – Exemplo de histograma gerado pelo HORUS.



Fonte – Elaborada pelo autor.

$$P(s|x) = \prod_{i=1}^{k} P(s_i|x) \tag{4}$$

A modelagem das distribuições de forma paramétrica, por sua vez, é realizada adaptando-se os histogramas dos APs para distribuições gaussianas na fase *off-line* e armazenando-as em um mapa. Utilizando o método de estimativa por máxima verossimilhança, a média e o desvio padrão da função de densidade de probabilidade gaussiana (Equação 5) são obtidos, como mostram as Equações 6 e 7, onde $s_i(j)$ é a j-ésima amostra de potência de sinal do AP i.

$$fdp(q) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(q-\mu)^2}{2\sigma^2}} \tag{5}$$

$$\hat{\mu} = \frac{1}{n}\sum_{j=1}^{n} s_i(j) \tag{6}$$

$$\hat{\sigma} = \sqrt{\frac{1}{n}\sum_{j=1}^{n} [s_i(j) - \hat{\mu}]^2} \tag{7}$$

Na fase *online*, a Equação 4 é maximizada obtendo-se cada $P(s_i|x)$ a partir da integral correspondente na distribuição gaussiana armazenada, processo este representado pela equação

8. Ambas as abordagens de modelagem do problema são comprovadamente ótimas dentre as técnicas de localização a partir de mapas construídos de forma *off-line*, de acordo com o erro médio de localização. O HORUS, portanto, consegue ser mais eficiente que o RADAR em termos de acurácia, dadas as hipóteses tomadas.

$$P(s_i|x) = \int_{s_i-0.5}^{s_i+0.5} fdp(q)\, dq \tag{8}$$

O HORUS apresenta, adicionalmente, quatro outros módulos de otimização da localização, tais como um módulo de compensação das características de pequena escala das redes WLAN e um módulo que lida com a correlação entre sucessivas amostras de um mesmo ponto de acesso. Apesar de ser um sistema único, o fato do HORUS ser extremamente modularizado implica no uso irrestrito de seus módulos em outros sistemas de localização, o que foi uma das grandes contribuições do trabalho.

O acoplamento dos módulos extras do HORUS à parte básica do sistema implica em um ganho significativo de performance deste. Complementarmente, ao acoplar os módulos do HORUS no RADAR, também é observado um incremento considerável na performance, de acordo com os testes realizados pelos autores.

Por ter sido um trabalho com uma forte base teórica, bem detalhado, e que trouxe diversas melhorias em relação ao estado da arte na época, o HORUS foi um enorme passo na área de localização *indoor*. Diversas outras pesquisas posteriores utilizam os resultados do sistema como um *benchmark* para avaliação das técnicas propostas, e desenvolvem novas abordagens baseadas nas ideias nele apresentadas. O conceito de *dead reckoning*, entretanto, não é abordado pelo HORUS.

3.4 UnLoc

Uma característica comum ao HORUS e ao RADAR, assim como a diversas técnicas desenvolvidas com base neles, é a necessidade de uma fase *off-line* para armazenar um mapa de sinais do estabelecimento no qual deseja-se lançar o sistema de localização. Um foco de pesquisa na área passou a ser, consequentemente, relaxar essa etapa de mapeamento sem sacrificar muita precisão e aumentando a escalabilidade dos sistemas. O sistema UnLoc (WANG et al., 2012), abreviação de *Unsupervised Localization*, propõe uma das mais notórias estratégias para lidar com o problema supracitado de forma não-supervisionada.

A principal hipótese levantada pelo UnLoc é a possibilidade de detectar padrões de sinais em determinados locais dentro de um estabelecimento e, consequentemente, de utiliza-los para localizar pessoas. Por exemplo, o movimento e os materiais de um elevador podem causar leituras características no acelerômetro e magnetômetro de um *smartphone*, assim como uma escada rolante ou uma escada. Dado que um certo padrão de leituras é detectado, é seguro localizar o usuário nas proximidades desses locais, chamados pelo UnLoc de *landmarks*.

O posicionamento de usuários em *landmarks* por si só, entretanto, é insuficiente para justificar um sistema completo de localização *indoor*, devido à inconstância destes no ambiente. Isto é, a densidade de *landmarks* no local, apesar de considerável, não é suficiente para fornecer localização precisa em tempo real. Para complementar esse aspecto, o UnLoc propõe o uso de *dead reckoning* baseado nas leituras do giroscópio e magnetômetro.

O UnLoc foi um dos trabalhos pioneiros na proposição com sucesso do uso de *dead reckoning* em ambientes *indoor*. Como mostrado no Capítulo 1 deste trabalho, as leituras ruidosas oriundas dos sensores de *smartphones* geram um acúmulo de erro em técnicas de localização com o passar do tempo. Um uso comum de *dead reckoning* para ambientes abertos na época do UnLoc era interpolar as localizações obtidas por meio dele com outras oriundas do GPS, para evitar o acúmulo de erro.

Devido à degradação do GPS em ambientes fechados, O UnLoc utiliza os *landmarks* para reinicializar o cálculo de localização por *dead reckoning*. Um usuário do sistema, portanto, é localizado prioritariamente por navegação inercial e é relocado para a posição de algum *landmark* quando ele é detectado nas proximidades. Para ser possível utilizar os *landmarks* da forma pretendida, entretanto, é primeiro necessário propor formas de detectá-los e posicioná-los no espaço.

Os *landmarks* podem ainda ser divididos em dois grupos: Os que possuem padrões bem definidos e podem ter sua localização determinada previamente a partir da planta baixa do local, e os que devem ter sua localização detectada dinamicamente de forma não supervisionada. Os pertencentes ao primeiro grupo são denominados *seed landmarks* (SLMs), enquanto os membros do segundo são chamados pelos autores de *organic landmarks* (OLMs).

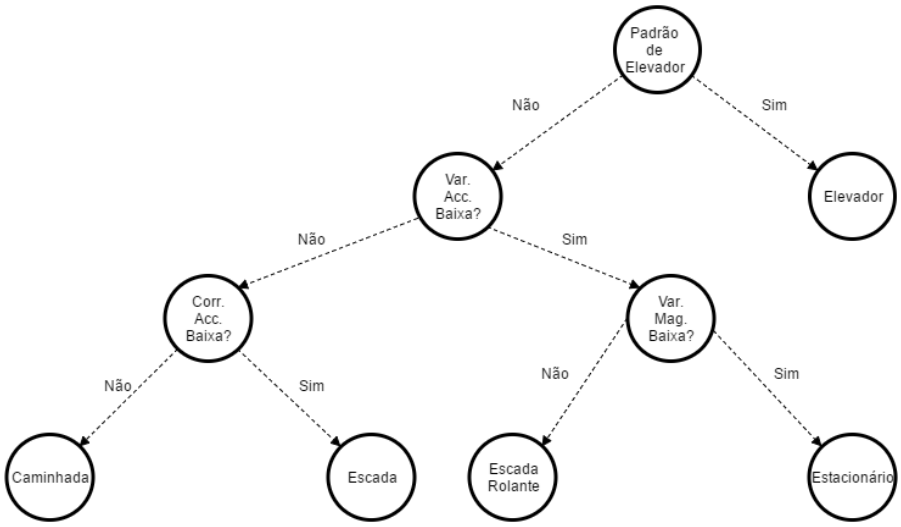
As subseções a seguir descrevem as técnicas desenvolvidas pelo UnLoc para tratar SLMs e OLMs separadamente. Por fim, é mostrada uma esquematização do sistema que incrementa sobre a ideia básica até então apresentada. As especificações técnicas do *dead reckoning* desenvolvido são dadas no Capítulo 4 deste trabalho.

3.4.1 Seed Landmarks

Os *seed landmarks* são locais comuns a uma variedade de estabelecimentos que geram assinaturas específicas nos sinais detectados pelos aparelhos dos usuários. Na localização em tempo real, os SLMs devem ser detectados de acordo com o padrão de sinais a eles atrelados. Os três tipos de SLMs abordados pelo UnLoc, por serem extremamente comuns, são elevadores, escadas e escadas rolantes.

A verificação de SLMs realizada pelo sistema pode ser representada a partir de uma árvore de decisão, mostrada na Figura 6. A partir da raiz, cada uma das checagens nos nós é realizada até que se atinja uma folha, que vai dizer se o usuário está localizado em um *landmark* e, em caso positivo, qual deles.

Figura 6 – Árvore de Decisão para SLMs.



Fonte - Elaborada pelo autor, adaptando de Wang et al. (2012).

A primeira checagem realizada é a por padrões gerados por elevadores. Um uso típico de elevadores consiste em um período de caminhada até o elevador, seguido de um período de espera por este, uma pequena caminhada para dentro, um período de espera já dentro do elevador e, por fim, uma caminhada para fora. Durante o período estacionário dentro do elevador, ocorre uma diminuição ou aumento no peso do usuário, dependendo do sentido da viagem, que causa picos no acelerômetro.

Para detectar a sequência dos movimentos típicos de uso de elevadores, é utilizada uma máquina de estados finita cujas transições dependem de limites empiricamente definidos. É importante notar que não somente é possível detectar quando o usuário está em um elevador, como também em qual sentido o usuário está se movimentando dentro dele.

Caso o padrão comum a elevadores não seja detectado, o sistema diferencia as classificações de velocidade constante (escadas rolantes e o caso estacionário) das demais (escadas e usuário andando) a partir da variância do acelerômetro. Uma baixa variância implica em casos de velocidade constante, porém, ainda é necessário diferenciar o caso de usuários em escadas rolantes de usuários em repouso.

Para realizar a diferenciação nos casos de velocidade constante, parte-se do princípio de que escadas rolantes causam perturbações nas medições do magnetômetro, implicando em uma maior variância nas leituras. Logo, uma alta variância no magnetômetro implica no caso de escadas rolantes, enquanto que uma baixa variância implica em um usuário totalmente em repouso.

Por fim, sendo o caso de velocidade constante descartado, restam duas possibilidades: Usuários caminhando em escadas ou usuários caminhando normalmente. Para separar os dois casos, os autores notaram que ao subir ou descer escadas há um acréscimo ou decréscimo de velocidade. Essa variação implica em uma maior correlação entre a aceleração no eixo Y e no eixo Z do sistema de coordenadas locais do *smartphone*. Portanto, uma maior correlação implica no caso de escadas enquanto uma menor correlação leva ao caso de caminhadas comuns.

Para cada um dos possíveis casos de SLMs, podem ocorrer falsos positivos ou falsos negativos. Enquanto falsos negativos fazem com que se perca uma chance de reiniciar o cálculo de *dead reckoning* a partir do posicionamento do usuário no SLM, falsos positivos geram a situação de posicionamento dos usuários em *landmarks* incorretos.

Caso a planta baixa do local no qual se deseja implementar o UnLoc esteja disponível previamente, os SLMs são localizados manualmente e servem, então, como um conjunto inicial sobre o qual o sistema irá operar. Em casos de indisponibilidade, porém, a posição dos SLMs precisa ser estimada dinamicamente de forma similar à que será mostrada na seção seguinte.

3.4.2 Organic Landmarks

Diferentemente dos *seed landmarks*, que possuem padrões bem definidos e são facilmente detectáveis em uma planta baixa, os *organic landmarks* podem representar lugares quaisquer dentro de um estabelecimento. É possível, por exemplo, que uma certa sala de um estabelecimento gere um padrão específico nas leituras do magnetômetro devido à presença de equipamentos de informática.

As etapas a seguir são descritas levando em conta que o sistema coleta leituras de diversos sensores do aparelho do usuário, e as atrela a *timestamps* e a localizações estimadas por *dead reckoning* naquele momento. As informações de todos os usuários são armazenadas em uma matriz.

Para detectar OLMs, o UnLoc primeiramente calcula diversas métricas sobre os sinais de cada um dos sensores, tais como valores máximos, mínimos e medianos, as normaliza e em seguida as processa com um algoritmo de *clustering* por K-médias[7]. Após a formação das partições, é calculada a similaridade entre elas com o objetivo de detectar partições com alto grau de significância, i.e.: que possuem similaridades com as demais partições abaixo de um limite.

A segunda etapa na detecção de OLMs consiste na verificação de proximidade dos membros da partição. Inicialmente, é verificado se todos os membros estão na mesma área de WiFi, isto é, se todos detectam as mesmas redes. Essa verificação elimina partições com uma baixa granularidade, agilizando a execução de uma segunda, de alta granularidade, que verifica a pertinência de todos os membros da partição a uma mesma área confinada. O valor dessa área foi escolhido como $4m^2$ pelos autores do sistema.

Caso uma partição passe pelas duas etapas de eliminação, ela é dita um OLM e sua posição é então calculada como o centroide da localização de todos os membros a ela pertencentes. Para executar o algoritmo de K-médias e o cálculo de similaridade entre partições, o UnLoc propõe diferentes métodos variando com o sensor utilizado como fonte dos dados: WiFi, magnetômetro ou uma combinação de sensores inerciais.

No caso dos *landmarks* detectados via WiFi, cada ponto de acesso corresponde a uma dimensão para o particionamento, e a similaridade é calculada a partir das relações entre as potências dos sinais para pares de cada uma das partições geradas. Sendo A o conjunto de todos

[7] https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html

os APs detectados em ambas as partições e $f_1(a)$ e $f_2(a)$ a potência do sinal do AP a nas partições 1 e 2, respectivamente, a Equação 9 mostra a similaridade entre duas partições de WiFi.

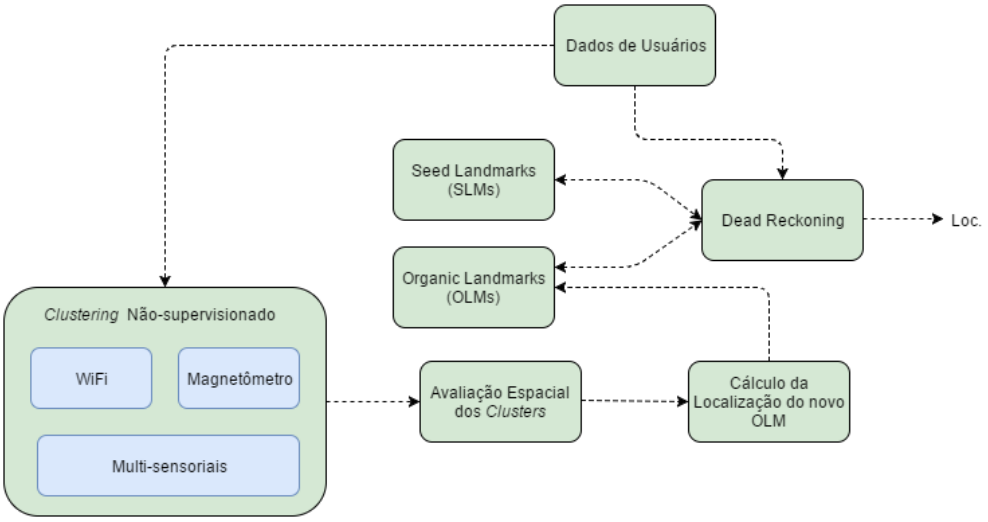
$$S = \frac{1}{|A|} \sum_{\forall a \in A} \frac{\min(f_1(a), f_2(a))}{\max(f_1(a), f_2(a))} \tag{9}$$

No caso de *landmarks* gerados via assinaturas magnéticas ou inerciais, são geradas métricas extras, tais como um coeficiente de curva para o giroscópio, a partir das obtidas previamente, e calcula-se a similaridade entre partições utilizando-as. Equações específicas, porém, não são mostradas pelos autores para esses casos.

3.4.3 Esquematização

Tendo sido detalhados os processos de detecção de *landmarks*, é possível descrever o UnLoc por um viés mais técnico, incrementando a descrição básica apresentada no início desta seção. Um esquema completo do UnLoc pode ser visto na Figura 7. Analisando-se as diferenças entre o sistema esquematizado e os apresentados anteriormente, i.e. HORUS e RADAR, é fácil notar a grande evolução que as pesquisas em localização *indoor* proporcionaram.

Figura 7 – Esquema geral do UnLoc.



Fonte – Elaborada pelo autor, adaptando de Wang et al. (2012).

A primeira etapa do UnLoc consiste na coleta de informações dos sensores de *smartphones* juntamente com o tempo no qual foram realizadas as medições. Essas coletas são realizadas constantemente, e os dados de cada uma são processados simultaneamente por dois módulos: Um módulo de *dead reckoning* e um de particionamento não-supervisionado para detecção de *landmarks*.

O módulo de *clustering* cria partições de acordo com os dados dos diversos sensores avaliados, seguindo as técnicas descritas na subseção 3.4.2. Caso haja partições consideradas relevantes de acordo com os critérios de similaridade, estas são encaminhadas para um módulo que calcula a coerência espacial de seus membros. Caso a partição seja também coerente no espaço, ela é considerada um novo OLM e então armazenada no sistema.

O módulo de *dead reckoning*, por sua vez, é o principal responsável pelo cálculo da localização do usuário em tempo real. Após calculada a posição do usuário a partir do *dead reckoning*, o módulo realiza ainda uma checagem pela presença de SLMs e OLMs de acordo com as leituras dos sensores e as técnicas explicitadas nas subseções passadas.

Caso seja detectado algum *landmark*, a posição dele é ajustada combinando-a com a estimativa atual. Após esse ajuste, o usuário é relocado para a nova posição do *landmark* e o *dead reckoning* é reiniciado. Caso não sejam detectados SLMs ou OLMs, a posição do usuário é puramente a estimativa atual.

É possível notar, portanto, que o UnLoc tende a um estado ótimo com o tempo, pois é esperado que o número de usuários cresça e, consequentemente, que a quantidade de informação disponível também. A partir de detecções subsequentes de *landmarks*, obtém-se estimativas mais precisas das posições destes, enquanto que estimativas mais precisas de *landmarks* melhoram o *dead reckoning*, resultando em um processo de aprimoramento recursivo.

Testes realizados pelos autores mostram que o UnLoc consegue atingir um erro médio de 1.69m quando 28 OLMs foram detectados, e de 1.9m quando apenas 10 destes foram localizados. Adicionalmente, os testes comprovam o comportamento esperado de aprimoramento recursivo do sistema. Comparado ao HORUS completo, o UnLoc atinge erros similares sem necessitar de uma etapa *off-line*.

3.5 Estado da Arte

O desenvolvimento de técnicas de localização em ambientes *indoor* livres de infraestrutura extra segue, majoritariamente, a tendência observada no UnLoc de eliminar etapas *off-line* e dependência em mapas a partir de *crowdsourcing*. Uma das possíveis razões para isso é o fortalecimento da independência de tais técnicas, visando maneiras totalmente escaláveis de localizar pessoas com facilidade a nível global. Exemplos de trabalhos nessa linha são os conduzidos por Bello (2015), por Yoo et al. (2016) e pela empresa Indoo.rs[8].

Os trabalhos supracitados levam a característica em comum de combinar leituras de diversos sensores e demais fontes para localizar os usuários, em um processo denominado atualmente de *sensor fusion*. Além disso, é comum tais trabalhos serem classificados como técnicas de mapeamento e localização simultâneos (SLAM), por construírem mapas de estabelecimentos em tempo real.

Complementarmente, também existe um foco, ainda que menor, em técnicas que não necessariamente relaxam a necessidade de plantas baixas de estabelecimentos, mas que eliminam a dependência em mapeamentos manuais. A técnica apresentada por Elbakly e Youssef (2016),

[8] https://indoo.rs

por exemplo, considera o uso de diagramas de Voronoi[9] para localizar usuários baseando-se em uma planta baixa do estabelecimento e relações de ordem entre as potências de sinais de APs.

É importante notar que há uma diferença gritante entre o foco das técnicas nesta seção apresentadas e outras técnicas que utilizam hardware extra, que visam principalmente baratear o custo dos equipamentos necessários e alcançar precisões muito superiores (AHMED; DIAZ; KAISER, 2016). Atualmente, são desenvolvidas técnicas, inclusive, que utilizam robôs para mapear estabelecimentos com precisões na ordem de grandeza dos milímetros (QUINTANA et al., 2016).

Uma limitação em comum entre todos os trabalhos até então citados é a falta de uma plataforma de testes em comum. Visto que cada técnica utiliza locais e bases de dados diferentes, análises comparativas envolvem a execução de técnicas anteriores em ambientes diferentes, gerando resultados inesperados. Adicionalmente, essa ausência facilita a escolha de um conjunto de testes que arbitrariamente fortaleça a técnica sendo descrita.

Trabalhos recentes na área (TORRES-SOSPEDRA et al., 2014; 2015) procuram fornecer bases de dados padronizadas para a avaliação de técnicas de localização *indoor*, sendo estas *open source*. Esses trabalhos, porém, ainda oferecem bases de baixa escala e carentes de detalhes mais específicos sobre a construção dos dados, como, por exemplo, a forma de calibração de sensores e especificações dos aparelhos utilizados.

3.6 Considerações Finais

Foram apresentados nesse capítulo trabalhos pioneiros na área de localização *indoor* independente de infraestrutura extra. O primeiro deles, RADAR, pode ser considerado o artigo seminal da área. O HORUS aprimora o RADAR com uma abordagem probabilística e fornecendo diversos módulos externos que incrementam a precisão do sistema e podem ser acoplados a quaisquer outros sistemas.

O UnLoc, por sua vez, foi um pioneiro na combinação de *dead reckoning* e *crowd sourcing* para eliminar a necessidade de etapas *off-line*. Por fim, foi dada uma breve descrição sobre o estado da arte na área e os problemas atualmente enfrentados pela academia. É importante notar que nesse campo do conhecimento, as indústrias seguem a academia rapidamente e, muitas vezes, propõem técnicas comparáveis ao estado da arte.

O principal objetivo do Capítulo 3 foi a contextualização do trabalho, que é focado em *dead reckoning*. A próxima seção fala especificamente do processo de *dead reckoning* e técnicas desenvolvidas, sendo uma delas a do UnLoc, que foi explicada apenas brevemente em favor da detecção de *landmarks*. Além disso, o próximo capítulo apresenta um módulo de confiabilidade para técnicas de *dead reckoning*, desenvolvido durante este trabalho.

[9] http://mathworld.wolfram.com/VoronoiDiagram.html

4. Dead Reckoning

O conceito de *dead reckoning* foi introduzido neste trabalho simplesmente como o processo de estimar a posição de um objeto carregado de sensores a partir de uma posição inicial e consequentes incrementos a esta, baseados em uma velocidade estimada. Esse processo é utilizado, por exemplo, em navegação marítima e aérea[10]. Neste capítulo, é dada uma descrição mais formal de *dead reckoning* no contexto de localização *indoor*.

Na literatura sobre localização de humanos portadores de dispositivos com sensores inerciais em ambientes fechados, o processo de *dead reckoning* recebe uma terminologia diferenciada: *Pedestrian Dead Reckoning* (PDR). O PDR pode ser particionado em dois desafios principais, sendo eles a determinação do deslocamento do usuário em um certo intervalo de tempo e a inferência da orientação do movimento no mesmo intervalo.

Dado que o movimento humano é compreendido por uma sequência de passos, o desafio de determinar o deslocamento de usuários em PDR pode ser reduzido para a detecção de passos e estimativas das distâncias percorridas por cada um deles. Existem pesquisas dedicadas a cada uma dessas partes, tais como as apresentadas nas respectivas seções deste capítulo.

Apesar das variadas técnicas utilizadas para o PDR atingirem boas precisões e reduzirem o acúmulo de erro com o tempo, um sistema de localização *indoor* baseado apenas em PDR não alcança otimalidade. Esse fato deve-se ao erro não eliminado dos sensores e a perturbações externas, tais como as mostradas no Capítulo 1 deste trabalho. Por essa razão, sistemas de localização *indoor* modernos costumam combinar PDR com alguma outra técnica de posicionamento, tais como os *landmarks* utilizados pelo UnLoc (WANG et al., 2012)

Uma habilidade valiosa a esses tipos de sistema é reconhecer momentos nos quais os resultados dos sensores para o PDR são confiáveis e, complementarmente, momentos nos quais eles estão sendo afetados por fatores externos ou estão desregulados e, logo, geram potenciais erros nos cálculos. Com essa informação disponível, o sistema é capaz de, por exemplo, decidir quando utilizar os resultados do PDR, quando é necessário compensa-los com outra técnica e quando é necessário aplicar uma nova calibração nos sensores.

Baseado na importância da informação supracitada, é apresentado também no atual capítulo um módulo de confiabilidade para *dead reckoning*, que detecta momentos de alta e baixa confiança nos seus resultados. As seções deste capítulo detalham, respectivamente, técnicas desenvolvidas para detecção de passos, para o cálculo do comprimento dos passos, para o cálculo da orientação e, por fim, o módulo de confiabilidade.

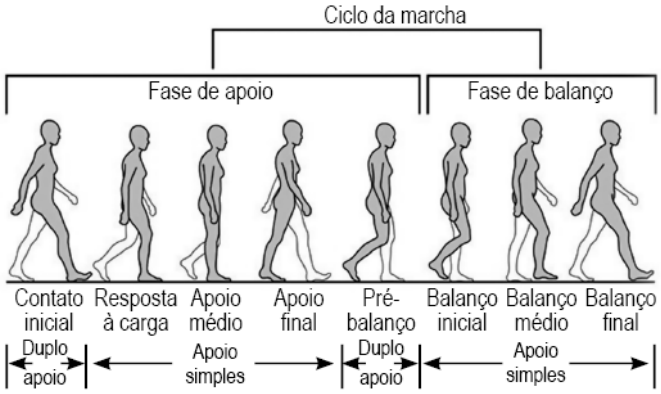
4.1 Detecção de Passos

A primeira etapa no desenvolvimento de um algoritmo para detecção de passos é o estudo do ciclo de caminhada humano. Apesar de possuir uma aparente trivialidade, o padrão com o qual andamos foi um importante alvo de estudos nos campos biológicos (ÖBERG et al., 1997) e

[10] http://www.experimentalaircraft.info/flight-planning/pilotage-dead-reckoning.php

animação computadorizada (FAURE et al., 1997). Adicionalmente, uma análise focada no caso de localização *indoor* é fornecia em ROY et al. (2014). O ciclo de caminhada é sumarizado pela Figura 8.

Figura 8 – Ciclo de caminhada humana.



Fonte – Obtida de http://www.portalsecad.com.br/demoArtigo.php?programa=38.

A partir da Figura 8, é possível notar que o ciclo de caminhada é dividido em duas fases principais: Uma fase de apoio, que compõe cerca de 60% do ciclo, e uma fase de balanço que representa os 40% restantes. Ambas as fases ainda podem ser divididas em múltiplas subfases de acordo com os ângulos de flexão dos joelhos e tornozelos, porém, para a análise deste trabalho é suficiente distinguir entre as duas fases principais e o ponto de início delas: O impacto do calcanhar.

O impacto do calcanhar no chão ao início da fase de apoio gera um padrão específico no acelerômetro, sendo este o momento de maior amplitude no sinal. O padrão gerado consiste em um pico, seguido de um vale e, por fim, um pico de menor amplitude. Por ser muito distinto dos demais momentos no ciclo de caminhada, percebe-se que este é um bom ponto para detectar a presença de passos.

Com o intuito de detectar o padrão gerado pelo impacto do calcanhar no chão, diversas abordagens foram propostas na academia. Uma das primeiras e mais simples delas é a técnica de *zero-crossing* (BEAUREGARD; HAAS, 2006), que consiste em detectar o momento no qual o módulo da aceleração própria detectada pelo sensor passa do módulo positivo para o negativo, desconsiderando a gravidade.

A abordagem de *zero-crossing*, porém, não é tão precisa na detecção dos passos por reduzir o padrão completo a uma simples checagem. Além de deixar de detectar passos eventualmente, o principal ponto negativo dela é a ocorrência de falsos positivos, ainda que incrementada com outras métricas.

Outra abordagem proposta para a detecção de passos é a verificação de picos (LADETTO, 2000), sendo um pouco mais robusta que a de *zero-crossing* por simplificar menos o padrão de caminhada. A detecção de picos consiste, primeiramente, na aplicação de um filtro passa-baixa nos eixos do acelerômetro e, em seguida, na detecção de pontos cuja aceleração é superior a um limiar empiricamente definido.

A estratégia proposta pelo UnLoc é similar à detecção de picos, porém, consiste em detectar dois mínimos locais consecutivos e buscar por um ponto de pico apenas entre eles. Caso a diferença entre o máximo local e os mínimos locais supere um limiar, o sistema considera que um passo foi dado. Com essa abordagem, o UnLoc conseguiu uma acurácia superior tanto em relação à detecção de picos quanto à técnica de *zero-crossing*.

Todas as estratégias até então citadas possuem uma limitação em comum, pois lidam com uma simplificação de um modelo muito mais complexo e, portanto, podem deixar de levar certos detalhes em consideração devido à baixa granularidade da solução. Uma abordagem que modela o problema com uma maior granularidade é apresentada em Alzantot e Youssef (2012), no sistema UPTIME de *pedestrian dead reckoning* (PDR).

O UPTIME observa que existem diversos estados presentes nos sinais do acelerômetro para cada passo, e que estes são mais capazes de representar o problema do que simples picos. Segundo o sistema, pode se considerar que um passo possivelmente foi iniciado ao ultrapassar-se um limiar positivo e, em seguida, um limiar negativo.

Após essas etapas, o passo pode ser confirmado caso o limiar negativo seja ultrapassado uma segunda vez, mas no outro sentido, e chegue-se a um valor terminal. Baseando-se nessas observações, o UPTIME propõe uma máquina de estados finita para a detecção de passos, onde cada uma das transições é governada por um limiar definido de forma empírica

Uma contribuição importantíssima do UPTIME foi a identificação da dependência entre os padrões gerados no acelerômetro e o posicionamento do celular no corpo humano. Técnicas anteriores consideravam, majoritariamente, as assinaturas geradas a partir de experimentos com o aparelho celular em mãos como válidas para outros casos, tais como aparelhos em bolsos de calças, em bolsos de camisas e em mochilas. Outra parte das técnicas, adicionalmente, apenas restringia o caso de uso ao de aparelhos em mãos, desconsiderando os restantes.

A máquina de estados finita proposta pelos autores do sistema foi desenvolvida levando-se em conta essa diferenciação. Por essa razão e por modelar o problema de forma inteligente, a abordagem proposta consegue uma porcentagem de falsos negativos cerca de seis vezes menor que a obtida pelo método de Beauregard e Haas (2006). No sistema de localização *indoor* baseado em *crowdsourcing* proposto por Bello (2015), a mesma máquina de estados é utilizada para a contagem de passos, sendo reportado um erro de seis a sete passos não detectados a cada cem.

O sistema apresentado em Bello (2015), entretanto, utiliza a máquina de estados apresentada no UPTIME apenas para os casos nos quais o *smartphone* está nas mãos do usuário. Apesar da máquina mostrar bons resultados para casos de celulares no bolso, estes não foram suficientes para os objetivos do sistema pretendido por Bello (2015). Detectar passos quando *smartphones* estão em bolsos de calça de seus respectivos usuários, entretanto, é um caso crucial para sistemas baseados em *crowdsourcing*, visto que boa parte das pessoas monitoradas mantém os aparelhos em algum bolso.

Devido à importância da detecção de passos nos casos de aparelhos em bolsos para sistemas baseados em *crowdsourcing*, e ao desempenho não satisfatório do UPTIME em experimentos, um novo método de detecção de passos para tais casos foi proposto por Bello (2015). Ele partiu da observação que a influência do impacto do calcanhar no chão é mais

eminente quando o celular se encontra em bolsos laterais, por ser mais influenciado pelas forças atuando na perna.

O método proposto interpreta o sinal como uma sequência de números discretos, dividindo-o em frações de onda e utilizando programação dinâmica para detectar os passos. É indicado no trabalho que o método atinge precisão similar ao utilizado para o caso de aparelhos em mãos, ou seja, de seis a sete passos não detectados a cada cem caminhados, o que é uma taxa aceitável de erros para o caso do sistema.

Para distinguir entre os momentos de utilizar diferentes métodos de acordo com o posicionamento do *smartphone*, o sistema proposto utiliza informações do sensor de proximidade, visto que bolsos não são espaçosos. Essa abordagem faz parte de um ramo acadêmico que visa diferenciar o aparelho em diferentes posições, assim como o trabalho de Susi, Renaudin e Lachapelle (2013).

Um outro ramo da academia, entretanto, busca fornecer técnicas independentes do posicionamento dos aparelhos no corpo para detecção de passos ou, ao menos, fornece-las para o maior número de casos possíveis. Um trabalho recente em tal ramo é apresentado por Kammoun et al. (2015). Este é denominado daqui em diante como FL, devido ao uso de *fuzzy logic* na detecção de passos proposta por ele.

No FL, os autores inicialmente processam os sinais com um filtro passa-baixa. Em seguida, detecta-se uma sequência específica no sinal: Uma passagem por um limiar positivo, seguida de *zero-crossing* no sentido negativo e, por fim, uma nova passagem pelo limiar. Caso a sequência seja respeitada, é gerado um candidato a passo. Após esse processamento rudimentar do sinal, é adicionada uma camada de lógica difusa sobre o problema.

O sistema FL leva em conta quatro características dos passos, sendo elas a duração total, a duração do ciclo de balanço, a amplitude da aceleração e a variância da aceleração. De acordo com esses discriminantes e valores *a priori* sobre eles, atribuem-se graus de pertinência a cada candidato a passo. A partir desses graus de pertinência, os candidatos são processados por um conjunto de cinco regras, consecutivamente. Ao fim da quinta regra, o sistema indica a validade do candidato.

O uso de lógica difusa na resolução deste desafio tem como intuito lidar eficientemente com falsos positivos e negativos, partindo da observação que eles são oriundos de casos degenerados que não são levados em conta pelos limiares propostos. Em comparação com o UPTIME, o FL apresenta uma menor taxa de falsos negativos e taxa similar de falsos positivos. O FL foi testado com seis posicionamentos diferentes de aparelhos.

Sumarizando as abordagens para detecção de passos, nota-se que as principais dificuldades consistem em modelar o problema de forma a não perder detalhes, processar os sinais de maneira ótima e, por fim, desenvolver invariância ao posicionamento dos aparelhos. Independentemente da estratégia utilizada, é necessário que o tamanho de cada passo seja estimado para que a distância percorrida pelas pessoas seja obtida por um sistema de PDR. A próxima seção explicita abordagens utilizadas para atacar esse desafio.

4.2 Estimativa do Comprimento de Passos

A tarefa de estimar o tamanho de cada passo dado é relativamente mais complexa do que a tarefa anterior, de detectar passos. Um estudo extensivo sobre passadas realizado em Öberg et al. (1997) apresenta diferentes tamanhos de passo, frequência de passadas e velocidade média de acordo com oito faixas etárias diferentes, para pessoas de ambos os gêneros e em diferentes ritmos de caminhada.

A gama de diferentes possibilidades mostradas no estudo é suficiente para notar-se a complexidade do problema enfrentado pelos desenvolvedores de sistemas de localização *indoor* ao atacarem a estimativa do tamanho de passos. O tamanho médio de passo para homens em uma caminhada normal, por exemplo, pode variar em até 6cm para diferentes faixas etárias, erro esse que, ao acumular-se, pode representar uma grande perda de precisão para o *dead reckoning*.

A abordagem mais rudimentar para estimar o tamanho dos passos para o PDR consiste em assumir um tamanho de passo constante para todas as pessoas, ultimamente ignorando as variações apontadas por Öberg et al. (1997). Exemplos de sistemas que utilizam essa estratégia são os propostos por Constandache, Choudhury e Rhee (2010) e Bylemans, Weyn e Klepal (2009).

Uma solução mais elaborada para o mesmo problema é modelar matematicamente o tamanho do passo como uma função da altura e peso do usuário. Essa estratégia, porém, é muito impeditiva para sistemas de localização *indoor*, visto que a altura e peso do usuário não são informações facilmente obtidas a partir de leituras de sensores apenas. É possível, entretanto, criar modelos a partir de métricas sobre a aceleração do usuário, informação facilmente disponível para os sistemas.

O estudo apresentado por Jahn et al. (2010) agrega diversas funções utilizadas para estimar o tamanho do passo a partir de dados do acelerômetro. Dentre as métricas necessárias, destacam-se os valores de máximo e mínimo e a variância da aceleração. O sistema criado por Bello (2015), utiliza uma dessas funções, explicitada na Equação 10 deste trabalho com fins de exemplificação, sendo $a_{peak,diff}$ a diferença entre os picos de aceleração e K uma constante empiricamente determinada.

$$StepLength = K\sqrt[4]{a_{peak,diff}} \tag{10}$$

O UnLoc também menciona o uso de um modelo matemático para atacar o referido problema, porém, os autores utilizam uma abordagem diferenciada em detrimento de um modelo fixo a longo prazo. A abordagem proposta por eles consiste em calcular o tamanho do passo para cada usuário baseando-se na distância percorrida entre dois *landmarks* e o número de passos dados. Isto é, após um usuário ter sua localização reiniciada ao menos duas vezes pela detecção de *landmarks*, ele pode ter uma estimativa muito mais precisa da sua passada.

O sistema UPTIME, apresentado na seção 4.1, também propõe uma estratégia diferente para estimar o comprimento dos passos para o PDR. O sistema em questão utiliza um

classificador SVM multi-classe[11] que, a partir de certas métricas da aceleração e da duração do passo, aponta qual é o tipo de caminhada sendo analisado, podendo este ser uma caminhada normal, um trote ou uma corrida.

Seguindo a detecção do tipo de caminhada, o sistema utiliza uma tabela de valores empiricamente obtidos para cada um dos tipos para obter o comprimento de cada passo. Apesar de reduzir os usuários a apenas três grupos específicos, a técnica consegue evitar o acúmulo de erro gerado pelo uso de um valor constante em até 71% para uma caminhada de 470m em um período de 6.3 minutos, segundo os testes realizados pelos autores.

Após a realização da contagem de passos e da estimativa do tamanho de cada um deles, é possível que um sistema de PDR estime a distância percorrida pela pessoa monitorada. Para complementar esse dado e indicar a localização da pessoa, porém, é necessário ao sistema estimar a orientação do movimento realizado. A próxima seção deste capítulo trata sobre o problema de determinação da orientação dos passos.

4.3 Determinação da Orientação

A orientação do movimento é o último componente aqui apresentado necessário a um módulo de *pedestrian dead reckoning* para inferir a localização de seus usuários. Assim como as tarefas anteriormente mostradas, a complexidade desta depende do quão generalizado o sistema de localização *indoor* pretende ser, vez que diferentes posicionamentos de *smartphones* no corpo podem levar a diferentes abordagens.

Uma distinção entre as estimativas de orientação do movimento é o sistema de coordenadas no qual elas são realizadas. Enquanto que é possível obter a orientação no sistema de coordenadas locais do *smartphone*, isto nem sempre é suficiente para localizar pessoas. Por essa razão, sistemas completos de localização *indoor* costumam globalizar a orientação do movimento comparando-a com o norte magnético e levando em conta a declinação em relação ao norte geográfico.

As duas subseções a seguir mostram técnicas propostas na literatura para um caso base relativo à posição do aparelho e, posteriormente, para casos mais elaborados. É importante notar que solucionar o caso básico é suficiente para diversos sistemas de localização *indoor* que oferecem soluções menos generalizadas.

4.3.1 Caso Base

Um caso mais simples de detecção da orientação ocorre quando o *smartphone* sendo rastreado está nas mãos do usuário, orientado verticalmente e com a tela oposta à palma da mão. Nessa situação, a orientação do aparelho replica a do próprio usuário e a influência do movimento nos sensores é bem definida, portanto, é suficiente basear-se nas medidas do magnetômetro e do acelerômetro do aparelho. Essa estratégia é utilizada em Bello (2015).

[11] http://www.cs.cornell.edu/people/tj/svm_light/svm_multiclass.html

Extraindo-se o vetor da gravidade a partir do acelerômetro, é possível obter o plano ortogonal que representa a tela do *smartphone*. Em posse dessa informação, projeta-se então o vetor tridimensional do norte magnético apontado pelo magnetômetro para tal plano, levando a orientação ao sistema de coordenadas local e explicitando a inclinação do aparelho em relação ao norte magnético.

É importante notar, entretanto, que o vetor da gravidade não é trivialmente obtido a partir das leituras puras do acelerômetro, pois este só fica evidente com o aparelho em repouso e sem a presença de muitos ruídos. Existem sensores mais recentes em *smartphones* que obtém o vetor de gravidade, porém, criar uma dependência em sensores específicos e pouco abrangentes é um ponto negativo para sistemas de localização *indoor* (ANDROID, 2016).

O UnLoc (WANG et al., 2012) apresenta outra estratégia para o problema no caso de uso supracitado. Ao invés de projetar as medidas do magnetômetro, é proposto que o giroscópio seja utilizado diretamente a partir de uma orientação inicial para detectar os movimentos do aparelho em pequenos incrementos. Como os incrementos não são sempre na direção absoluta e sim em coordenadas locais, essa estratégia fornece diversas possibilidades de trajetórias que diferem apenas por um viés θ na direção inicial.

Para combater o viés inicial das trajetórias e estimar a verdadeira dentre elas, o UnLoc aproveita novamente os *landmarks* detectados no sistema. Desde que um usuário saia de um *landmark* L_1, caminhe até um ponto X_2 e seja detectado então em um *landmark* L_2, θ é estimado como o ângulo entre os vetores $\overrightarrow{L_1L_2}$ e $\overrightarrow{L_1X_2}$. Essa estimativa pode também ser atualizada a cada nova detecção do usuário em um *landmark*.

Um caso degenerado da abordagem do UnLoc ocorre no início da localização, quando o usuário ainda não foi detectado em pelo menos dois *landmarks*. Uma heurística utilizada para lidar com a situação consiste em considerar o ângulo do vetor do norte magnético apontado pelo magnetômetro como a orientação do movimento até que se encontre o segundo *landmark*. É importante notar que a abordagem do UnLoc não retorna uma orientação globalizada de movimento.

Para descobrir em quais momentos o magnetômetro pode ser utilizado, por estar livre de interferências, é necessário ter noção do quão confiável é o resultado oferecido pelo sensor. Esse problema encontrado pelo UnLoc, inclusive, foi uma das fontes de inspiração para o desenvolvimento de um módulo de confiabilidade para o *dead reckoning*, apresentado na seção 4.4 deste capítulo.

4.3.2 Generalização

Uma das técnicas explicitadas na subseção passada foi a apresentada no sistema de Bello (2015). Na seção 4.1, também foi mostrada a forma com a qual esse sistema generaliza a contagem de passos nos casos onde os aparelhos se encontram em bolsos laterais de calças. Para generalizar também a determinação de orientação para o caso de *dead reckoning* em bolsos, tal sistema propõe o uso de *Principal Component Analysis* (PCA)[12] sobre as medidas obtidas.

[12] http://setosa.io/ev/principal-component-analysis/

Em adição, para extrair o verdadeiro vetor de gravidade sem depender de sensores mais recentes (ANDROID, 2016), Bello propõe utilizar o giroscópio, que fornece medidas relativas com o tempo, para calibrar os resultados do acelerômetro e do magnetômetro. Nessa abordagem, os vetores de gravidade e do magnetômetro são inicializados e então atualizados gradativamente pelo giroscópio. As medidas são então lidadas pelo PCA.

Processando os vetores obtidos com PCA, obtém-se a direção aparente do movimento. Por fim, foi observado que o sentido do movimento pode ser inferido a partir da análise da média do vetor de aceleração restrito ao último vale de onda do sinal, ou seja, o vetor resultante nessa área do sinal será o mais próximo do sentido correto caminhado pelo usuário

Outra abordagem recente, proposta por Roy et al. (2014) no sistema WalkCompass, parte da ideia inicial do UnLoc de utilizar o giroscópio para estimar a orientação do movimento de usuários. O WalkCompass conta com dois módulos principais separados para a tarefa de detectar a orientação do movimento: Um módulo responsável por estimar a direção em coordenadas locais e outro responsável por globalizá-la.

O primeiro módulo do WalkCompass utiliza o oposto da distância angular obtida através das medições do giroscópio para rotacionar as amostras de sinal extraídas para um sistema de coordenadas estável. As amostras processadas por esse módulo constituem os sinais em um intervalo que se inicia um pouco antes do impacto do calcanhar no chão e termina um pouco depois dele, durante a caminhada do usuário.

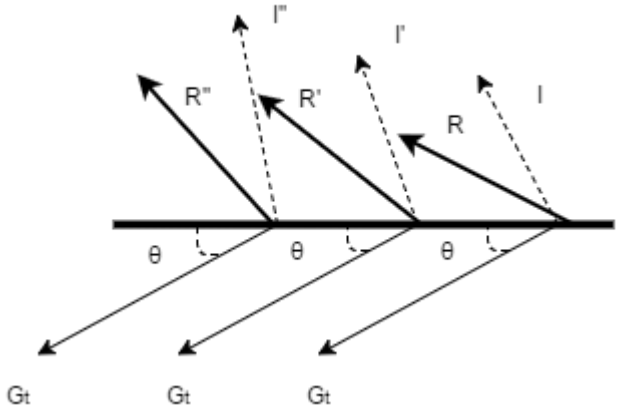
Após obter a direção do movimento em um sistema de coordenadas estável, ela é então projetada no plano ortogonal ao vetor de gravidade, que é por sua vez extraído pelo WalkCompass a partir de um filtro complementar, similar ao apresentado no Capítulo 2 deste trabalho. O vetor de direção em coordenadas locais é processado por mais alguns filtros para suavizar o resultado e é encaminhado consequentemente ao módulo de globalização.

O módulo de globalização do WalkCompass precisa solucionar dois problemas para entregar uma orientação em coordenadas globais: Encontrar o norte magnético verdadeiro e então calcular o ângulo entre o norte verdadeiro e a saída do módulo de orientação local. O segundo problema a ser atacado é trivial e, em um caso idealizado, o primeiro também.

Em casos reais, entretanto, interferências magnéticas tornam a tarefa de encontrar o norte magnético complexa. A partir de experimentos conduzidos por Roy et al., observou-se que usuários andando em um corredor próximo a laboratórios de informática em um edifício universitário estão propensos a erros de até 23° nas medições do magnetômetro. O módulo de globalização do sistema, portanto, foca em melhorias à precisão do magnetômetro.

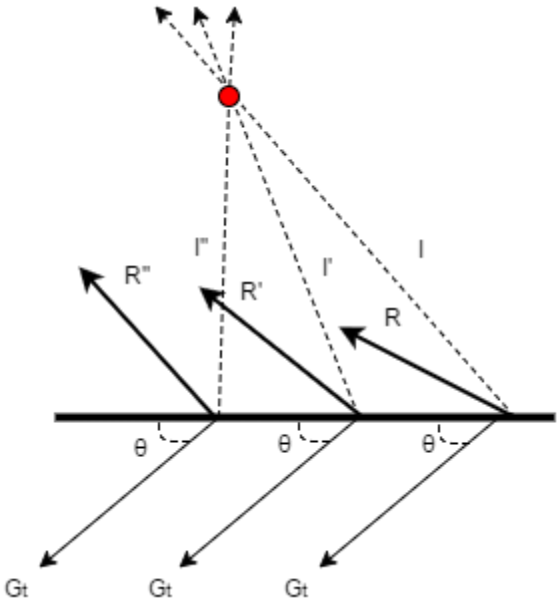
A ideia básica da qual esse módulo do WalkCompass parte é a de que os vetores mensurados pelo magnetômetro podem ser decompostos no vetor do norte magnético verdadeiro e um vetor de interferências. Considerando-se o caso de apenas um gerador de interferência no ambiente, é possível encontrar o vetor de norte magnético fazendo com que os componentes de interferência de sequentes medições do magnetômetro convirjam para um ponto único. Essa ideia é expressada graficamente nas Figuras 9 e 10, e pode ser estendida para mais de um gerador de interferência.

Figura 9 – Decomposição do vetor de norte magnético.



Fonte – Elaborada pelo autor, adaptando de Roy et al. (2014).

Figura 10 – Convergência dos vetores de interferência a partir da escolha de θ.



Fonte – Elaborada pelo autor, adaptando de Roy et al. (2014).

Para encontrar o vetor que gera componentes de interferência convergentes, o WalkCompass itera por diferentes valores de θ para triplas de vetores do magnetômetro. A cada iteração, os vetores de interferência correspondentes são computados. Em seguida, o sistema cria *clusters* de interseções dos vetores e encontra o *cluster* menos rígido, utilizando a soma das distâncias entre pares de pontos nessa região como uma medida de rigidez. Essa abordagem é necessária pois, na prática, os vetores nem sempre intersectam em um mesmo ponto.

A magnitude do vetor estimado é obtida a partir de tabelas de referência de acordo com a localização crua do usuário, baseando-se, por exemplo, no código postal do local onde ele se

encontra. Caso o WalkCompass detecte durante o processo de iteração que o *locus*[13] dos vetores de interferência cria uma forma irregular, o sistema reduz a magnitude do vetor estimado até que seja criado um elipsoide. Esse fenômeno foi observado empiricamente pelos autores.

Finalmente, o WalkCompass calcula o ângulo entre o norte magnético estimado e o vetor de direção em coordenadas locais e o utiliza como a orientação globalizada do movimento do usuário. A orientação obtida em coordenadas globais produz erros médios de apenas 10° em relação à situação real.

4.4 Módulo de Confiabilidade

Ao estudar as técnicas de *pedestrian dead reckoning* propostas na academia e neste trabalho apresentadas, é possível notar que estas enfrentam um problema em comum, relativo às distorções sofridas pelos sensores dos *smartphones*. Especificamente, distorções de *hard iron* e *soft iron* sofridas pelo magnetômetro aparentam ser grandes fontes de dificuldade para as técnicas, como por exemplo no UnLoc, no momento de detecção da orientação enquanto os dois primeiros *landmarks* não são encontrados, e no WalkCompass, na globalização de coordenadas.

Percebendo a dificuldade em conhecer os momentos nos quais é possível confiar nas medições de sensores para realizar o processo de *dead reckoning* e, principalmente, no cálculo da orientação do movimento, esta seção apresenta um módulo de confiabilidade. Apesar de ter sido desenvolvido sobre uma implementação do sistema de localização *indoor* apresentado em Bello (2015), o módulo aqui disposto, seguindo a filosofia do HORUS (YOUSSEF, 2004), é desacoplado e modularizado, podendo ser utilizado por qualquer sistema.

As aplicações de um módulo de confiabilidade são amplas. O módulo pode ser, por exemplo, utilizado por um sistema para detectar momentos nos quais é necessário obter a posição do usuário por outros meios diferentes do *dead reckoning*. Outra possibilidade é a detecção da necessidade de se calibrar algum sensor que por ventura tenha sofrido interferências que o levaram a um estado desregulado.

O processo de desenvolvimento deste módulo passou por diversas iterações, durante as quais foram realizados experimentos com dois aparelhos cujas especificações técnicas são explicitadas na Tabela 2. Ambos os aparelhos funcionam sobre o sistema operacional (SO) Android. As subseções a seguir detalham o funcionamento do módulo de confiabilidade, seus fundamentos, os experimentos realizados tanto para a obtenção de *insights* como para validação do código, e os resultados obtidos.

Tabela 2 – Descrição dos aparelhos utilizados nos testes.

Aparelho	Modelo	Versão do SO	Fabricante
Gran Duos 2	GT-I9082L	4.2.2	Samsung
Zenfone 2	ZE551ML	5.0.0	Asus

Fonte – Elaborada pelo autor.

[13] http://mathworld.wolfram.com/Locus.html

4.4.1 Fundamentos

A primeira tarefa realizada ao se desenvolver o módulo de confiabilidade foi buscar uma métrica que representasse suficientemente bem os momentos durante os quais é possível confiar nas medições dos sensores para o *dead reckoning*. Ou seja, inicialmente buscava-se uma métrica cujos valores fossem proporcionais à confiabilidade dos sensores. Também era desejado focar no giroscópio e no magnetômetro, que se mostraram durante o estudo da literatura como os sensores mais problemáticos.

Após a avaliação de algumas métricas, foi notado que a correlação parecia atender aos requisitos desejados. Nesse ponto, é importante explicitar a inspiração no UnLoc e sua sugestão do uso da correlação entre o magnetômetro e o giroscópio para detectar mudanças no movimento dos usuários. O uso da correlação no UnLoc, entretanto, é apresentado de forma extremamente superficial e pouco analisado nos experimentos, não dando espaço para avaliações externas.

Com o intuito de validar a correlação como uma boa métrica, foram coletados dados do giroscópio e magnetômetro a partir de uma aplicação simples para os aparelhos *Android* descritos na Tabela 2. A fórmula de correlação escolhida foi a de Pearson, mostrada na Equação 11. Apesar de simples, a correlação de Pearson aparentou ser suficiente para o caso de uso do trabalho.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{11}$$

Na Equação 11, x e y são conjuntos de dados separados e suas respectivas médias amostrais são $\bar{x}$ e $\bar{y}$. A coleta de valores foi realizada por um usuário caminhando com o *smartphone* em mãos e o aplicativo aberto durante algum tempo e, posteriormente, aproximando-se de computadores, com o intuito de gerar perturbações magnéticas no aparelho. Os valores coletados foram então salvos em um *log*, particionados, e processados por uma calculadora de correlação disponível *online*[14]. Análises iniciais desses valores confirmaram a capacidade da métrica.

Após a escolha da correlação entre os valores do giroscópio e do magnetômetro como a métrica a ser utilizada, foram definidos os comportamentos esperados pelo módulo a partir da análise dela. Esses comportamentos são apresentados na Tabela 3. Em seguida, foi necessário buscar uma maneira de calcular a correlação de acordo com restrições comumente impostas por um sistema de localização *indoor*.

Tabela 3 – Comportamento desejado da correlação.

Estado / Interferência	Forte Interferência Próxima	Sem Interferência Próxima
Em Movimento	Baixa Correlação	Alta Correlação
Em Repouso	Baixa Correlação	Alta Correlação

Fonte – Elaborada pelo autor.

[14] http://www.socscistatistics.com/tests/pearson/

O principal problema envolvendo o cálculo da correlação para um sistema de localização *indoor* reside no fato de que a maioria destes necessita localizar usuários em tempo real. Isso implica na impossibilidade de pós-processamento de métricas, logo, é crucial que estas sejam calculadas recorrentemente. Em adição, não pode se assumir armazenamento de todos os valores disponíveis para o cálculo das métricas, devido a limitações espaciais dos dispositivos.

A solução encontrada para calcular a correlação em tempo real para ambos os sensores requeridos, sem entrar em conflitos de armazenamento, foi a criação de janelas temporais sobre os sinais. A ideia básica das janelas para os sinais obtidos é analisar apenas um subconjunto diminuto de amostragens de sinal para o cálculo da correlação, sendo esse subconjunto atualizado com novas amostras e o descarte de amostras mais antigas.

Em código, as janelas podem ser representadas trivialmente como filas FIFO de um tamanho fixo w. Após serem totalmente preenchidas, as filas dispensam o seu elemento mais antigo para dar espaço a um novo. Dessa forma, a análise da correlação é realizada de maneira eficiente em termos de tempo e espaço, além de ser mantida local ao invés de global, o que facilita a detecção pontual dos comportamentos descritos na Tabela 3.

Uma janela foi criada para cada sensor analisado, funcionando de maneira independente em ambos os casos. Uma das razões para essa escolha de arquitetura é a existência de diferentes taxas de amostragem para diferentes sensores em diferentes momentos. Em termos simples, um sensor nem sempre irá mensurar um valor ao mesmo tempo que outro. Uma razão extra e provavelmente mais importante foi a necessidade de separar os dois tipos de valores entre os sensores.

Adicionalmente, foi decidido que o módulo de confiabilidade só iniciaria os cálculos após cada sensor providenciar uma quantidade mínima (w) de amostras. Até essa condição ser respeitada, seria retornado o valor *default* 0, que representa dados não correlacionados na fórmula de Pearson. O módulo aqui proposto, portanto, necessita de um curto tempo de inicialização, geralmente menor que um segundo, para poder fornecer boas indicações.

Percebeu-se também que o tamanho w das janelas influenciava principalmente em dois fatores: A granularidade da correlação e o tempo de inicialização do módulo. Valores pequenos de w implicam em cálculos de alta granularidade e curtos períodos de inicialização, enquanto que valores mais altos implicam em cálculos de baixa granularidade e períodos de inicialização mais longos. Experimentos posteriores foram conduzidos para estimar um bom valor para w.

Durante a concepção deste módulo, foi questionada a necessidade de realizar alguma forma de pareamento de amostras dos dois sensores sendo analisados, devido, novamente, às possíveis taxas de amostragem discrepantes. Uma abordagem inicial nas linhas de um algoritmo de *Dynamic Time Warping* (Ann; Keogh, 2016) foi considerada. Experimentos acerca disso, todavia, revelaram que as diferenças na amostragem não são muito relevantes para a correlação, dado que elas não foram muito discrepantes.

Outro ponto essencial no desenvolvimento do módulo foi a conversão dos valores mensurados para orientações em um sistema em comum. Inicialmente, os valores incrementais do eixo Z do giroscópio foram convertidos a partir de integração numérica e consequente somatório dos incrementos, enquanto os valores do magnetômetro foram convertidos projetando-

se o vetor do norte magnético no plano do celular e calculando-se o ângulo para o vetor canônico Y no sistema local.

Em conjunto à criação de janelas para o cálculo da correlação, buscou-se utilizar uma relação de recorrência. A principal razão por esse requisito foi a otimização dos cálculos e do espaço. Como a correlação de Pearson envolve basicamente o cálculo de covariâncias e médias, um primeiro passo seria encontrar relações para essas duas métricas. A primeira tentativa foi utilizar a relação de recorrência de Welford (1962), apresentada no conjunto de Equações 12, 13 e 14. Ela destacou-se por ser numericamente estável e calcular tanto a média (M_k) como a variância (Var_k).

$$\mathrm{M_k} = \mathrm{M_{k-1}} + \frac{(\mathrm{x_k} - \mathrm{M_{k-1}})}{k} \tag{12}$$

$$S_k = S_{k-1} + (x_k - M_{k-1})(x_k - M_k) \tag{13}$$

$$Var_k = \frac{S_k}{k-1} \tag{14}$$

A relação de Welford, entretanto, sofre de uma grave restrição para o caso de uso do módulo; Ela lida apenas com sucessivos incrementos de valores, enquanto que as janelas temporais constantemente descartam valores antigos. Por essa razão, outra solução precisou ser avaliada. A abordagem encontrada partiu da expansão da fórmula da variância populacional, mostrada na Equação 15.

$$Var(X) = \left(\sum_{i=1}^{n} x_i^2\right) - \left(2\bar{x}\sum_{i=1}^{n} x_i\right) + n\bar{x}^2 \tag{15}$$

Na Equação 15, percebe-se que é possível calcular a variância armazenando-se apenas algumas métricas mais simples recorrentemente, sendo elas o somatório dos valores, o somatório dos quadrados dos valores e a média. A média, no caso, é facilmente calculada utilizando a soma dos valores, que também é armazenada. Foram criadas estruturas, portanto, para armazenar esses valores e atualiza-los a cada nova iteração das janelas.

Não se atingiu o objetivo, porém, de realizar o cálculo da correlação completamente em tempo constante. Mesmo com as variâncias e médias sendo calculadas em O(1), sucessivas tentativas de expandir e transformar a fórmula da correlação de Pearson para evitar cálculo em tempo linear falharam, devido à presença de um termo dependente de ambos os sensores. Foi necessário, portanto, armazenar também os valores nas janelas, e não só suas métricas.

Por fim, foi utilizado o módulo da correlação ao invés do valor puro. Enquanto que os valores puros poderiam indicar uma correlação em sentido inverso, esta não era uma informação muito relevante para o caso de uso do módulo. A razão para isso pode ser explicada a partir de um exemplo: Caso o magnetômetro apontasse para o norte geográfico e o usuário girasse o *smartphone* em 45° no sentido anti-horário, o giroscópio iria acompanhar a rotação, porém, o magnetômetro iria variar no sentido horário, implicando em uma correlação negativa indesejada.

Após as decisões de lógica e arquitetura tomadas acima, foi implementada uma versão inicial do módulo. Experimentos básicos foram realizados, e logo percebeu-se que os dados puros de ambos os sensores estavam causando certos comportamentos indesejados. Diversas abordagens, apresentadas na próxima subseção, foram exploradas para incrementar os resultados obtidos até então.

4.4.2 Modificações Posteriores

A primeira etapa para incrementar o módulo foi avaliar as fontes de instabilidade dos sensores, que estavam causando resultados imprecisos nas análises da correlação. Além das interferências magnéticas no ambiente, percebeu-se que o magnetômetro providenciava resultados muito ruidosos devido também à imprecisão do próprio sensor. Essa instabilidade do magnetômetro, porém, era esperada e até desejada. No caso do giroscópio, o principal culpado era o *drift*.

O *drift* do giroscópio ocorre com adições e subtrações de um valor inicialmente constante a cada amostra obtida, variando com o tempo de acordo com fatores externos tais como a temperatura do giroscópio. Experimentos realizados durante o desenvolvimento do módulo mostraram, em ambos os aparelhos, que o giroscópio em repouso mensurava valores entre $\pm d$ até $\pm 5d$, sendo d a constante inicial de *drift*.

Dado que os valores do giroscópio estavam sendo integrados numericamente e então acumulados, o *drift* ocasionava uma tendência ao infinito no acumulador ou, caso fosse utilizado o valor módulo 2π, um padrão cíclico. Portanto, foi necessário buscar estratégias para lidar com o *drift* do giroscópio.

A abordagem clássica para lidar com o *drift* seria aplicar um filtro passa-baixa nos sinais do giroscópio para descobrir o valor e então subtraí-lo de leituras subsequentes. O problema dessa abordagem, contudo, é a dependência na permanência do celular em repouso por pelo menos algum tempo no início do módulo, o que é uma restrição muito forte para muitos sistemas de localização *indoor*. Além disso, essa abordagem desconsidera a variação do *drift* com fatores externos ao decorrer do tempo.

A estratégia escolhida para lidar com o *drift* neste módulo foi a aplicação de um filtro passa-alta nos sinais do giroscópio. Devido ao decaimento dos valores em um filtro passa-alta, a tendência a 0 em casos de sinais constantes, e mudanças de sinal matemático indesejadas quando a variância do sinal é muito alta, foi necessário aplicar o filtro também aos sinais do magnetômetro, para que ele obedecesse ao mesmo comportamento. Apesar de não eliminar o *drift* por completo, a redução foi suficiente para causar uma grande melhoria nos resultados.

Para a escolha do valor α utilizado no filtro, foi levado em conta que quanto menor o valor, maior é a variância necessária para estimular o filtro. Inicialmente, era esperado que valores mais próximos de 1 desempenhassem melhor. A partir de experimentos, entretanto, descobriu-se que a faixa entre 0.2 e 0.3 apresentava um bom balanceamento para o módulo.

Após o valor de α ser estimado, foi calculada uma constante de tempo τ levando em conta um intervalo médio de 0.01s, baseando-se nas taxas de amostragem indicadas pelos sensores dos aparelhos analisados. O valor de α passou a ser calculado a cada etapa dos filtros de acordo com

a constante de tempo e o intervalo entre a amostra atual e a anterior, possibilitando um maior grau de independência do filtro em relação ao aparelho utilizado.

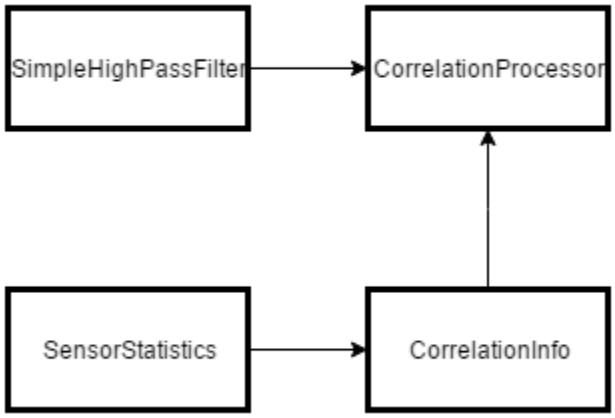
Um ponto adicional de ruído nos cálculos era a integração numérica realizada sobre os valores do giroscópio. A integração numérica foi eliminada do processo e os valores de velocidade angular do giroscópio passaram a ser comparados diretamente com a derivada dos valores do magnetômetro. Essa abordagem reduziu ainda mais as incertezas ocasionadas por ruído no cálculo da correlação.

Por fim, foi realizada a adição de um tempo máximo de espera pelos sensores. A cada nova amostra recebida por um dos sensores, passou a ser verificado se a diferença de tempo entre essa amostra e a amostra anterior, tanto do mesmo sensor como do outro, era menor que a constante. Caso essa condição não fosse respeitada, o módulo era reinicializado. O intuito dessa adição foi prevenir casos degenerados de sensores defasados. O valor empiricamente escolhido foi de 0.1s.

4.4.3 Estrutura e Desempenho

A estrutura final do módulo é detalhada graficamente no diagrama da Figura 11. Objetivou-se desenvolver o sistema da forma mais esparsa possível, para que este pudesse ser utilizado em quaisquer sistemas de localização *indoor* e que, caso necessário, os autores de tais sistemas pudessem modificar seletas partes de acordo com suas necessidades específicas. O módulo foi desenvolvido em Java para Android, e o código está disponível no Apêndice B deste trabalho, com exceção das classes de utilidade geral.

Figura 11 – Diagrama do módulo de confiabilidade.



Fonte – Elaborada pelo autor.

A classe principal do módulo é a *CorrelationProcessor*, que é responsável por receber as novas amostras do giroscópio e do magnetômetro, unificar os sinais em um sistema de coordenadas em comum, e processa-los com instâncias de *SimpleHighPassFilter*. A classe *CorrelationInfo*, por sua vez, aplica as janelas no domínio do tempo a valores de ambos os sensores e calcula a correlação de acordo com as métricas armazenadas em instâncias de *SensorStatistics*.

O tamanho das janelas temporais para os sensores foi determinado empiricamente como 20. Um tamanho maior de janela implica em uma necessidade de mais tempo de inicialização dos cálculos e em uma degradação maior dos resultados devido ao ruído. Ou seja, janelas grandes possuem mais valores ruidosos que prejudicam os cálculos. Enquanto isso, janelas muito pequenas podem realizar cálculos de muito alta granularidade, o que pode gerar resultados imprecisos devido a *outliers*.

O módulo de confiabilidade funciona em tempo real e demora muito pouco tempo para inicializar os cálculos com o tamanho escolhido para as janelas. O módulo foi testado sobre uma implementação do sistema de localização *indoor* apresentado em Bello (2015), e o impacto no tempo de execução do mesmo foi negligenciável.

Os experimentos finais foram conduzidos de forma similar à dos apresentados no início desta seção, ou seja, tanto com simulações de casos de uso reais como com simulações de picos de interferência magnética. Todos os testes foram conduzidos em um ambiente fechado com forte presença de geradores de interferência, mesmo para os padrões de ambientes *indoor*. A interferência gerada no ambiente era tão forte que apenas o ato de colocar e levantar em 5cm o *smartphone* em uma mesa já alterava as medidas do magnetômetro em até 10°. Por conta disso, pode se considerar que os resultados obtidos são pessimistas.

É importante notar que nenhum conjunto de testes completo foi construído com sensores desregulados, devido à dificuldade de reprodução destes casos, apesar de alguns poucos testes terem sido realizados sob esta condição e obtido resultados satisfatórios. É esperado, porém, que estes sejam casos mais triviais de se detectar pelo módulo, visto que a correlação deveria indicar valores baixos com muito mais frequência, correspondendo ao comportamento desejado.

Ao analisar brevemente os resultados dos experimentos com as novas modificações, ficou claro que uma análise pontual da correlação, apesar de mandatória para muitos sistemas, poderia ser substituída em detrimento de uma análise da distribuição de seus valores. Além de facilitar o entendimento dos resultados, essa abordagem de análise probabilística pode ser utilizada em sistemas de requerimento *near real-time*. Para tal análise, foram definidas faixas de correlação, expressadas na Tabela 4.

Tabela 4 – Faixas de correlação e suas respectivas classificações.

Faixa do Módulo da Correlação (C)	Classificação
$0.0 \leq C \leq 0.3$	Baixa
$0.3 < C < 0.8$	*Fuzzy*
$0.8 \leq C \leq 1.0$	Alta

Fonte – Elaborada pelo autor.

Na Tabela 4, nota-se que uma das faixas definidas pertence à classificação *fuzzy*. Essa classificação significa que os valores nessa faixa pouco informam sobre o real estado da confiabilidade dos sensores quando analisados separadamente, devido aos ruídos não eliminados no sistema. As outras classificações indicam, respectivamente, valores esperados de baixa confiabilidade e de alta confiabilidade

As análises dos experimentos em três casos chave separados são mostrados na Tabela 5. Os casos chave em questão são com o *smartphone* em repouso em cima de uma mesa (1), com o *smartphone* muito mais próximo a um gerador de interferência (2) e com o *smartphone* em um caso de uso normal com movimentos laterais e frontais, simulando caminhadas (3). É importante notar que o uso de uma análise de faixas nesta tabela não desclassifica o uso pontual da correlação como um caso válido para o uso em sistemas de localização *indoor*.

Tabela 5 – Faixas de correlação por caso de uso.

Caso de Uso / Correlação	Baixa Correlação	Alta Correlação	*Fuzzy*
Caso 1	82.8%	0%	17.2%
Caso 2	36.2%	7%	56.8%
Caso 3	12.4%	45.5%	42.1%

Fonte – Elaborada pelo autor.

O caso (1) é classificado majoritariamente como um caso de baixa confiabilidade, diferentemente do esperado, pois a grande presença de interferência magnética no ambiente de testes causava ruído constante no magnetômetro. Com o celular em repouso, não há movimento o qual a correlação possa detectar e apenas o ruído acaba sendo levado em consideração. Repete-se aqui o fato de que o experimento foi realizado em um ambiente restritivo, fornecendo uma análise pessimista. É esperado que esse caso seja classificado corretamente em ambientes menos impeditivos, entretanto, testes adicionais não foram realizados.

Apesar dos resultados inesperados no caso (1), é possível ver que eles ainda dizem uma informação importante sobre a confiabilidade real dos sensores. Em ambientes de muita interferência, se a classificação esperada da correlação é aquela mostrada na Tabela 5, o sistema de localização *indoor* pode simplesmente considerar casos de concentração de baixas correlações como de alta confiabilidade. Isto é, apesar de indicar um valor não esperado, a característica do caso (1) ainda pode ser utilizada para detectar a confiabilidade verdadeira.

Para os casos (2) e (3), nota-se que a distribuição dos valores da correlação pende para o lado esperado, ou seja, alta correlação no caso (3) e baixa correlação no caso (2). Essa tendência, porém, é ofuscada em parte pela grande quantidade de valores na faixa *fuzzy*. Neste ponto, percebeu-se que uma análise dos histogramas normalizados de cada um dos casos poderia ter sido mais reveladora, o que pode ser considerado um defeito nos testes.

Os valores *fuzzy*, todavia, não devem ser analisados como uma métrica principal. Isolando apenas os valores de alta e baixa correlação, obtém-se uma imagem muito mais nítida da capacidade do módulo. Não obstante, a taxa considerável de valores na faixa *fuzzy* e na faixa oposta ao desejado mostram que ainda existem muitas melhorias a serem realizadas sobre este módulo. No geral, entretanto, o módulo cumpre o objetivo de detectar momentos de alta e baixa confiabilidade, de forma eficiente em questões de tempo e espaço e para qualquer sistema de localização *indoor*.

4.5 Considerações Finais

No início deste capítulo foi apresentada uma definição mais formal de *dead reckoning* e, especificamente, o caso de uso dele para pedestres. Em seguida, foi dado um esquema geral do processo e dissertou-se então sobre as três principais partes necessárias: Contagem de passos, determinação do tamanho dos passos e determinação da orientação do movimento. O capítulo foi finalizado com a apresentação de um módulo de confiabilidade, incluindo seu processo de desenvolvimento, estrutura geral e resultados.

O tópico de *dead reckoning* é ativamente pesquisado pela academia e, por isso, espera-se que um estudo sobre os fundamentos e estado da arte dele, tal como o apresentado neste capítulo, sirva de base para novos pesquisadores interessados na área. Adicionalmente, o módulo de confiabilidade aqui fornecido mostra uma aplicação clara dos conceitos da área e detalha um processo de desenvolvimento usual dela. Possíveis melhorias para o módulo são apresentadas nas conclusões deste trabalho.

5. Conclusão

O principal objetivo deste trabalho foi conduzir um estudo sobre técnicas de *dead reckoning* para localização *indoor* de pedestres (PDR). Esse estudo é de grande importância devido ao crescimento da computação ubíqua na sociedade, fato este atrelado também ao aumento do número de dispositivos eletrônicos com capacidades para tal. Uma das principais capacidades necessárias para a computação ubíqua é a localização de pessoas, que, porventura, passam a maior parte de seus dias em ambientes *indoor*.

Em adição ao estudo realizado, foi desenvolvido um módulo de confiabilidade para técnicas de PDR. O desenvolvimento desse módulo foi baseado na percepção de que a informação por ele providenciada seria de extrema utilidade para diversas técnicas estudadas durante o trabalho. O módulo foi desenvolvido de forma modularizada, independente, e é fornecido em código neste trabalho, possibilitando seu uso em qualquer sistema de localização *indoor* interessado.

Inicialmente, foram mostrados conceitos fundamentais para o entendimento do trabalho, referentes a sinais, sensores, e ideias básicas da área. Mais especificamente, discorreu-se sobre os três principais sensores utilizados para técnicas de localização *indoor*: Magnetômetros, giroscópios e acelerômetros. Posteriormente, foi mostrado como processar os valores por eles mensurados a partir de filtros.

Após a apresentação dos conceitos fundamentais, o trabalho prosseguiu com a primeira parte do estudo, referente à área de localização *indoor*. Essa primeira parte serviu como um meio de contextualizar o trabalho e mostrar algumas das principais técnicas da área, dando ao leitor a possibilidade de entender como *dead reckoning* pode ser utilizado por estas para atingir o objetivo de localizar pessoas em ambientes fechados.

Previamente à apresentação de sistemas da área, dissertou-se sobre uma das principais características diferenciadoras destes: A dependência em infraestrutura. Este trabalho apresentou as principais vantagens e desvantagens de criar técnicas de localização independentes de infraestrutura adicional e explicou as razões de apenas técnicas desse estilo serem contabilizadas no capítulo.

Dentre os trabalhos apresentados no estudo sobre localização *indoor*, constam o RADAR, uma das técnicas pioneiras em localização *indoor* baseada em *fingerprinting*, o HORUS, que apresentou diversos incrementos sobre o RADAR, e o UnLoc, um dos primeiros sistemas a abordar o *crowdsourcing* e *dead reckoning* como técnicas de localização *indoor*, sendo estas estudadas até a atualidade. Por fim, foram citadas algumas técnicas do estado da arte.

O trabalho seguiu então com a apresentação de um estudo e análise sobre as formalidades e principais etapas de PDR. O estudo foi dividido em quatro etapas, sendo uma de fundamentos, uma sobre a contagem de passos, outra sobre a determinação do tamanho dos passos e, por fim, uma etapa sobre a determinação da orientação do movimento. Cada etapa foi descrita minuciosamente, com exemplo de técnicas clássicas e do estado da arte.

O capítulo sobre *dead reckoning* é finalizado com a apresentação do módulo de confiabilidade desenvolvido durante o trabalho. Ele é apresentado de maneira linear, seguindo o processo de desenvolvimento desde a percepção de sua importância, sua concepção e

ideias/experimentos inicias até sua versão final, experimentos realizados e resultados obtidos. Espera-se, a partir desta abordagem, fornecer uma perspectiva sobre a condução de projetos na área de localização *indoor*, os tipos de desafios encontrados e possíveis resultados.

O módulo de confiabilidade proposto atingiu resultados satisfatórios, detectando momentos de alta e baixa confiabilidade com certa precisão em boa parte dos casos. Um dos principais desafios encontrados durante o desenvolvimento deste também foi a razão de quedas de precisão nos resultados obtidos: Lidar com a presença constante de ruídos no ambiente. Essa dificuldade foi acentuada pelo contato com uma área até então desconhecida pelo autor, que se interessou por localização *indoor* alguns meses antes da elaboração deste trabalho.

Apesar do ambiente de testes ter sido ainda mais complexo do que um ambiente *indoor* comum, ele serve de exemplo sobre como as diferenças entre espaços abertos e espaços fechados comuns podem afetar técnicas de localização. Por essa razão, foi percebido que os casos e métricas de teste elaborados não foram capazes de expressar toda a complexidade do ambiente. A avaliação de vistas adicionais dos testes e a criação de uma plataforma de métricas seriam incrementos fantásticos ao trabalho, que abririam portas para novas modificações e, possivelmente, possibilitariam resultados melhores dos que foram apresentados.

Ainda há muito espaço para melhorias nesse módulo, além das relativas aos testes. Dentre elas, a principal é encontrar uma maneira mais eficiente de lidar com o ruído dos sensores, respeitando as limitações dos sistemas de localização, visto que as medidas ruidosas foram os fatores mais prejudiciais ao cálculo da correlação. Esta não é uma tarefa trivial, e, como foi mostrado durante o trabalho, é um tópico de constante pesquisa na academia.

Além de eliminar o ruído, outra melhoria a ser realizada no módulo consiste em avaliar o comportamento de outros sensores, como por exemplo o acelerômetro, para o cálculo da correlação. Em relação à correlação, ainda é necessário tornar seu cálculo constante em tempo e avaliar outras fórmulas de correlação para séries temporais mais elaboradas que a de Pearson.

Em geral, o trabalho cumpriu seus objetivos, provendo um estudo detalhado e contextualizado sobre técnicas de PDR e disponibilizando um módulo de confiabilidade para quaisquer dessas técnicas. Como foi dissertado neste trabalho, a localização de pessoas em ambientes *indoor* é de extrema importância para o avanço da computação ubíqua. Portanto, espera-se também que o trabalho, a partir de seus objetivos cumpridos, contribua na divulgação e estímulo a pesquisas na área, principalmente no local de sua publicação.

Apêndice A – Aproximação do Filtro Passa-Alta

Sejam $\hat{S}_i$ o valor desejado após a i-ésima amostra de sinal,

S_i o valor puro dessa amostra,

L_i o valor da amostra processada por um filtro passa-baixa

Temos que:

$$\hat{S}_i = S_i - L_i$$
$$\hat{S}_i = S_i - (\alpha L_{i-1} + (1-\alpha)S_i)$$
$$\hat{S}_i = S_i - \alpha L_{i-1} - (1-\alpha)S_i$$
$$\hat{S}_i = S_i - \alpha L_{i-1} - S_i + \alpha S_i$$
$$\hat{S}_i = -\alpha L_{i-1} + \alpha S_i \equiv \alpha S_i - \alpha L_{i-1}$$

Expandindo consecutivamente os termos de L, obtemos:

$$\hat{S}_i = \alpha S_i - \alpha(\alpha L_{i-2} + (1-\alpha)S_{i-1})$$
$$\hat{S}_i = \alpha S_i - \alpha(\alpha^2 L_{i-3} + \alpha(1-\alpha)S_{i-2} + (1-\alpha)S_{i-1})$$
$$\hat{S}_i = \alpha S_i - \alpha(\alpha^3 L_{i-4} + \alpha^2(1-\alpha)S_{i-3} + \alpha(1-\alpha)S_{i-2} + (1-\alpha)S_{i-1})$$
...
$$\hat{S}_i = \alpha S_i - \alpha \sum_{k=0}^{\infty} \alpha^k (1-\alpha) S_{i-1-k} \quad (1)$$

Isolando o primeiro termo do somatório:

$$\hat{S}_i = \alpha S_i - \alpha\big((1-\alpha)S_{i-1}\big) - \alpha \sum_{k=1}^{\infty} \alpha^k (1-\alpha) S_{i-1-k}$$

$$\hat{S}_i = \alpha S_i - \alpha\big((1-\alpha)S_{i-1}\big) - \alpha\sum_{k=0}^{\infty}\alpha^{k+1}(1-\alpha)S_{i-2-k}$$

$$\hat{S}_i = \alpha S_i - \alpha(S_{i-1} - \alpha S_{i-1}) - \alpha\sum_{k=0}^{\infty}\alpha^{k+1}(1-\alpha)S_{i-2-k}$$

$$\hat{S}_i = \alpha S_i - \alpha S_{i-1} + \left(\alpha^2 S_{i-1} - \alpha\sum_{k=0}^{\infty}\alpha^{k+1}(1-\alpha)S_{i-2-k}\right) \quad (2)$$

Isolando e reorganizando a parte entre parênteses:

$$\alpha^2 S_{i-1} - \alpha\sum_{k=0}^{\infty}\alpha^{k+1}(1-\alpha)S_{i-2-k}$$

$$\equiv \alpha\left(\alpha S_{i-1} - \sum_{k=0}^{\infty}\alpha^{k+1}(1-\alpha)S_{i-2-k}\right)$$

$$\equiv \alpha\left(\alpha S_{i-1} - \alpha\sum_{k=0}^{\infty}\alpha^{k}(1-\alpha)S_{i-2-k}\right) \quad (3)$$

Substituindo (1) em (3):

$$\alpha\left(\alpha S_{i-1} - \alpha\sum_{k=0}^{\infty}\alpha^{k}(1-\alpha)S_{i-2-k}\right) \equiv \alpha\hat{S}_{i-1}$$

Logo, substituindo de volta em (2):

$$\hat{S}_i = \alpha S_i - \alpha S_{i-1} + \alpha\hat{S}_{i-1}$$

$$\hat{S}_i = \alpha\big(S_i - S_{i-1} + \hat{S}_{i-1}\big)$$

■

Apêndice B – Implementação do Módulo de Confiabilidade

Todas as classes estão implementadas em Java SE 7. A implementação de classes de utilidade geral, tais como *Vector3*, *SensorEvent* e *ObjectPool*, não é providenciada.

Código 3 – CorrelationProcessor.java

```
1. /* package and imports */
2.
3. public class CorrelationProcessor {
4.   private static final double MAX_SENSOR_DT = 0.1;
5.   private static final double DEFAULT_DT = 0.01;
6.   private static final double TAU = 2.5e-3;
7.   private static final Vector3 PHONE_Y_VECTOR =
8.       Vector3.getNew(0.0, 1.0, 0.0);
9.   private static final Vector3 PHONE_Z_VECTOR =
10.      Vector3.getNew(0.0, 0.0, 1.0);
11.
12.   private SimpleHighPassFilter mMagFilter;
13.   private SimpleHighPassFilter mGyroFilter;
14.   private CorrelationInfo mCorrelationInfo;
15.
16.   private double mLastGyroTs;
17.   private double mLastMagTs;
18.   private double mGyroAngle;
19.   private double mMagAngle;
20.   private double mLastMagMeasure;
21.
22.   private boolean mMagInitialized;
23.   private boolean mGyroInitialized;
24.
25.   public CorrelationProcessor() {
26.     reset();
27.   }
28.
29.   private void reset() {
30.     mGyroAngle = 0.0;
31.     mMagAngle = 0.0;
32.     mLastMagMeasure = 0.0;
33.     mGyroInitialized = false;
34.     mMagInitialized = false;
35.
36.     mGyroFilter = new SimpleHighPassFilter(TAU);
37.     mMagFilter = new SimpleHighPassFilter(TAU);
38.     mCorrelationInfo = new CorrelationInfo();
39.   }
40.
41.   private void initializeMagnetometer(Vector3 magVector,
42.                                       double magTs){
43.     mLastMagTs = magTs;
44.
45.     Vector3 magVectorProjZ = magVector.proj(PHONE_Z_VECTOR);
```

```
46.     Vector3 north = magVector.diff(magVectorProjZ);
47.
48.     mLastMagMeasure = PHONE_Y_VECTOR.ang(north);
49.     mMagAngle = mLastMagMeasure * 180.0 / Math.PI;
50.     mCorrelationInfo.addMagAngle(mMagAngle);
51.   }
52.
53.   private void processMagnetometer(Vector3 magVector, double magTs) {
54.     double magDt = magTs - mLastMagTs;
55.     double measureDt = magTs - mLastGyroTs;
56.
57.     if (magDt > MAX_SENSOR_DT ||
58.        (mGyroInitialized && measureDt > MAX_SENSOR_DT)) {
59.         reset();
60.         return;
61.     }
62.
63.     mLastMagTs = magTs;
64.
65.     Vector3 magVectorProjZ = magVector.proj(PHONE_Z_VECTOR);
66.     Vector3 north = magVector.diff(magVectorProjZ);
67.
68.     // Prevent NaN results when the time difference
69.     // results in 0.0 due to being extremely low
70.     if (magDt > 0.0) {
71.         double magDelta = PHONE_Y_VECTOR.ang(north) - mLastMagMeasure;
72.         mLastMagMeasure = PHONE_Y_VECTOR.ang(north);
73.         double magOmega = magDelta / magDt;
74.         mMagFilter.receive(magOmega, magDt);
75.         mMagAngle += mMagFilter.getOutput() * magDt * 180.0 / Math.PI;
76.         mCorrelationInfo.addMagAngle(mMagAngle);
77.     }
78.   }
79.
80.   private void initializeGyroscope(Vector3 gyroVector, double gyroTs) {
81.     mLastGyroTs = gyroTs;
82.     mGyroFilter.receive(gyroVector.getZ(), DEFAULT_DT);
83.   }
84.
85.   private void processGyroscope(Vector3 gyroVector, double gyroTs) {
86.     double gyroDt = gyroTs - mLastGyroTs;
87.     double measureDt = gyroTs - mLastMagTs;
88.
89.     if (gyroDt > MAX_SENSOR_DT ||
90.        (mMagInitialized && measureDt > MAX_SENSOR_DT)) {
91.         reset();
92.         return;
93.     }
94.
95.     mLastGyroTs = gyroTs;
96.
97.     mGyroFilter.receive(gyroVector.getZ(), gyroDt);
98.     mGyroAngle += mGyroFilter.getOutput() * gyroDt * 180.0 / Math.PI;
```

```
99.      mCorrelationInfo.addGyroAngle(mGyroAngle);
100.   }
101.
102.   public synchronized void process(final Sensor3Event event) {
103.     switch(event.getType()) {
104.       case Sensor.TYPE_MAGNETIC_FIELD:
105.         Vector3 magVector = Vector3.getNew(event.getX(),
106.                                            event.getY(),
107.                                            event.getZ());
108.         if (mMagInitialized) {
109.           processMagnetometer(magVector, event.getTsSec());
110.         } else {
111.           initializeMagnetometer(magVector, event.getTsSec());
112.           mMagInitialized = true;
113.         }
114.         mCorrelationInfo.updateSampleCorrelation();
115.         magVector.recycle();
116.         break;
117.       case Sensor.TYPE_GYROSCOPE:
118.         Vector3 gyroVector = Vector3.getNew(event.getX(),
119.                                             event.getY(),
120.                                             event.getZ());
121.         if (mGyroInitialized) {
122.           processGyroscope(gyroVector, event.getTsSec());
123.         } else {
124.           initializeGyroscope(gyroVector, event.getTsSec());
125.           mGyroInitialized = true;
126.         }
127.         mCorrelationInfo.updateSampleCorrelation();
128.         gyroVector.recycle();
129.         break;
130.     }
131.   }
132.
133.   public double getCorrelation() {
134.     return mCorrelationInfo.getCorrelation();
135.   }
136. }
```

Código 4 – CorrelationInfo.java

```
1. /* package and imports */
2.
3. public class CorrelationInfo {
4.   private static final int TIME_WINDOW_SIZE = 20;
5.
6.   private SensorStatistics mGyroStatistics;
7.   private SensorStatistics mMagStatistics;
8.
9.   private double mCorrelation;
10.
11.   public CorrelationInfo() {
12.       reset();
```

```
13.   }
14.
15.   public void reset() {
16.     mCorrelation = 0.0;
17.     if (mGyroStatistics != null) {
18.         mGyroStatistics.recycle();
19.     }
20.     mGyroStatistics = SensorStatistics.getNew();
21.     if (mMagStatistics != null) {
22.       mMagStatistics.recycle();
23.     }
24.     mMagStatistics = SensorStatistics.getNew();
25.   }
26.
27.   private void addSensorAngle(double sensorAngle,
28.                               SensorStatistics sensor) {
29.     double offset = 0.0;
30.     if (sensor.getNumSamples() == TIME_WINDOW_SIZE) {
31.         offset = sensor.getAngles().remove();
32.     }
33.
34.     sensor.getAngles().add(sensorAngle);
35.     sensor.setSum(sensor.getSum() + sensorAngle - offset);
36.     sensor.setSquaresSum(sensor.getSquaresSum()
37.                          + (sensorAngle * sensorAngle)
38.                          - (offset * offset));
39.     sensor.setMean(sensor.getSum() / sensor.getNumSamples());
40.   }
41.
42.   public void addGyroAngle(double gyroAngle) {
43.     addSensorAngle(gyroAngle, mGyroStatistics);
44.   }
45.
46.   public void addMagAngle(double magAngle) {
47.     addSensorAngle(magAngle, mMagStatistics);
48.   }
49.
50.   public void updateSampleCorrelation() {
51.     if (!isInitialized()) {
52.       return;
53.     }
54.
55.     double nominator = 0.0;
56.     Iterator<Double> gyroIterator = mGyroStatistics.getAngles()
57.         .iterator();
58.     Iterator<Double> magIterator = mMagStatistics.getAngles()
59.         .iterator();
60.
61.     while (gyroIterator.hasNext() && magIterator.hasNext()) {
62.       nominator += (gyroIterator.next() - mGyroStatistics.getMean()) *
63.           (magIterator.next() - mMagStatistics.getMean());
64.     }
65.
```

```
66.     mCorrelation = nominator
67.         / Math.sqrt(getVariance(mGyroStatistics)
68.         * getVariance(mMagStatistics));
69.   }
70.
71.   public boolean isInitialized() {
72.     return mGyroStatistics.getNumSamples() == TIME_WINDOW_SIZE
73.         && mMagStatistics.getNumSamples() == TIME_WINDOW_SIZE;
74.   }
75.
76.   public double getCorrelation() {
77.     return mCorrelation;
78.   }
79.
80.   private double getVariance(SensorStatistics sensor) {
81.     return sensor.getSquaresSum()
82.         - 2 * sensor.getMean() * sensor.getSum()
83.         + sensor.getNumSamples() * sensor.getMean() * sensor.getMean();
84.   }
```

Código 5 – SensorStatistics.java

```
1. /* package and imports */
2.
3. public class SensorStatistics {
4.   private double mSquaresSum;
5.   private double mSum;
6.   private double mMean;
7.   private Queue<Double> mAngles;
8.
9.   protected SensorStatistics() {
10.     reset();
11.   }
12.
13.   public void recycle() {
14.     SensorStatisticsPool.getInstance().put(this);
15.   }
16.
17.   public static SensorStatistics getNew() {
18.     SensorStatistics sensorStatistics = SensorStatisticsPool
19.         .getInstance().get();
20.     sensorStatistics.reset();
21.     return sensorStatistics;
22.   }
23.
24.   public void reset() {
25.     mMean = 0.0;
26.     mSum = 0.0;
27.     mSquaresSum = 0.0;
28.     mAngles = new LinkedList<>();
29.   }
30.
31.   public double getMean() {
```

```
32.     return mMean;
33.   }
34.
35.   public Queue<Double> getAngles() {
36.     return mAngles;
37.   }
38.
39.   public long getNumSamples() {
40.     return mAngles.size();
41.   }
42.
43.   public void setMean(double mean) {
44.     mMean = mean;
45.   }
46.
47.   public double getSquaresSum() {
48.     return mSquaresSum;
49.   }
50.
51.   public void setSquaresSum(final double mSumSquared) {
52.     this.mSquaresSum = mSumSquared;
53.   }
54.
55.   public double getSum() {
56.     return mSum;
57.   }
58.
59.   public void setSum(final double mSum) {
60.     this.mSum = mSum;
61.   }
62. }
63.
64. final class SensorStatisticsPool {
65.   private static final int SIZE = 200;
66.   private static ObjectPool<SensorStatistics> sBuffer = null;
67.
68.   // Suppress constructor to ensure noninstantiability
69.   private SensorStatisticsPool() {
70.     throw new AssertionError();
71.   }
72.
73.   static ObjectPool<SensorStatistics> getInstance() {
74.     if (sBuffer == null) {
75.       init();
76.     }
77.     return sBuffer;
78.   }
79.
80.   static synchronized void init() {
81.     if (sBuffer != null) {
82.       return;
83.     }
84.     sBuffer = new ObjectPool<>(new Generator<SensorStatistics>() {
```

```
85.         @Override
86.         public SensorStatistics getNew() {
87.             return new SensorStatistics();
88.         }
89.     }, SIZE);
90.   }
91. }
```

Código 6 – SimpleHighPassFilter.java

```
1. /* package and imports */
2.
3. public class SimpleHighPassFilter {
4.   private double mOutput;
5.   private double mLastInput;
6.   private double mTimeConstant;
7.   private boolean mInitialized;
8.
9.   public SimpleHighPassFilter(double timeConstant) {
10.    mOutput = 0.0;
11.    mLastInput = 0.0;
12.    mTimeConstant = timeConstant;
13.    mInitialized = false;
14.  }
15.
16.  public void receive(double v, double dt) {
17.    if (!mInitialized) {
18.      mOutput = v;
19.      mLastInput = v;
20.      mInitialized = true;
21.      return;
22.    }
23.    double alpha = mTimeConstant / (mTimeConstant + dt);
24.    mOutput = alpha * (mOutput + v - mLastInput);
25.    mLastInput = v;
26.  }
27.
28.  public double getOutput() {
29.    return mOutput;
30.  }
31. }
```

Referências Bibliográficas

AHMED, Dina Bousdar; DIAZ, Estefania Munoz; KAISER, Susanna. **Performance comparison of foot- and pocket-mounted inertial navigation systems**. 2016 International Conference On Indoor Positioning And Indoor Navigation (ipin), [s.l.], out. 2016. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/ipin.2016.7743673.

ALZANTOT, Moustafa; YOUSSEF, Moustafa**. UPTIME: Ubiquitous pedestrian tracking using mobile phones**. 2012 Ieee Wireless Communications And Networking Conference (wcnc), [s.l.], abr. 2012. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/wcnc.2012.6214359.

ANDROID, **Motion Sensors**, 2002. Disponível em: <https://developer.android.com/guide/topics/sensors/sensors_motion.html#sensors-motion-accel>. Acesso em 08 jul. 2016.

ANN, Chotirat; KEOGH, Ratanamahatana Eamonn. **Everything you know about Dynamic Time Warping is Wrong**. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.73.3623>. Acesso em: 27 nov. 2016.

BAHL, P.; PADMANABHAN, V.n.. **RADAR: an in-building RF-based user location and tracking system**. Proceedings Ieee Infocom 2000. Conference On Computer Communications. Nineteenth Annual Joint Conference Of The Ieee Computer And Communications Societies (cat. No.00ch37064), [s.l.], v. 2, p.775-784, 2000. Institute of Electrical & Electronics Engineers (IEEE). http://dx.doi.org/10.1109/infcom.2000.832252.

BEAUREGARD, Stéphane; HAAS, Harald. **Pedestrian dead reckoning: A basis for personal positioning**. Proceedings Of The 3rd Workshop On Positioning, Navigation And Communication (wpnc'06), [s.l.], jan. 2006.

BELLO, P. A.. **Localização Indoor: Dos Princípios Ao Crowdsourcing**, Dissertação (Mestrado) - Curso de Ciência da Computação, Centro de Informática, Universidade Federal de Pernambuco, Recife, 2015.

BYLEMANS, Inge; WEYN, Maarten; KLEPAL, Martin. **Mobile Phone-Based Displacement Estimation for Opportunistic Localisation Systems**. 2009 Third International Conference On Mobile Ubiquitous Computing, Systems, Services And Technologies, [s.l.], out. 2009. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/ubicomm.2009.23.

CONSTANDACHE, Ionut; CHOUDHURY, Romit Roy; RHEE, Injong. **Towards Mobile Phone Localization without War-Driving**. 2010 Proceedings Ieee Infocom, [s.l.], mar. 2010. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/infcom.2010.5462058.

DAI, Wenhan; SHEN, Yuan; WIN, Moe Z.. **Distributed Power Allocation for Cooperative Wireless Network Localization**. Ieee Journal On Selected Areas In Communications, [s.l.], v. 33, n. 1, p.28-40, jan. 2015. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/jsac.2014.2369631.

ELBAKLY, Rizanne; YOUSSEF, Moustafa. **A Robust Zero-Calibration RF-based Localization System for Realistic Environments**. SECON 2016, [s. L.], maio 2016.

ENGE, P.; MISRA, P**.. Special Issue on Global Positioning System**. Proceedings Of The Ieee, [s.l.], v. 87, n. 1, p.3-15, jan. 1999. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/jproc.1999.736338.

FAURE, François et al. **Dynamic analysis of human walking**. Eurographics, [s.l.], p.53-65, 1997. Springer Science + Business Media. http://dx.doi.org/10.1007/978-3-7091-6874-5_4.

FRIEDMAN, A., **1.2 billion smartphones sold in 2014, slowdown in growth seen for 2015**, PhoneArena. 2015. Disponível em: <http://www.phonearena.com/news/1.2-billionsmartphones-sold-in-2014-slowdown-in-growth-seen-for-2015_id66085>. Acesso em 19 jul. 2016.

HIGGINS, W.. **A Comparison of Complementary and Kalman Filtering**. Ieee Trans. Aerosp. Electron. Syst., [s.l.], v. -11, n. 3, p.321-325, maio 1975. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/taes.1975.308081.

JAHN, Jasper et al**. Comparison and evaluation of acceleration based step length estimators for handheld devices**. 2010 International Conference On Indoor Positioning And Indoor Navigation, [s.l.], set. 2010. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/ipin.2010.5646888.

KAMMOUN, Soufien; POTHIN, Jean-baptiste; COUSIN, Jean-christophe**. An efficient fuzzy logic step detection algorithm for unconstrained smartphones**. 2015 Ieee 26th Annual International Symposium On Personal, Indoor, And Mobile Radio Communications (pimrc), [s.l.], ago. 2015. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/pimrc.2015.7343646.

LADETTO, Quentin. **On Foot Navigation: Continuous Step Calibration Using Both Complementary Recursive Prediction and Adaptive Kalman Filtering**., 2000. Disponível em: <https://www.researchgate.net/publication/37409574_On_foot_navigation_Continuous_step_calibration_using_both_complementary_recursive_prediction_and_adaptive_Kalman_filtering>. Acesso em: 27 nov. 2016.

LADETTO, Q.; MERMINOD, B.. **Digital Magnetic Compass and Gyroscope Integration for Pedestrian Navigation**. 9th International Conference on Integrated Navigation Systems, p. 27 – 29, St. Petersburg, 2002.

LYMBEROPOULOS, D. et al. **A realistic evaluation and comparison of indoor location technologies**. Proceedings Of The 14th International Conference On Information Processing In Sensor Networks - Ipsn '15, [s.l.], p.178-189, 2015. Association for Computing Machinery (ACM). http://dx.doi.org/10.1145/2737095.2737726.

MAUTZ, Rainer. **Indoor Positioning Technologies**. 2012. 128 f. Tese (Habilitação) - Curso de Geodesy And Photogrammetry, Department Of Civil, Environmental And Geomatic Engineering, Eth Zurich, Zurique, 2012.

MIRANDA, E. R.; WANDERLEY, M. M.; KIRK, Ross, **New Digital Musical Instruments: Control and Interaction Beyond the Keyboard (Computer Music and Digital Audio Series)**, Wisconsin: A-R Editions, 2006. 295 p.

MITRA, Abhijit. **Lecture Notes on Mobile Communication**. 2009. Disponível em: <http://www.iitg.ernet.in/scifac/qip/public_html/cd_cell/EC632.pdf>. Acesso em: 07 nov. 2016.

NRC. **The global positioning system: a shared national asset: recommendations for technical improvements and enhancements**. National Academies Press. p. 16, Washington, 2014.

ÖBERG, Tommy; KARSZNIA, Alek; ÖBERG, Kurt. **Basic gait parameters: Reference data for normal subjects**, 10-79 years of age. Journal Of Rehabilitation Research, Uppsala, v. 30, n. 2, p.210-223, 1993.

PRIYANTHA, Nissanka B.; CHAKRABORTY, Anit; BALAKRISHNAN, Hari. **The Cricket location-support system**. Proceedings Of The 6th Annual International Conference On Mobile Computing And Networking - Mobicom '00, [s.l.], p.32-43, 2000. Association for Computing Machinery (ACM). http://dx.doi.org/10.1145/345910.345917.

QUINTANA, B. et al. **Door detection in 3D colored laser scans for autonomous indoor navigation**. 2016 International Conference On Indoor Positioning And Indoor Navigation (ipin), [s.l.], out. 2016. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/ipin.2016.7743677.

RAJAGOPAL, Niranjini; ROWE, Anthony; SINOPOLI, Bruno. **Beacon Placement for Range-Based Indoor Localization**. The 7th International Conference On Indoor Positioning And Indoor Navigation: IPIN 2016, Alcala de Henares, out. 2016.

ROY, Nirupam; WANG, He; CHOUDHURY, Romit Roy. **I am a smartphone and i can tell my user's walking direction**. Proceedings Of The 12th Annual International Conference On Mobile Systems, Applications, And Services - Mobisys '14, [s.l.], p.329-342, 2014. Association for Computing Machinery (ACM). http://dx.doi.org/10.1145/2594368.2594392.

SONG, Jinseon et al. **An improved RSSI of geomagnetic field-based indoor positioning method involving efficient database generation by building materials**. 2016 International Conference On Indoor Positioning And Indoor Navigation (ipin), [s.l.], out. 2016. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/ipin.2016.7743605.

SUSI, Melania; RENAUDIN, Valérie; LACHAPELLE, Gérard. **Motion Mode Recognition and Step Detection Algorithms for Mobile Phone Users**. Sensors, [s.l.], v. 13, n. 2, p.1539-1562, 24 jan. 2013. MDPI AG. http://dx.doi.org/10.3390/s130201539.

TEKINAY, S.. **Wireless Geolocation Systems and Services**. Ieee Commun. Mag., [s.l.], v. 36, n. 4, p.28-28, abr. 1998. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/mcom.1998.667408.

TORRES-SOSPEDRA, Joaquin et al. **UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems**. 2014 International Conference On Indoor Positioning And Indoor Navigation (ipin), [s.l.], out. 2014. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/ipin.2014.7275492.

TORRES-SOSPEDRA, Joaquin et al. **UJIIndoorLoc-Mag: A new database for magnetic field-based localization problems**. 2015 International Conference On Indoor Positioning And Indoor Navigation (ipin), [s.l.], out. 2015. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/ipin.2015.7346763.

VAN EXTER, M.. **Noise and Signal Processing**, 2003. Disponível em: <http://home.physics.leidenuniv.nl/~exter/SVR/noise.pdf>. Acesso em: 19 jul. 2016.

WAHDAN, Ahmed; GEORGY, Jacques; NOURELDIN, Aboelmagd. **Three-Dimensional Magnetometer Calibration With Small Space Coverage for Pedestrians**. Ieee Sensors J., [s.l.], v. 15, n. 1, p.598-609, jan. 2015. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/jsen.2014.2348552.

WANG, H.; SEN S.; ELGOHARY, A.; FARID, M.; YOUSSEF, M.; CHOUDHURY R. R., **Unsupervised Indoor Localization**. MobiSys'12 - Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, Ambleside, p. 197 – 210, 2012.

WANT, Roy et al. **The active badge location system**. Acm Transactions On Information Systems, [s.l.], v. 10, n. 1, p.91-102, 2 jan. 1992. Association for Computing Machinery (ACM). http://dx.doi.org/10.1145/128756.128759.

WEISER, M., **The Computer for The 21st Century**, Sigmobile Mob. Comput. Commun. Rev.,vol. 3, n. 3, p. 3 – 11, New York, 1999. Association for Computing Machinery (ACM). http://dx.doi.org/10.1145/329124.329126.

WELFORD, B. P.. **Note on a Method for Calculating Corrected Sums of Squares and Products**. Technometrics, [s.l.], v. 4, n. 3, p.419-420, ago. 1962. JSTOR. http://dx.doi.org/10.2307/1266577.

YOO, Jaehyun; KIM, H. Jin; JOHANSSON, Karl H.. **Mapless indoor localization by trajectory learning from a crowd**. 2016 International Conference On Indoor Positioning And Indoor Navigation (ipin), [s.l], out. 2016. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/ipin.2016.7743685.

YOUSSEF, M., **Horus: A WLAN-Based Indoor Location Determination System**, Tese (Doutorado) - Curso de Computer Science, Department Of Computer Science, University Of Maryland, College Park, 2004