

UNIVERSIDADE ESTADUAL DE PONTA GROSSA
MESTRADO EM COMPUTAÇÃO APLICADA

ROBERSON JUNIOR FERNANDES ALVES

RASTREAMENTO DE *AGROBOTS* EM ESTUFAS AGRÍCOLAS
USANDO MODELOS OCULTOS DE MARKOV:
Comparação do desempenho e da correção dos algoritmos de Viterbi e
Viterbi com janela de observações deslizante

PONTA GROSSA

2015

ROBERSON JUNIOR FERNANDES ALVES

**RASTREAMENTO DE *AGROBOTS* EM ESTUFAS AGRÍCOLAS
USANDO MODELOS OCULTOS DE MARKOV:
Comparação do desempenho e da correção dos algoritmos de Viterbi e
Viterbi com janela de observações deslizando**

Dissertação submetida ao Programa de Pós-Graduação em Computação Aplicada - Área de concentração Computação para Tecnologias em Agricultura - da Universidade Estadual de Ponta Grossa como requisito para a obtenção do título de mestre.

Orientador: Profº Dr. José Carlos Ferreira da Rocha

PONTA GROSSA

2015

Ficha Catalográfica
Elaborada pelo Setor de Tratamento da Informação BICEN/UEPG

A474 Alves, Roberson Junior Fernandes
Rastreamento de agrobots em estufas agrícolas usando modelos ocultos de Markov: comparação do desempenho e da correção dos algoritmos de Viterbi e Viterbi com janela de observações deslizante/ Roberson Junior Fernandes Alves. Ponta Grossa, 2015.
78f.

Dissertação (Mestrado em Computação Aplicada - Área de Concentração: Agricultura), Universidade Estadual de Ponta Grossa.

Orientador: Prof. Dr. José Carlos Ferreira da Rocha.

1. Robôs móveis. 2. Estufas agrícolas. 3. Modelos Ocultos de Markov. I. Rocha, José Carlos Ferreira da. II. Universidade Estadual de Ponta Grossa. Mestrado em Computação Aplicada. III. T.

CDD: 006.3

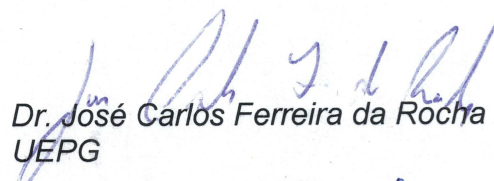
TERMO DE APROVAÇÃO

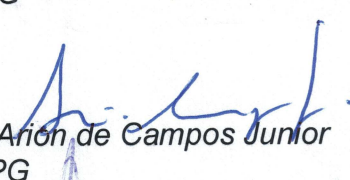
Roberson Junior Fernandes Alves

**“RASTREAMENTO DE AGROBOTS EM ESTUFAS AGRÍCOLAS USANDO
MODELOS OCULTOS DE MARKOV: COMPARAÇÃO DO DESEMPENHO E DA
CORREÇÃO DOS ALGORITMOS DE VITERBI E VITERBI COM JANELA DE
OBSERVAÇÕES DESLIZANTE”**

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre no Programa de Pós-Graduação em Computação Aplicada da Universidade Estadual de Ponta Grossa, pela seguinte banca examinadora:

Orientador:


Dr. José Carlos Ferreira da Rocha
UEPG


Dr. Arion de Campos Junior
UEPG


Dr^a Fabiana Costa de Araujo Schutz
UTFPR

Ponta Grossa, 17 de setembro de 2015.

AGRADECIMENTOS

A Deus, pelo dádiva da vida, sem a qual não estaria concluindo este curso.

A minha esposa e filhos, pelo apoio e carinho nas horas que mais precisei.

À minha família, pelo apoio irrestrito, mesmo nas horas mais difíceis, sempre me incentivando e apoiando de todas as formas possíveis.

Ao meu orientador e amigo Prof. Dr. José Carlos Ferreira da Rocha, pelas contribuições e orientações desta dissertação, mas acima de tudo pela oportunidade dada e o conhecimento compartilhado comigo.

A todos os professores, colegas e amigos da UEPG, os quais compartilharam as alegrias e tristezas durante esta caminhada, que de alguma forma contribuíram para minha formação acadêmica.

RESUMO

O desenvolvimento de *agrobots* móveis e autônomos para operar em estufas agrícolas depende da implementação de procedimentos que permitam o rastreamento do robô no ambiente. O problema do rastreamento pode ser modelado como a determinação da sequência de estados mais prováveis de um modelo oculto de Markov cujos estados indicam posições de uma grade de ocupação. Esta sequência pode ser estimada pelo algoritmo de Viterbi. No entanto, o tempo de processamento e a memória consumida, por esse algoritmo, crescem com as dimensões da grade e com a duração do rastreamento, e isto pode limitar seu uso no rastreamento de *agrobots* em estufas. Considerando o exposto, este trabalho apresenta um procedimento de rastreamento que utiliza implementações aproximadas do algoritmo de Viterbi denominadas de Viterbi-JD(Viterbi com janela deslizante) e Viterbi-JD-MTE(Viterbi com janela deslizante sobre um modelo oculto de Markov com matriz de transição esparsa). Os experimentos mostram que o desempenho de tempo e memória do rastreamento baseado nessas implementações aproximadas é significativamente melhor que aquele do algoritmo original. A hipótese de rastreamento gerada é sub ótima em relação àquela calculada pelo algoritmo original, contudo, não há um aumento substancial do erro. Os experimentos foram realizados utilizando dados simulados de RSSI (Received Signal Strength Indicator).

Palavras-chave: Robôs móveis. Estufas agrícolas. Modelos Ocultos de Markov.

ABSTRACT

Developing mobile and autonomous agrobots for greenhouses requires the use of procedures which allow robot autolocalization and tracking. The tracking problem can be modeled as finding the most likely sequence of states in a hidden Markov model, whose states indicate the positions of an occupancy grid. This sequence can be estimated with Viterbi's algorithm. However, the processing time and consumed memory, of this algorithm, grows with the dimensions of the grid and tracking duration, and, this can constraint its use for tracking agrobots. Considering it, this work presents a tracking procedure which uses two approximated implementations of Viterbi's algorithm called Viterbi-JD(Viterbi's algorithm with a sliding window) and Viterbi-JD-MTE(Viterbi's algorithm with a sliding window over an hidden Markov model with sparse transition matrix). The experimental results show that the time and memory performance of tracking with this two approximated implementations are significantly higher than the Viterbi's based tracking. The reported tracking hypothesis is suboptimal, when compared to the hypothesis generated by Viterbi, but the error does not grows substantially. Th experimentos was performed using RSSI(Received Signal Strength Indicator) simulated data.

Keywords: Mobile robot, Greenhouse, Hidden Markov Models.

LISTA DE ILUSTRAÇÕES

1	Estufas da UEPG	19
2	Representação genérica de uma grade de ocupação	21
3	Processo genérico de um MOM	24
4	Histograma da <i>assinatura</i> de duas células diferentes	31
5	Grade G com 3×3 células	32
6	Grades de ocupação G_1 e G_2	39
7	Caminhamentos gerados para as grade G_1 e G_2 com velocidade de $90cm/s$ e $\gamma = 5$	45
8	Gráficos comparando a memória consumida para cada w e velocidade do Viterbi-JD X Viterbi	54
9	Gráficos comparando o tempo para cada w e velocidade do Viterbi-JD X Viterbi	57
10	Rastreamento com Viterbi-JD na grade G_1 sem ruído: $w \times E_r$	60
11	Rastreamento com Viterbi-JD na grade G_2 sem ruído: $w \times E_r$	60
12	Rastreamento com Viterbi-JD na grade G_1 com ruído: $w \times E_r$	62
13	Rastreamento com Viterbi-JD na grade G_2 com ruído: $w \times E_r$	62
14	Grades de ocupação G_1 e G_2 com a sobreposição dos pontos de predição(PPs)	73
15	AR da grade GD_2	76
16	Dados da simulação de caminhada na gade G_1	78

LISTA DE TABELAS

1	Consumo médio de memória no rastreamento com o Viterbi nas grades GD_1 e GD_2	51
2	Consumo médio de memória para o rastreamento com o Viterbi-JD para grade GD_1	52
3	Consumo médio de memória para o rastreamento com o Viterbi-JD para grade GD_2	52
4	Tempo médio do rastreamento com o Viterbi nas grades GD_1 e GD_2 . . .	54
5	Tempo médio do rastreamento com o Viterbi-JD na grade GD_1	55
6	Tempo médio do rastreamento com o Viterbi-JD na grade GD_2	55
7	Tempo médio do rastreamento com o Viterbi-JD-MTE na grade GD_1 . . .	56
8	Tempo médio do rastreamento com o Viterbi-JD-MTE na grade GD_2 . . .	56
9	Tempos (mínimo e máximo) para o rastreamento com o Viterbi-JD nas grades GD_1 e GD_2	56
10	Correlação resultante para os valores de w e o tempo de processamento para cada velocidade	58
11	Correlação resultante para os valores de w e a memória consumida para cada velocidade	58
12	Taxa de erro(%) no rastreamento com o Viterbi nas grades G_1 e G_2	59
13	Taxa de erro(%) no rastreamento com Viterbi-JD para $\gamma = 3$ na grade G_1 .	59
14	Taxa de erro(%) no rastreamento com Viterbi-JD para $\gamma = 5$ na grade G_1 .	59
15	Taxa de erro(%) no rastreamento com Viterbi-JD para $\gamma = 3$ na grade G_2 .	60
16	Taxa de erro(%) no rastreamento com Viterbi-JD para $\gamma = 5$ na grade G_2 .	60
17	Taxa de erro(%) no rastreamento com o Viterbi nas grades G_1 e G_2	61
18	Taxa de erro(%) no rastreamento com Viterbi-JD para $\gamma = 3$ na grade G_1 .	61

19	Taxa de erro(%) no rastreamento com Viterbi-JD para $\gamma = 5$ na grade G_1 .	61
20	Taxa de erro(%) no rastreamento com Viterbi-JD para $\gamma = 3$ na grade G_2 .	62
21	Taxa de erro(%) no rastreamento com Viterbi-JD para $\gamma = 5$ na grade G_2 .	62
22	Correlação entre os valores de w e a taxa de erro, sem ruído, para cada velocidade e γ	64
23	Correlação entre os valores de w e a taxa de erro, com ruído, para cada velocidade	64

LISTA DE QUADROS

1	Notação formal de um MOM genérico	24
2	Atributos que fazem parte das ARs	42
3	Parâmetros do MOM das grades de ocupação G_1 e G_2	74
4	Parâmetros do MOM das grades de ocupação GD_1 e GD_2	75

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS	17
1.1.1	Objetivo Geral	17
1.1.2	Objetivos Específicos	17
2	REVISÃO BIBLIOGRÁFICA	18
2.1	ESTUFAS AGRÍCOLAS E AGROBOTS	19
2.2	REPRESENTAÇÃO DO ESPAÇO DE NAVEGAÇÃO DE UM ROBÔ USANDO GRADES DE OCUPAÇÃO	21
2.3	MODELOS OCULTOS DE MARKOV	22
2.3.1	Algoritmo de Viterbi	25
2.3.2	Algoritmo de Viterbi com janela deslizante	27
2.4	AUTOLOCALIZAÇÃO PROBABILÍSTICA COM MOM E INFORMA- ÇÕES DE RSSI	29
2.4.1	Rastreamento de robôs móveis com o algoritmo de Viterbi	34
2.5	CONSIDERAÇÕES FINAIS	35
3	METODOLOGIA	36
3.1	RASTREAMENTO USANDO O ALGORITMO DE VITERBI COM JA- NELA DESLIZANTE	36
3.1.1	MOM com Matriz de Transição Esparsa	38
3.2	MATERIAIS E MÉTODOS	38
3.2.1	Infraestrutura: laboratório, dispositivos de hardware e software	39
3.2.2	Assinatura de RSSI	40

3.2.3	Experimentos	42
3.2.3.1	Experimentos sobre o desempenho computacional	43
3.2.3.2	Experimentos de rastreamento	44
3.2.4	Análise dos Resultados	46
3.2.4.1	Análise dos resultados de desempenho	46
3.2.4.2	Análise de resultados de correção	47
3.3	CONSIDERAÇÕES FINAIS	49
4	RESULTADOS	50
4.1	RESULTADOS DOS EXPERIMENTOS DE DESEMPENHO	50
4.1.1	Resultados dos Experimentos sobre Desempenho Computacional: consumo de memória	51
	Análise Estatística	53
4.1.2	Resultados dos Experimentos sobre Desempenho Computacional: tempo de processamento	54
	Análise Estatística	57
4.2	AVALIAÇÃO DA CORREÇÃO DO RASTREAMENTO	58
4.2.1	Resultados do rastreamento para dados de RSSI sem a presença de ruído .	59
4.2.2	Resultados do rastreamento para dados de RSSI com a presença de ruído .	61
4.2.3	Análise dos resultados de erro de rastreamento	63
4.3	CONSIDERAÇÕES FINAIS	64
5	CONCLUSÕES E TRABALHOS FUTUROS	65
	REFERÊNCIAS	66
	APÊNDICE A – GRADES DE OCUPAÇÃO COM ARS PLOTADAS . .	73
	APÊNDICE B – PARÂMETROS DO MOM PARA RASTREAMENTO	74
	APÊNDICE C – PARÂMETROS DOS MOMs PARA EXPERIMEN-	
	TOS DE DESEMPENHO	75

APÊNDICE D - AR GERADA PARA A GRADE GD_2 E $\gamma = 3$	76
APÊNDICE E - GERAÇÃO DE DADOS DE CAMINHAMENTO . . .	77
APÊNDICE F - ARQUIVO COM DADOS DA SIMULAÇÃO DE CA- MINHAMENTO NA GRADE G_1	78

1 INTRODUÇÃO

O desenvolvimento de robôs móveis autônomos para atuar em estufas agrícolas tem visado, principalmente, diminuir a exposição de seres humanos ao ambiente interno da estufa, reduzir os custos de produção e aumentar a produtividade. Uma das funcionalidades associadas aos robôs móveis autônomos envolve a capacidade de navegar e planejar sua locomoção. Essas tarefas dependem das informações fornecidas por um sistema que seja capaz de determinar a posição do robô e rastrear seu deslocamento a partir de dados coletados por sensores.

Os robôs móveis são dispositivos capazes de navegar através de um determinado ambiente de trabalho, dotados de certo nível de autonomia. O desenvolvimento de robôs móveis para a agricultura, os *agrobots* (*Agricultural Robots*), tem procurado criar dispositivos capazes de realizar tarefas que são, normalmente, cansativas ou arriscadas para os seres humanos. Segundo Bergerman *et al.* (2015), Yaghoubi *et al.* (2014), Reynolds (2007) os *agrobots* devem ter a capacidade de comunicar-se com sensores remotos de forma a se transformarem em *agricultores autônomos* com disponibilidade para executar suas atividades durante as 24 horas do dia. Assim, o conceito de autonomia associado a esse tipo de robô se refere à sua capacidade de perceber, modelar, planejar e atuar sem a intervenção humana.

Graças às novas tecnologias de *hardware* e *software*, que permitem reduzir o tamanho e incrementar a capacidade de processamento dos robôs, o uso de *agrobots* se estendem às estufas agrícolas. Contudo, se por um lado, o uso de *agrobots* capazes de se mover livremente permite a criação de dispositivos para realizar tarefas que exijam deslocamentos em estufas agrícolas, seu desenvolvimento requer a implementação de procedimentos para a navegação e localização autônoma. O procedimento de navegação deve prover funcionalidades que permitam que o robô se oriente e se locomova no ambiente sem colidir em obstáculos e sem a intervenção humana. Já a localização autônoma trata mais especificamente da capacidade do robô se localizar no ambiente, seja interno ou externo, dadas as leituras sensoriais disponíveis, sem considerar sua orientação.

Uma forma de abordar o problema da localização é a autolocalização, assim chamada porque o robô implementa um subsistema (*software* e *hardware*) que é capaz de determinar sua posição atual. Em termos práticos, a ideia básica é coletar o sinal de sensores que forneçam dados que estejam, de alguma forma, relacionados à posição do robô no ambiente e então usá-los para estimar a localização correta. Um subproblema associado à autolocalização é o rastreamento de robôs móveis. Isto é, o acompanhamento do desloca-

mento do robô ao longo do tempo. De acordo com Thrun, Burgard e Fox (2005), Nikovski e Nourbakhsh (2002) o rastreamento é uma técnica robusta e oferece bons resultados.

Quando da implementação de sistemas de rastreamento, é necessário prover uma descrição do ambiente de operação em que o robô irá se locomover. Uma abstração frequentemente utilizada é o da grade de ocupação. Uma grade de ocupação representa o ambiente em que o robô irá navegar como uma partição formada por sub-regiões da área de trabalho. As quais são chamadas de células e simbolizam uma posição que o robô pode assumir. Neste esquema, é possível definir o rastreamento como a determinação da sequência de células visitadas pelo robô ao longo do tempo.

No que se refere à coleta de dados, diversos autores tem investigado informações fornecidas por dispositivos de comunicação de rádio como os pontos de acesso de redes sem fio. A informação mais comumente utilizada é o RSSI (*Received Signal Strength Indicator* ou Indicador de Intensidade de Sinal Recebido). O RSSI é uma medida numérica produzida por diversos dispositivos de comunicação de rádio que quantifica a intensidade do sinal de rádio percebido.

As redes sem fio são flexíveis (disponíveis para vários ambientes e configurações), fáceis de serem instaladas, possuem um custo baixo comparadas com outras tecnologias e para atender aplicações específicas podem ser configuradas em diferentes topologias. Esses são alguns dos fatores que motivam seu uso em sistemas de localização e rastreamento. O principal problema é que os dados de RSSI, coletados nessas redes sem fio, podem estar sujeitos a interferências, atenuações e ruídos que afetam os sinais de rádio de forma geral.

Dado o exposto, tem-se que os sistemas que usam o RSSI na localização de robôs precisam adotar técnicas para representar a relação entre os dados e a localização. Uma das técnicas utilizadas é aquela que explora o conceito de *Assinatura de RSSI*¹ (AR). A função da AR é prover um modelo formal para expressar o relacionamento entre várias leituras de RSSI e a localização do robô.

Também deve ser notado que a incerteza associada aos dados de RSSI pode fazer com que as estimativas das posições associadas ao rastreamento sejam incertas. Logo, surge a demanda pelo emprego de técnicas capazes de tratar incerteza quando da realização de inferências sobre o rastro do robô. Um formalismo matemático usado para tanto é aquele definido pelos modelos ocultos de Markov (MOM), no qual o rastreamento de um robô é visto como o monitoramento de um sistema (robô) cujo estado (localização) muda ao longo do tempo. Nesta abordagem, o cômputo do rastreamento equivale à estimação

¹Em inglês, *fingerprinting*. Tradução, impressão digital.

da sequência mais provável de estados, dado um conjunto de observações onde o valor depende do estado. O que pode ser obtido com o algoritmo de Viterbi ().

Assim, de forma resumida, no que se refere o rastreamento de robôs móveis a partir de dados de RSSI, os MOMs permitem:

- representar a incerteza na interação entre a AR e cada localização (célula) da grade de ocupação;
- descrever o deslocamento do robô ao longo do tempo; e
- executar inferências probabilísticas para estimar a hipótese mais provável de seu rastro - algoritmo de Viterbi.

A complexidade de tempo do algoritmo de Viterbi é diretamente proporcional ao número de transições e ao quadrado do número de estados do MOM. Como o número de estados de um MOM que descreve um problema de rastreamento é igual ao número de células da grade de ocupação e como a sequência de dados a serem processados pode ser arbitrariamente longa, o tempo gasto na execução do algoritmo de Viterbi aumenta à medida que o rastreamento prossegue. Além disso, representações mais detalhadas do espaço de navegação (grades com um maior número de células) podem ter um impacto importante sobre o tempo de processamento. Esse comportamento pode ser extremamente limitante do ponto de vista prático quando se considera que um procedimento de navegação para robôs autônomos está sujeito às restrições de um sistema de tempo real.

A quantidade de memória utilizada para armazenar um MOM é influenciada pelo tamanho da matriz de transição, número de estados e parâmetros das distribuições marginais. O tamanho da matriz de transição cresce quadraticamente com o número de estados do MOM, contudo, a matriz de transição usada no modelo para rastreamento tem muitos elementos iguais a zero. Com a informação da velocidade de locomoção do robô se define a quantidade de elementos com valores maiores que zero. Portanto, a esparsidade da matriz pode ser explorada para reduzir a quantidade de memória consumida pelo MOM, já que os elementos com valor zero em uma matriz esparsa são eliminados. Isso pode ser importante, do ponto de vista da implementação de sistemas reais, porque robôs móveis usualmente empregam dispositivos que têm uma quantidade de memória limitada. Além disso, a economia de memória poderia liberar recursos para outros componentes do sistema do robô.

Com base no descrito acima, este trabalho apresenta um estudo sobre uso de três diferentes implementações do algoritmo de Viterbi no rastreamento de *agrobots*. A

primeira implementação é a versão padrão do algoritmo de Viterbi descrita em Viterbi (1967). A segunda, descrita por Jackson (2008), usa o conceito de janelas deslizantes no processamento dos dados com o algoritmo de Viterbi. Na terceira implementação foi testado o uso de matrizes esparsas na representação do MOM com o objetivo de reduzir a memória ocupada pelo MOM. Na terceira implementação foi combinado ainda o uso do Viterbi com janelas deslizantes. A segunda e a terceira versões foram denominadas, respectivamente, de Viterbi-JD e Viterbi-JD-MTE.

Foram realizados experimentos sobre o consumo de memória e o tempo gasto a partir de simulações da tarefa de rastreamento em uma estufa, com o objetivo de comparar as diferenças no desempenho dos seguintes pares de algoritmos:

- a) Viterbi e Viterbi com Janela Deslizante (Viterbi-JD); e
- b) Viterbi e Viterbi com Janela Deslizante sobre um MOM com matriz de transição esparsa (Viterbi-JD-MTE).

Os dados usados nos experimentos de memória e tempo consumidos foram gerados pela simulação de sinais de RSSI aleatórios. Os desempenhos foram comparados usando o teste estatístico t de *Student*.

Também foram realizados experimentos que objetivaram comparar a correção dos algoritmos propostos no rastreamento do robô. Esses experimentos envolveram a criação, por meio de simulação, de alguns caminhos dentro da grade de ocupação. Realizou-se o percorrimto dos caminhos e para cada célula, pertencente ao caminho, armazenou-se o resultado da inferência dos algoritmos de Viterbi e Viterbi-JD. Os dados de RSSI do experimento foram simulados com o procedimento proposto por Narzullaev e Park (2013). Os tempos de computação e a correção do rastreamento de ambos os algoritmos foram comparados com o t de *Student*.

Foi ainda estudada a influência do tamanho da janela deslizante do algoritmo Viterbi-JD sobre o tempo de computação, o consumo de memória e a taxa de erro das hipóteses de rastreamento. Para tanto foi realizada uma análise da correlação linear entre o tamanho da janela e cada uma das medidas de desempenho. O objetivo principal foi verificar a possibilidade de usar o tamanho da janela para explorar uma possível relação de custo $\times$ benefício entre o tempo de computação, consumo de memória e a correção dos resultados, quando do uso do Viterbi-JD no rastreamento.

A seguir, na Seção 1.1, são apresentados os objetivos, gerais e específicos, que orientaram o desenvolvimento do presente trabalho.

1.1 OBJETIVOS

A seguir são descritos os objetivos que nortearam o desenvolvimento da presente dissertação, os quais foram divididos em geral e específicos.

1.1.1 Objetivo Geral

O objetivo geral do trabalho é:

- Avaliar o desempenho computacional e a correção dos algoritmos de Viterbi, Viterbi-JD e Viterbi-JD-MTE no rastreamento de *agrobots* em estufas agrícolas.

1.1.2 Objetivos Específicos

Com a intenção de estabelecer as etapas para atingir o objetivo geral proposto, foram definidos os seguintes objetivos específicos:

- a) Especificar um procedimento para rastreamento e autolocalização de robôs em estufas agrícolas por meio do processamento de dados de RSSI provenientes de pontos de acesso de redes sem fio com modelos ocultos de Markov;
- b) Aplicar o teste t de *Student* para determinar se existe uma diferença significativa, do ponto de vista estatístico, no desempenho dos algoritmos de Viterbi, Viterbi-JD e Viterbi-JD-MTE em termos de tempo de processamento e memória consumida; e
- d) Comparar os resultados referentes as taxas de erro das hipóteses de rastreamento calculadas pelas implementações do algoritmo de Viterbi e Viterbi-JD.

A estrutura do trabalho foi assim organizada: no Capítulo 2 são apresentados os principais conceitos que fundamentam o trabalho; o Capítulo 3 apresenta a metodologia e a estrutura experimental utilizada; no Capítulo 4 são descritos os resultados e contribuições obtidas pelo trabalho; no Capítulo 5 são apresentadas as conclusões e sugerem-se trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

De forma crescente importantes resultados das evoluções tecnológicas e de investigações científicas estão sendo utilizadas na agricultura, especialmente na produção intensiva de culturas em ambientes externos e também em estufas agrícolas (BELFORTE *et al.*, 2006). Dentre estas tecnologias destacam-se o desenvolvimento de *agrobots* com a finalidade de automatizar inúmeras tarefas agrícolas. Para tanto, são investigadas técnicas de autolocalização que permitam aumentar e/ou assegurar a autonomia do robô na navegação em um ambiente interno ou externo.

O desenvolvimento de sistemas de autolocalização para o rastreamento de robôs móveis em estufas agrícolas é uma etapa importante para a construção de *agrobots* capazes de navegar de forma autônoma nestes ambientes (BURGARD *et al.*, 1996). A implementação desse tipo de sistema exige:

- a) a elaboração de uma abstração matemática para descrever o ambiente;
- b) a seleção de um formalismo matemático que permita a representação do problema de rastreamento de robôs móveis;
- c) a definição de um modelo, baseado na técnica de assinatura, para associar as posições do robô com os dados coletados dos sensores;
- d) a utilização de algoritmos para estimar as hipóteses de rastreamento a partir dos dados coletados; e
- e) a especificação dos parâmetros físicos e operacionais da estufa e dos dispositivos utilizados no sistema.

Com o propósito de definir melhor o escopo do problema de pesquisa e a abordagem probabilística proposta, este capítulo faz uma descrição dos quatro primeiros itens da enumeração supracitada e de sua aplicação em estufas agrícolas. Para tanto, a Seção 2.1 descreve o ambiente das estufas agrícolas assim como os problemas específicos para a navegação de *agrobots* nesse ambiente.

A Seção de 2.2 descreve como o ambiente de navegação na estufa agrícola pode ser representado na forma de uma grade de navegação. Em seguida, a Seção 2.3 apresenta uma formulação probabilística da autolocalização e do rastreamento usando modelos ocultos de Markov. No que se refere ao cálculo do rastreamento este capítulo descreve, nas seções 2.3.1 e 2.3.2, os algoritmos de Viterbi e Viterbi-JD.

A Seção 2.4 disserta sobre o uso de dados de RSSI na determinação da localização do robô. Já a subseção 2.4.1 mostra como o problema do rastreamento pode ser especificado por um MOM e como o algoritmo de Viterbi pode ser usado na sua determinação.

2.1 ESTUFAS AGRÍCOLAS E AGROBOTS

As estufas agrícolas são construções que oferecem um ambiente apropriado para abrigar culturas especiais, como hortaliças, flores e certas frutas, entre outras. Podem ser construídas ou fabricadas em uma série de dimensões e formas diferentes. Geralmente são edificadas de plástico ou de vidro translúcido, principalmente pelo seu baixo custo (COSTA, 2003; ACACCIA *et al.*, 1998).

Normalmente possuem uma área retangular e a disposição do ambiente interno da estufa está diretamente associada ao tipo de planta sendo cultivado. Contudo, geralmente as plantas são organizadas em carreiras (ou linhas) paralelas com corredores estreitos as separando. Esta organização facilita a circulação de pessoas e dispositivos ao redor dos carreiras (ACACCIA *et al.*, 1998; MANDOW *et al.*, 1996). A Figura 1 apresenta três estufas agrícolas localizadas no campus de Uvaranas da Universidade Estadual de Ponta Grossa - UEPG - Ponta Grossa (PR).

Figura 1: Estufas da UEPG



Fonte: UEPG (2014).

O cultivo de algumas plantas (tais como frutas e vegetais), em estufas agrícolas, requer uma enorme quantidade de trabalho manual, quando comparado com outras culturas mais extensivas (ACACCIA *et al.*, 1998). Os tipos de trabalhos disponíveis não são atrativos e os resultados em termos de lucros são baixos. Além disso, o trabalho manual realizado em uma estufa agrícola e as condições climáticas internas afetam a saúde hu-

mana, aumentando o risco de doenças (HENTEN *et al.*, 2002). O trabalho dentro da estufa é considerado um dos principais fatores de custo. Assim, esse conjunto de elementos associado ao trabalho dentro da estufa tem impacto direto sobre a qualidade dos produtos e o resultado econômico (BELFORTE *et al.*, 2006).

Essa situação motiva pesquisas que objetivam automatizar as tarefas tediosas e repetitivas no cultivo e colheita de plantas em estufas agrícolas (KAMRAN; MIRASI; SAMADI, 2014; SCHÜTZ *et al.*, 2014; MESTER, 2011; ACACCIA *et al.*, 1998). Um grande número delas têm direcionado seus estudos para o desenvolvimento de aplicações baseadas em *agrobots* (EMMI *et al.*, 2014; CORRELL *et al.*, 2010; MANDOW *et al.*, 1996; EDAN, 1995; TILLET, 1993). Neste sentido, os estudos remontam à década de 1980 quando foram realizadas pesquisas relacionadas ao uso de robôs agrícolas em atividades como colheita de frutos, inspeção de plantas, irrigação, transporte, enxerto, entre outras (ACACCIA *et al.*, 1998; BOURELY *et al.*, 1995; SANDINI *et al.*, 1991; GILBERT; IVAN, 1991). Os resultados obtidos permitiram identificar algumas vantagens associadas ao uso de robôs móveis para realização de tarefas em estufas agrícolas. Dentre elas destacam-se (YAGHOUBI *et al.*, 2014; SAMMONS; FURUKAWA; BULGIN, 2005):

- um robô pode trabalhar durante 24 horas sem parar;
- em tarefas repetitivas é possível adicionar melhorias tecnológicas de forma a qualificar o processo; e
- incremento da eficiência e produtividade.

Duas tarefas frequentemente realizadas pelos *agrobots* móveis são a pulverização e a fertilização (BELFORTE *et al.*, 2006). Na pulverização o robô precisa se locomover nos corredores e pulverizar, uma a uma, as plantas. Na fertilização o robô deposita uma quantidade precisa de fertilizante em cada uma das plantas. Ambas as tarefas, assim como outras atividades realizadas em uma estufa agrícola, exigem que os robôs percorram os corredores entre as plantas organizadas em fileiras. Este é o tipo de deslocamento considerado neste trabalho.

Segundo Liu, Yuan e Wang (2006) e Mandow *et al.* (1996), o desenvolvimento de um *agrobot* capaz de percorrer, autonomamente, as fileiras de uma estufa agrícola, depende da implementação de um sistema para planejar rotas trafegáveis. Isto é, rotas que percorram os caminhos da estufa de forma segura e evitando obstáculos. Um dos principais componentes deste sistema é aquele que determina a localização do robô auxiliando assim na tomada de decisões na tarefa de navegação (POL, 2014).

2.2 REPRESENTAÇÃO DO ESPAÇO DE NAVEGAÇÃO DE UM ROBÔ USANDO GRADES DE OCUPAÇÃO

A determinação da localização de um robô no ambiente, usualmente, objetiva o cômputo de suas coordenadas no espaço em relação a um determinado referencial (GONZALEZ-ARJONA *et al.*, 2013; THRUN; BURGARD; FOX, 2005). Logo, é necessário empregar um formalismo que forneça: (a) uma maneira de descrever o espaço de navegação; e (b) um sistema de coordenadas que permita a representação da posição e dos deslocamentos do robô.

Um formalismo utilizado em aplicações que tratam da locomoção de robôs em ambientes fechados e superfícies planas é o das grades de ocupação. Neste esquema de representação, o espaço de operação é particionado¹ em regiões (MARTIN; MORAVEC, 1996; BORENSTEIN; KOREN, 1991; MORAVEC, 1989; MORAVEC; ELFES, 1985). Desta forma, uma grade de ocupação pode ser abstraída como uma matriz bidimensional $G_{n \times p}$ em que cada elemento g_{ij} representa uma única região, usualmente quadrangular, do espaço de locomoção. As regiões da grade são denominadas de células.

A Figura 2, apresenta uma representação genérica de uma grade de ocupação G com regiões $\{g_{0,0} \dots g_{n,p}\}$. As regiões da grade com o X representam a presença de obstáculos, como paredes, mesas e cadeiras; naquela posição do ambiente. As demais regiões representam espaços livres onde não há presença de obstáculos.

Figura 2: Representação genérica de uma grade de ocupação

(0,0)	(0,1)	(0,2)	...	...	...	(0,i)
(1,0)	X					X
(2,0)		X				
...			X		X	
X						
...		X				
X					X	(n,p)

Nas últimas duas décadas a representação baseada em grades de ocupação tornou-se um paradigma dominante tratando-se de modelagem de ambientes para robótica móvel

¹Os elementos da grade de ocupação representam posições que descrevem o espaço de navegação de forma exaustiva e mutuamente exclusiva.

(THRUN, 2003). Uma das principais razões para isto é que a representação do ambiente por meio de grades de ocupação oferece algumas vantagens, como (ELFES, 1987; ELFES, 1989a; CHEN *et al.*, 2006):

- a) abordagem robusta e unificada para representar o espaço de navegação;
- b) a possibilidade de representação multidimensional do ambiente;
- c) a possibilidade de explorar modelos baseados em multisensores para inferir regiões de espaço que estão ocupadas ou não ocupadas; e
- d) define o rastreamento a partir do histórico de células visitadas.

Assim, as principais motivações para o uso de grades de ocupação no desenvolvimento deste trabalho foram o fato de poderem ser aplicadas em problemas de localização que envolvem o uso de múltiplos sensores e o fato de fornecerem uma abstração simples para o rastreamento. A possibilidade de associar grades de ocupação a procedimentos que combinam dados de multisensores é adequada porque sistemas de localização baseados em RSSI processam, simultaneamente, dados oriundos de vários emissores de sinal de rede. Adicionalmente, a abstração fornecida para o problema de rastreamento, sequência de células visitadas, pode ser vista como uma série temporal, e daí, manipulada por procedimentos baseados em MOMs.

2.3 MODELOS OCULTOS DE MARKOV

Os Modelos Ocultos de Markov (MOMs)² provêem uma ferramenta para modelagem probabilística de problemas que envolvem o processamento de séries temporais. Sua aplicabilidade abrange diversas áreas da inteligência artificial e problemas relacionados ao reconhecimento de fala, bioinformática, biologia molecular e compressão de dados, entre outros (CAPPÉ; MOULINES; RYDEN, 2005; GHAHRAMANI, 2001; DUDEK; JENKIN, 2000). Seu uso na autolocalização e navegação de robôs móveis é motivado principalmente pela possibilidade de utilizá-los para modelar o relacionamento entre dados coletados por sensores (como o RSSI) e o estado de um sistema (a localização de um *agrobot*) quando ambos variam com o tempo.

Segundo Russell e Norvig (2009) e Zucchini e MacDonald (2009), um MOM representa a evolução do estado de um sistema ao longo do tempo usando uma variável

²Em inglês: hidden Markov model (HMM).

discreta S_t associada a intervalos de tempo contíguos. Os valores da variável de estado $s_1 \dots s_N$ são os possíveis estados do sistema, o índice $t = 1..T$ determina o intervalo de tempo discreto a que S_t se refere. A distribuição de probabilidades $\pi(S_1)$ é chamada de distribuição inicial e informa a distribuição marginal *a priori* dos estados do sistema. A probabilidade de uma transição de estados entre o tempo t e o tempo $t + 1$ é descrita pela matriz de transição $A(S_{t+1} | S_t) = a_{i,j}$, em que $a_{i,j}$ informa a probabilidade do sistema ir do estado i para o estado j entre dois intervalos de tempo consecutivos. As probabilidades de transição precisam satisfazer as restrições de $0 \leq a_{ij} \leq 1$, para $1 \leq i, j \leq N$, e $\sum_{j=1}^N a_{ij} = 1$, para $1 \leq i \leq N$. Esse trabalho assume que a matriz de transição não varia em função do tempo, ou seja, assume que o MOM é homogêneo.

A evidência observável, cujo valor depende de S_t , é representada por uma variável aleatória O_t . O relacionamento entre S_t e O_t é quantificado por uma distribuição $B(O_t | S_t)$ de probabilidades condicionais chamada de distribuição de emissão. Cada O_t pode ser representada pelos valores de um conjunto de variáveis definidas por $R = \{r_1, \dots, r_M\}$ de tamanho M . A distribuição de probabilidades, em cada estado, é definida por $B = \{b_{ij}(k)\}$ na qual $b_{ij}(k)$ é a probabilidade do símbolo r_k ser emitido no estado S_j e compor a observação O_i (RABINER, 1989; RABINER; JUANG, 1986). Se existem múltiplas variáveis que representam as observações coletadas em cada estado, a relação entre aquelas e S_t pode ser descrita por uma distribuição/função de densidade multivariada. De acordo com a natureza do domínio de O_t , os MOMs podem ser classificados como discretos ou contínuos (RABINER; JUANG, 1993). Para MOMs discretos, os valores de O_t são símbolos associados a elementos de conjuntos discretos (finitos ou infinitos). No caso de MOMs contínuos, os valores de O_t e pertencem a $\mathbb{R}$.

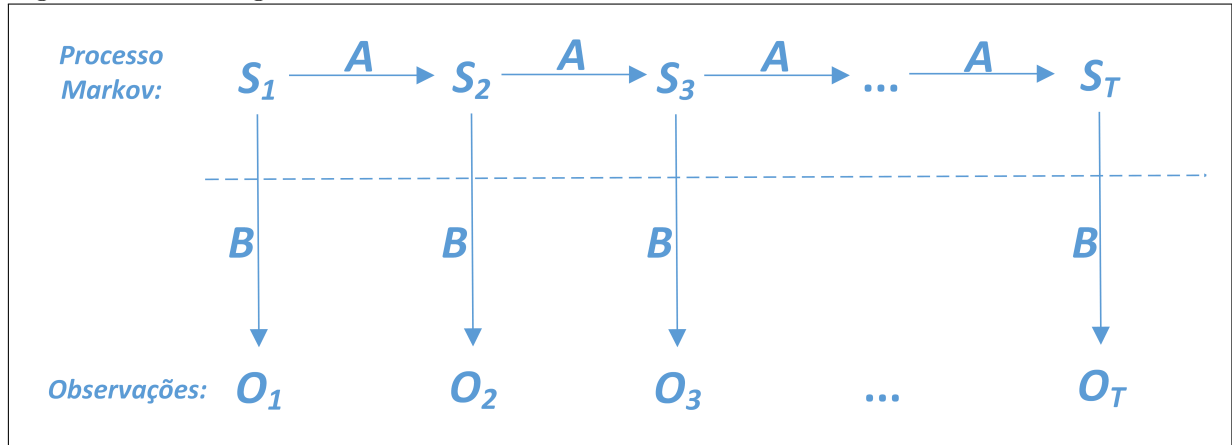
No todo, a estrutura de um MOM pode ser representado como uma tripla $\lambda = (A, B, \pi)$ (RABINER, 1989). O Quadro 1 destaca os elementos de um modelo oculto de Markov (ver Stamp (2004)):

Quadro 1: Notação formal de um MOM genérico

Parâmetro	Variável
Número de estados no modelo.	N
Número de símbolos observados.	M
Diferentes estados do processo de Markov.	S
Distribuição de probabilidades do estado inicial.	π
Distribuição de probabilidades de emissão das observações tal que $b_j(k) = P(O_t = r_k \mid S_t = sj)$.	B
Distribuição de probabilidade da transição do estado S tal que $a_{ij} = P(S_t + 1 = j \mid S_t = i)$.	A
Tamanho da sequência de observações.	T

A Figura 3 ilustra a estrutura de um MOM genérico. Ela mostra os principais elementos do processo de Markov representado pelo MOM. Assim, $S_1, S_2 \dots S_T$ simboliza a sequência dos estados (ocultos) do sistema representado pelo MOM. As variáveis $O_1 \dots, O_T$ indicam os eventos que podem ser observados. Os arcos que conectam S_t a O_t ilustram o fato de que o MOM assume que a t -ésima observação é condicionada diretamente por S_t . A linha pontilhada destaca a separação entre as variáveis ocultas e observáveis.

Figura 3: Processo genérico de um MOM



Fonte: Adaptado de Stamp (2004).

Os dois principais problemas de inferência com MOMs são o problema da avaliação e da decodificação. O primeiro problema, da avaliação, considera que dada uma sequência de observações $\mathbf{O} = o_1, o_2 \dots o_T$ e um MOM $\lambda = (A, B, \pi)$, como é possível computar, com eficiência, a probabilidade desta sequência de observações ser gerada pelo modelo, ou seja, $P(\mathbf{O} \mid \lambda)$. Para solução desse primeiro problema é utilizado o algoritmo

de programação dinâmica *Forward-Backward*. No segundo problema da decodificação, a ideia é estimar a sequência mais provável de estados do sistema, dado um conjunto de observações $\mathbf{O}$. Isto é, tendo-se uma série de observações $\mathbf{O} = o_1, o_2 \dots o_T$ e um MOM $\lambda = (A, B, \pi)$, como é possível encontrar a melhor sequência $S = s_1, s_2 \dots s_T$ de estados para representar as observações (STAMP, 2004; RABINER; JUANG, 1986). Para resolver o problema da decodificação utiliza-se o algoritmo de Viterbi que também é baseado em conceitos de programação dinâmica. A Seção 2.3.1 detalha o funcionamento deste algoritmo.

2.3.1 Algoritmo de Viterbi

Dada uma sequência de eventos observados $\mathbf{O} = o_1, o_2 \dots o_T$ e assumindo que os eventos sejam gerados por um MOM, $\lambda = (A, B, \pi)$, com um espaço de observações O , o algoritmo processa os seguintes passos para calcular sequência mais provável dos estados (SLADE, 2013; ZUCCHINI; MACDONALD, 2009; JURAFSKY; MARTIN, 2006):

- 1) **Inicialização:** neste passo é feita a declaração dos vetores δ e ψ , ambos com o tamanho $N * T$. O vetor δ é utilizado para armazenar as probabilidades de cada s_t e é inicializado com $a_{ij} \times b_{ij}$ para t_1 . O vetor ψ armazena os índices de cada s_t e é inicializado com 0 para t_1 (Equações 1 e 2).

$$\delta_1(j) = a_{ij}b_{ij}, \quad 1 \leq j \leq N \quad (1)$$

$$\psi_1(j) = 0 \quad (2)$$

- 2) **Indução/Recursão:** o segundo passo envolve o percorrimento de todos os estados, de forma recursiva e exaustiva, para cada observação do conjunto (O) iniciando em t_2 . Armazena-se no vetor δ e ψ , respectivamente, o valor da máxima probabilidade e o índice no vetor δ com a máxima probabilidade para um estado s_k . Os cálculos de probabilidade novamente utilizam a matriz de transição, emissão e as probabilidades acumuladas em δ (Equações 3 e 4).

$$\delta_t(k) = \max_{1 \leq j \leq N} [\delta_{t-1}(j)a_{jk}b_{jk}(O_t)], \quad 2 \leq t \leq T \quad e \quad 1 \leq k \leq N \quad (3)$$

$$\psi_t(k) = \arg \max_{1 \leq j \leq N} [\delta_{t-1}(j)a_{jk}], \quad 2 \leq t \leq T \quad e \quad 1 \leq k \leq N \quad (4)$$

- 3) **Finalização:** neste passo é, utilizando o vetor de apoio P^* , para armazenar o índice

do estado, dado o tempo $T - 1$ com a máxima probabilidade e que equivale ao primeiro estado inferido (Equação 5).

$$P_T^* = \arg \max_{1 \leq k \leq N} [\delta_T(k)] \quad (5)$$

- 4) **Reconstrução:** no passo da reconstrução são armazenados os demais estados, observando a máxima probabilidade, em P^* . Inicia-se a reconstrução a partir do tempo $T - 2$ (Equação 5).

$$P_t^* = \{\psi_{t+1}, P_{t+1}^*\}, \quad t = T - 2, T - 3, \dots, 1 \quad (6)$$

No algoritmo 1 é apresentada uma versão do pseudocódigo do Viterbi utilizado no presente trabalho usando uma notação algorítmica.

Algoritmo 1: VITERBI

Entrada: λ, O, T, N

Saída: melhor caminho - P^*

Dados: $P^*[T], \delta[N * T], \psi[N * T]$

inicio

```

para cada  $s \in 1..N$  hacer                                     /* Inicialização */
     $\delta[s, 1] \leftarrow a_{1,s} * b_s(O_1)$ 
     $\psi[s, 1] \leftarrow 0$ 

```

fin

```

para cada  $t \in 2..T$  hacer                                     /* Recursão */

```

```

    para cada  $s' \in 1..N$  hacer
         $\delta[s, t] \leftarrow \max_{s'=1}^N \delta[s', t-1] * a_{s',s} * b_s(O_t)$ 
         $\psi[s, t] \leftarrow \arg \max_{s'=1}^N \psi[s', t-1] * a_{s',s}$ 

```

fin

fin

```

 $P^*[T] \leftarrow \arg \max_{s=1}^N \delta[s, T] * a_{s, s_F}$                                      /* Finalização */

```

```

para cada  $t \in T - 1..1$  hacer                                     /* Reconstrução */

```

```

     $P^*[t] \leftarrow \psi[(t+1) + (P^*[t+1])]$ 

```

fin

fin

retorna P^*

/* $a_{s',s}$ prob. de transição do estado s' para o estado s */

/* $b_s(O_t)$ prob. de emissão do estado s dada a observação O no tempo t */

Como pode ser observado o algoritmo de Viterbi tem uma complexidade de tempo pessimista $O(s^2T)$ (FELZENSZWALB; HUTTENLOCHER; KLEINBERG, 2004). Logo, o tempo

de processamento cresce proporcionalmente ao número de transições e ao quadrado do número de estados previstos no modelo.

Do exposto, tem-se que a influência do número de estados sobre o tempo de computação do algoritmo de Viterbi pode limitar sua utilização em problemas em que a variável de estado tem uma grande cardinalidade e o sistema sob consideração exige processamento em tempo real. No caso de sistemas de tempo real para o rastreamento de robôs móveis (SRáMEK, 2007), a função de complexidade do algoritmo de Viterbi pode inviabilizar seu uso com grades de ocupação que têm um grande número de células. Outro fator a ser considerando é que o rastreamento de robôs pode exigir o processamento de um conjunto de observações que cresce arbitrariamente ao longo do tempo. Segue daí que o processo de coleta de dados fará com que o desempenho do sistema de rastreamento degrade ao longo de sua operação.

O total de memória necessária para o processamento é proporcional ao número de estados (FORNEY G.D., 1973). Assim, a principal dificuldade é que as rotinas de rastreamento são, usualmente, implementadas em sistemas embarcados, neste caso, deve se considerar as limitações impostas pela memória disponível. Se ela não for suficiente para armazenar as observações, a operação do rastreamento pode ser inviabilizada.

2.3.2 Algoritmo de Viterbi com janela deslizante

Uma alternativa para contornar as limitações práticas relacionadas às complexidades de tempo e espaço do algoritmo de Viterbi é o uso de algoritmos aproximados. Uma implementação aproximada do algoritmo de Viterbi é denominada Viterbi com Janela Deslizante³ (Viterbi-JD) (ZHENG *et al.*, 2009; JACKSON, 2008; RYBSKI; VELOSO, 2005; PILU, 2004). A ideia do Vitebi-JD é executar o algoritmo de Viterbi sobre uma subsequência de observações $\mathbf{O}'$ contíguas de tamanho w . Este processo é executado para cada valor de (t) de forma a se obter estimativas da sequência, com respeito a $\mathbf{O}'$ no intervalo de tempo entre $t - w$ até t (FUJIWARA; SAKURAI, 2011).

Como destacado, no algoritmo Viterbi-JD a janela de observações deslizante $\mathbf{O}'$ é um subconjunto do conjunto de observações $\mathbf{O}$ tal que $\mathbf{O}'$ mantém w leituras dos sensores. Isto é, $\mathbf{O}' = \{(r_{t,1}, r_{t,2} \dots r_{t,k}) : k = 1..K, t = (t' - w)..t', t' \in \{1..T\}\}$. À medida que as observações são atualizadas o algoritmo de Viterbi é executado apenas sobre os dados da janela. Portanto, seu uso evita o processamento da sequência completa, com todas de

³Em inglês, *Sliding Vitebi. Abordagem algorítmica para redução de tempo processamento implementado que considera somente uma parte do conjunto de observações* (DATAR *et al.*, 2002)

observações coletados dos sensores, o que reduz o tempo de processamento e faz com que ele se torne uma alternativa para sistemas de tempo real (ZHENG *et al.*, 2009; RYBSKI; VELOSO, 2005).

Existem várias formas de usar janelas deslizantes para o processamento de um MOM. A principal diferença entre elas é a maneira como a janela é analisada. Em uma delas, a janela avança no conjunto de dados e considera observações que estão adiante do índice de tempo sob análise (PILU, 2004). Uma segunda opção, é fazer a janela processar observações anteriores ao estado atual (JACKSON, 2008). A implementação do Viterbi-JD adotada neste trabalho emprega a segunda alternativa e pode ser descrita da seguinte maneira:

- 1) Seleciona-se o tamanho da janela w e determina-se a distribuição de probabilidades inicial (π) para os N estados, utilizando: $\pi_s = \frac{1}{N}$;
- 2) É verificado se o número total de observações T é maior ou menor que a janela w :
 - a) Se T for menor que w , todo conjunto de observações $\mathbf{O}$ é processado pelo algoritmo de Viterbi;
 - b) Se T for maior que w , um subconjunto de observações $\mathbf{O}'$ de tamanho w é processado pelo algoritmo de Viterbi;
- 3) O melhor caminho, $s_1, s_2, \dots, s_w$ é retornado pelo algoritmo de Viterbi e armazenado em memória;
- 4) A janela avança para $w + 1$ enquanto $w < T$ e define-se que $\pi_{s=(T-(w-1))} = 1$ e $\pi_{s \neq (T-(w-1))} = 0$; e
- 5) Os passos 3 e 4 precisam ser repetidos até o algoritmo analisar todo conjunto de observações $\mathbf{O}$.

No algoritmo 2 é apresentado o pseudocódigo do Viterbi-JD.

Algoritmo 2: VITERBI-JD

Entrada: λ, O, T, N, w
Saída: melhor caminho - P^*
Dados: $P^*[T], O', inicial, \pi$
inicio
 enquanto $t < T$ **faça**
 se $t < w$ **então**
 $P^* \leftarrow VITERBI(\lambda, O'[1..t + 1], t + 1, N)$
 senão
 $inicial \leftarrow t - (w - 1)$
 $\pi[i = P^*[inicial + 1]] \leftarrow 1$
 $\pi[i! = P^*[inicial + 1]] \leftarrow 0$
 $P^* \leftarrow VITERBI(\lambda, O'[inicial..t + 1], w, N)$
 fim
 fim
fin
retorna P^*

2.4 AUTOLOCALIZAÇÃO PROBABILÍSTICA COM MOM E INFORMAÇÕES DE RSSI

Se colocados em um ambiente desconhecido, humanos, animais ou mesmo robôs recebem informações incompletas sobre sua localização atual por meio de seus sentidos. No entanto, humanos e animais de forma rápida e precisa são capazes de desenvolver conhecimento sobre sua localização e que direção seguir. Tais habilidades podem ser obtidas de robôs autônomos adicionando-se a eles uma combinação de informações obtidas a partir de sensores específicos (FRANZ; MALLOT, 2000).

A autolocalização, também chamada de localização global, é definida como a capacidade do robô reconhecer sua localização através da determinação de suas coordenadas no ambiente, seja este interno ou externo (THRUN; BURGARD; FOX, 2005; DUCKETT; NEHMZOW, 2001). Dado que a autolocalização objetiva que o robô seja capaz de determinar sua posição, é necessário empregar métodos de inferência sobre informações obtidas a partir de sensores.

Pela sua popularidade, facilidade de implantação e sua composição caracterizar-se de nodos sensores que formam uma grande rede colaborativa, as redes sem fio⁴, tem

⁴Do inglês *Wireless Sensor Networks* (WSN).

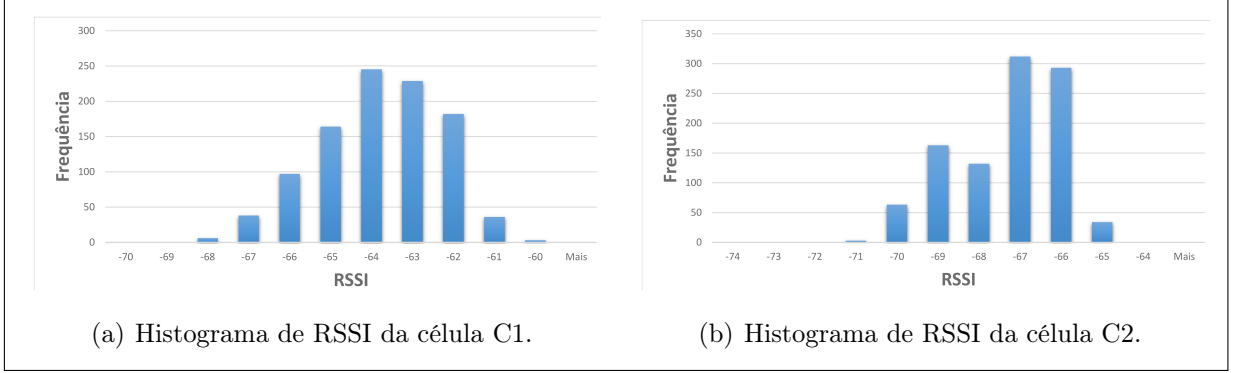
sido exploradas em vários trabalhos como alternativa para a construção ou modelagem de sistemas de autolocalização para robôs móveis (IBRAHIM; YOUSSEF, 2011; PAL, 2010; RAGHAVAN *et al.*, 2010; CABRERA-MORA; XIAO, 2008; RAMADURAI; SICHITIU, 2003). E uma das informações utilizadas na autolocalização de robôs móveis é o RSSI (*Received Signal Strength Indicator*) (LINA; YU; ZHENGQI, 2013; MAHAJAN; CHANANA, 2012; LO; WU; CHUNG, 2007). Esta informação é fornecida por diversos equipamentos de comunicação sem fio, como em redes *WiFi* (*Wireless Fidelity*), *Bluetooth*, *RFID* (*Radio-Frequency Identification*), *ZigBee*, torres de celular, entre outros.

O RSSI é uma medida que fornece uma indicação da intensidade do sinal de rede recebido (IEEE, 2012). Conforme a norma da IEEE 802.11 no caso de redes sem fio *WiFi*, padrão 802.11, o RSSI é a medida da energia observada entre a antena que emite o sinal (emissor) e a antena que recebe o sinal (receptor). Ele mede a potência do sinal recebido e varia entre 0 e R_{max} , sendo que o valor máximo (R_{max}) do sinal é determinado pelo modelo da placa de rede sem fio. A unidade de medida do RSSI é o *decibel miliwatt* - dBm, em sua forma negativa.

As medidas de RSSI podem ser usadas para estimar a localização de um robô em um ambiente, pois o valor coletado está associado à sua distância de cada ponto emissor. Portanto, havendo uma alteração de distância entre receptor e emissor acontece uma variação também no sinal de RSSI, para mais ou para menos. Não existe uma relação direta, documentada, entre o RSSI e a possível distância, ou seja, isso pode variar de modelo de equipamento ou mesmo pelo tipo de rede sem fio utilizada, no entanto, é possível perceber-se um padrão (RAPPAPORT, 2001).

Um esquema de representação usado para associar os valores de RSSI à uma posição s_l é denominado de *assinatura de RSSI* (AR) e sua função é prover um modelo formal para expressar o relacionamento entre os dados coletados e a localização do robô (MILIORIS *et al.*, 2014). As Figuras 4(a) e 4(b) apresentam a *assinatura* das leituras de RSSI de um mesmo ponto de acesso para duas células diferentes. É possível observar que existe uma diferença nas leituras para cada célula o que permite criar uma assinatura do RSSI para cada célula.

Figura 4: Histograma da *assinatura* de duas células diferentes



Pode ser notado que, em adição ao fato de que a relação entre o RSSI e a distância emissor-receptor não ser determinística, os dados de RSSI podem ser permeados por ruído ou mesmo estarem incompletos. Esses fatores podem causar incerteza quanto a correção das leituras e isto pode fazer com que os parâmetros do modelo que relaciona os dados de RSSI com uma certa posição no ambiente também sejam incertos. Portanto, um sistema de rastreamento deve usar procedimentos de inferência capazes de tratar a incerteza que surge em decorrência de ruído nos dados assim como aquela inerente ao processo de modelagem (MILIORIS *et al.*, 2014; LADD *et al.*, 2004).

Uma maneira de lidar com as incertezas inerentes a autolocalização é a adoção de técnicas ou modelos probabilísticos (SCHMITT *et al.*, 2002; OLSON, 2000; THRUN *et al.*, 1998). O emprego desta abordagem envolve a realização de duas tarefas. Primeiro, a definição de uma função de probabilidades para codificar a AR de cada célula. Segundo, a seleção e implementação de um procedimento para realizar inferências probabilísticas.

O emprego da abordagem probabilística na codificação da AR permite usar uma distribuição condicional, da forma $P(R_{k,1} \dots, R_{k,t} | S_k)$, para quantificar a incerteza inerente ao relacionamento entre as leituras de RSSI e cada posição s_l (JEKABSONS; KAIRISH; ZURAVLYOV, 2011). Com isto, e dada a distribuição a priori $P(S_k)$, é possível usar o teorema de Bayes para calcular $P(S_k | R_{k,1} \dots, R_{k,t})$. Em seguida é possível usar a regra de decisão bayesiana para estimar a hipótese mais provável para a posição do robô (TRAN *et al.*, 2008). Assumindo-se que cada leitura de RSSI é independente das demais é possível fazer $P(R_{k,1} \dots, R_{k,t} | S_k) = P(R_{k,1} | S_k) P(R_{k,2} | S_k) \dots P(R_{k,t} | S_k)$ - modelo de sensor de Markov (MILIORIS *et al.*, 2014; HOANG *et al.*, 2013).

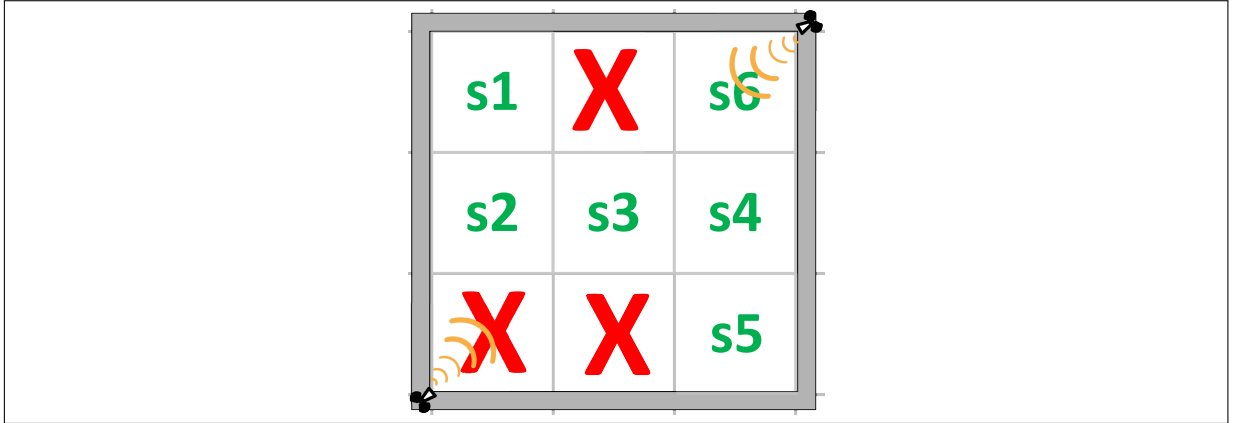
Quando se considera o problema do rastreamento de robôs móveis, o modelo de sensor de Markov pode ser integrado à uma cadeia de Markov de forma a compor um MOM. Essa abordagem para modelar probabilisticamente um sistema de navegação e localização é proposta em vários trabalhos (ARTHI; DEVARAJ; MURUGAN, 2013; HERRMANN;

PAWELZIK; GEISEL, 1999), os quais provêem uma abordagem para modelar e apoiar decisões em problemas de autolocalização pois permitem (IBRAHIM; YOUSSEF, 2011; IBRAHIM; YOUSSEF, 2010; ZHOU; SAKANE, 2008):

1. representar do ambiente utilizando grades de ocupação, em que cada célula corresponde a um estado do modelo;
2. modelar as informações coletadas em sensores;
3. inferir a localização do robô utilizando as informações dos sensores combinada com a representação do ambiente por grades de ocupação; e
4. manter o histórico das inferências passadas como forma de aumentar a precisão de novas inferências.

A Figura 5 ilustra uma forma de combinar a representação baseada em grades de ocupação, MOMs e informações de RSSI coletadas de pontos de acesso *Wifi* para realizar rastreamentos. Nesta figura, a grade de ocupação tem $N = 6$ células livres e um ambiente com 2 PAs de rede sem fio. O MOM associado a esta grade tem seis estados que são denotados por $S_t = \{s_i, \dots, s_6\}$, um para cada célula da grade. Isto é, os estados S_t do MOM correspondem as possíveis localizações do robô (NIKOVSKI; NOURBAKHS, 2002).

Figura 5: Grade G com 3×3 células



A partir da configuração do ambiente são definidos os parâmetros do MOM: matriz de probabilidades inicial (π), matriz de transição (A) e matriz de emissão dos símbolos (B). No exemplo, a matriz de probabilidades π foi definida como uma distribuição uniforme e, portanto, $\pi_{[i]} = \frac{1}{N} = 1/6$. A matriz de transição descreve as regras de evolução dos estados como resultado das ações do robô. No exemplo da Figura 5 A é uma matriz de dimensões 6×6 . Se for assumido que o MOM é homogêneo e que no intervalo de

tempo entre duas observações consecutivas o robô é capaz de deslocar-se para uma célula vizinha, A é uma matriz banda da forma:

$$A = \begin{pmatrix} 0.34 & 0.33 & 0.33 & 0 & 0 & 0 \\ 0.33 & 0.34 & 0.33 & 0 & 0 & 0 \\ 0 & 0.33 & 0.34 & 0.33 & 0 & 0 \\ 0 & 0 & 0.33 & 0.34 & 0.33 & 0 \\ 0 & 0 & 0 & 0.33 & 0.34 & 0.33 \\ 0 & 0 & 0 & 0.33 & 0.33 & 0.34 \end{pmatrix}$$

O robô não pode observar sua localização por inspeção direta, contudo, pode usar os dados de RSSI para estimá-la. Para tanto, basta associar cada S_t a um modelo sensor de Markov que o relacione com as leituras de RSSI. Em seguida, basta definir a matriz de emissão de forma que o sensor de Markov possa ser usado para expressar em termos probabilísticos a AR. Neste trabalho, a AR de cada célula foi representada por uma distribuição gaussiana $\mathcal{N}(\mu_{j(k)}, \sigma_{j(k)})$, em que $\mu_{j(k)}$ é a média de um conjunto de leituras do RSSI de cada ponto de acesso coletadas em cada célula (assinatura de RSSI). $\sigma_{j(k)}$ é o desvio padrão dos valores de RSSI (NARZULLAEV; PARK, 2013).

Dados esses parâmetros como entradas para o MOM, conforme o robô se desloca ele processa os dados obtidos dos sensores e sua crença em algumas hipóteses de localização são reforçadas enquanto outras têm sua probabilidade reduzida. O sistema de localização então utiliza a regra de decisão *bayesiana* e seleciona a hipótese com a máxima probabilidade. Basicamente, calcula-se a probabilidade de o robô estar ocupando a posição, que corresponde a uma célula na representação por grades de ocupação, dadas as evidências coletadas, ao longo do tempo (LIM; KANG, 2002; BURGARD; FOX; HENNIG, 1997).

Utilizando a técnica de *AR* é possível dividir o processo de construção do modelo em duas fases, denominadas de treinamento (*off-line*) e posicionamento (*on-line*) (WIDYAWAN; KLEPAL; PESCH, 2007). Na fase *off-line* são coletadas uma série de leituras de RSSI que associam o RSSI as respectivas células. Na fase *on-line* é onde de fato acontece a inferência a partir de novas leituras.

A modelagem supracitada permite explorar procedimentos que usam MOMs, informações de RSSI e o algoritmo de Viterbi para estimar a hipótese mais provável para o rastreamento do robô. A Seção 2.4.1 descreve o processo de rastreamento utilizando o algoritmo de Viterbi. Neste trabalho é assumido ainda que os sensores utilizados mensu-

ram o RSSI de pontos de acesso de uma rede sem fio, portanto, cada variável $R_{t,k}$ pode assumir valores inteiros entre -100 e 0 .

2.4.1 Rastreamento de robôs móveis com o algoritmo de Viterbi

Quando usado para o rastreamento de robôs a partir de dados de RSSI, o algoritmo de Viterbi recebe como entrada uma sequência $\mathbf{O}$ de K -*uplas* com os valores observados ao longo do tempo de rastreamento (RODRIGUES; VIEIRA; CAMPOS, 2012). Em seguida estima qual a sequência de estados $\mathbf{P}^*$ (células visitadas) tem a maior probabilidade *a posteriori* dadas as leituras de RSSI e a estrutura do MOM. Esta sequência é assumida como a hipótese de rastreamento. O algoritmo 3 apresenta o pseudocódigo do procedimento de rastreamento baseado no algoritmo de Viterbi.

Algoritmo 3: RASTREAMENTO-VITERBI

Entrada: λ, N

Saída: melhor caminho - P^*

Dados: $P^*[T], O, LIGADO$

início

$LIGADO \leftarrow VERDADEIRO$

enquanto $LIGADO$ **faça**

$o_t \leftarrow \{RSSI_1, \dots, RSSI_k\}$

 Adiciona o_t em O

$P^* \leftarrow VITERBI(\lambda, O, T, N)$

 Informa P^* para sistema de navegação e controle

$t \leftarrow t + 1$

fim

fin

Observando o pseudocódigo apresentado acima o rastreamento é realizado enquanto o *flag* (sinalizador) estiver com o valor da variável $LIGADO \leftarrow VERDADEIRO$. A cada iteração, do laço de repetição, os seguintes passos são executados: a) uma nova leitura (o) dos PAs; b) adiciona-se a nova observação (o) no conjunto de observações O ; c) procede-se com a inferência do melhor caminho utilizando o algoritmo de Viterbi; e d) informa-se o novo melhor caminho (P^*) ao sistema de navegação. Aqui é importante observar que o conjunto de observações O cresce em função do tempo.

2.5 CONSIDERAÇÕES FINAIS

As estufas agrícolas podem ser alternativas eficazes para a produção agrícola. E como pôde ser observado, os robôs móveis são empregados na agricultura como forma de automatizar estufas agrícolas, principalmente em tarefas que oferecem risco e que são repetitivas e dispendiosas para os seres humanos, contribuindo de forma positiva na qualidade e custo da produção.

Este capítulo apresenta uma solução para o problema do rastreamento de *agrobots* em estufas usando modelos ocultos Markov MOMs. Para tanto foi descrito como os parâmetros do MOM se relacionam com o esquema de representação espacial fornecido pelas grade de ocupação e com informações sensoriais baseadas em assinaturas de RSSI. Também foi descrito como o algoritmo de Viterbi pode ser usado para calcular o rastro de um robô.

3 METODOLOGIA

O emprego de MOMs e do algoritmo de Viterbi permite a integração de informações para estimar o rastro de deslocamento de um robô móvel em um ambiente fechado. O uso desta estratégia estabelece um mecanismo para determinar a hipótese de rastreamento mais provável, dado um conjunto de informações coletadas pelos sensores do robô. Os MOMs oferecem ainda um esquema para tratar a incerteza inerente ao problema da autolocalização.

Do ponto de vista prático, é importante que as hipóteses de rastreamento geradas sejam condizentes com os dados disponíveis e que o procedimento de inferência seja eficiente em termos de tempo de processamento e consumo de memória. Assim, este capítulo descreve duas versões de um procedimento para o rastreamento de *agrobots* em estufas agrícolas que utilizam o algoritmo de Viterbi-JD e o Viterbi-JD-MTE na tentativa de reduzir o esforço computacional necessário para realizar a tarefa. Também são descritos os experimentos realizados para comparar a eficiência e a eficácia desses algoritmos no que se refere ao tempo de processamento, consumo de memória e a correção dos resultados do rastreamento.

Nesse capítulo, as Seções 3.1 e 3.1.1 descrevem, respectivamente, as versões do procedimento de rastreamento que utilizam os algoritmos Viterbi-JD e Viterbi-JD-MTE. A Seção 3.2 apresenta a metodologia experimental que orientou os experimentos de rastreamento.

3.1 RASTREAMENTO USANDO O ALGORITMO DE VITERBI COM JANELA DESLIZANTE

O procedimento de rastreamento aqui descrito usa o algoritmo de Viterbi-JD, apresentado na Seção 2.3.2, com o objetivo de reduzir o consumo de memória e o tempo de processamento gasto no rastreamento em relação à versão que usa o algoritmo de Viterbi.

Assim, como o procedimento baseado em Viterbi, o procedimento aqui descrito representa o ambiente na forma de grades de ocupação, o que permite que ele também explore a maneira simplificada com que essa representação relaciona as percepções coletadas pelos sensores às localizações do espaço de navegação (ELFES, 1989b), ao mesmo tempo que procura economizar recursos computacionais. O Algoritmo 4 apresenta o pseu-

docódigo do rastreamento com o algoritmo de Viterbi-JD:

Algoritmo 4: RASTREAMENTO-VITERBI-JD

Entrada: λ, N, w

Saída: melhor caminho - P^*

Dados: $P^*[T], P'[w], O'[w], LIGADO$

inicio

$LIGADO \leftarrow VERDADEIRO$

enquanto $LIGADO$ **faça**

$o_t \leftarrow \{RSSI_1, \dots, RSSI_k\}$

Adiciona o_t em O'

$P' \leftarrow VITERBI - JD(\lambda, O', T, N, w)$

Concatena $P'P^*$ Informa P^* para sistema de navegação e controle

$t \leftarrow t + 1$

fim

fin

O procedimento de rastreamento proposto neste trabalho emprega o algoritmo Viterbi-JD da seguinte maneira: à medida que o *agrobot* se locomove na estufa e coleta os dados RSSI, ele encaminha essas informações para a função de autolocalização que executa o algoritmo de Viterbi-JD. A cada iteração a função compara o índice de tempo t , $t = 1..T$ do MOM, o qual também indica o número de observações T coletadas, com um limiar $w \in \mathbb{N}^+$. Se $t \leq w$, o rastreamento segue de forma similar ao procedimento descrito na Seção 2.3.2.

Se $t > w$, o procedimento é executado sobre uma janela de observações que armazena apenas as últimas w leituras, ou seja, $\mathbf{O}' = \{(r_{t,1}, r_{t,2}, \dots, r_{t,k}) : k = 1..K, t = (t' - w)..t', t' \in \{1..T\}\}$ é um subconjunto do conjunto de observações $\mathbf{O}$ tal que $\mathbf{O}'$ contém as últimas w leituras dos pontos de acesso. Quando t é incrementado, a janela de observações $\mathbf{O}'$ é atualizada e o algoritmo de Viterbi é executado somente sobre as evidências da janela. A cada execução do Viterbi-JD a probabilidade *marginal* do primeiro elemento da janela, $t - w$, é ajustada de forma que $P(S_{t-w} = s_i) = 1$, se o sistema estimou que o robô estava na posição s_i na última vez que os dados referentes ao índice de tempo $t - w - 1$ foram processados. Caso contrário, $p(s_i) = 0$.

Após cada chamada do procedimento de Viterbi-JD obtêm-se uma sequência $\mathbf{P}^*$ com a hipótese de rastreamento relativa a janela $\mathbf{O}'$. Para todo $S_{t'}$, tal que $t' < t - w$, o procedimento aqui proposto considera como resultado do rastreamento a hipótese que foi associada a $S_{t'}$ na última vez em que uma leitura de RSSI foi realizada. Desta forma, o conjunto de estados $S_1 \dots S_{t-w-1}$ é relacionado a uma sequência $\mathbf{P}'$ de estados que indica

o resultado do rastreamento para o período anterior a $t - w$. O resultado final do algoritmo 4 acima é a concatenação¹ $\mathbf{P}_{jd} = \mathbf{P}'\mathbf{P}^*$.

A sequência de rastreamento $\mathbf{P}_{jd}$ não é necessariamente igual a sequência reportada pelo procedimento de rastreamento que emprega o algoritmo de Viterbi. Isso acontece porque quando o processamento de observações relacionadas ao cômputo da sequência $\mathbf{P}^*$ não permite a atualização de $\mathbf{P}'$. No algoritmo de Viterbi, todas as observações podem ter impacto sobre as hipóteses de estado. Em um caso extremo, a última observação poderia alterar a inferência sobre o primeiro estado da sequência. Disso segue que hipótese $\mathbf{P}_{jd}$ é subótima.

3.1.1 MOM com Matriz de Transição Esparsa

Como descrito na Seção 2.3.1, o algoritmo de Viterbi tem como uma de suas entradas a matriz de transição do MOM, a qual é a maior estrutura de dados manipulada no algoritmo e tem a dimensão de $N \times N$, em que N é o número de estados do MOM. Durante o processamento do algoritmo essa matriz é usada em uma série de operações como percorrimto e multiplicação de elementos.

Dado que no problema do rastreamento, muitos elementos da matriz de transição, denotada por A , são nulos, o uso de representações computacionais que explorem a esparsidade² reduzirá a memória consumida para armazenar o MOM.

Portanto, uma alternativa é executar o rastreamento sobre uma implementação do MOM em que a matriz A é codificada usando estruturas de dados para matrizes esparsas. Neste trabalho, a alteração em questão foi obtida fazendo com que o procedimento descrito no algoritmo 2 manipule um objeto A da classe *CRSMatrix* que utiliza uma modelagem matricial chamada *Compressed Row Storage* (CRS)³. A classe *CRSMatrix* está disponível na biblioteca *Linear Algebra for Java* (la4j). Essa versão do rastreamento é denominada de Viterbi-JD-MTE.

3.2 MATERIAIS E MÉTODOS

Essa seção descreve os experimentos realizados neste trabalho. Inicialmente, é apresentado o ambiente físico (Seção 3.2.1) e o procedimento (Seção 3.2.2) que foi usado

¹Aqui a concatenação de duas cadeias u e v é indicada como uv .

²Uma matriz é esparsa se a maioria de seus elementos é igual 0 (SRÁMEK, 2007).

³Nesta estratégia não são armazenados os valores 0 da matriz e todos os valores diferentes de 0 são armazenados em posições de memória contíguas.

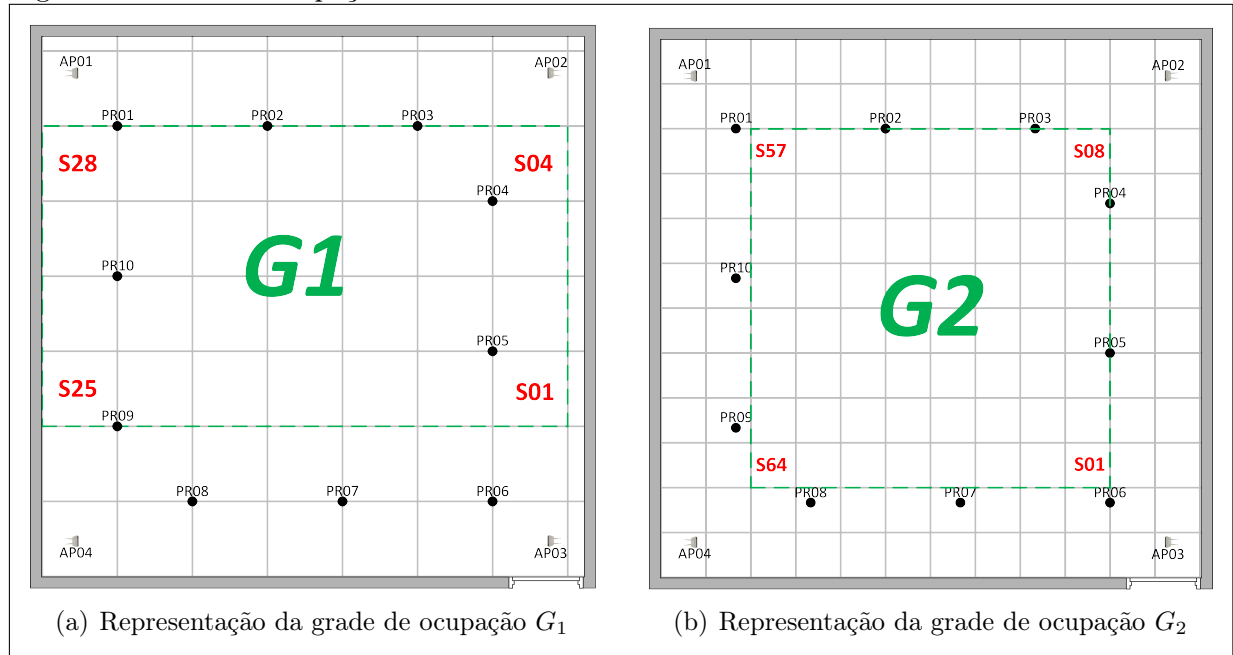
para elaborar as simulações usadas nos testes. A Seção 3.2.3 descreve os experimentos propriamente ditos. A Seção 3.2.4 mostra a abordagem usada para analisar os resultados obtidos.

3.2.1 Infraestrutura: laboratório, dispositivos de hardware e software

Os experimentos de autolocalização exigiram a definição de uma grade de ocupação para uma estufa agrícola. Para atender a essa exigência, duas grades de ocupação foram definidas sobre a estrutura física de um laboratório localizado no Bloco D da Universidade do Oeste de Santa Catarina - UNOESC - Campus de São Miguel do Oeste, na cidade São Miguel do Oeste - SC. Nesse laboratório, com uma área total de $52,05m^2$ foram configuradas duas grades. A primeira grade, identificada como G_1 , possuía 28 células que representavam regiões de $1m^2$. Aa segunda grade, denominada de G_2 , formada por 64 células, sendo que cada célula representava uma região de $0,36m^2$.

A Figura 6 mostra a planta baixa do ambiente de testes com a sobreposição das grades de ocupação (linha pontilhada em verde). Nessa representação é assumido que cada grade de ocupação está associada a um conjunto $G_i = s_1, \dots, s_N$, onde, s_i é a identificação de cada célula válida para o percurso do robô e $N = 28$ para G_1 e $N = 64$ para G_2 . A célula s_1 é a primeira da grade e indica o ponto de saída do robô, considerando-se uma ordem crescente na enumeração das células.

Figura 6: Grades de ocupação G_1 e G_2



A figura exhibe ainda a disposição dos pontos de acesso (PAs) da rede sem fio

padrão IEEE (2012), que foram distribuídos no ambiente para fornecer a infraestrutura necessária para as simulações. Foram instalados 4 PAs, modelos DWL-3200AP da marca D-Link, com suporte as classes 802.11b/g e frequência de operação do sinal de rádio de 2.4GHz. Os PAs foram instalados sobre mesas a aproximadamente 80cm do chão. As configurações padrões de fábrica foram mantidas e os PAs foram nomeados como *AP01*, *AP02*, *AP03* e *AP04*. Para a coleta dos valores de RSSI foi utilizado um *smartphone* modelo Moto G com tela HD de 4,5", processador Snapdragon Quad-Core de 1,2GHz e 1GB de RAM. O *smartphone* executava o sistema operacional Android versão 4.4.4 (*KitKat*). Uma aplicação, denominada de *JBWReader*, desenvolvida utilizando Java para a plataforma Android, permitia ler valores de RSSI de redes *WiFi* e *bluetooth*.

Este trabalho empregou o procedimento proposto por Narzullaev e Park (2013) para gerar as leituras sintéticas de RSSI que foram usadas para simular as assinaturas de RSSI. O procedimento de simulação requer a coleta de uma amostra de valores reais de RSSI para geração de uma base de dados referência. Nessa fase é necessário definir alguns pontos de referência (PR), distribuídos pelo ambiente, e nestes realizar as leituras de RSSI. Para este estudo foram escolhidos 10 PRs. Os pontos foram selecionados de forma aleatória e identificados como *PR01* à *PR10*. Em cada PR foram realizadas 500 leituras de RSSI para alimentar a base de dados de referência. A mesma base de dados referência foi utilizada para gerar 02 assinaturas de RSSI diferentes, uma para a grade de ocupação G_1 e outra para a grade G_2 (Seção 3.2.2).

3.2.2 Assinatura de RSSI

Conforme citado na Seção 2.4, uma das técnicas mais utilizadas para associar os valores de RSSI às células da grade de ocupação é denominada de *fingerprinting* ou assinatura de RSSI (AR). Uma dificuldade no emprego dessa técnica é que para estimar a *AR* é necessário coletar leituras de RSSI para cada posição da grade (fase *off-line*). Como o número de leituras de uma amostra cresce com a área do ambiente, o tempo gasto na geração de uma grande base de dados com leituras de RSSI pode ser inviável em grandes estufas agrícolas.

Uma alternativa proposta por diversos autores é estimar as *ARs* a partir de bases de dados com leituras de RSSI obtidas por simulação (WIDYAWAN; KLEPAL; PESCH, 2007; CHEUNG; SAU; MURCH, 1998; PANJWANI; ABBOTT; RAPPAPORT, 1996). Uma abordagem para simular dados de RSSI é usar modelos que utilizam informações referentes a média de perda do sinal(γ) no ambiente. Este é caso do *modelo de uma encosta* ou *One-Slope*

Model (OSM) (PANJWANI; ABBOTT; RAPPAPORT, 1996).

Este trabalho utilizou uma versão simplificada do modelo OSM proposta por Narzullaev e Park (2013). A adaptação proposta pelos autores define conceitos como simulação de RSSI, pontos de referência (PR) e pontos de predição (PP). A simulação de RSSI é a variável alvo do OSM. Os pontos de referência indicam posições em que foram realizadas as leituras de RSSI na fase *off-line*. São recomendados poucos PRs distribuídos pelo ambiente para reduzir o tempo gasto nesta fase. Os PPs representam os pontos de predição, ou seja, cada PP é uma coordenada (x,y) no plano cartesiano do ambiente para o qual será gerado um valor de RSSI.

No OSM o valor da n -ésima simulação de RSSI, indicada por sR_n é obtido pela Equação 1:

$$sR_n[dBm] = P_{0_k} - 10\gamma \log_{10}\left(\frac{d_{p_n}}{d_{0_k}}\right) \quad (1)$$

Nessa equação, P_{0_k} representa o valor médio de RSSI das leituras realizadas no PR mais próximo. Já d_{p_n} e d_{0_k} e n referem-se, respectivamente, a distância do PP e do PR para um ponto de acesso (PA) e ao número do PP. A fórmula precisa ser aplicada para cada PA. γ expressa a média de perda do sinal⁴

Definidos os elementos do modelo, os autores sugerem os seguintes passos para implementação do OSM:

- a) seleção de M pontos de referência (PRs) distribuídos no ambiente com suas coordenadas (x,y) no plano cartesiano;
- b) realização de R leituras de RSSI para cada PR;
- c) definição de N pontos de predição (PPs) em cada célula com informações como: coordenadas (x,y) no plano cartesiano e o valor médio de RSSI do PR mais próximo;
- d) definição do valor de perda de sinal representado por γ ; e
- e) uso da Equação 1 para gerar o *RSSI* de cada PP.

Neste estudo, os parâmetros do OSM foram configurados como segue: (a) $M = 10$ pontos de referência (PRs); (b) $R = 500$ leituras de RSSI em cada PR; (c) $N = 500$ pontos de predição (PPs); e (d) $\gamma = 3$ e $\gamma = 5$. Os valores de γ foram definidos a partir dos resultados descritos por Sabri *et al.* (2013), que apresentaram um gráfico com

⁴O OSM utiliza uma versão da fórmula de estimação de RSSI do modelo *Log-Normal Shadow* (XU *et al.*, 2010).

a variação da atenuação provocada pela presença de vegetação, para a frequência de $2.4GHz$, utilizando o padrão da ITU (2015).

Pode-se observar que o OSM não considera a presença de ruído nos dados de RSSI (NARZULLAEV; PARK, 2013). Com a finalidade de contornar a limitação descrita utilizou-se a seguinte versão modificada da Equação 1.

$$sR_n[dBm] = P_{0_k} - 10\gamma \log_{10}\left(\frac{d_{p_n}}{d_{0_k}}\right) + U \quad (2)$$

Nessa equação, U representa o ruído aditivo que é amostrado da distribuição $N(0, \sigma)$, uma distribuição gaussiana com média 0. O valor de σ foi ajustado para 1,09. Este valor foi selecionado a partir do trabalho de Xu *et al.* (2010), que estuda a relação entre a variação do sinal de RSSI e a distância dos PAs. O estudo descreve o comportamento do ruído considerando-se a presença de vegetação no ambiente, que é uma característica de estufas agrícolas.

As amostras de RSSI resultantes do processo de geração das *ARs* foram armazenados em 02 arquivos, formato texto, com a extensão CSV (*Comma-Separated Values*) (Apêndice D). O Quadro 2 apresenta os atributos que compõem cada uma das *ARs*.

Quadro 2: Atributos que fazem parte das *ARs*

Atributo	Descrição	Tipo/Valores
PONTO	Identificador do PP onde foi realizada a leitura.	Valor alfanumérico.
PA1	RSSI do PA1.	Valor inteiro negativo.
PA2	RSSI do PA2.	Valor inteiro negativo.
PA3	RSSI do PA3.	Valor inteiro negativo.
PA4	RSSI do PA4.	Valor inteiro negativo.

Além das informações supracitadas, o processo de geração das *ARs* exigiu ainda a criação de dois arquivos com informações auxiliares. O primeiro arquivo continha as coordenadas (x,y) de cada PP e a respectiva célula.

3.2.3 Experimentos

Mais formalmente, para um MOM onde o ambiente é representado com uma grade de ocupação, o rastreamento pode ser visto como o problema de determinar a sequência de células ocupadas pelo robô ao longo do tempo. Assim, seja $\mathbf{T} = 1 \dots T$, com $T \in \mathbb{N}^+$, um conjunto de índices que representa intervalos de tempo consecutivos. Seja S_t , $t \in \mathbf{T}$, uma

variável aleatória com valores $s_1 \dots, s_N$ tal que cada s_i , $i \in 1..N$ e $N = n \times p$, simboliza uma única célula de uma grade M . As informações coletadas pelos sensores no tempo t , a partir das quais se pretende estimar o valor de S_t , são descritas por um conjunto de variáveis aleatórias $R_{t,1} \dots, R_{t,K}$, onde K é o número de sensores. Para implementar os experimentos utilizando MOMs com tais características, foram definidos os seguintes experimentos.

3.2.3.1 Experimentos sobre o desempenho computacional

Os experimentos de desempenho avaliaram a eficiência do procedimento de rastreamento com os algoritmos de Viterbi, Viterbi-JD e Viterbi-JD-MTE. Os algoritmos foram avaliados em termos de tempo e memória usados durante as simulações de rastreamento. Os testes foram executados considerando-se uma estufa agrícola de $315m^2$ apresentada no trabalho de Gaspareto (2013) por se caracterizar como uma referência real a qual o sistema proposto estará sujeito. Sobre a área total da estufa, independentemente de paredes e outros obstáculos existentes, foram definidas duas grades de ocupação, denominadas de GD_1 e GD_2 .

A motivação para a especificação de duas grades de ocupação com tamanhos diferentes foi avaliar, empiricamente, como o número de células da grade afeta o desempenho dos algoritmos Viterbi, Viterbi-JD e Viterbi-JD-MTE. A grade GD_1 tem 315 células com $1m^2$ de área e a grade GD_2 tem 875 células de $0,36m^2$. Os tamanhos das células foram definidas em função do tempo para simular/executar os experimentos. Foram usadas três velocidades de deslocamento⁵: 30 cm/s, 60 cm/s e 90 cm/s.

Os seguintes passos foram executados neste experimento:

- para cada velocidade $v \in \{30cm/s; 60cm/s; 90cm/s\}$
 - para cada janela $w \in \{90, 120, 150, 180\}$
 - * foi gerado um MOM, λ , cuja topologia e parâmetros atendam as restrições referentes à estrutura da grade de ocupação;
 - * os parâmetros do MOM foram inicializados aleatoriamente. Foram observadas as restrições impostas pela arquitetura da grade e velocidade de navegação v ;

⁵As velocidades foram selecionadas a partir de trabalhos que utilizaram robôs móveis para automação de tarefas nas mais diversas áreas, incluindo sua aplicação em estufas agrícolas (SILVA, 2013; GONZALEZ *et al.*, 2009; SAMMONS; FURUKAWA; BULGIN, 2005; OTTJES; RIJSENBRILJ; GROENENBOOM, 2005; MANDOW *et al.*, 1996).

- * foram geradas 30 sequências de RSSI, com 600 observações cada. As sequências foram geradas a partir do MOM λ usando o procedimento de simulação de séries temporais;
- * os algoritmos de rastreamento com Viterbi, Viterbi-JD e Viterbi-JD-MTE foram executados sobre cada uma das sequências; e
- * a média, o desvio padrão, valor máximo e mínimo do tempo de processamento e do consumo de memória foram calculados.

As distribuições iniciais dos MOM λ foram definidas como uniformes. Assim, a probabilidade inicial de cada estado para a grade GD_1 é $\frac{1}{315}$. Para GD_2 as probabilidades iniciais são $\frac{1}{875}$. As probabilidades de emissão dos estados também foram geradas de forma aleatória para cada MOM (Apêndice C). Essa decisão não afetou os testes de desempenho porque neles a correção das hipóteses de rastreamento não estava sob análise, apenas o consumo de memória e o tempo de processamento.

Como descrito na Seção 2.4, para o problema proposto, a matriz de probabilidades de transição (A) do MOM é uma matriz banda. Adicionalmente, a especificação dessa matriz depende da área da célula de cada grade de ocupação, da topologia da grade e da velocidade de navegação (GNANAJEYARAMAN; SEENIVASAN, 2012; RAMAGE, 2007). Assim, para a velocidade de 90 cm/s a largura da banda definida foi 5. Para as demais usou-se uma matriz cuja banda tinha tamanho 3. Restringiu-se a quantidade mínima e máxima de memória RAM à 1 GB para o processamento de cada algoritmo.

3.2.3.2 Experimentos de rastreamento

Os experimentos de rastreamento tiveram como objetivo obter dados sobre o erro de cada um dos algoritmos na estimativa da sequência de células visitadas. Seja $c_1, c_2, \dots, c_N$, a sequência correta de células visitadas pelo robô e seja $q_1, q_2, \dots, q_N$ a sequência estimada pelo algoritmo de rastreamento, o erro e_k relacionado a k -ésima posição da sequência foi definido como 1, se $q_k \neq c_k$, e zero caso contrário. A taxa de erro é dada pela expressão $E = \frac{100 * \sum_{k=1}^N e_k}{N}$.

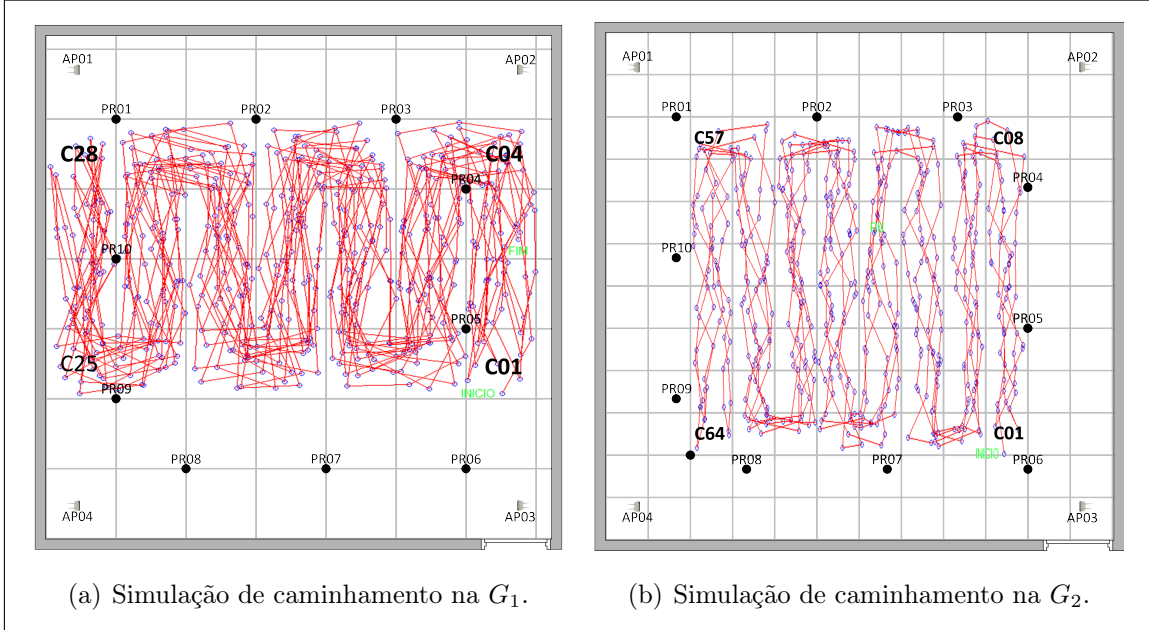
Um vez definido como o erro seria mensurado, foram executados os seguintes passos:

- para cada velocidade $v \in \{30cm/s; 60cm/s; 90cm/s\}$
 - para cada janela $w \in \{90, 120, 150, 180\}$

- * foi gerado um MOM, λ , cuja topologia e parâmetros atendam as restrições referentes à estrutura da grade de ocupação;
- * os parâmetros do MOM da matriz de transição e da distribuição inicial foram inicializados de acordo com a arquitetura da grade e velocidade de navegação v ;
- * os parâmetros das distribuição de emissão foram treinados a partir dos dados de RSSI gerados com o procedimento da Seção 3.2.2;
- * foram simuladas 30 sequências de caminhada de um robô, por um período de 10 min (600 observações), em cada sequência, usando o procedimento descrito no Apêndice E;
- * os algoritmos de rastreamento com Viterbi e Viterbi-JD foram executados sobre cada uma das sequências; e
- * a média, desvio padrão, maior e menor valores de E foram calculados.

A Figura 7 ilustra um exemplo das simulações de caminhada plotadas na planta baixa utilizada como referência para os experimentos.

Figura 7: Caminhamentos gerados para as grade G_1 e G_2 com velocidade de $90cm/s$ e $\gamma = 5$.



Nesses testes, a distribuição inicial também foi definida como sendo uniforme. Logo, cada uma das células de G_1 e G_2 tinha a probabilidade inicial igual a $1/28$ e $1/64$, respectivamente. A matriz de transição foi definida de forma similar àquelas dos experimentos de desempenho.

As distribuições de emissão $P(R_k|s_l)$ da assinatura de RSSI, R_k , em que $k = 1..4$

indica o PA e l é o índice de uma célula da grade de ocupação, foram assumidas como gaussianas (HOANG *et al.*, 2013). Assim, para a G_1 (G_2) foi necessário especificar a média e o desvio padrão do RSSI para cada uma de suas 28 (64) células. Esses parâmetros foram calculadas a partir dos dados de RSSI simulados conforme o procedimento descrito na Seção 3.2.2. Dessa maneira, os parâmetros simulados para a l -ésima célula e k -ésimo PA foram usados para ajustar a distribuição $P(R_k|s_l)$. O Apêndice B mostra os valores destes parâmetros para as duas grades.

Para os experimentos de correção também configuraram-se duas grades de ocupação utilizando os modelos simulados e descritos na Seção 3.2.1. Nesse caso também os MOMs, para as grades G_1 e G_2 , diferem no parâmetro de número de estados (N). A configuração completa é apresentada no Apêndice B. Da mesma forma que nos experimentos de desempenho consideraram-se as velocidades de 30 cm/s e 60 cm/s com uma banda de tamanho 3 para a matriz de transição; e 90 cm/s com uma banda de tamanho 5.

3.2.4 Análise dos Resultados

Nessa seção descritos os procedimentos de análise de desempenho e correção dos algoritmos Viterbi, Viterbi-JD e Viterbi-JD-MTE no processo de rastreamento de um robô móvel.

3.2.4.1 Análise dos resultados de desempenho

Durante a execução dos experimentos foi gerado um arquivo texto com as seguintes informações: Número da sequência de execução, nome do algoritmo, velocidade, w , tempo (em segundos), memória do MOM e a memória do processo (ambas em *MBytes*). Os resultados do rastreamento com o algoritmo de Viterbi foram submetidos a análise estatística descritiva, na qual foram calculados a média, o desvio padrão, o mínimo e o máximo dos valores registrados para o consumo de memória e para o tempo de processamento. Esses valores foram calculados para cada grade de ocupação e cada velocidade.

Similarmente, os resultados obtidos com os algoritmos Viterbi-JD e Viterbi-JD-MTE foram sujeitos à análise estatística descritiva, conforme segue:

- para cada valor de v, w ;
- calculou-se média, o desvio padrão, o mínimo e o máximo dos valores registrados para o tempo de processamento do rastreamento; e

- calculou-se média, o desvio padrão, o mínimo e o máximo dos valores registrados para o consumo de memória do rastreamento.

Adicionalmente, a fim de verificar se as diferenças observadas no desempenho dos algoritmos são significativas, do ponto de vista estatístico, o teste t de *Student* foi usado sobre os resultados dos testes de desempenho.

Nesta avaliação foram comparados os pares de algoritmos **Viterbi X Viterbi-JD**, **Viterbi X Viterbi-JD-MTE**, para todos os valores w , v e em cada grade de ocupação.

Os resultados foram submetidos a análise estatística com o teste t de *Student* e foram realizadas ainda as análises de correlação que permitem avaliar a relação entre o tamanho da janela (w) do algoritmo Viterbi-JD com o tempo de processamento e o consumo de memória. O objetivo da primeira destas análises era determinar se a diferença nas médias do desempenho dos procedimentos testados era significativa do ponto de vista estatístico. Na análise de correlação a intenção era verificar se o parâmetro w tem influência direta sobre o desempenho computacional dos algoritmos Viterbi-JD e Viterbi-JD-MTE.

As análises estatísticas para o desempenho testaram as hipóteses:

- 1) a média de memória consumida pelo algoritmo Viterbi-JD ($\mu_{M_{jd}}$) é igual a média de memória consumida pelo algoritmo de Viterbi (μ_{M_v});
- 2) a média de memória consumida pelo algoritmo Viterbi-JD-MTE ($\mu_{M_{mte}}$) é igual a média de memória consumida pelo algoritmo de Viterbi (μ_{M_v});
- 3) a média de tempo de processamento do algoritmo Viterbi-JD ($\mu_{T_{jd}}$) é igual a média de tempo de processamento do algoritmo de Viterbi (μ_{T_v}); e
- 4) a média de tempo de processamento do algoritmo Viterbi-JD-MTE ($\mu_{T_{mte}}$) é igual a média de tempo de processamento do algoritmo de Viterbi (μ_{T_v}).

3.2.4.2 Análise de resultados de correção

A comparação dos resultados de correção foi aplicada aos algoritmos de Viterbi e Viterbi-JD. Para esses experimentos também considerou-se que os dados entrados eram pareados e os caminhamentos (Apêndice E) eram os mesmos.

As grades de ocupação, testadas nos experimentos eram a G_1 e G_2 . Durante a execução eram armazenados em um arquivo texto, utilizando o separador #, as seguintes

informações: nome do algoritmo, velocidade, γ , σ , w , sequência do caminharmento e a taxa de erro. Foi gerado um arquivo separado para cada grade com todas as execuções.

Da mesma forma que, para a comparação dos resultados dos experimentos de desempenho, após finalizar a execução dos algoritmos e coletar os dados resultantes geraram-se as sumarizações estatísticas utilizando o software estatístico *R*.

Nestas sumarizações geraram-se arquivos de dados com as seguintes informações, tanto para memória como tempo:

- a) Estatística Descritiva: média geral, das 30 sequências, de erro dos caminharmentos, desvio padrão, menor valor de erro, maior valor de erro; e
- b) Estatística *t* de *Student*: o *p-value*⁶ do teste e o intervalo das médias baseado no nível de confiança.

Para a correção foram testadas as seguintes hipóteses:

- 1) a) a taxa de erro(%), sem a presença de ruído, do algoritmo de Viterbi-JD ($\mu_{E_{jd}}$) é igual a taxa de erro do algoritmo de Viterbi (μ_{E_v});
- 2) b) a taxa de erro(%), com a presença de ruído, do algoritmo de Viterbi-JD ($\mu_{E_{jd}^1}$) é igual a taxa de erro do algoritmo de Viterbi ($\mu_{E_v^1}$);
- 3) c) a taxa de erro(%) para o algoritmo de Viterbi, sem a presença de ruído(μ_{E_v}), é igual a sua taxa com a presença de ruído ($\mu_{E_v^1}$); e
- 4) d) a taxa de erro(%) para o algoritmo de Viterbi-JD, sem a presença de ruído($\mu_{E_{jd}}$), é igual a sua taxa com a presença de ruído ($\mu_{E_{jd}^1}$).

Todos as implementações dos algoritmos e aplicações descritas neste capítulo foram realizadas utilizando a linguagem Java e o ambiente de desenvolvimento integrado Eclipse. A versão da aplicação para a coleta das leituras de RSSI foi desenvolvida para o Android versão 2.3.3 (API 10). As análises estatísticas foram conduzidas utilizando o software R e a IDE R Studio. Foram utilizadas ainda as seguintes bibliotecas:

- *Apache Commons Math*, versão 3.3, da APACHE (c2003-2015) para implementação de distribuições normais (*gaussianas*);

⁶O *p-value* é um valor de probabilidade que permite ao usuário rejeitar ou aceitar uma determinada conclusão. O uso desta abordagem é extensiva na área da estatística aplicada (WALPOLE *et al.*, 2006).

- *Object Explorer Measurer* para realizar as medições de memória no processo de *benchmark* dos algoritmos; e
- JCSV, *Simple CSV library for Java*, na versão 1.4.0 para gravação e leitura de arquivos CSV; e
- *Linear Algebra for Java*, la4j, para implementação da matriz de transição esparsa do MOM.

Em todos os experimentos os algoritmos foram executados em um computador pessoal com processador *Core i7-2600* de 3.40GHz, 8 GB de memória RAM e com o sistema operacional *Microsoft Windows 8.1*.

3.3 CONSIDERAÇÕES FINAIS

Este capítulo apresentou uma descrição detalhada da metodologia experimental adotada neste trabalho e que culminou nos resultados que serão mostrados a seguir. Neste capítulo foram apresentadas também as ferramentas utilizadas para as atividades práticas relacionadas à metodologia experimental.

O Capítulo 4 mostra os resultados alcançados a partir da execução dos experimentos. É realizada, ao final do capítulo, uma discussão a respeito desses resultados.

4 RESULTADOS

Este capítulo apresenta os resultados obtidos nos experimentos descritos na metodologia. Inicialmente, são mostrados os resultados dos experimentos que comparam o consumo de memória dos algoritmos Viterbi X Viterbi-JD e Viterbi X Viterbi-JD-MTE na realização de inferências. Em seguida, são relatados os resultados dos experimentos que comparam o tempo de processamento das diferentes implementações do algoritmo de Viterbi.

Na sequência são apresentados os resultados dos experimentos que avaliaram a eficácia dos procedimentos de rastreamento baseados em MOMs, Seção 4.2. Para tanto, as médias das taxas de erro dos procedimentos de rastreamento com Viterbi e Viterbi-JD foram comparados com o teste t de *Student*. Os testes foram executados considerando as duas grades de ocupação descritas na metodologia (G_1 e G_2) e diferentes velocidades de locomoção. O impacto da presença de ruído na correção do rastreamento também é reportado.

No final das seções 4.1 e 4.2 é analisada a correlação entre o parâmetro w (largura da janela) e os resultados sobre desempenho computacional e correção do rastreamento. Essa seção também aborda o efeito do ruído sobre a taxa de erro. Neste sentido discute-se o resultado do teste das seguintes hipóteses: (a) a taxa média de erro, μ_{E_v} , do rastreamento com o algoritmo de Viterbi em dados sem a presença de ruído é igual a média $\mu_{E_v^1}$, obtida por este algoritmo no rastreamento a partir de dados permeados de ruído; e (b) a taxa média de erro, $\mu_{E_{jd}}$, do rastreamento com o algoritmo de Viterbi-JD em dados sem ruído, é igual a média $\mu_{E_{jd}^1}$, obtida por este algoritmo no rastreamento a partir de dados com ruído. Adicionalmente, os resultados obtidos são comparados com aqueles relatados por outros autores.

4.1 RESULTADOS DOS EXPERIMENTOS DE DESEMPENHO

A seguir são apresentados os resultados para os experimentos de consumo de memória e tempo de processamento realizados para as grades GD_1 e GD_2 . Para esses experimentos foram consideradas as velocidades de 30 cm/s, 60 cm/s e 90 cm/s para todos os algoritmos avaliados. No caso dos algoritmos de Viterbi-JD e Viterbi-JD-MTE foram utilizadas ainda janelas de observações (w) de 90, 120, 150 e 180.

4.1.1 Resultados dos Experimentos sobre Desempenho Computacional: consumo de memória

Os resultados dos experimentos de consumo de memória para o rastreamento com o algoritmo de Viterbi, nas grades GD_1 e GD_2 , são apresentados na Tabela 1. Os resultados reportados se referem à quantidade total de memória utilizada pelo algoritmo de rastreamento. Na tabela, a primeira coluna informa a grade de ocupação, e as demais colunas indicam a média e o desvio padrão da memória consumida para cada velocidade. A unidade de medida utilizada para a apresentação dos valores de memória é o *MB* (*MegaBytes*).

Tabela 1: Consumo médio de memória no rastreamento com o Viterbi nas grades GD_1 e GD_2

Grade	30 cm/s	60 cm/s	90 cm/s
	$\mu(\sigma)$	$\mu_t(\sigma)$	$\mu(\sigma)$
GD_1	110,84 (0,003)	110,84 (0,003)	124,43 (30,180)
GD_2	227,10 (0,820)	227,65 (1,670)	229,49 (28,175)

Dividindo-se a média da memória consumida para cada velocidade na grade GD_2 pela mesma média na grade GD_1 obtêm-se as razões 2,05, 2,05 e 1,84 para as velocidades de 30 cm/s, 60 e 90 cm/s, respectivamente. Assim, nota-se que o aumento do número de células da grade (de 315 para 875) afetou a memória consumida pelo rastreamento com Viterbi. Isto é, o tamanho da grade de ocupação fez crescer os requisitos de memória para o rastreamento quando se usa o algoritmo de Viterbi. Este resultado era esperado e está de acordo com a função de complexidade de espaço do algoritmo de Viterbi - $\Theta(N \times M)$ (SRÁMEK, 2007).

O menor consumo de memória na grade GD_1 para as velocidades de 30 cm/s, 60 cm/s e 90 cm/s foi de 110,83, 110,83 e 107,85, respectivamente. O consumo máximo foi de 110,84, 110,84 e 188,96 para as velocidades de 30 cm/s, 60 cm/s e 90 cm/s. Na grade GD_2 o consumo mínimo (máximo) em cada velocidade foi de 226,80 (231,32), 226,80 (231,32) e 111,18 (295,05) para as mesmas velocidades. Como se nota, a alteração da velocidade de 30 cm/s para 60 cm/s não produziu uma grande variação da memória consumida. Já para a velocidade de 90 cm/s, a diferença entre a velocidade máxima e mínima chegou a 183,87 MB. Isso sugere que a dimensão da matriz de transição do MOM, a qual é afetada pela velocidade de deslocamento, pode influir na variância do consumo de memória.

As Tabelas 2 e 3 mostram os resultados obtidos pelo procedimento de rastreamento que empregou o algoritmo Viterbi-JD. Nessas tabelas, a primeira e a quarta colunas informam a velocidade de deslocamento considerada durante a simulação, a segunda

e quinta colunas indicam a largura da janela deslizante e a terceira e a sexta colunas apresentam a média e o desvio padrão do tempo de processamento.

Tabela 2: Consumo médio de memória para o rastreamento com o Viterbi-JD para grade GD_1

Velocidade	Janela	Memória	Velocidade	Janela	Memória
cm/s	w	Média(Desvio)	cm/s	w	Média(Desvio)
30	90	8,00(0,012)	60	150	32,45(0,070)
30	120	15,84(0,007)	60	180	61,23(0,002)
30	150	32,45(0,070)	90	90	8,01(0,100)
30	180	61,23(0,001)	90	120	15,86(0,087)
60	90	8,00(0,013)	90	150	32,45(0,083)
60	120	15,84(0,007)	90	180	61,24(0,009)

Tabela 3: Consumo médio de memória para o rastreamento com o Viterbi-JD para grade GD_2

Velocidade	Janela	Memória	Velocidade	Janela	Memória
cm/s	w	Média(Desvio)	cm/s	w	Média(Desvio)
30	90	7,30(0,052)	60	150	31,69(0,127)
30	120	14,69(0,107)	60	180	69,42(0,147)
30	150	31,69(0,127)	90	90	7,03(0,524)
30	180	69,38(0,131)	90	120	14,60(0,092)
60	90	7,30(0,063)	90	150	31,53(0,051)
60	120	14,69(0,110)	90	180	69,36(0,094)

Seja $m_{\frac{JD}{V}w,v}$ a razão entre o consumo de memória no processamento do rastreamento com Viterbi-JD e a memória consumida pelo rastreamento com o algoritmo de Viterbi na grade GD_1 para cada v e w . Os resultados da Tabela 2 mostram que na grade GD_1 os valores desta razão para cada janela w estão próximos da média aritmética¹. $\sum_{v=\{30,60,90\}} m_{\frac{JD}{V}w,v}$. O mesmo comportamento foi observado na grade GD_2 ². Este resultado sugere que a economia de memória conseguida com o uso de *Viterbi-JD*, se mantém com a alteração da velocidade de deslocamento (e a consequente mudança na matriz de transição).

Os resultados apresentados pelo Viterbi-JD-MTE, em termos de consumo de memória, são melhores se comparados ao algoritmo de Viterbi. No entanto, a comparação dos resultados do rastreamento com Viterbi-JD-MTE com os resultados do Viterbi-JD

¹Na grade GD_1 , as médias dos valores definida por $m_{\frac{JD}{V}w,v}$ para cada $w = 90, 120, 150, 170$ foram 0,07, 0,14, 0,29 e 0,55

²Na grade GD_1 , as médias dos valores de $m_{\frac{JD}{V}w,v}$ para cada $w = 90, 120, 150, 170$ foram 0,03, 0,06, 0,14 e 0,30.

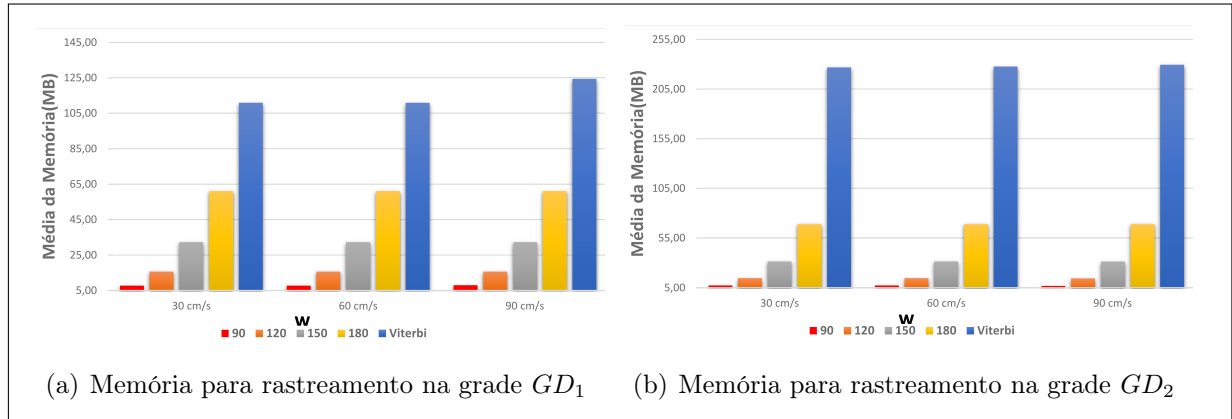
mostra que há situações em que há redução e situações em que há um acréscimo do consumo. No caso da grade GD_1 houve uma redução de memória que chegou a 1,61%. Com a grade GD_2 , que possui mais células, houve um aumento de aproximadamente 2,55% na memória consumida pelo processo de rastreamento. Apesar disso, é importante destacar que a memória usada para armazenar o MOM na CPU foi mensurada separadamente e apresentou para a grade GD_1 uma redução de mais de 90% quando do uso de uma matriz de transição esparsa. Já para a GD_2 a redução alcançou mais de 97%.

Análise Estatística As considerações acima são apoiadas pelo resultado de análise estatística. Como descrito na metodologia, o teste t de *Student* foi usado para comparar diferenças na média do consumo de memória do rastreamento com Viterbi-JD em relação ao rastreamento com Viterbi. O teste mostrou que para todas as velocidades e tamanhos da janela deslizante, a hipótese alternativa foi aceita, ou seja, $\mu_{M_{jd}} < \mu_{M_v}$. Os resultados para o teste estatístico entre os algoritmos Viterbi versus Viterbi-JD-MTE também aceitaram a hipótese alternativa ($\mu_{M_{mte}} < \mu_{M_v}$).

Foi realizado ainda um teste t de *Student* para analisar se a diferença entre as médias da memória consumida pelos algoritmo de Viterbi-JD e Viterbi-JD-MTE é significativa estatisticamente. Os resultados, para a grade GD_1 mostraram que hipótese nula foi aceita somente para os pares $(w = 90, v = 30)$ e $(w = 90, v = 60)$. Com a grade GD_2 os resultados aceitaram a hipóteses alternativa somente para a velocidade de 30 cm/s e $w = 90$.

Os gráficos da Figura 8 apresentam um resumo dos dados referentes ao consumo de memória do rastreamento com Viterbi-JD. Como pode ser notado, nos gráficos e nas tabelas, o algoritmo Viterbi-JD é mais econômico em termos de memória do que o de Viterbi na execução do rastreamento - para todas as configurações de velocidade e janela de observação testadas.

Figura 8: Gráficos comparando a memória consumida para cada w e velocidade do Viterbi-JD X Viterbi



4.1.2 Resultados dos Experimentos sobre Desempenho Computacional: tempo de processamento

A Tabela 1 mostra os resultados referentes ao tempo de execução do algoritmo de Viterbi. Nessa tabela são apresentadas as médias do tempo gasto no processamento do procedimento de rastreamento baseado no algoritmo de Viterbi. Na primeira coluna da tabela informa-se a grade de ocupação, na segunda coluna indica-se a média e o desvio padrão e as demais colunas apresentam os valores de tempo para as velocidades utilizadas.

Tabela 4: Tempo médio do rastreamento com o Viterbi nas grades GD_1 e GD_2

Grade	Medida	30 cm/s	60 cm/s	90 cm/s
GD_1	μ	633,0	627,9	627,1
	σ	13,0	0,6	1,2
GD_2	μ	4857,63	4867,62	4860,62
	σ	3,51	57,91	46,21

Esses resultados indicam que o algoritmo de Viterbi não é adequado quando se considera a necessidade de processamento em tempo real associado aos problemas de rastreamento e autolocalização. Isso pode ser inferido do fato que, em ambas as grades o algoritmo gastou mais de 600 segundos para processar os dados coletados ao longo de 10 minutos. No caso mais demorado, grade GD_2 , o algoritmo precisou em média de 8,11s para realizar o reprocessamento decorrente da leitura de uma observação.

Além disso, como esperado, o tamanho da grade de ocupação é um dos fatores que influencia neste comportamento. Para as velocidades de 30 cm/s, 60 cm/s e 90 cm/s as proporções entre o tempo gasto para processar as sequências na grade GD_1 e na grade GD_2 é de aproximadamente 7,7. Isto é, mais sete vezes a proporção entre o número de

células de GD_2 e GD_1 . Esse fato mostra que, a manutenção da implementação padrão do algoritmo de Viterbi, impossibilita o emprego de grades de ocupação que representam a localização do robô de forma mais precisa (aumentando o número de células e reduzindo a área de cada célula).

As Tabelas 5 e 6 mostram os resultados de tempo obtidos pelo procedimento de rastreamento que empregou o algoritmo Viterbi-JD. Os resultados obtidos nos experimentos de tempo para o algoritmo Viterbi-JD-MTE são descritos nas Tabelas 7 e 8. Naquelas tabelas, a primeira e quinta) colunas indicam a velocidade, a segunda e sexta os valores de w e a terceira e sétima as médias dos tempos. A quarta e oitava colunas mostram a proporção $t_{\frac{JD}{V}}^{v,w}(t_{\frac{JD}{V}}^{v,w})$ entre os tempos de processamento do algoritmo de Viterbi-JD e Viterbi, para cada w e v . A unidade de medida utilizada para apresentação dos valores de tempo é o segundo (s).

Tabela 5: Tempo médio do rastreamento com o Viterbi-JD na grade GD_1

Velocidade	Janela	Tempo	Proporção	Velocidade	Janela	Tempo	Proporção
cm/s	w	Média(Desvio)	$t_{\frac{JD}{V}}^{v,w}$	cm/s	w	Média(Desvio)	$t_{\frac{JD}{V}}^{v,w}$
30	90	166,3(0,7)	0,26	60	150	275,8(3,3)	0,44
30	120	216,0(0,1)	0,34	60	180	327,3(9,5)	0,52
30	150	256,2(1,5)	0,40	90	90	172,5(0,3)	0,28
30	180	298,5(0,1)	0,47	90	120	224,3(0,3)	0,36
60	90	172,6(0,3)	0,27	90	150	273,0(0,3)	0,44
60	120	227,8(7,4)	0,36	90	180	318,5(0,3)	0,52

Tabela 6: Tempo médio do rastreamento com o Viterbi-JD na grade GD_2

Velocidade	Janela	Tempo	Proporção	Velocidade	Janela	Tempo	Proporção
cm/s	w	Média(Desvio)	$t_{\frac{JD}{V}}^{v,w}s$	cm/s	w	Média(Desvio)	$t_{\frac{JD}{V}}^{v,w}$
30	90	1296,1(11,2)	0,26	60	150	2115,7(1,8)	0,43
30	120	1664,0(21,6)	0,34	60	180	2491,4(54,4)	0,51
30	150	1996,0(15,8)	0,41	90	90	1341,9(13,5)	0,28
30	180	2340,8(29,8)	0,48	90	120	1730,3(6,4)	0,36
60	90	1335,6(1,6)	0,27	90	150	2120,4(38,2)	0,44
60	120	1737,4(1,4)	0,36	90	180	2468,1(25,7)	0,51

Tabela 7: Tempo médio do rastreamento com o Viterbi-JD-MTE na grade GD_1

Velocidade	Janela	Tempo	Proporção	Velocidade	Janela	Tempo	Proporção
cm/s	w	Média(Desvio)	$t_{\frac{JDM}{V}}^{v,w}$	cm/s	w	Média(Desvio)	$t_{\frac{JDM}{V}}^{v,w}$
30	90	186,4(0,1)	0,29	60	150	327,6(10,4)	0,52
30	120	242,5(0,1)	0,38	60	180	369,3(23,1)	0,59
30	150	295,2(0,1)	0,47	90	90	202,4(0,4)	0,32
30	180	344,1(0,1)	0,54	90	120	263,1(0,5)	0,42
60	90	206,7(8,5)	0,33	90	150	320,1(0,5)	0,51
60	120	265,3(9,1)	0,42	90	180	373,6(0,7)	0,6

Tabela 8: Tempo médio do rastreamento com o Viterbi-JD-MTE na grade GD_2

Velocidade	Janela	Tempo	Proporção	Velocidade	Janela	Tempo	Proporção
cm/s	w	Média(Desvio)	$t_{\frac{JDM}{V}}^{v,w}$	cm/s	w	Média(Desvio)	$t_{\frac{JDM}{V}}^{v,w}$
30	90	1428,1(12,8)	0,29	60	150	2432,7(70,6)	0,50
30	120	1859,4(23,9)	0,38	60	180	2763,8(52,6)	0,57
30	150	2259,1(21,2)	0,47	90	90	1558,3(22,1)	0,32
30	180	2504,3(36,0)	0,52	90	120	2009,2(45,1)	0,41
60	90	1518,9(2,1)	0,31	90	150	2444,2(24,7)	0,50
60	120	1976,1(2,7)	0,41	90	180	2870,5(60,6)	0,59

Os tempos mínimo (máximo) para o processamento com o algoritmo de Viterbi-JD na grade GD_1 e GD_2 para as velocidades de 30 cm/s, 60 cm/s e 90 cm/s e w são relacionados na Tabela 9. A primeira e segunda colunas correspondem as grades de ocupação e as velocidades. Da terceira a sexta coluna estão representados os w por meio de seus valores mínimos e máximos, como t_{min} e t_{max} .

Tabela 9: Tempos (mínimo e máximo) para o rastreamento com o Viterbi-JD nas grades GD_1 e GD_2

Grade	Velocidade	90	120	150	180
		$t_{min}(t_{max})$			
GD_1	30 cm/s	165,88(169,63)	215,82(216,18)	255,68(262,92)	298,39(298,67)
	60 cm/s	172,18(173,38)	224,03(251,20)	273,37(286,02)	319,95(351,03)
	90 cm/s	171,91(173,16)	223,57(224,85)	272,31(273,50)	317,81(319,61)
GD_2	30 cm/s	1290,94(1350,17)	1633,39(1682,66)	1988,38(2055,53)	2319,82(2431,39)
	60 cm/s	1332,97(1339,89)	1734,30(1740,29)	2112,25(2119,94)	2465,05(2707,92)
	90 cm/s	1331,64(1372,49)	1726,48(1749,63)	2101,57(2286,27)	2453,97(2596,60)

Para a grade GD_1 a maior diferença de $t_{max}-t_{min}$ foi de 32s e concentrou-se na

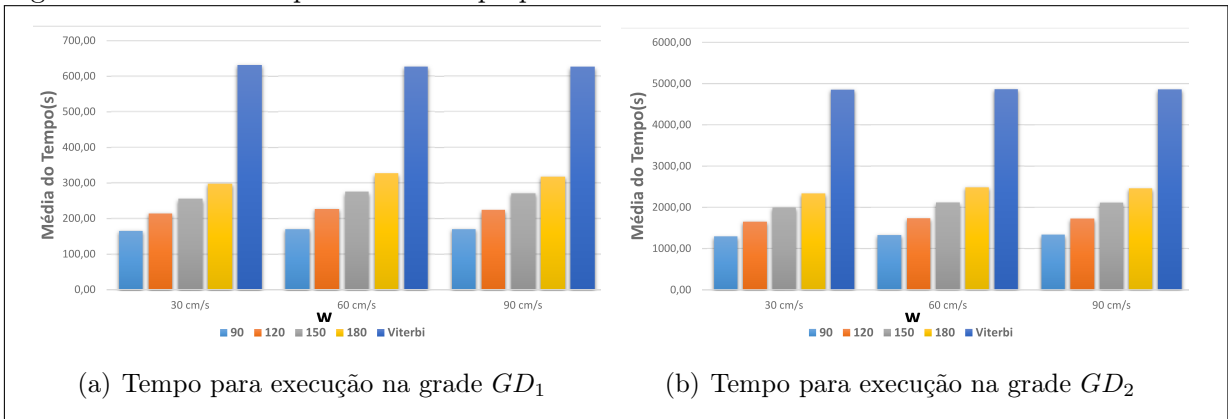
velocidade de 60 cm/s. Já para a grade GD_2 a mesma diferença ficou em 242,87s também na velocidade de 60 cm/s. Mesmo o algoritmo de Viterbi-JD-MTE apresentando melhores resultados que o algoritmo de Viterbi, houve um acréscimo em seu tempo se comparado com o algoritmo de Viterbi-JD, algo em torno de 49,74%.

Análise Estatística O teste t de *Student*, para as grades GD_1 e GD_2 , aceitou as hipóteses alternativas para $\mu_{T_{jd}} < \mu_{T_v}$ e $\mu_{T_{mte}} < \mu_{T_v}$, considerando $\alpha = 0,05$, em todos os experimentos. Isso indica que, na média, o ganho no tempo de processamento pelo uso dos algoritmos de Viterbi-JD e Viterbi-JD-MTE no rastreamento de robôs é significativo.

Quando aplicado para comparar o desempenho entre os algoritmos Viterbi-JD e Viterbi-JD-MTE, o teste t de *Student* mostrou que na grade GD_1 aceitou-se a hipótese nula para todas as velocidades e tamanhos de janela. O mesmo ocorreu para a grade GD_2 . Assim, o acréscimo (*overhead*) notado no tempo de processamento do Viterbi-JD-MTE, que está associado ao uso de matrizes de transição esparsa, não causou uma perda significativa de desempenho.

Os gráficos da Figura 9 resumem os dados descritos nas tabelas desta seção. Como pode ser notado, nos gráficos e nas tabelas, tanto para Viterbi-JD quanto para Viterbi-JD-MTE, os resultados indicam que estes algoritmos são mais rápidos do que o de Viterbi na execução do rastreamento - para todas as configurações de velocidade e janela de observação testadas.

Figura 9: Gráficos comparando o tempo para cada w e velocidade do Viterbi-JD X Viterbi



Os resultados permitem constatar que houve uma forte redução em termos de tempo de processamento e memória consumida no rastreamento com o emprego dos algoritmos Viterbi-JD e Viterbi-JD-MTE. A economia de memória dos algoritmos baseados em janela ficou na faixa de 40% e 90% para uma redução de 70% a 85% no tamanho

da janela. A economia de tempo ficou na faixa de 70% a 90% para a mesma redução no tamanho da janela.

Apesar do ganho de desempenho, deve ser notado que, na grade GD_2 , as configurações testadas para Viterbi-JD e Viterbi-JD-MTE não permitiriam seu uso em uma situação de rastreamento real. Para ver isso basta observar que, naquela grade, o tempo mínimo para processar cada entrada é superior a dois segundos. Contudo, uma análise de correlação sobre os dados (ver Tabelas 10 e 11) permite notar que existe uma forte relação linear entre o tamanho da janela, w , e o tempo de processamento. Segue daí que é possível definir uma função linear que pode ser usada para ajustar o tamanho o valor de w de maneira a atender a requisitos de processamento em tempo real. Por exemplo, a regressão linear dos dados da Tabela 6 para a velocidade de 30cm/s determina a expressão $t = 11,6w + 264$. A partir dessa expressão é possível mostrar que para produzir respostas de rastreamento a cada 0,90s seria suficiente estabelecer $w = 30$.

Tabela 10: Correlação resultante para os valores de w e o tempo de processamento para cada velocidade

Grade	30 cm/s	60 cm/s	90 cm/s
	ρ	ρ	ρ
GD_1	0,998	0,999	0,999
GD_2	0,999	0,999	0,945

Similarmente, os resultados dos testes de memória também mostram uma forte correlação entre w e consumo de memória. Da mesma forma que acontece com o tempo de processamento, essa relação pode ser usada para atender requisitos sobre o uso da memória de um determinado hardware.

Tabela 11: Correlação resultante para os valores de w e a memória consumida para cada velocidade

Grade	30 cm/s	60 cm/s	90 cm/s
	ρ	ρ	ρ
GD_1	0,966	0,966	0,966
GD_2	0,947	0,947	0,948

4.2 AVALIAÇÃO DA CORREÇÃO DO RASTREAMENTO

Os experimentos desta seção comparam os erros das sequências de células associadas às hipóteses de rastreamento calculadas com os algoritmos de Viterbi e Viterbi-JD.

Os experimentos consideraram dados de RSSI simulados usando o modelo OSM em duas grades (G_1 e G_2), conforme descrito na Seção 3.2.2.

4.2.1 Resultados do rastreamento para dados de RSSI sem a presença de ruído

Inicialmente, a Tabela 12 apresenta as taxas de erros($E\%$) obtidos pelos rastreamentos com o algoritmo de Viterbi usando dados de RSSI sem a presença de ruído e para $\gamma \in \{3, 5\}$. As Tabelas 13 e 14 mostram, para o mesmo conjunto de dados, a taxa de erro nas hipóteses de rastreamento calculadas com o algoritmo Viterbi-JD na grade G_1 . As Tabelas 15 e 16 mostram a taxa de erro nas hipóteses de rastreamento calculadas com o algoritmo Viterbi-JD na grade G_2 .

Tabela 12: Taxa de erro(%) no rastreamento com o Viterbi nas grades G_1 e G_2

Grade	γ	E	30 cm/s	60 cm/s	90 cm/s
GD_1	3	μ	1,17	0,85	2,36
	3	σ	0,49	0,48	0,57
	5	μ	1,70	1,00	4,08
	5	σ	0,60	0,34	0,80
GD_2	3	μ	3,77	2,93	5,83
	3	σ	0,71	0,91	1,07
	5	μ	2,98	2,19	4,38
	5	σ	0,72	0,81	0,73

Tabela 13: Taxa de erro(%) no rastreamento com Viterbi-JD para $\gamma = 3$ na grade G_1

Velocidade	Janela	E	Velocidade	Janela	E	Velocidade	Janela	E
m/s	w	$\mu(\sigma)$	m/s	w	$\mu(\sigma)$	m/s	w	$\mu(\sigma)$
30	90	1,70(0,52)	60	90	1,59(0,53)	90	90	2,71(0,60)
30	120	1,67(0,50)	60	120	1,54(0,51)	90	120	2,67(0,61)
30	150	1,65(0,50)	60	180	1,51(0,51)	90	150	2,66(0,63)
30	180	1,65(0,50)	60	180	1,46(0,50)	90	180	2,61(0,60)

Tabela 14: Taxa de erro(%) no rastreamento com Viterbi-JD para $\gamma = 5$ na grade G_1

Velocidade	Janela	E	Velocidade	Janela	E	Velocidade	Janela	E
m/s	w	$\mu(\sigma)$	m/s	w	$\mu(\sigma)$	m/s	w	$\mu(\sigma)$
30	90	2,46(0,81)	60	90	2,45(0,65)	90	90	4,65(0,84)
30	120	2,44(0,80)	60	120	2,42(0,65)	90	120	4,60(0,82)
30	150	2,32(0,76)	60	180	2,25(0,63)	90	150	4,56(0,80)
30	180	2,30(0,78)	60	180	2,16(0,66)	90	180	4,51(0,81)

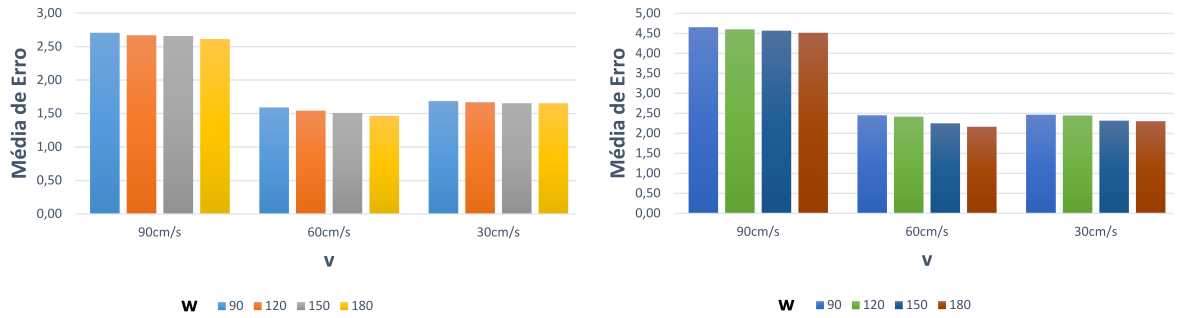
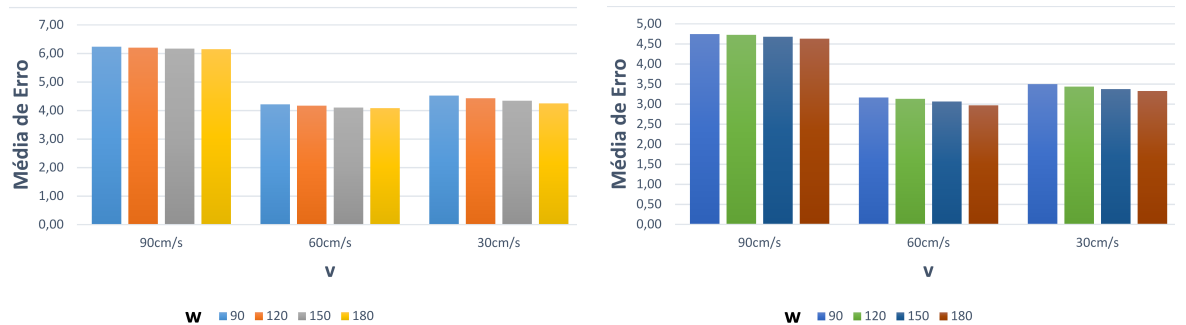
Tabela 15: Taxa de erro(%) no rastreamento com Viterbi-JD para $\gamma = 3$ na grade G_2

Velocidade	Janela	E	Velocidade	Janela	E	Velocidade	Janela	E
m/s	w	$\mu(\sigma)$	m/s	w	$\mu(\sigma)$	m/s	w	$\mu(\sigma)$
30	90	4,52(0,80)	60	90	4,22(0,99)	90	90	6,23(1,07)
30	120	4,43(0,81)	60	120	4,17(1,00)	90	120	6,20(1,05)
30	150	4,34(0,77)	60	180	4,10(1,01)	90	150	6,17(1,07)
30	180	4,25(0,70)	60	180	4,08(1,01)	90	180	6,15(1,04)

Tabela 16: Taxa de erro(%) no rastreamento com Viterbi-JD para $\gamma = 5$ na grade G_2

Velocidade	Janela	E	Velocidade	Janela	E	Velocidade	Janela	E
m/s	w	$\mu(\sigma)$	m/s	w	$\mu(\sigma)$	m/s	w	$\mu(\sigma)$
30	90	3,49(0,66)	60	90	3,16(0,85)	90	90	4,74(0,83)
30	120	3,43(0,66)	60	120	3,13(0,84)	90	120	4,73(0,84)
30	150	3,37(0,65)	60	180	3,06(0,85)	90	150	4,68(0,83)
30	180	3,32(0,64)	60	180	2,97(0,85)	90	180	4,63(0,83)

Os gráficos das Figuras 10 e 11 ilustram os resultados apresentados nas tabelas acima.

Figura 10: Rastreamento com Viterbi-JD na grade G_1 sem ruído: $w \times E_r$ (a) Taxa de erro(%) para $\gamma = 3$ (b) Taxa de erro(%) para $\gamma = 5$ Figura 11: Rastreamento com Viterbi-JD na grade G_2 sem ruído: $w \times E_r$ (a) Taxa de erro para $\gamma = 3$ (b) Taxa de erro para $\gamma = 5$

4.2.2 Resultados do rastreamento para dados de RSSI com a presença de ruído

A Tabelas 17, 18, 19, 20 e 21 apresentam as taxas de erro obtidas pelo rastreamento com os algoritmos de Viterbi e Viterbi-JD usando dados de RSSI com a presença de ruído. Os testes mostram os resultados para as duas grades de ocupação (G_1 e G_2) e $\gamma = \{3, 5\}$. Nesses experimentos foi utilizado um ruído aditivo gaussiano com distribuição $N(0, 1.09)$.

Tabela 17: Taxa de erro(%) no rastreamento com o Viterbi nas grades G_1 e G_2

Grade	γ	E	30 cm/s	60 cm/s	90 cm/s
GD_1	3	μ	5,15	3,38	8,64
	3	σ	1,06	0,65	1,27
	5	μ	3,93	3,12	7,89
	5	σ	0,76	0,78	0,95
GD_2	3	μ	15,22	13,00	21,02
	3	σ	1,23	1,48	2,08
	5	μ	10,95	7,99	13,66
	5	σ	4,83	1,42	1,21

Tabela 18: Taxa de erro(%) no rastreamento com Viterbi-JD para $\gamma = 3$ na grade G_1

Velocidade	Janela	E	Velocidade	Janela	E	Velocidade	Janela	E
m/s	w	$\mu(\sigma)$	m/s	w	$\mu(\sigma)$	m/s	w	$\mu(\sigma)$
30	90	6,79(1,09)	60	90	6,03(0,96)	90	90	10,41(1,39)
30	120	6,74(1,11)	60	120	6,01(1,03)	90	120	10,22(1,41)
30	150	6,69(1,11)	60	180	5,85(1,00)	90	150	10,09(1,43)
30	180	6,61(1,13)	60	180	5,71(0,99)	90	180	9,99(1,44)

Tabela 19: Taxa de erro(%) no rastreamento com Viterbi-JD para $\gamma = 5$ na grade G_1

Velocidade	Janela	E	Velocidade	Janela	E	Velocidade	Janela	E
m/s	w	$\mu(\sigma)$	m/s	w	$\mu(\sigma)$	m/s	w	$\mu(\sigma)$
30	90	5,78(0,93)	60	90	5,96(0,93)	90	90	9,38(1,13)
30	120	5,67(0,91)	60	120	5,83(0,95)	90	120	9,31(1,10)
30	150	5,48(0,94)	60	180	5,55(0,89)	90	150	9,22(1,06)
30	180	5,46(0,92)	60	180	5,34(0,95)	90	180	9,13(1,09)

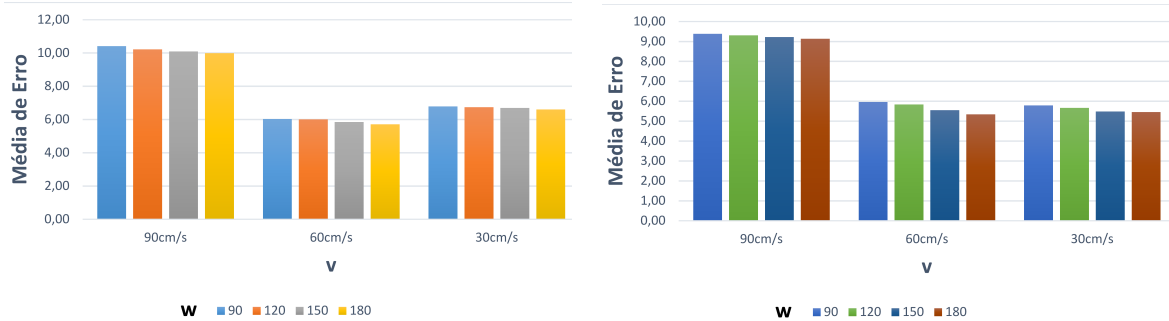
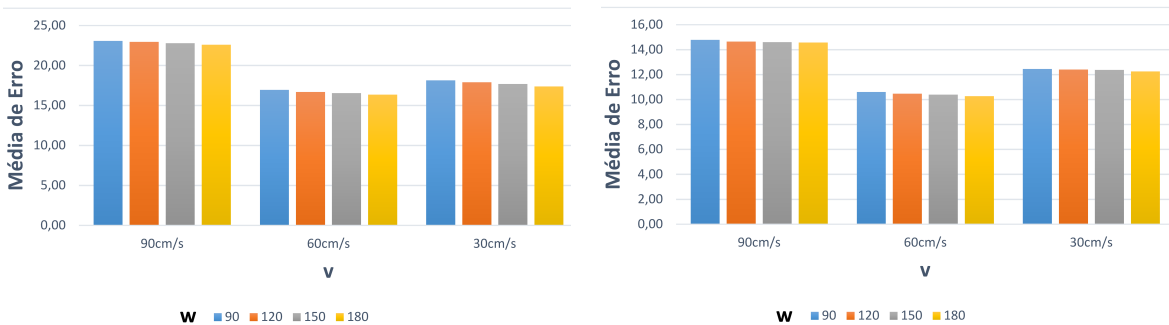
Tabela 20: Taxa de erro(%) no rastreamento com Viterbi-JD para $\gamma = 3$ na grade G_2

Velocidade	Janela	E	Velocidade	Janela	E	Velocidade	Janela	E
m/s	w	$\mu(\sigma)$	m/s	w	$\mu(\sigma)$	m/s	w	$\mu(\sigma)$
30	90	18,14(1,31)	60	90	16,94(1,97)	90	90	23,08(2,09)
30	120	17,90(1,28)	60	120	16,69(1,96)	90	120	22,96(2,12)
30	150	17,68(1,32)	60	180	16,53(1,93)	90	150	22,80(2,10)
30	180	17,38(1,23)	60	180	16,34(1,90)	90	180	22,61(2,04)

Tabela 21: Taxa de erro(%) no rastreamento com Viterbi-JD para $\gamma = 5$ na grade G_2

Velocidade	Janela	E	Velocidade	Janela	E	Velocidade	Janela	E
m/s	w	$\mu(\sigma)$	m/s	w	$\mu(\sigma)$	m/s	w	$\mu(\sigma)$
30	90	12,45(3,04)	60	90	10,60(1,27)	90	90	14,78(1,21)
30	120	12,41(3,76)	60	120	10,47(1,29)	90	120	14,65(1,18)
30	150	12,38(4,44)	60	180	10,39(1,28)	90	150	14,60(1,20)
30	180	12,26(4,80)	60	180	10,27(1,30)	90	180	14,56(1,17)

Os gráficos das Figuras 12 e 13 ilustram os resultados obtidos em ambas as grades.

Figura 12: Rastreamento com Viterbi-JD na grade G_1 com ruído: $w \times E_r$ Figura 13: Rastreamento com Viterbi-JD na grade G_2 com ruído: $w \times E_r$ 

4.2.3 Análise dos resultados de erro de rastreamento

O teste t de *Student*, para dados pareados e $\alpha = 0,05$, foi aplicado sobre os resultados das tabelas acima. Esse teste foi realizado a fim de comparar a taxa de erro dos algoritmos. Para tanto, definiu-se $H_0 \equiv \mu_{E_v} - \mu_{E_{jd}} = 0$ como hipótese nula e $H_1 \equiv \mu_{E_{jd}} > \mu_{E_v}$ como hipótese alternativa, para casos onde há ausência de ruído. Para casos onde há presença de ruído as hipóteses, H_0 e H_1 , foram definidas como $\mu_{E_v^1} - \mu_{E_{jd}^1} = 0$ e $\mu_{E_{jd}^1} > \mu_{E_v^1}$. A análise estatística aceitou a hipótese alternativa para todos os valores de v , w e γ para ambas as grades simuladas. Tal resultado mostra que a taxa de erro do rastreamento com o algoritmo Viterbi-JD foi efetivamente superior àquela obtida pelo algoritmo Viterbi.

A diferença mínima absoluta entre as taxas de erro do algoritmo de Viterbi-JD e Viterbi foi de 0,25 na grade G_2 , para $w = 180$, $v = 90$ cm/s e $\gamma = 5$. Já a maior diferença absoluta foi de 1,44 na grade G_1 , para $w = 90$, $v = 60$ cm/s e $\gamma = 5$. Na presença de ruído, a menor diferença absoluta entre as taxas do erro foi 0,90 e ocorreu na grade G_2 , para $w = 180$, $v = 90$ e $\gamma = 5$. A maior diferença foi 3,94 e também ocorreu nas taxas da grade G_2 para $w = 90$, $v = 60$ e $\gamma = 3$. O fato dessas diferenças serem pequenas em termos absolutos mostra que o incremento na taxa de erro foi muito menor do que o ganho observado no tempo de processamento e no consumo de memória durante os testes de desempenho. Na próxima seção essa observação é explorada na proposição de um mecanismo que pode ser útil para ajustar o procedimento de rastreamento de forma a obter um desempenho em que a taxa de erro é aceitável e o esforço computacional é viável.

A análise de correlação entre w e a taxa de erro para dados sem e com ruído, cujos resultados estão listados nas Tabelas 22 e 23, também indica a existência de uma relação linear entre w e E . Além disso, uma análise de regressão mostra que, em ambas as grades e para todo v e γ , o maior valor absoluto da inclinação da função linear $E = aw + b$ não ultrapassa 0,008. Isto é, nas simulações realizadas, a taxa de erro aumenta em oito milésimos quando w decresce uma unidade. Isso sugere que o emprego de valores pequenos para w não causa uma queda substancial na correção do rastreamento. Por exemplo, o uso de uma janela com 30 observações no rastreamento em G_2 , com $v = 30$, $\gamma = 2$ e presença de ruído, elevaria a expectativa do erro para 18,65. Um incremento de apenas 0,5%.

Tabela 22: Correlação entre os valores de w e a taxa de erro, sem ruído, para cada velocidade e γ

Grade	γ	30 cm/s	60 cm/s	90 cm/s
		ρ	ρ	ρ
GD_1	3	-0,944	-0,997	-0,978
	5	-0,940	-0,973	-0,996
GD_2	3	-1,000	-0,984	-0,990
	5	-0,999	-0,979	-0,982

Tabela 23: Correlação entre os valores de w e a taxa de erro, com ruído, para cada velocidade

Grade	γ	30 cm/s	60 cm/s	90 cm/s
		ρ	ρ	ρ
GD_1	3	-0,991	-0,969	-0,991
	5	-0,965	-0,990	-1,000
GD_2	3	-0,998	-0,995	-0,996
	5	-0,941	-0,997	-0,950

Finalmente, no que se refere à correção das hipóteses de rastreamento, se observa que a presença do ruído influencia negativamente a correção das hipóteses de rastreamento. Esta conclusão foi corroborada com o teste t de *Student* que mostrou que para todas as velocidades e valores de w , e em todas as grades, foi aceita a hipótese alternativa de que a taxa de erro(%), com a presença de ruído($\mu_{E_{jd}}$), é maior que a taxa de erro sem a presença de ruído ($\mu_{E_{jd}^1}$).

4.3 CONSIDERAÇÕES FINAIS

Este capítulo mostra os resultados da eficiência e eficácia computacional dos algoritmos de Viterbi, Viterbi-JD e Viterbi-JD-MTE. Esses algoritmos foram comparados em termos de desempenho computacional aos algoritmos de Viterbi $\times$ Viterbi-JD e Viterbi $\times$ Viterbi-JD-MTE. Em termos de correção foram comparados os algoritmos de Viterbi $\times$ Viterbi-JD. Além de serem apresentados os resultados obtidos, uma discussão em torno destes e sua relação com outros trabalhos é realizada ao final.

5 CONCLUSÕES E TRABALHOS FUTUROS

Esse trabalho abordou o problema do rastreamento de *agrobots*, *robôs agricultores*, em estufas agrícolas utilizando leituras de RSSI de pontos de acesso de redes *Wifi*. Para tanto, o ambiente interno da estufa agrícola foi representado como uma grade de ocupação e o problema do rastreamento foi modelado como a determinação da sequência das células visitadas que tinha a máxima probabilidade posterior, dadas as leituras de RSSI. Formalmente, o deslocamento do robô ao longo do tempo foi modelado por um modelo oculto de Markov e as inferências sobre seu rastro foram implementadas com o algoritmo de Viterbi.

Nos experimentos, o algoritmo Viterbi-JD teve um desempenho computacional significativamente superior ao do algoritmo de Viterbi no processamento do rastreamento, o que sugere que ele é uma alternativa para o desenvolvimento de sistemas de rastreamento sujeitos à restrições de tempo e memória. Mais objetivamente, a correlação entre o desempenho computacional e o tamanho da janela permite a criação de um mecanismo para ajustar o esforço computacional dedicado à tarefa de rastreamento de *agrobots*.

Em termos de correção na geração das hipóteses de rastreamento, ambos os algoritmos baseados em janelas deslizantes apresentaram um erro maior do que aquele produzido pelo algoritmo de Viterbi. Contudo, em todos os casos, o emprego dos algoritmos Viterbi-JD e Viterbi-JD-MTE não levou a um incremento substancial do erro na determinação do rastro. Com isso, espera-se que o ajuste do tamanho da janela, como proposto, não prejudique a qualidade das informações fornecidas ao sistema de navegação.

Trabalhos Futuros

Os próximos trabalhos planejam verificar a possibilidade de integrar informações coletadas em diversos tipos de sensores no procedimento proposto com o objetivo de melhorar a precisão do rastreamento. Além disso, se propõe abordar alguns aspectos práticos do rastreamento de *agrobots* em ambientes reais, nos quais há presença de ruído nos dados de RSSI e a variabilidade do sinal em decorrência do crescimento das plantas. Também se pretende estudar o desempenho computacional de implementações paralelas do Viterbi-JD combinadas com o aumento do número de pontos de acesso na melhora da correção do algoritmo de Viterbi-JD.

REFERÊNCIAS

- ACACCIA, G. M. *et al.* Mobile robots in greenhouse cultivation: Inspection and treatment of plants. *ACTA Horticulture, Vol.*, v. 453, p. 52–60, 1998.
- APACHE. *The Apache Commons Mathematics Library*. c2003–2015. Disponível em: <<http://commons.apache.org/proper/commons-math/>>.
- ARTHI, R.; DEVARAJ, P.; MURUGAN, K. Rss based localization of sensor nodes by learning movement model. *WSEAS Transactions on Communications*, v. 12, November 2013.
- BELFORTE, G. *et al.* Robot design and testing for greenhouse applications. *Biosystems Engineering*, v. 95, n. 3, p. 309 – 321, 2006. ISSN 1537-5110.
- BERGERMAN, M. *et al.* Robot farmers: Autonomous orchard vehicles help tree fruit production. *Robotics Automation Magazine, IEEE*, v. 22, n. 1, p. 54–63, March 2015. ISSN 1070-9932.
- BORENSTEIN, J.; KOREN, Y. The vector field histogram-fast obstacle avoidance for mobile robots. *Robotics and Automation, IEEE Transactions on*, v. 7, n. 3, p. 278–288, Jun 1991. ISSN 1042-296X.
- BOURELY, A. *et al.* Rosal, a grafting robot for woody plants. *Proc. of IARP 4th Workshop on Robotics in Agriculture and the Food Industry*, 1995.
- BURGARD, W.; FOX, D.; HENNIG, D. Fast grid-based position tracking for mobile robots. In: *In Proc. of the 21th German Conference on Artificial Intelligence*. [S.l.]: Springer Verlag, 1997. p. 289–300.
- BURGARD, W. *et al.* Position tracking with position probability grids. In: *Advanced Mobile Robot, 1996., Proceedings of the First Euromicro Workshop on*. [S.l.: s.n.], 1996. p. 2–9.
- CABRERA-MORA, F.; XIAO, J. Preprocessing technique to signal strength data of wireless sensor network for real-time distance estimation. In: *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. [S.l.: s.n.], 2008. p. 1537–1542. ISSN 1050-4729.
- CAPPÉ, O.; MOULINES, E.; RYDEN, T. *Inference in Hidden Markov Models (Springer Series in Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005. ISBN 0387402640.
- CHEN, C. *et al.* Dynamic environment modeling with gridmap: A multiple-object tracking application. In: *Control, Automation, Robotics and Vision, 2006. ICARCV '06. 9th International Conference on*. [S.l.: s.n.], 2006. p. 1–6.

- CHEUNG, K.-W.; SAU, J.-M.; MURCH, R. A new empirical model for indoor propagation prediction. *Vehicular Technology, IEEE Transactions on*, v. 47, n. 3, p. 996–1001, Aug 1998. ISSN 0018-9545.
- CORRELL, N. *et al.* Indoor robot gardening: design and implementation. *Intelligent Service Robotics*, Springer-Verlag, v. 3, n. 4, p. 219–232, 2010. ISSN 1861-2776.
- COSTA, J. da. *Dicionário Rural do Brasil*. [S.l.]: Campus, 2003.
- DATAR, M. *et al.* Maintaining stream statistics over sliding windows: (extended abstract). In: *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2002. (SODA '02), p. 635–644. ISBN 0-89871-513-X.
- DUCKETT, T.; NEHMZOW, U. Mobile robot self-localisation using occupancy histograms and a mixture of gaussian location hypotheses. *Robotics and Autonomous Systems*, v. 34, n. 2–3, p. 119–130, 2001.
- DUDEK, G.; JENKIN, M. *Computational Principles of Mobile Robotics*. New York, NY, USA: Cambridge University Press, 2000. ISBN 0521560217.
- EDAN, Y. Design of an autonomous agricultural robot. *Applied Intelligence*, Kluwer Academic Publishers, v. 5, n. 1, p. 41–50, 1995. ISSN 0924-669X.
- ELFES, A. Sonar-based real-world mapping and navigation. *Robotics and Automation, IEEE Journal of*, v. 3, n. 3, p. 249–265, June 1987. ISSN 0882-4967.
- ELFES, A. Occupancy grids: a stochastic spatial representation for active robot perception. 1989.
- ELFES, A. Using occupancy grids for mobile robot perception and navigation. *Computer*, v. 22, n. 6, p. 46–57, June 1989. ISSN 0018-9162.
- EMMI, L. *et al.* New trends in robotics for agriculture: Integration and assessment of a real fleet of robots. *The Scientific World Journal*, p. 21, March 2014.
- FELZENSZWALB, P. F.; HUTTENLOCHER, D. P.; KLEINBERG, J. M. Fast algorithms for large-state-space hmms with applications to web usage analysis. In: *Advances in Neural Information Processing Systems 16*. [S.l.]: MIT Press, 2004. p. 409–416.
- FORNEY G.D., J. The viterbi algorithm. *Proceedings of the IEEE*, v. 61, n. 3, p. 268–278, March 1973. ISSN 0018-9219.
- FRANZ, M. O.; MALLOT, H. A. Biomimetic robot navigation. *Robotics and autonomous Systems*, v. 30, p. 133–153, 2000.
- FUJIWARA, Y.; SAKURAI, Y. Fast algorithm for monitoring data streams by using hidden markov models. v. 9, n. 12, 2011.
- GASPARETO, M. J. *SELEÇÃO DE ATRIBUTOS EM IMAGENS COLETADAS SOB CONDIÇÕES DE ILUMINAÇÃO NÃO CONTROLADA E SUA INFLUÊNCIA NO DESEMPENHO DE CLASSIFICADORES NAIVE BAYES PARA IDENTIFICAÇÃO DE OBJETOS EM ESTUFAS AGRÍCOLAS*. Dissertação (Dissertação de Mestrado) — Universidade Estadual de Ponta Grossa - UEPG, 2013.

- GHAHRAMANI, Z. Hidden markov models. In: . River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2001. cap. An Introduction to Hidden Markov Models and Bayesian Networks, p. 9–42. ISBN 981-02-4564-5.
- GILBERT, G.; IVAN, L. Handling robots for horticultural greenhouses. *Proc. of the 2nd International Workshop on Robotics in Agriculture & the Food Industry IARP*, p. 9–17, June 1991.
- GNANAJEYARAMAN, R.; SEENIVASAN, G. Diagonalization matrix method of solving the first problem of hidden markov model in speech recognition system. *CoRR*, abs/1202.6256, 2012.
- GONZALEZ-ARJONA, D. *et al.* Simplified occupancy grid indoor mapping optimized for low-cost robots. *ISPRS International Journal of Geo-Information*, v. 2, n. 4, p. 959, 2013. ISSN 2220-9964.
- GONZALEZ, R. *et al.* Navigation techniques for mobile robots in greenhouses. *Applied engineering in agriculture*, v. 25, p. 153–165, 2009.
- HENTEN, E. van *et al.* An autonomous robot for harvesting cucumbers in greenhouses. *Autonomous Robots*, Kluwer Academic Publishers, v. 13, n. 3, p. 241–258, 2002. ISSN 0929-5593.
- HERRMANN, J.; PAWELZIK, K.; GEISEL, T. Self-localization of autonomous robots by hidden representations. *Autonomous Robots*, Kluwer Academic Publishers, v. 7, n. 1, p. 31–40, 1999. ISSN 0929-5593.
- HOANG, M. *et al.* A hidden markov model for indoor user tracking based on wifi fingerprinting and step detection. In: *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European*. [S.l.: s.n.], 2013. p. 1–5.
- IBRAHIM, M.; YOUSSEF, M. Cellsense: A probabilistic rssi-based gsm positioning system. In: *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*. [S.l.: s.n.], 2010. p. 1–5. ISSN 1930-529X.
- IBRAHIM, M.; YOUSSEF, M. A hidden markov model for localization using low-end gsm cell phones. In: *ICC*. [S.l.]: IEEE, 2011. p. 1–5. ISBN 978-1-61284-232-5.
- IEEE. *802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. 2012.
- ITU. *About International Telecommunication Union(ITU)*. March 2015. Disponível em: <<http://www.itu.int/en/about/Pages/default.aspx>>.
- JACKSON, R. C. *Data Detection Algorithms for Perpendicular Magnetic Recording in the Presence of Strong Media Noise*. Dissertação (Mestrado) — University of Warwick, 2008.
- JEKABSONS, G.; KAIRISH, V.; ZURAVLYOV, V. An analysis of wi-fi based indoor positioning accuracy. *Scientific Journal of Riga Technical University*, v. 47, 2011.
- JURAFSKY, D.; MARTIN, J. H. *Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition*. [S.l.: s.n.], 2006.

- KAMRAN, E.; MIRASI, A.; SAMADI, M. Navigation techniques of mobile robots in greenhouses. *International journal of Advanced Biological and Biomedical Research*, v. 2, p. 1438–1449, 2014.
- LADD, A. *et al.* On the feasibility of using wireless ethernet for indoor localization. *Robotics and Automation, IEEE Transactions on*, v. 20, n. 3, p. 555–559, June 2004. ISSN 1042-296X.
- LIM, J.-H.; KANG, C.-U. Grid-based localization of a mobile robot using sonar sensors. *KSME International Journal*, Korean Society of Mechanical Engineers, v. 16, n. 3, p. 302–309, 2002. ISSN 1738-494X.
- LINA, C.; YU, Z.; ZHENGQI, Z. A new approach of rssi probability distributions for indoor positioning. *Journal of Theoretical and Applied Information Technology - ISSN: 1992-8645*, v. 51, n. 1, May 2013.
- LIU, X.; YUAN, J.; WANG, K. A problem-specific genetic algorithm for path planning of mobile robot in greenhouse. In: WANG, K. *et al.* (Ed.). *Knowledge Enterprise: Intelligent Strategies in Product Design, Manufacturing, and Management*. [S.l.]: Springer US, 2006, (IFIP International Federation for Information Processing, v. 207). p. 211–216. ISBN 978-0-387-34402-7.
- LO, S.-H.; WU, C.-H.; CHUNG, Y.-C. Cooperative localization with pre-knowledge using bayesian network for wireless sensor networks. In: *Parallel Processing Workshops, 2007. ICPPW 2007. International Conference on*. [S.l.: s.n.], 2007. p. 69–69. ISSN 1530-2016.
- MAHAJAN, A.; CHANANA, M. Wi-fi localization using rssi in indoor environment via a smartphone. *International Journal Of Engineering And Computer Science ISSN:2319-7242*, v. 1, p. 94–98, Nov. 2012.
- MANDOW, A. *et al.* The autonomous mobile robot aurora for greenhouse operation. *Robotics Automation Magazine, IEEE*, v. 3, n. 4, p. 18–28, Dec 1996. ISSN 1070-9932.
- MARTIN, M. C.; MORAVEC, H. P. *Robot evidence grids*. [S.l.], 1996.
- MESTER, G. Fuzzy-logic sensor-based navigation of autonomous wheeled mobile robots in the greenhouse environments. 2011.
- MILIORIS, D. *et al.* Low-dimensional signal-strength fingerprint-based positioning in wireless lans. *Ad Hoc Networks*, v. 12, p. 100 – 114, 2014. ISSN 1570-8705.
- MORAVEC, H. Sensor fusion in certainty grids for mobile robots. In: CASALS, A. (Ed.). *Sensor Devices and Systems for Robotics*. [S.l.]: Springer Berlin Heidelberg, 1989, (NATO ASI Series, v. 52). p. 253–276. ISBN 978-3-642-74569-0.
- MORAVEC, H.; ELFES, A. High resolution maps from wide angle sonar. In: *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*. [S.l.: s.n.], 1985. v. 2, p. 116–121.

- NARZULLAEV, A.; PARK, Y. Novel calibration algorithm for received signal strength based indoor real-time locating systems. *{AEU} - International Journal of Electronics and Communications*, v. 67, n. 7, p. 637 – 644, 2013. ISSN 1434-8411.
- NIKOVSKI, D.; NOURBAKHSI, I. Learning probabilistic models for state tracking of mobile robots. In: *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*. [S.l.: s.n.], 2002. v. 1, p. 1026–1031 vol.1.
- OLSON, C. Probabilistic self-localization for mobile robots. *IEEE Transactions on Robotics and Automation*, v. 16, n. 1, p. 55–66, Feb 2000. ISSN 1042-296X.
- OTTJES, J. A.; RIJSENBRIJ, J. C.; GROENENBOOM, A. Greenhouse automation: A model for design and evaluation. In: *3rd International Industrial simulation Conference ISC*. Berlin Germany: [s.n.], 2005.
- PAL, A. Localization algorithms in wireless sensor networks: Current approaches and future challenges. *Network Protocols and Algorithms - ISSN 1943-3581*, v. 2, n. 1, 2010.
- PANJWANI, M. A.; ABBOTT, A. L.; RAPPAPORT, T. s. Interactive computation of coverage regions for wireless communication in multifloored indoor environments. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, v. 14, n. 3, 1996.
- PILU, M. Video stabilization as a variational problem and numerical solution with the viterbi method. In: *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. [S.l.: s.n.], 2004. v. 1, p. I-625–I-630 Vol.1. ISSN 1063-6919.
- POL, M. S. *Navigation Of mobile robot in cluttered environment*. Dissertação (Master Thesis) — Department of Mechanical Engineering, National Institute of Technology Rourkela, Rourkela, Odisha, 769 008, India, May 2014.
- RABINER, L. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, v. 77, n. 2, p. 257–286, Feb 1989. ISSN 0018-9219.
- RABINER, L.; JUANG, B.-H. *Fundamentals of Speech Recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993. ISBN 0-13-015157-2.
- RABINER, L. R.; JUANG, B. H. An introduction to hidden markov models. *IEEE ASSP Magazine*, 1986.
- RAGHAVAN, A. *et al.* Accurate mobile robot localization in indoor environments using bluetooth. In: *IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2010. p. 4391–4396. ISSN 1050-4729.
- RAMADURAI, V.; SICHITIU, M. L. *Localization in Wireless Sensor Networks: A Probabilistic Approach*. 2003.
- RAMAGE, D. Hidden markov models fundamentals. CS229 Section Notes. December 2007.

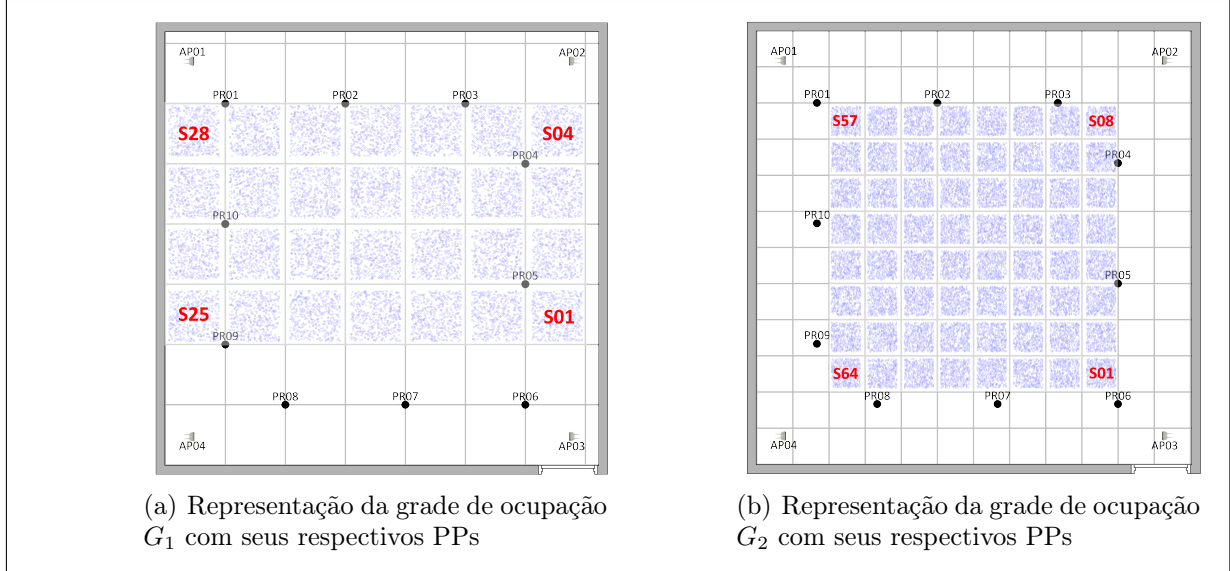
- RAPPAPORT, T. *Wireless Communications: Principles and Practice*. 2nd. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001. ISBN 0130422320.
- REYNOLDS, L. Agrobots - robots in agriculture. In: . [S.l.: s.n.], 2007.
- RODRIGUES, M.; VIEIRA, L.; CAMPOS, M. Indoor wireless sensor localization using mobile robot and rssi. *10th International Conference on Mobile Ad-Hoc and Sensor Systems*, IEEE Computer Society, Los Alamitos, CA, USA, v. 0, p. 1–6, 2012.
- RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 3rd. ed. [S.l.]: Pearson, 2009.
- RYBSKI, P.; VELOSO, M. Robust real-time human activity recognition from tracked face displacements. In: BENTO, C.; CARDOSO, A.; DIAS, G. (Ed.). *Progress in Artificial Intelligence*. [S.l.]: Springer Berlin Heidelberg, 2005, (Lecture Notes in Computer Science, v. 3808). p. 87–98. ISBN 978-3-540-30737-2.
- SABRI, N. *et al.* Path loss analysis of wsn wave propagation in vegetation. *Journal of Physics: Conference Series* 423, 2013.
- SAMMONS, P. J.; FURUKAWA, T.; BULGIN, A. Autonomous pesticide spraying robot for use in a greenhouse. *Australasian Conference on Robotics and Automation*, September 2005.
- SANDINI, G. *et al.* Vision-based operations in greenhouses. *Proc. of the 2nd International Workshop on Robotics in Agriculture & the Food Industry*, 1991.
- SCHMITT, T. *et al.* Cooperative probabilistic state estimation for vision-based autonomous mobile robots. *IEEE Transactions on Robotics and Automation*, v. 18, n. 5, p. 670–684, Oct 2002. ISSN 1042-296X.
- SCHÜTZ, C. *et al.* A modular robot system for agricultural applications. In: *International Conference of Agricultural Engineering*. [S.l.: s.n.], 2014.
- SILVA, D. M. da. *Navigo: Robô autônomo para navegação agrícola*. Dissertação (Mestrado) — Universidade Estadual de Santa Catarina - UEPG, 2013.
- SLADE, G. *The viterbi algorithm desmystified*. 2013.
- SRÁMEK, R. *The on-line Viterbi algorithm*. Dissertação (Mestrado) — Faculty of Mathematics, Physics and Informatics Comenius University, Bratislava, 2007.
- STAMP, M. *A revealing introduction to hidden Markov models*. 2004.
- THRUN, S. Learning occupancy grid maps with forward sensor models. *Auton. Robots*, Kluwer Academic Publishers, Hingham, MA, USA, v. 15, n. 2, p. 111–127, set. 2003. ISSN 0929-5593.
- THRUN, S.; BURGARD, W.; FOX, D. *Probabilistic Robotics*. [S.l.]: Cambridge, MA: The MIT Press, 2005.
- THRUN, S. *et al.* A probabilistic approach to concurrent mapping and localization for mobile robots. In: *Machine Learning*. [S.l.: s.n.], 1998. p. 29–53.

- TILLET, N. Robotic manipulators in horticulture: A review. *Journal of Agricultural Engineering Research*, v. 55, n. 2, p. 89 – 105, 1993. ISSN 0021-8634.
- TRAN, K. *et al.* Indoor location prediction using multiple wireless received signal strengths. In: RODDICK, J. F. *et al.* (Ed.). *Seventh Australasian Data Mining Conference*. Glenelg, South Australia: ACS, 2008. (CRPIT, v. 87), p. 187–192.
- UEPG. *Casas de Vegetação incrementam pesquisas na Agronomia*. Fev. 2014. Disponível em: <<http://portal.uepg.br/noticias.php?id=5527>>.
- VITERBI, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, v. 13, n. 2, p. 260–269, April 1967. ISSN 0018-9448.
- WALPOLE, R. *et al.* Probability and statistics for engineers and scientists (8th ed.). In: . NJ, USA: Prentice-Hall, 2006.
- WIDYAWAN; KLEPAL, M.; PESCH, D. Influence of predicted and measured fingerprint on the accuracy of rssi-based indoor location systems. In: *Positioning, Navigation and Communication, 2007. WPNC '07. 4th Workshop on*. [S.l.: s.n.], 2007. p. 145–151.
- XU, J. *et al.* Distance measurement model based on rssi in wsn. *Wireless Sensor Network*, v. 2, p. 606, August 2010.
- YAGHOUBI, S. *et al.* *Autonomous Robots for Agricultural Tasks and Farm Assignment and Future Trends in Agro Robots*. 2014.
- ZHENG, Y. *et al.* Real-time generation of interactive virtual human behaviours. In: *Computer Vision and Computer Graphics. Theory and Applications*. [S.l.]: Springer Berlin Heidelberg, 2009, (Communications in Computer and Information Science, v. 24). p. 70–82. ISBN 978-3-642-10225-7.
- ZHOU, H.; SAKANE, S. Sensor planning for mobile robot localization—a hierarchical approach using a bayesian network and a particle filter. *Robotics, IEEE Transactions on*, v. 24, n. 2, p. 481–487, April 2008. ISSN 1552-3098.
- ZUCCHINI, W.; MACDONALD, I. *Hidden Markov models for time series*. [S.l.]: Chapman and Hall/CRC, 2009.

APÊNDICE A – GRADES DE OCUPAÇÃO COM ARS PLOTADAS

Alguns valores de RSSI gerados para cada ARs , plotados na planta baixa do ambiente de testes, são apresentadas nas Figuras 15(a) e 15(b).

Figura 14: Grades de ocupação G_1 e G_2 com a sobreposição dos pontos de predição (PPs)



APÊNDICE B – PARÂMETROS DO MOM PARA RASTREAMENTO

O Quadro 3 apresenta os parâmetros assumidos para o MOM nos experimentos de rastreamento.

Quadro 3: Parâmetros do MOM das grades de ocupação G_1 e G_2

Parâmetro	G_1	G_2
M	4	4
X	28	64
π	1/28	1/64
$B_{N \times M}$	Para cada célula c , $\mathcal{N}_c(\mu, \sigma)$	Para cada célula c , $\mathcal{N}_c(\mu, \sigma)$
$A_{N \times N}$	$A_{28 \times 28}$	$A_{64 \times 64}$

Na tabela $\mathcal{N}_c(\mu, \sigma)$ é uma distribuição normal onde μ e σ foram computadas a partir da AR de cada célula.

APÊNDICE C – PARÂMETROS DOS MOMS PARA EXPERIMENTOS DE DESEMPENHO

O Quadro 4 apresenta os parâmetros utilizados para o MOM dos experimentos de desempenho.

Quadro 4: Parâmetros do MOM das grades de ocupação GD_1 e GD_2

Parâmetro	GD_1	GD_2
M	4	4
X	315	875
π	1/315	1/875
$B_{N \times M}$	Para cada célula c , $\mathcal{N}_c(\mu, \sigma)$	Para cada célula c , $\mathcal{N}_c(\mu, \sigma)$
$A_{N \times N}$	$A_{315 \times 315}$	$A_{875 \times 875}$

Na tabela $\mathcal{N}_c(\mu, \sigma)$ é uma distribuição normal onde μ e σ foram computadas a partir da AR de cada célula.

APÊNDICE D – AR GERADA PARA A GRADE GD_2 E $\gamma = 3$

A Figura 15 apresenta parte de um arquivo com uma AR gerada para a grade GD_2 utilizando o parâmetro $\gamma = 3$ e sem considerar ruído.

Figura 15: AR da grade GD_2

rssi_osm_3_0_060.csv	
1	ponto;ap1;ap2;ap3;ap4
2	PP1;53.279933456444;48.9804135540368;48.2588018199558;63.951129114961
3	PP1;53.0847630100933;48.5993748339755;49.2658478569029;64.0147709931751
4	PP1;53.4131353740153;49.3927268689327;47.5853534016715;63.7618492378797
5	PP1;53.6893849321929;49.4061685405625;45.991666580549;64.1802965202602
6	PP1;53.0324294473757;48.6266511366851;49.4701196189028;63.9169917786495
7	PP1;53.4820781825237;49.7283652256335;47.3915856964087;63.5360664498948
8	PP1;52.7929320832252;48.9723028131912;50.4481955489768;63.2566450735206
9	PP1;52.9348135788304;49.1636388352119;49.8876675771344;63.2703184092597
10	PP1;53.284342029225;49.0593731819895;48.2242398765785;63.8843585015267
11	PP1;53.2272744694694;49.2228094887825;48.5163747412402;63.6437907088976
12	PP1;52.9423361467097;48.8818592123244;49.8025973640189;63.5547646691449
13	PP1;53.4528514883544;49.2195979077787;47.3480330461859;63.9881664046671
14	PP1;53.425669357029;49.3675790311617;47.5092210288157;63.8055573380967
15	PP1;53.4420948860665;49.7267786902458;47.6271514876972;63.4738529579961
16	PP1;53.4773027130469;49.503577214563;47.2661770434824;63.7534048481742
17	PP1;53.0792135125053;48.8957844528656;49.2013668105712;63.7371393038973
18	PP1;53.2339603662087;49.2935955560047;48.5018345886561;63.5849522575033
19	PP1;53.5429333341738;49.1055960181512;46.8988829571127;64.2309113407303
20	PP1;53.6948161689419;49.4601121912354;45.9571720037266;64.1380886848669
21	PP1;53.2337061741876;48.7944834213643;48.5349983856472;64.0529349881559
22	PP1;53.6645255836261;49.6580311866192;46.2079256691135;63.8990319106991
23	PP1;53.4941411631538;49.2995560805583;47.1218005726661;63.9761729914443
24	PP1;53.1253555016931;49.0904423905918;48.9899310270726;63.6200713690882
25	PP1;53.0544378317549;48.7880864212958;49.3278585578008;63.8014457865354

APÊNDICE E – GERAÇÃO DE DADOS DE CAMINHAMENTO

O coeficiente γ , perda de sinal no caminho¹, foi configurado com os valores extremos, Sabri *et al.* (2013), $\gamma = 3$ e $\gamma = 5$ durante a simulação. As distribuições de emissão e a matriz de transição do MOM para cada uma das velocidades foram definidas a partir das *ARs*. A partir destas informações os caminhamentos foram simulados (Algoritmo 5).

Algoritmo 5: CAMINHAMENTO

Entrada: μ, σ

Dados: $x, y, celula, \{RSSI_1, \dots, RSSI_k\}$

inicio

para cada $i \in 1..600$ **fazer**

repita

$x \leftarrow$ calcula coordenada x

$y \leftarrow$ calcula coordenada y

$celula \leftarrow$ recuperar a célula do ponto(x, y)

até $celula \in \{\text{células válidas}\}$;

 Gera $\{RSSI_1, \dots, RSSI_k\}$ utilizando $\mu[celula]$ e $\sigma[celula]$

 Grava $celula$ e $\{RSSI_1, \dots, RSSI_k\}$

fin

fin

/* O conjunto de células válidas envolve células consideradas

vizinhas, dadas as regras da matriz de transição

*/

Na rotina de geração dos caminhamentos também foi considerado a possibilidade existir ou não ruído no ambiente. Os valores adotados para o ruído, σU , foram de 0 e 1,09. Para cada velocidade, γ e σU foram geradas 30 amostras de caminhada de 10 minutos. Totalizaram-se no final 480 arquivos CSV para cada uma das grades, G_1 e G_2 . Cada arquivo de caminhada continha em cada uma de suas linhas os seguintes valores: *ponto*, x , y , $RSSI_{PA_1}$, $RSSI_{PA_2}$, $RSSI_{PA_3}$ e $RSSI_{PA_4}$ (Apêndice F).

¹Em inglês, *path loss*

APÊNDICE F – ARQUIVO COM DADOS DA SIMULAÇÃO DE CAMINHAMENTO NA GRADE G_1

A Figura 16 apresenta os dados simulados contidos em um arquivo de caminhada para a grade G_1 , com $\gamma = 5$, velocidade de 60 cm/s e sem considerar ruído.

Figura 16: Dados da simulação de caminhada na grade G_1

caminhamentos_100_0.6_5_0_1.csv	
1	PP1;0;6.5228;2.0445;55.28716;51.132008;35.61429;57.005
2	PP1;1;6.374126;2.6257885;57.529804;50.2423;29.239893;55.2451
3	PP2;2;6.3325834;3.2243485;55.541817;47.33807;33.04803;57.654102
4	PP2;3;6.312949;3.8240273;53.32486;44.856594;36.2901;57.980206
5	PP3;4;6.288385;4.4235244;55.833252;41.064415;44.090916;60.041595
6	PP4;5;6.15881;5.009366;56.169407;26.043264;47.259537;62.097157
7	PP5;6;5.559883;5.0452323;59.09667;44.489937;44.111343;59.544796
8	PP6;7;5.84881;4.5193796;52.173336;42.765022;42.1848;58.447227
9	PP7;8;5.775106;3.9239237;52.238636;45.832336;38.87893;52.42043
10	PP7;9;5.775106;3.9239237;50.193962;46.38107;38.477184;53.74868
11	PP7;10;5.9190183;3.3414383;49.577374;47.881348;40.7436;53.212494
12	PP8;11;5.8366637;2.747117;53.903606;52.039806;34.255054;52.927944
13	PP8;12;5.3058586;2.4673917;53.61263;53.861374;32.892994;52.71995
14	PP9;13;4.706062;2.4830146;45.350174;46.59408;42.574455;61.033344
15	PP10;14;4.6927133;3.0828662;47.881172;51.560787;44.628902;51.172535
16	PP10;15;4.6702995;3.6824474;47.533028;49.968292;43.353535;50.687492
17	PP11;16;4.8056407;4.266984;56.151894;61.144016;38.97094;53.69242
18	PP11;17;4.880719;4.862268;48.833378;53.508984;45.702515;54.704037
19	PP12;18;4.941505;5.459181;54.08305;54.982124;40.841915;55.263245
20	PP12;19;4.36755;5.6340404;56.25919;52.605553;40.97826;56.089596
21	PP13;20;3.7845545;5.492208;48.901226;57.42306;44.558582;49.46411
22	PP14;21;3.7954493;4.892307;53.117607;58.331783;40.080547;46.650658
23	PP14;22;3.8299687;4.2933006;53.04181;59.692604;40.125008;48.36674
24	PP15;23;3.6457195;3.7222908;53.907963;50.999832;55.350876;51.381725
25	PP15;24;3.8554022;3.1601224;50.67836;59.34581;39.529194;37.912865
26	PP16;25;3.8198957;2.561174;40.902077;47.122063;58.242905;54.699776