



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
IFCE *CAMPUS* FORTALEZA
PRÓ-REITORIA DE PESQUISA, PÓS-GRADUAÇÃO E INOVAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELECOMUNICAÇÕES
MESTRADO ACADÊMICO EM ENGENHARIA DE TELECOMUNICAÇÕES

FRANCISCO ALAN XAVIER DA MOTA

PROPOSTA DE UM MÉTODO DE LOCALIZAÇÃO E MAPEAMENTO
SIMULTÂNEOS PARA ROBÔS MÓVEIS EM AMBIENTES INTERNOS BASEADO EM
VISÃO COMPUTACIONAL E REDES DE PETRI

FORTALEZA

2023

FRANCISCO ALAN XAVIER DA MOTA

**PROPOSTA DE UM MÉTODO DE LOCALIZAÇÃO E MAPEAMENTO
SIMULTÂNEOS PARA ROBÔS MÓVEIS EM AMBIENTES INTERNOS BASEADO EM
VISÃO COMPUTACIONAL E REDES DE PETRI**

Dissertação submetida à Coordenação de Pós-Graduação em Engenharia de Telecomunicações do Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE) - *Campus* Fortaleza, como requisito parcial para obtenção do Título de Mestre em Engenharia de Telecomunicações. Área de concentração: Processamento Digital de Sinais

Orientador: Prof. Dr. Auzuir Ripardo de Alexandria

FORTALEZA

2023

Dados Internacionais de Catalogação na Publicação
Instituto Federal do Ceará - IFCE
Sistema de Bibliotecas - SIBI
Ficha catalográfica elaborada pelo SIBI/IFCE, com os dados fornecidos pelo(a) autor(a)

M917p Mota, Francisco Alan Xavier da.
Proposta de um Método de Localização e Mapeamento Simultâneos para Robôs Móveis em Ambientes Internos Baseado em Visão Computacional e Redes de Petri / Francisco Alan Xavier da Mota. - 2024.
66 f. : il.

Dissertação (Mestrado) - Instituto Federal do Ceará, Mestrado em Engenharia de Telecomunicações, Campus Fortaleza, 2024.

Orientação: Prof. Dr. Auzuir Ripardo de Alexandria.

1. Robótica Móvel. 2. Rede de Petri. 3. Visão Computacional. 4. SLAM. I. Título.

CDD 621.382

FRANCISCO ALAN XAVIER DA MOTA

**PROPOSTA DE UM MÉTODO DE LOCALIZAÇÃO E MAPEAMENTO
SIMULTÂNEOS PARA ROBÔS MÓVEIS EM AMBIENTES INTERNOS BASEADO EM
VISÃO COMPUTACIONAL E REDES DE PETRI**

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia de Telecomunicações, sendo aprovada pelo Programa de Pós-Graduação em Engenharia de Telecomunicações do Instituto Federal de Educação, Ciência e Tecnologia do Ceará e pela banca examinadora:

Aprovado em: ____/____/____.

BANCA EXAMINADORA

Prof. Dr. Auzuir Ripardo de Alexandria
(Orientador)
Instituto Federal de Educação, Ciência e
Tecnologia do Ceará (IFCE)

Prof. Dr. Glendo de Freitas Guimarães
Instituto Federal de Educação, Ciência e
Tecnologia do Ceará (IFCE)

Prof. Dr. Giovanni Cordeiro Barroso
Universidade Federal do Ceará (UFC)

Dedico esse trabalho a Deus, o Criador de todas as coisas, aos meus queridos pais, ao meu irmão, à minha esposa, aos meus amigos e ao meu orientador.

AGRADECIMENTOS

A Deus, o Criador de todas as coisas, meu Pai, meu Mestre, meu Senhor e Salvador.

Aos meus pais Antônio e Édna, e ao meu irmão David, pelo incentivo, amizade e por sempre estarem dispostos a me ajudarem, por tudo que têm providenciado para mim ao longo de toda minha caminhada.

A minha esposa Beatriz, sobretudo por dedicar seu tempo e sua escuta em cada momento de nossa jornada.

Aos professores, que muito contribuíram com minha formação acadêmica, em especial ao meu orientador e amigo, prof. Auzuir, que ao longo dessa última década tem me conduzido no universo da pesquisa e tem me orientado na minha formação acadêmica desde minha graduação.

Ao PPGET, Programa de Pós-Graduação em Engenharia de Telecomunicações, em especial coordenação e professores por todo suporte, parceria e conhecimento proporcionado.

Ao IFCE, Instituto Federal de Educação, Ciência e Tecnologia do Ceará, por proporcionar, além de uma sólida construção do conhecimento, inúmeras oportunidades profissionais. Especialmente, a toda a equipe do Laboratório de Ensaios Mecânicos (LEM), que nos últimos 12 anos me acolheram e me ajudaram inúmeras vezes.

À CAPES, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, pelo apoio financeiro prestado ao longo deste mestrado em Engenharia de Telecomunicações.

“Porque o Senhor é o que dá a Sabedoria; da
Sua boca é que sai o conhecimento e o entendi-
mento.”

(Provérbios 2:6)

RESUMO

Os estudos na área de robótica móvel têm avançado nos últimos anos, sobretudo devido à evolução da tecnologia e à necessidade crescente de soluções automatizadas e dinâmicas em setores como indústria, transportes e agricultura. Precisão, rapidez e eficiência são características cada vez mais exigidas e descrevem os principais atributos de um robô móvel. Entretanto, esses dispositivos são complexos e, apesar da vasta literatura publicada, o método ideal para localização, mapeamento do ambiente explorado e navegação dos robôs móveis autônomos muda conforme a aplicação. Considerando este contexto, o objetivo geral deste trabalho é propor um método de localização e mapeamento simultâneos para robôs móveis autônomos em ambientes internos, utilizando Visão Computacional (VC) e Rede de Petri (RP). O algoritmo foi desenvolvido em Python, utilizando a biblioteca *Snakes* e o plugin *gv* (Graphviz) para gerar a RP e a biblioteca *OpenCV* para manipulação e tratamento das imagens capturadas no ambiente. Ao lado de cada porta da região analisada foi colocada uma baliza e são adquiridas imagens à medida que os cômodos do ambiente são explorados. O algoritmo faz o processamento das imagens obtidas, utilizando técnicas de visão computacional, para contabilizar e identificar as portas presentes na região analisada. Após essa etapa, é criada uma transição na RP para cada porta encontrada. Os cômodos conectados por essas portas são representados pelos lugares da RP. Em seguida, uma das portas é atravessada, novas imagens são obtidas e o processo se repete até que todos os cômodos sejam explorados. Ao fim do processo de navegação, o algoritmo gera uma rede de Petri, que pode ser representada por um arquivo de imagem (.png) ou por um arquivo na extensão .pnml. Os resultados comparam o *layout* de cinco ambientes com as respectivas RPs geradas e mostram ser possível modelar um mapa topológico de um ambiente utilizando esta técnica, bem como é possível se localizar nesse mapa utilizando visão computacional. Ainda, são propostos seis critérios avaliativos para validação das redes de Petri enquanto mapa topológico de ambientes. Conclui-se que utilizar RP para esta finalidade apresenta originalidade e potencial inovação, sendo uma técnica de SLAM para ambientes internos, que demanda baixo custo computacional.

Palavras-chave: robótica móvel. rede de Petri. visão computacional. SLAM.

ABSTRACT

Researches in the field of mobile robotics have advanced in recent years, primarily due to the evolution of technology and the growing need for automated and dynamic solutions in sectors such as industry, transportation, and agriculture. Accuracy, quickness, and efficiency are increasingly demanded characteristics that describe the main attributes of a mobile robot. However, these devices are complex, and despite the extensive published literature the ideal method for localization, mapping of the explored environment and navigation of autonomous mobile robots changes according to the application. Considering this context, the aim of this work is to propose a simultaneous localization and mapping method for autonomous mobile robots in indoor environments using Computer Vision (CV) and Petri Net (PN). The algorithm was developed in Python, using the *Snakes* library and the *gv* plugin (Graphviz) to generate the RP and the *OpenCV* library for manipulation and treatment of images captured in the environment. Next to each door in the analyzed region, a landmark was placed and images are acquired as the rooms of the environment are explored. The algorithm processes the images obtained, using computer vision techniques, to count and identify the ports present in the analyzed region. After this, a transition is created in the PN for each door found. The rooms connected by these doors are represented by the places of the PN. Then, one of the doors is crossed, new images are obtained and the process is repeated until all the rooms are explored. At the end of the navigation process, the algorithm generates a Petri net, which can be represented by an image file (.png) or by a file with the extension .pnml. The results compare the *layout* of five environments with the respective PNs generated and show that it is possible to model a topological map of an environment using this technique, as well as it is possible to locate yourself in this map using computer vision. Additionally, six evaluation criteria are proposed to validate Petri nets as a topological map of environments. It is concluded that using PN for this purpose presents originality and potential innovation, being a SLAM technique for indoor environments, which demands low computational cost.

Keywords: mobile robotic. Petri net. computer vision. SLAM.

LISTA DE ILUSTRAÇÕES

Figura 1 – Carro autônomo Waymo driver.	13
Figura 2 – Exemplos de robô do tipo manipulador fixo utilizado (a) em linhas de produção e (b) para deslocamento de objetos.	14
Figura 3 – Exemplo de robô móvel (a) explorador de ambientes e (b) impressora de estruturas de concreto.	14
Figura 4 – Principais métodos para localização e mapeamento utilizados por robôs móveis.	16
Figura 5 – Módulos de um robô móvel típico.	21
Figura 6 – Modelo gráfico de uma Rede de Petri.	23
Figura 7 – Composição de um sistema de visão computacional típico.	25
Figura 8 – Material e Métodos.	32
Figura 9 – Material.	33
Figura 10 – Etapas do processo de mapeamento do ambiente por RP.	34
Figura 11 – Metodologia utilizada no mapeamento do ambiente.	35
Figura 12 – Porta com baliza e QRcode.	38
Figura 13 – Imagem 640x480 pixels (a) na escala HSV, (b) pós processo de dilatação e com filtro vermelho aplicado e (c) pós processo de binarização.	38
Figura 14 – Imagem do ambiente explorado (a) na escala RGB e (b) segmentada.	42
Figura 15 – Taxa de sucesso na leitura de QRcodes em diferentes ângulos.	42
Figura 16 – Representação do ambiente simples mapeado em (a) Planta baixa e (b) sua respectiva rede de Petri.	44
Figura 17 – Representação do modelo 1 de um ambiente complexo em (a) Planta baixa e (b) sua respectiva rede de Petri.	45
Figura 18 – Representação do modelo 2 de um ambiente complexo em (a) Planta baixa e (b) sua respectiva rede de Petri.	46
Figura 19 – Representação do modelo 3 de um ambiente complexo em (a) Planta baixa e (b) sua respectiva rede de Petri.	47
Figura 20 – Representação do modelo 4 de um ambiente complexo em (a) Planta baixa e (b) sua respectiva rede de Petri.	48

LISTA DE TABELAS

Tabela 1 – Estado da arte sobre Robótica Móvel associada a Rede de Petri e Visão Computacional.	30
Tabela 2 – Métricas Avaliativas para os Experimentos II e III.	49
Tabela 3 – Comparação entre o método proposto nesta pesquisa e demais métodos para localização e mapeamento.	51

LISTA DE ABREVIATURAS E SIGLAS

IFR	<i>International Federation of Robotics</i>
AGV	<i>Automated Guided Vehicle</i>
QRcode	<i>Quick Response Code</i>
RFID	Identificação por Rádio Frequência
SLAM	<i>Simultaneous Localization and Mapping</i>
RP	Redes de Petri
NASA	Agência Espacial Norte-Americana
LIDAR	<i>Light Detection and Ranging</i>
BCI	<i>Brain-Computer Interface</i>
AUV	<i>autonomous underwater vehicle</i>
VC	Visão Computacional
ZED	Câmera Estéreo Binocular
ATE	<i>Average Trajectory Error</i>
VIO	<i>Visual Inertial Odometry</i>
MCR	<i>Map Change Rate</i>
CRI	<i>Correctness Rate of Initialization</i>
MRCL	<i>Mileage Rate of Correct Localization</i>
AMS	<i>Average Matching Score</i>
IMU	Unidade de Medição Inercial
LOAM	Odometria e Mapeamento em Tempo Real Usando LIDAR
ROS	Sistema Operacional Robótico
ALM	Algoritmo de Localização e Mapeamento
AIP	Algoritmo de Identificação das Portas

LISTA DE SÍMBOLOS

p_x	Lugar x de uma rede de Petri
t_x	Transição x de uma rede de Petri
E_i	Estado i de uma rede de Petri
M_i	Matriz de marcações de uma rede de Petri
P	Conjunto finito de lugares de uma rede de Petri
T	Conjunto finito de transições de uma rede de Petri
A	Matriz de incidência de uma rede de Petri
a_{ij}^+	Peso do arco de saída que conecta a transição i ao lugar j
a_{ij}^-	Peso do arco de entrada que conecta o lugar j à transição i
id	Identificação
t_d	Matriz de transições disparadas
$idPE$	Índice da porta de entrada
lv	Vetor lugares visitados
$lugAtual$	Lugar da rede de Petri onde o robô está

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Motivação	17
1.2	Objetivos	17
1.3	Escopo e Limitações	17
1.4	Contribuições	18
1.5	Organização da Dissertação	18
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	Robôs Móveis	19
2.1.1	Definição	19
2.1.2	Aplicações Gerais	20
2.1.3	Subsistemas	20
2.1.4	Localização e Mapeamento Simultâneos	21
2.2	Redes de Petri	22
2.2.1	Definição	22
2.2.2	Aplicações Gerais	24
2.3	Visão Computacional	25
2.3.1	Definição	25
2.3.2	Aplicações Gerais	26
2.4	Estado da Arte	26
2.5	Considerações Finais do Capítulo	31
3	MATERIAL E MÉTODOS	32
3.1	Material	32
3.1.1	Biblioteca SNAKES	32
3.1.2	Biblioteca OpenCV	33
3.2	Método de Construção da Rede de Petri	33
3.2.1	Identificação das Portas no Ambiente	37
3.3	Método de Testes	39
3.4	Métricas de Avaliação	39
3.5	Considerações Finais do Capítulo	40
4	RESULTADOS E ANÁLISE DAS MÉTRICAS DE AVALIAÇÃO	41
4.1	Experimento I: Algoritmo de Identificação das Portas	41
4.2	Experimento II: Algoritmo de Geração do Mapa de Ambientes Simples	43
4.3	Experimento III: Algoritmo de Geração do Mapa de Ambientes Complexos	43
4.4	Discussões	45

5	CONCLUSÃO E TRABALHOS FUTUROS	53
	REFERÊNCIAS	54
	APÊNDICE A – ALGORITMOS	64

1 INTRODUÇÃO

A evolução tecnológica é um processo que ocorre em todo o mundo e atinge diversos ramos de estudo, como a Robótica, cujas aplicações têm expandido e impactam cada vez mais o cotidiano das pessoas (FENERICK; VOLANTE, 2020). Os estudos na área de Robótica envolvem conteúdos como eletrônica, mecânica, programação, inteligência artificial, desenvolvimento e aprimoramento de robôs e seus subsistemas.

Segundo dados da IFR (2021), do inglês, *International Federation of Robotics*, existem mais de 3 milhões de robôs realizando serviços em fábricas ao redor do mundo. Em 2021, o crescimento de unidades instaladas foi de 10%, tendo sido investido mais de US\$13 bilhões nos últimos anos nesse setor, sobretudo por países como EUA, Japão, Coreia do Sul, Alemanha e China. Mundialmente, estima-se que esse investimento alcance US\$260 bilhões em 2030, sendo US\$30 bilhões em robôs do tipo AGV (do inglês, *automated guided vehicle*) (LÄSSIG MARKUS LORENZ; BUCHNER, 2021).

No Brasil, seguindo a tendência global, a Robótica segue em expansão, mas de forma lenta. Apesar de, em 2019, o Brasil possuir o maior estoque operacional de robôs da América do Sul, com 15,3 mil unidades (PEREIRA, 2020), a quantidade de robôs instalados é menor que 1%, quando comparada com os demais países do mundo. Estima-se que haja 10 robôs para cada 10 mil operários, no Brasil, enquanto em países como a Coreia do Sul, esse número é 50 vezes maior (FENERICK; VOLANTE, 2020).

Entre as diversas aplicações de robôs, uma das mais evidentes que impacta diretamente a dinâmica do trânsito são os carros autônomos, conforme representado na Figura 1, automóveis equipados com sensores diversos capazes de se deslocar pela cidade com critérios de segurança, eficiência e com mínima ou nenhuma intervenção humana (SCHWALL et al., 2020).

Figura 1 – Carro autônomo Waymo driver.

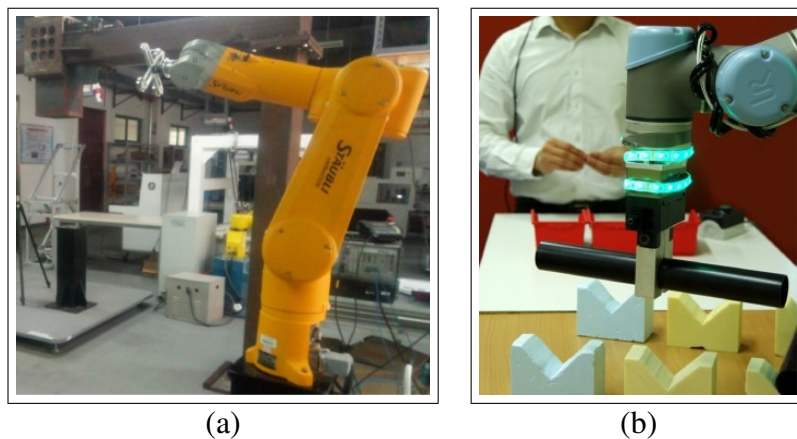


Fonte: Schwall et al. (2020).

Robô é todo dispositivo mecatrônico capaz de ser controlado remotamente ou programado para exercer alguma função de forma autônoma (MOTA et al., 2018). Nesse contexto, quanto à locomoção, os robôs podem ser classificados como manipuladores fixos e robôs móveis. Respectivamente, eles podem executar funções sem haver movimentos de translação no ambiente, enquanto os demais podem se deslocar de um ponto a outro para realizar alguma atividade (SIEGWART; NOURBAKHS, 2004).

Existem manipuladores fixos utilizados em linhas de produção (BANAS et al., 2015) nos mais variados setores da indústria, tendo em vista que aumentam a produtividade e reduzem os custos de produção (URREA; KERN; ALVARADO, 2020). São utilizados, ainda, na indústria automobilística (BARTOŠ et al., 2021; FENG et al., 2020), deslocam peças de uma esteira para outra em uma linha de produção industrial (NEJAD et al., 2019), auxiliam na execução de cirurgias (GOTO, 2022), classificam objetos (ALI et al., 2018) e erguem objetos pesados (QIN et al., 2022; ENGEMANN et al., 2020). Na Figura 2 são representados modelos de manipuladores fixos.

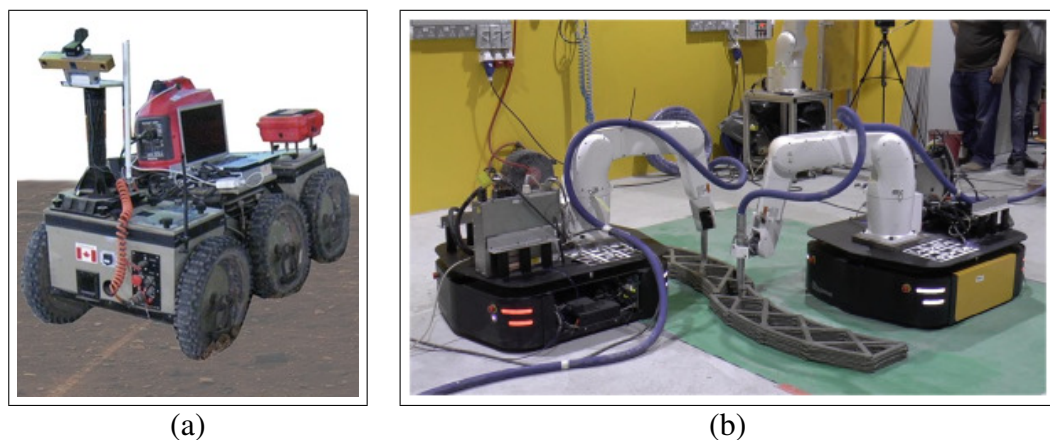
Figura 2 – Exemplos de robô do tipo manipulador fixo utilizado (a) em linhas de produção e (b) para deslocamento de objetos.



Fonte: (a) Chen, Zhang e Sun (2019) e (b) Tang, Webb e Thrower (2019).

Com relação aos robôs móveis, atuam em ações logísticas (FRAGAPANE et al., 2021), monitoram plantações e incêndios florestais (SANTOS et al., 2020), exploram ambientes (GUL et al., 2021), desarmam bombas (AZIM et al., 2021), entretêm (NEERINCX; HIWAT; GRAAF, 2021), realizam serviços domésticos (LEE, 2021), entre outras aplicações. Na Figura 3 são representados modelos de robôs móveis.

Figura 3 – Exemplo de robô móvel (a) explorador de ambientes e (b) impressora de estruturas de concreto.



Fonte: (a) Ostafew et al. (2015) e (b) Zhang et al. (2018).

Os robôs móveis podem atuar em ambientes externos (*outdoor*) e em ambientes internos (*indoor*) (KHATIB; JARADAT; ABDEL-HAFEZ, 2020). Ambientes *outdoor* exigem robôs

mais robustos, devido à dinamicidade do ambiente como condições climáticas, luminosidade, deformações no solo e outras instabilidades (NILOY et al., 2021). Por outro lado, ambientes *indoor* apesar de, em geral, serem estruturados e mais estáveis com relação à disposição de objetos, luminosidade e tipo de solo, apresentam problemas críticos como necessidade de precisão na localização do robô (DOPPLER et al., 2022). Estas características não possuem tanta importância quando esses ambientes são navegados por robôs controlados remotamente, entretanto são relevantes para robôs móveis autônomos ou veículos autoguiados (AGVs).

Considerando o critério autonomia, os robôs podem ser autônomos (ZGHAIR; AL-ARAJI, 2021), guiados (MONIRUZZAMAN et al., 2022) e programáveis (AJAYKUMAR; STEELE; HUANG, 2022). Robôs autônomos são equipados com diversos sensores que auxiliam no desenvolvimento de suas tarefas de forma independente. É comum essa categoria possuir sensores que auxiliam na tomada de decisões para desvio de obstáculos, identificação de ambientes e orientação de trajetórias. Assim, a partir dos dados coletados pelos sensores, conseguem definir ações para cumprir a tarefa a que foram destinados sem necessitar de intervenções externas. Sobre os robôs guiados, conseguem realizar atividades, mas sob o controle constante de um usuário. São dispositivos mecatrônicos dependentes e, portanto, limitados. Por fim, os robôs programáveis são aqueles programados para realizarem uma sequência de ações previamente definidas por um usuário. Os principais exemplos desses tipos de robôs são aqueles que executam atividades repetitivas.

Siegwart e Nourbakhsh (2004) apresentam uma possível arquitetura de um robô móvel constituída pelos módulos de Percepção, Localização (construção do mapa), Cognição (planejamento de rota) e Controle de Movimento.

Sobre o módulo de percepção de robôs móveis, diversas técnicas são aplicadas para permitir que o robô extraia informações do ambiente e então se localize. Dentre elas, destaca-se o uso da visão computacional. Esta técnica possui como principal objetivo extrair características descritivas do meio externo por meio de imagens ou por uma sequência destas (FORSYTH; PONCE, 2011). A visão computacional aplicada aos sistemas robóticos, acrescenta robustez, maior número de técnicas para tratamento de imagens e possibilita maior eficiência para a execução de tarefas. No entanto, o custo computacional é elevado, devido aos componentes, que dependem diretamente da finalidade e podem apresentar elevado custo de aquisição, além de determinados algoritmos para o processamento de imagens exigirem alto desempenho de *hardware*. A visão computacional é largamente utilizada e é o foco de várias pesquisas acadêmicas (KUMAR et al., 2018; WEI et al., 2018; ONITA et al., 2018).

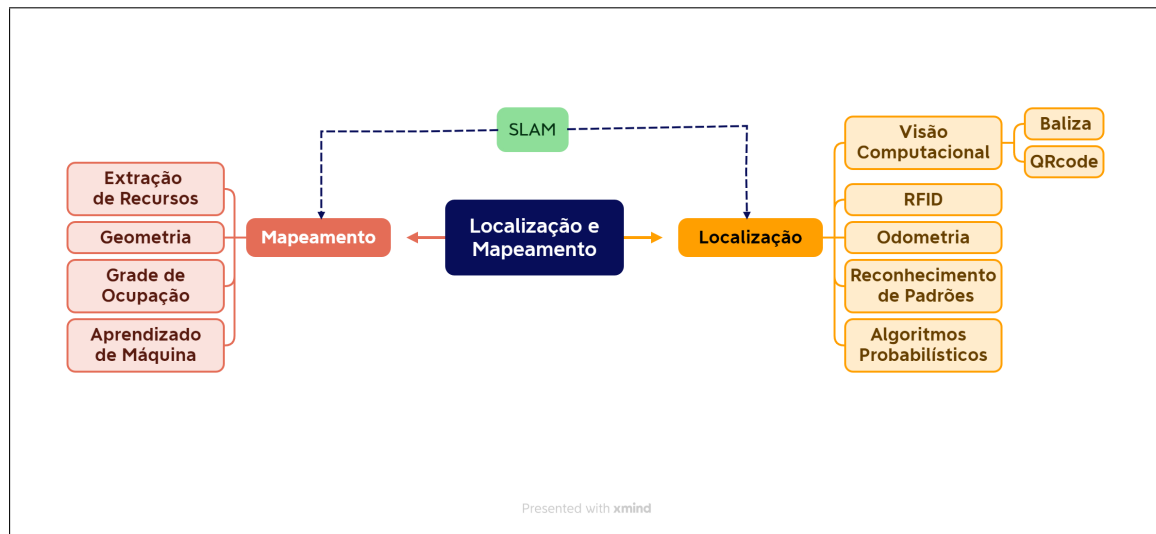
De posse dos dados provenientes do módulo de percepção, torna-se possível localizar-se. Na Figura 4 são representados os principais métodos citados na literatura para localização e mapeamento de robôs móveis.

A localização utilizando balizas, geralmente utilizada em conjunto com visão computacional, consiste em identificar a distância e o ângulo do robô em relação a um objeto (referência) conhecido (MORENO; HUSSEIN; GARCIA, 2021). A localização utilizando QRcode (do inglês, *quick response code*) é semelhante a este conceito, tendo em vista que o robô necessita de uma câmera para identificar os QRcode posicionados no ambiente (YANG et al., 2020b).

Outra técnica de localização utilizada por robôs móveis é a Identificação por Rádio Frequência (RFID). Neste modelo, o robô possui um leitor RFID e busca *tags* RFID espalhadas pelo ambiente para se localizar, conforme demonstra (MOTA et al., 2018).

Na literatura é citada também a localização por odometria baseada na distância percorrida pelo robô desde o seu ponto de partida (SUN, 2022). Entretanto, pode apresentar erros significa-

Figura 4 – Principais métodos para localização e mapeamento utilizados por robôs móveis.



Fonte: Elaboração Própria.

tivos, devido ao processo de derrapagem que ocorre com os mecanismos de locomoção, como as rodas, acumulando erro a cada leitura do sensor responsável pela quantificação do deslocamento.

O reconhecimento de padrões refere-se à capacidade de identificar e classificar padrões em um conjunto de dados. De posse dessas informações, por exemplo, um robô pode localizar-se no ambiente (KALINOV et al., 2019).

A localização baseada em algoritmos probabilísticos, como Monte Carlo, consiste em considerar a posição de um robô a partir da teoria da probabilidade, minimizando, assim, os erros provocados por ruídos na leitura dos sensores e outras incertezas na medição (SUN, 2022).

A técnica de Localização e Mapeamento Simultâneos (SLAM) consiste na construção de um mapa de um ambiente explorado por um robô a medida que este se localiza no ambiente (LLUVIA; LAZKANO; ANSUATEGI, 2021).

Sobre o mapeamento e o controle de movimento, eles podem ser desenvolvidos a partir de um conhecimento prévio do ambiente navegado ou à medida que o ambiente é explorado.

Mota et al. (2018) utilizam Redes de Petri (RP) para este fim. Esta técnica consiste na modelagem de Sistemas a Eventos Discretos, ou seja, sistemas que mudam de estado a partir da ocorrência de um evento. Assim, essa modelagem pode ser feita de forma matemática ou gráfica. É utilizada na modelagem de processos na área de Administração (DENIGRES; KOSE, 2018; MUŠIČ, 2021), Robótica (MOTA et al., 2018; KOTB; ABI-CHAR, 2017), Educação (LI; YU; HU, 2019), Indústria (SIMON; TEIXEIRA; da Costa, 2022; ZHOU; RENIERS, 2022), Mecânica (LU et al., 2019; FREITAS; BASILIO, 2021), Computação (YIN; FANG; SHAO, 2019; KAVITHA et al., 2022), Saúde (YU et al., 2020; GUPTA; FATIMA; KUMAWAT, 2021) entre outras aplicações.

O mapeamento baseado em extração de recursos ocorre a partir da detecção e identificação de características do ambiente, como linhas, bordas e superfícies pelos sensores do robô (TARDÓS et al., 2002). Esse método é utilizado em situações onde não há necessidade de um mapa detalhado. Enquanto no processo de criação de mapas baseado na geometria da região explorada são utilizados, por exemplo, lasers e câmeras 3D, para fornecer informações precisas

sobre o ambiente e a distância entre os objetos contidos na cena (THRUN; BURGARD; FOX, 2000).

O mapeamento baseado em grades de ocupação consiste em representar um ambiente por uma grade e atribuir um valor, que indica a probabilidade de ocupação, a cada célula dessa grade (THRUN, 2003).

Ainda sobre a criação de mapas, tem-se a abordagem baseada em aprendizado de máquina, em que um algoritmo recebe dados dos sensores do robô, constrói conhecimento sobre o ambiente e gera um mapa baseado nesse aprendizado (CHAVES et al., 2019).

Por fim, com relação ao planejamento de rotas, esse módulo está relacionado ao caminho pelo qual o robô deve navegar para sair de sua posição inicial e chegar ao seu destino. Considerando um mapa representado por uma Rede de Petri, o planejamento de rotas pode se dar a partir das matrizes de incidência desta RP (MOTA et al., 2018; GUNARDI et al., 2019).

1.1 Motivação

A robótica tem expandido e alcançou diversos setores da sociedade. Ainda, com o surgimento de novas tendências como a Inteligência Artificial, Indústria 4.0 e a internet 5G, a perspectiva é de que esse campo das ciências receba investimentos maiores a cada ano.

Dentre os tipos de robôs, aqueles que conseguem se deslocar de um lugar a outro no ambiente (robôs móveis) se sobressaem devido a sua versatilidade nos mais variados campos da indústria.

Entretanto, o problema da localização de robôs em ambientes *indoor*, bem como o mapeamento de ambientes desconhecidos, se destacam por, apesar de sua relevância, não possuírem uma solução definitiva na literatura, mas diversas soluções que mudam conforme a aplicação. Neste caso, propõem-se uma técnica com originalidade e potencial inovação, como solução para o problema supracitado, que demanda baixo custo computacional.

1.2 Objetivos

O objetivo geral deste trabalho é propor um método de localização e mapeamento simultâneos para robôs móveis autônomos em ambientes internos. A localização é baseada em imagens capturadas do ambiente explorado e o mapeamento deste ambiente é baseado em Redes de Petri (RP). Assim, são necessários os seguintes objetivos específicos:

- Desenvolver o algoritmo de identificação das portas em um ambiente interno;
- Desenvolver o algoritmo de construção da rede de Petri;
- Implementar e avaliar o algoritmo de SLAM;
- Avaliar o algoritmo em diferentes modelos de ambientes *indoor*.

1.3 Escopo e Limitações

De posse das classificações supracitadas e em coerência com o objetivo principal desta pesquisa, as técnicas propostas consideram um robô móvel, autônomo que atua em ambiente *indoor*. Este trabalho explora o módulo de localização com construção simultânea do mapa do ambiente.

Durante a realização dos testes e a validação do algoritmo, considera-se que: o mapa inicialmente está vazio, ou seja, o mapa é construído à medida que o ambiente é explorado; todas as portas dos cômodos estão abertas; as imagens foram adquiridas manualmente, ou seja, uma pessoa posicionou uma câmera em frente às portas de cada cômodo do ambiente navegado; este trabalho limita-se ao mapeamento de ambientes *indoor*, portanto, nos testes, não são utilizadas balizas nas portas que conectam o ambiente explorado ao meio externo.

1.4 Contribuições

A partir da efetivação dos objetivos mencionados, as principais contribuições deste trabalho são: originalidade quanto ao método proposto, que utiliza Redes de Petri (RPs) para mapeamento de ambientes indoor, bem como a possibilidade do uso da dinâmica das RPs para condução da navegação de um robô por um ambiente desconhecido. Além disso, propõem-se métricas que podem ser utilizadas para avaliar RPs, neste estudo e em pesquisas posteriores, que representem o mapa de um ambiente explorado.

Nesse contexto, considerando as contribuições mencionadas, este estudo desempenha um papel de destaque em Robótica Móvel. Apresenta-se um novo método de modelagem de ambientes que se destaca pela sua baixa demanda computacional e demonstra alta eficiência e intuitividade. Essa abordagem inovadora oferece uma solução promissora para a área, proporcionando uma alternativa acessível para a criação de mapas de ambientes internos.

1.5 Organização da Dissertação

Esta pesquisa está dividida em cinco capítulos. No primeiro capítulo é apresentada uma abordagem geral sobre robôs móveis e seus subsistemas. Enquanto no segundo capítulo são apresentados conceitos fundamentais sobre Robótica Móvel, Redes de Petri e Visão Computacional necessários para o desenvolvimento da pesquisa, além do estado da arte sobre a temática em questão. No terceiro capítulo são descritos o material e os métodos utilizados para o desenvolvimento desta pesquisa. No quarto capítulo são apresentados os resultados obtidos e são realizadas discussões sobre as técnicas utilizadas neste trabalho. Por fim, a conclusão da pesquisa e possibilidades de trabalhos futuros são apresentadas no último capítulo.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção contém informações relacionadas às definições e às características de Robótica Móvel, Redes de Petri e Visão Computacional. São apresentadas, ainda, informações sobre o Estado da Arte, relacionando a aplicação dos temas supracitados em robôs móveis.

2.1 Robôs Móveis

O ser humano se destaca por sua inteligência. Ações básicas realizadas no dia a dia são atributos desta habilidade como ver, se localizar, analisar, tomar decisões. Nesse contexto, conforme Zhang e Lu (2021), a inteligência artificial está relacionada com a inserção de atributos inteligentes a uma máquina, proporcionando, pois, que ela possa ser treinada para simular comportamentos humanos como aprender, julgar e classificar.

A inteligência artificial é aplicada em diversas áreas do conhecimento, sendo uma delas a Robótica (SARKER et al., 2021), que aborda o desenvolvimento de robôs e projetos mecatrônicos. Os robôs são dispositivos utilizados, sobretudo, para a execução de tarefas repetitivas (LIANG; KAMAT; MENASSA, 2020), insalubres (SMITH; CUCCO; FAIRBAIRN, 2020), atividades que exigem alto nível de precisão como cirurgias robóticas (HOLSINGER; BIRKELAND; TOPF, 2021), exploração de ambientes (LI, 2020) entre outras.

2.1.1 Definição

Neste trabalho, considera-se robô móvel todo dispositivo composto por uma estrutura mecânica e por circuitos eletrônicos que pode ser programado para realizar movimentos em um ambiente.

Uma das aplicações mais comuns da inteligência artificial na Robótica está presente nos robôs móveis autônomos (AGVs) (TSOLAKIS et al., 2021), na capacidade destes dispositivos captarem dados do ambiente em que estão inseridos, analisá-los e tomarem decisões a partir dos dados processados.

Os AGVs se destacam por sua versatilidade e suas inúmeras aplicações, sendo capazes de se deslocar para realizar alguma tarefa sem a intervenção humana (RUBIO; VALERO; LLOPIS-ALBERT, 2019). A importância desse tipo de robô é constatada, por exemplo, em situações em que vidas humanas estão expostas a perigos iminentes.

Um dos pontos críticos para o desenvolvimento satisfatório da navegação de robôs está relacionado com a plataforma móvel, ou seja, com o mecanismo responsável pelo deslocamento, que deve se adequar ao ambiente navegado (SUN et al., 2023). Para este fim, destaca-se o uso de rodas uni (KLANCAR et al., 2017) e omnidirecionais (TAHERI; ZHAO, 2020), pernas (REHMAN et al., 2016), esteiras (BIETRESATO et al., 2016), ventosas (PRADOS et al., 2022), além de drones (LI; SUN; CAI, 2019), barcos robóticos (LEONARD; BAHR, 2016) e submarinos (RAWASHDEH; ABU-ALRUB, 2021).

Farooq, Eizad e Bae (2023) descrevem que robôs móveis pertencem à classe dos robôs dotados da capacidade de navegar entre ambientes para a realização de uma tarefa. Zhang, Lin e Chen (2018) complementam que o planejamento de rotas é um dos problemas fundamentais para uma navegação segura e eficiente em ambientes complexos, tendo em vista que robôs do tipo móvel possuem um problema intrínseco que é o desvio de obstáculos.

2.1.2 Aplicações Gerais

No acidente nuclear de Fukushima, Japão, em 2011, e até hoje, robôs móveis são utilizados para explorar o ambiente interno das usinas e reatores nucleares, a fim de captar imagens, rastrear a origem do vazamento de radiação, bem como adquirir o maior número de dados possíveis para auxiliar na tomada de decisão das equipes que monitoram a região (NAGATANI et al., 2012).

Robôs móveis autônomos podem ser utilizados para explorar, também, ambientes externos. É o caso dos robôs da Agência Espacial Norte-Americana (NASA), como os robôs geólogos *Spirit* e *Opportunity*, que aterrissaram no planeta Marte em 2004, tendo sido enviados com a missão de explorar e observar a geologia da superfície marciana e coletar dados atmosféricos. Já em 2011, o *Curiosity* foi lançado para responder à pergunta: “As condições ambientais de Marte permitem a existência de pequenas formas de vida (chamadas micróbios)?”. Além deles, em julho de 2020, a NASA enviou também o *Perseverance*, que pousou em Marte em fevereiro de 2021. A proposta deste robô explorador é coletar amostras de uma extensa cratera, que teria sido um lago há cerca de 3,5 milhões de anos, na superfície do “planeta vermelho”, além de observar a geologia e realizar testes para validar a hipótese de converter dióxido de carbono em oxigênio e assim possibilitar uma missão humana e, posteriormente, a colonização do planeta.

Ainda sobre as aplicações dos robôs móveis, durante a pandemia de COVID-19, em 2020, foram usados para desinfetar o ar da China, desinfecção de locais como hospitais e, no Japão, para levar o medicamento até os pacientes (YANG et al., 2020a; ANDERSON; MCKEE; MOSSIALOS, 2020; REN; DING; MA, 2020).

Sun et al. (2023) destacam a crescente demanda e tendência de se utilizar robôs móveis na agricultura, tendo em vista a modernização do campo, a eficiência na execução das tarefas feitas por robôs, bem como a vantagem de tornar esse tipo de trabalho menos árduo para os seres humanos.

Além das aplicações de robôs móveis que se deslocam pelo solo, há também aquelas no qual o deslocamento acontece por meio aéreo e/ou aquático.

Rawashdeh e Abu-Alrub (2021) utilizam um submarino robótico para a exploração de corpos hídricos bem como para fins educacionais, relacionados a controle de trajetória e movimento. Enquanto Liu et al. (2022) relatam sobre os aerobarco, que se deslocam sobre as superfícies hídricas e sua aplicação para monitoramento ambiental e coleta de dados em conjunto com internet das coisas, inteligência artificial e computação em nuvem.

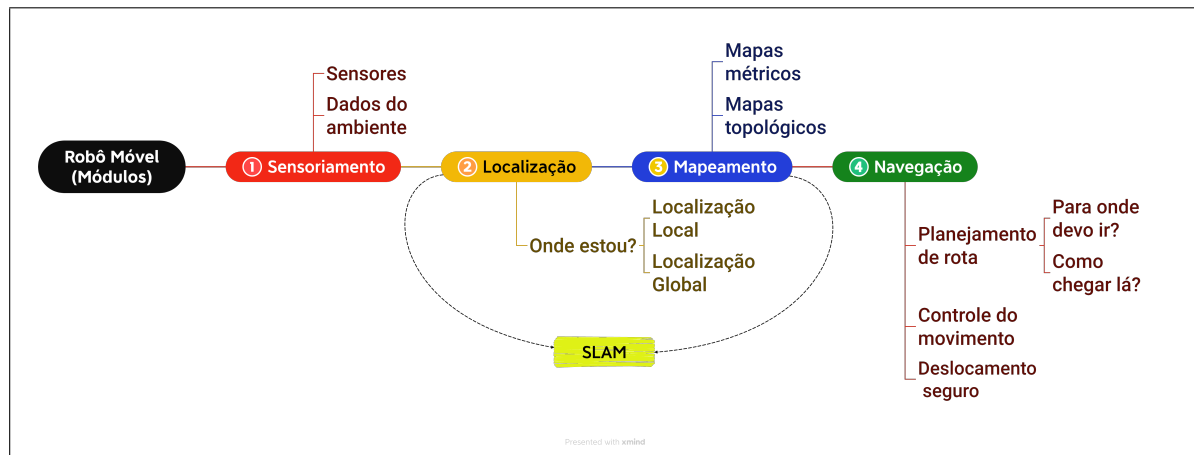
Com relação ao uso de robôs que navegam pelo espaço aéreo, têm-se, por exemplo, os drones utilizados para monitoramento e detecção de focos de incêndio (ALLAUDDIN et al., 2019), sobretudo em florestas, parques e outros ambientes ao ar livre de fácil propagação de fogo. Destaca-se, ainda, o uso de veículos aéreos não tripulados em sistemas de vigilância para identificar indivíduos suspeitos e/ou ações criminosas (SINGH; PATIL; OMKAR, 2018).

2.1.3 Subsistemas

Siegwart e Nourbakhsh (2004) apresentam um robô móvel típico composto pelos módulos de sensoriamento, localização, mapeamento e navegação, conforme Figura 5.

O módulo de Sensoriamento/Percepção é responsável por extrair dados do ambiente para que, a partir deles, o robô possa tomar decisões de modo a alcançar o objetivo para qual foi desenvolvido. É caracterizado pela presença de diversos sensores, que coletam dados do ambiente navegado pelo robô.

Figura 5 – Módulos de um robô móvel típico.



Fonte: Elaboração Própria.

Com relação ao módulo de Localização, a capacidade de localizar-se é fundamental não só para os seres humanos e animais. É imprescindível para um robô, sobretudo do tipo móvel, saber se localizar para identificar a rota que deve seguir até chegar ao seu destino. Para isso, podem ser utilizados diversos materiais e métodos como balizas e QRcode associados a técnicas de visão computacional; cartões ou tags RFID distribuídas pelo ambiente, geralmente fixadas no solo; antenas RFID; *bluetooth*; *wifi*; odometria e localização probabilística, que estima a posição do robô a partir de cálculos matemáticos e informações de sensores.

O módulo de Mapeamento, por sua vez, está relacionado com a construção do mapa que representa o ambiente a ser explorado. Esse mapa é classificado, em geral, em duas categorias: métricos e topológicos. Diz-se mapa métrico, aquele que considera medidas de distância e detalhes do ambiente, enquanto os mapas topológicos têm por objetivo apresentar a conectividade de lugares e a presença de obstáculos, que são as principais informações necessárias à navegação do robô (LLUVIA; LAZKANO; ANSUATEGI, 2021). Costumam-se utilizar para estas finalidades os sensores ultrassônicos; o LIDAR (do inglês, *light detection and ranging*), que utiliza a luz para detecção das barreiras presentes no ambiente e a visão computacional.

Por fim, o módulo de Navegação é responsável por analisar as informações providas do módulo de Percepção e, considerando o mapa e o objetivo da missão, definir as ações de deslocamento do robô. Robôs móveis podem navegar por um ambiente terrestre seguindo paredes; seguindo linhas, sejam magnéticas ou pintadas no chão; comparando sua posição atual com a coordenada de destino a partir de um mapa e guiados remotamente por meio de controles de rádio, aplicações mobile via *bluetooth* ou *wifi*.

2.1.4 Localização e Mapeamento Simultâneos

Além dos módulos discutidos anteriormente, há diversos trabalhos na literatura em que é apresentada uma fusão dos módulos de localização e mapeamento. É possível associar técnicas de localização e mapeamento, de modo que o robô construa um mapa enquanto navega pelo ambiente. É o caso do SLAM (do inglês, *simultaneous localization and mapping*), técnica usada por robôs móveis em ambientes desconhecidos para exploração e criação de uma representação interna deste ambiente, por meio de um mapa, simultaneamente (ROMERO ROSELI

APARECIDA F.; PRESTES; WOLF, 2014).

Inicialmente, o termo SLAM foi proposto por Leonard e Durrant-Whyte (1991), que propuseram um robô construir o mapa de um ambiente desconhecido concomitantemente ao processo de se localizar. Até então, os referidos autores realizaram estudos sobre mapeamento e localização como etapas distintas no processo de navegação, que consiste em rastrear recursos geométricos (obstáculos) em um ambiente para que, assim, o robô possa se deslocar sem colisões.

Uma questão pertinente quando se há necessidade de representar um ambiente é o tipo de mapa que deve ser utilizado. Thrun (2002) explica que mapas métricos exigem uma localização precisa do robô, o que exige maior custo computacional quando comparado aos mapas topológicos. Entretanto, a localização do robô em mapas topológicos é abstrata, tendo em vista que não é possível definir sua posição exata. Os mapas topológicos são estruturas representadas por grafos e fornecem informações a partir das quais pode-se inferir “regiões livres” pelas quais o robô pode navegar. Nesse tipo de mapa, o foco está em identificar a conectividade entre os lugares que compõem o ambiente.

Aguiar et al. (2020) destacam que dividir um espaço geométrico em regiões menores (mapas locais) é uma ação coerente que corrobora com o processo de SLAM. Nesse contexto, os mapas topológicos são soluções mais lógicas a serem usadas por robôs que realizam localização e mapeamento simultâneos.

2.2 Redes de Petri

As Redes de Petri (RP) foram propostas inicialmente por Petri (1962), em sua tese de doutorado. Desde então, é apresentada na literatura como uma ferramenta poderosa para representar processos de sistemas dinâmicos e estruturar sua modelagem. Estas redes podem ser descritas matematicamente e graficamente, sendo apresentadas, pois, com diversas funções e aplicadas em diferentes contextos e áreas.

2.2.1 Definição

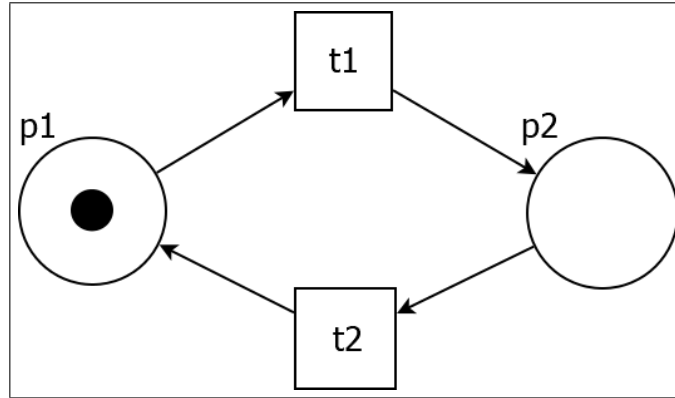
Conforme Murata (1989), RP é uma técnica de modelagem, gráfica e matemática, de sistemas de processamento de informações caracterizados como concorrentes, assíncronos, distribuídos, paralelos, não determinísticos e/ou estocásticos.

Cardoso e Valette (1997) destacam o uso de RP para especificar, implementar, analisar o comportamento lógico e avaliar o desempenho de sistemas a eventos discretos, ou seja, sistemas que mudam bruscamente seu estado em instantes precisos e cujo estado futuro pode ser previsto a partir das variáveis dos estados anteriores independentemente dos instantes de tempo analisados.

Considerando a abordagem gráfica, uma RP é composta por lugares, transições, arcos e fichas. Por exemplo, a rede representada na Figura 6 é composta por dois lugares (p_1 e p_2), duas transições (t_1 e t_2), quatro arcos e uma ficha no lugar p_1 . Assim, quando uma transição, que liga um lugar a outro na rede, é disparada, a ficha é retirada do lugar de entrada e é depositada instantaneamente no lugar de saída. Uma transição é ligada a um lugar por meio de um arco, que possui um peso e que pode ser de entrada, quando sai de um lugar e chega a uma transição, ou de saída, quando sai de uma transição e chega a um lugar. O peso de um arco de entrada indica quantas fichas são necessárias ao lugar para habilitar a transição a ele conectada, assim como o peso de um arco de saída indica quantas fichas serão depositadas no lugar de destino.

O evento de disparo de uma transição pode implicar na mudança de estado do sistema, sendo descrito como um processo de duas etapas: retirada de uma ou mais fichas de cada lugar

Figura 6 – Modelo gráfico de uma Rede de Petri.



Fonte: Elaboração Própria.

de entrada da transição, conforme peso dos arcos de entrada e depósito das fichas no lugar de saída da transição, conforme peso dos arcos de saída. O disparo das transições indica que novas atividades estão sendo realizadas e que o processo está fluindo.

Um sistema modelado por uma RP apresenta um comportamento dinâmico e seu estado pode mudar quando uma transição da rede é disparada. Assim, o estado E_{i+1} depende das variáveis (recursos disponíveis) no estado E_i . O estado E_i pode ser definido como uma representação da distribuição das fichas nos lugares da RP. Matematicamente, essa distribuição é representada por uma matriz de Marcação (M_i) do tipo coluna, em que cada elemento de M_i representa a quantidade de fichas contidas nos lugares da RP. Assim, considerando uma matriz de marcação qualquer $M_i = [1 \ 0 \ 3]^T$, infere-se que os lugares p_1 , p_2 e p_3 possuem 1, 0 e 3 fichas, respectivamente.

Considerando o contexto exposto, Murata (1989) define formalmente uma rede de Petri lugar-transição como uma quintupla, conforme equação 2.1.

$$RP = (P, T, F, W, M_0) \quad (2.1)$$

Em que: $P = \{p_1, p_2, \dots, p_m\}$ é um conjunto finito de lugares; $T = \{t_1, t_2, \dots, t_n\}$ é um conjunto finito de transições; $F \subseteq (P \times T) \cup (T \times P)$ é um conjunto de arcos (relação de fluxo); $W : F \rightarrow \{1, 2, 3, \dots\}$ é uma função peso; $M_0 : P \rightarrow \{0, 1, 2, 3, \dots\}$ é uma marcação inicial; $P \cap T = \emptyset$ e $P \cup T \neq \emptyset$.

Assim, uma rede de Petri é representada por uma matriz de incidência e descrita por uma equação de estados, conforme equação 2.2 proposta por Murata (1977):

$$M_k = M_{k-1} + A^T * u_k \quad (2.2)$$

Em que: $k = 1, 2, 3, \dots$; M_k é a matriz de marcações com dimensão $m \times 1$; A^T é a transposta da matriz de incidência com dimensão $m \times n$; u_k é a matriz de transições com dimensão $n \times 1$.

A matriz de incidência $A = [a_{ij}]$ composta por n transições e m lugares é uma matriz de números inteiros com dimensão $n \times m$, em que cada elemento a_{ij} é dado pela equação 2.3:

$$a_{ij} = a_{ij}^+ - a_{ij}^- \quad (2.3)$$

Em que: a_{ij}^+ é o peso do arco de saída que conecta a transição i ao lugar j e a_{ij}^- é o peso do arco de entrada que conecta o lugar j a transição i .

Quando se trabalha com Redes de Petri é importante observar as suas propriedades comportamentais para ser possível analisar o sistema modelo. Segundo Dias (2007), as principais propriedades são: vivacidade, que determina se a rede é viva, ou seja, se, para uma marcação inicial M_0 , todas as suas transições são vivas para M_0 , sendo que uma transição t_a é viva para M_0 se, para toda marcação acessível M_k , existe uma sequência de disparos que contém a transição t_a , a partir de M_k ; reinicialização, que determina se depois de uma sequência de disparos é possível voltar para M_0 ; alcançabilidade, que representa todas as possíveis marcações que a RP pode possuir a partir dos disparos das transições que levam M_0 até M_k ; limitação, que identifica se o número de fichas em cada lugar da RP é finito, ou seja, se o sistema físico é possível de ser modelado e persistência, que indica que mesmo que mais de uma transição esteja habilitada, quando uma delas disparar, as demais permanecem habilitadas.

2.2.2 Aplicações Gerais

A modelagem de sistemas e processos por redes de Petri é recorrente para diversas aplicações desde as áreas relacionadas à administração até a área de atuação da robótica. Esse tipo de modelagem, seja matemática ou por meio de grafos, apresenta-se como uma técnica robusta e eficiente.

A partir do avanço do processo de globalização que se intensificou nas últimas décadas, o deslocamento de pessoas entre países tornou-se uma prática cotidiana, demandando cada vez mais voos de aeronaves. Diante disso, Zhu (2017) utiliza RP para modelar a dinâmica do pátio de aeroportos. A proposta é otimizar o fluxo de pessoas e veículos, evitando acidente e outros conflitos entre pousos e decolagens das aeronaves.

Ainda nessa perspectiva de gerenciamento de processos, as RP são utilizadas na modelagem de fluxos de trabalho para autorizar com segurança a execução de determinadas tarefas em um processo e promover um gerenciamento eficiente (ZHONG; ZHAO; HU, 2023), bem como Zhu et al. (2023) relatam sobre a modelagem e avaliação de processos no gerenciamento de *data centers*. Du, Zhou e Hu (2023) destacam a importância de obter planos de alocação de funções e de usuários em processos de negócios e/ou de fluxos de trabalho e propõem uma técnica baseada em redes de Petri para alocações dinâmicas de tarefas e de pessoas.

Redes de Petri podem ser utilizadas também no projeto e implementação de sistemas BCI (do inglês, *brain-computer interface*) (YIN; FANG; SHAO, 2019), para diagnóstico, análise e classificação de diversos tipos de falhas em motores (GONG et al., 2017) e em sistemas de controle como redes elétricas, sistemas de transporte, redes de comunicação (LI et al., 2023).

Na robótica podem ser observados diferentes usos das redes de Petri. Feng, Liu e Wang (2006) propõem um novo tipo de redes de Petri para modelagem de AUVs (do inglês, *autonomous underwater vehicle*), que apresentam maior complexidade quando comparados a outros tipos de robôs em quesitos como controle da navegação, autoproteção e reconhecimento de padrões no meio explorado.

Gunardi e Hanafi (2022) desenvolvem a modelagem do sistema de movimentação de um robô móvel em curvas acentuadas utilizando RP, considerando ambiente que contém obstáculos dos quais um robô móvel deve desviar, mas com espaço reduzido para a realização das manobras.

Na indústria, sistemas de manufatura robustos e complexos são automatizados e constituídos por uma estrutura hierárquica, em que um sistema é formado por diversas linhas de produção e cada linha possui estações de trabalho, por sua vez compostas por máquinas. Assim, em processos industriais robotizados, Yasuda (2008) propõe uma metodologia de controle distribuído e hierárquico baseado em redes de Petri.

2.3 Visão Computacional

A visão é um dos sentidos mais importantes para os seres humanos e para os animais. É por meio dela que se torna possível coletar dados do ambiente e a partir do processamento desses dados pelo cérebro, podem-se evitar colisões, tomar decisões, identificar cores e formatos, reconhecer lugares, outros seres vivos e objetos e realizar tantas outras ações (HUTMACHER, 2019).

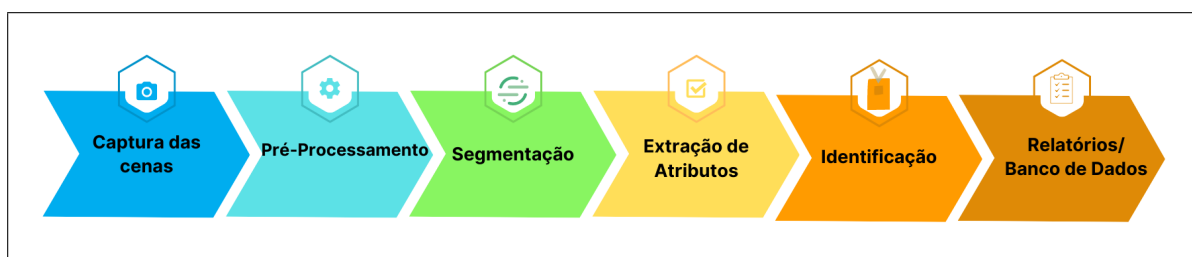
A visão nesse sentido é, pois, um atributo inteligente que animais e pessoas possuem e utilizam constantemente. Esse tipo de atributo proporciona realizar inúmeras tarefas, fazendo com que, ao longo dos anos, o ser humano tenha buscado formas de artificializar esse sentido. Nesse contexto, surgiram sistemas de visão computacional, que deram as máquinas a capacidade de enxergar, uma capacidade básica que auxilia na tomada de decisões a todo instante.

2.3.1 Definição

Um sistema de visão computacional (VC) típico é constituído pelas etapas captura da cena, pré-processamento, segmentação, extração de atributos, identificação e relatórios (GONZALEZ; WOODS, 2009), conforme é representado na Figura 7.

Os sistemas de visão computacional, associados sobretudo ao uso de inteligência artificial, têm se apresentado como uma tendência na literatura das últimas duas décadas. Os resultados expressivos e satisfatórios para a execução de diversas tarefas, como detecção de objetos, análise de cenas em vídeos e aprendizado supervisionado, têm ratificado essas expectativas (GUO et al., 2022).

Figura 7 – Composição de um sistema de visão computacional típico.



Fonte: Elaboração Própria.

Inicialmente uma câmera é utilizada para capturar cenas do ambiente. De posse das imagens, um algoritmo de programação pode ser usado para melhorar a qualidade e preparar as imagens de entrada (pré-processamento) do sistema de visão. Em seguida, no algoritmo é executada a etapa de segmentação para dividir a imagem em porções menores, onde estão os itens de interesse, os quais são elementos característicos do ambiente como marcações, objetos, balizas para que, assim, esses itens possam ser identificados, catalogados em relatórios e/ou banco de dados, se for o caso, e possam ser tomadas decisões.

2.3.2 Aplicações Gerais

Os sistemas de visão computacional têm recebido especial atenção de pesquisadores por sua alta capacidade de se adaptar em diversos contextos (CONSTÂNCIO; CARVALHO; TSUNODA, 2022). Dentre suas aplicações, podem-se destacar engenharias, reconhecimento facial, imagens médicas, agricultura e robótica.

Dan et al. (2022) utilizam visão computacional para propor um novo método, rápido e preciso, de verificação da profundidade da textura de um pavimento asfáltico, tendo em vista que as técnicas normalmente utilizadas são passíveis a erros por dependerem de aspectos subjetivos, como experiência do operador e condições ambientais.

Eyiokur et al. (2023) utilizam técnicas de VC para colaborar com o controle e mitigação da transmissão da COVID-19, diante da emergência sanitária que atingiu o mundo nos últimos anos. O sistema proposto avalia se em áreas públicas, por exemplo, medidas preventivas, como uso de máscara, distanciamento social e evitar levar a mão à boca, anunciadas pelos órgãos sanitários competentes, estão sendo cumpridas.

Na agricultura, Ma, Li e Peng (2023) apresentam que as técnicas de visão computacional são ferramentas de análise rápidas e não invasivas que podem ser usadas para monitorar a qualidade, desde o plantio até a armazenagem, de legumes. Enquanto Chakravartula et al. (2023) desenvolveram um sistema baseado em visão computacional para monitoramento da qualidade e controle do processo de secagem de alimentos, mais especificamente, no caso de fatias de cenoura.

Oliveira et al. (2016) propõem um sistema com inteligência artificial que utiliza a visão computacional para classificar grãos de café a partir da análise de sua cor e tamanho.

Na robótica, o uso de sistemas de visão computacional por robôs móveis é abordado com frequência ao longo dos anos. Esta técnica pode compor os módulos de Percepção/Sensoriamento e de Localização de um robô móvel. Conforme Bessa et al. (2015), equipar um robô com sistema de visão computacional é uma vantagem em relação a outras técnicas de localização, tendo em vista a captura contínua de imagens e o fornecimento da posição dos objetos no ambiente.

Li et al. (2020) desenvolveram um sistema baseado em visão computacional para drones autônomos que participam de corridas. Esse esporte tem se tornado popular entre pilotos e consiste em conduzir um drone por meio de um circuito composto por portões e outros obstáculos.

El-sayed et al. (2022) realizaram um estudo de caso, e obtiveram resultados com até 99,8% de acurácia para o uso de técnicas de VC na identificação e orientação de pacotes conduzidos por esteiras transportadoras omnidirecionais em galpões logísticos.

2.4 Estado da Arte

A Robótica Móvel é uma área em constante ascensão e os avanços tecnológicos proporcionam que esta área do conhecimento esteja em constante inovação. Assim, diversas técnicas são utilizadas por robôs para identificar sua localização, bem como navegar por um ambiente desconhecido e mapeá-lo. Nesta perspectiva, a seguir é apresentado o estado da arte relacionado as temáticas desta pesquisa. Os trabalhos apresentados foram selecionados a partir da busca nas bases Science Direct, IEEE Xplore, Google Acadêmico e Web of Science obedecendo a seguinte expressão de busca: “SLAM” AND “mobile robotic” AND (perception OR “computer vision”) AND “Petri net”. O filtro de tempo utilizado na busca abrange trabalhos publicados desde 2017 a 2022 e que sigam o seguinte critério de inclusão: trabalhos que abordem localização e mapeamento simultâneo de robôs móveis, além de redes de Petri, grafos ou equivalentes e

visão computacional. São excluídos os trabalhos repetidos e aqueles que não estão diretamente relacionados à localização e navegação de robôs móveis em ambientes indoor. Os resultados obtidos foram compilados na Tabela 1. É válido destacar que na literatura analisada somente os trabalhos de Mota et al. (2018) e Kotb e Abi-Char (2017) utilizam Rede de Petri em conjunto com robótica móvel.

Kotb e Abi-Char (2017) se propõem a formalizar uma estrutura para fluxo de trabalho usando algoritmo SLAM (localização e mapeamento simultâneos) probabilístico, baseado em dados de odometria do robô. É importante destacar que fluxo de trabalho é definido pelos autores como um tipo de Rede de Petri que possui um lugar sem conexão fonte (ponto de partida) e um lugar sem arco de saída (ponto de chegada), ou seja, um trajeto com início e fim. Assim, o robô poderia se deslocar de um ponto *A* a um ponto *B* do ambiente. Foi desenvolvido um *framework* que executa SLAM probabilístico e aprende as ações de navegação necessárias para alcançar qualquer localização desejada no ambiente. Este trabalho está em fase inicial de testes.

Ingrand e Ghallab (2017) apresentam um estudo sobre as principais funções executadas durante a tomada de decisão de robôs autônomos. Esta capacidade de escolha desses robôs está diretamente relacionada aos seus sensores e atuadores. A partir dos dados coletados do ambiente por meio dos sensores, o algoritmo executa sua lógica, que está associada a uma missão previamente definida, fazendo os atuadores intervirem no meio no qual o robô está inserido. Os autores elegeram o planejamento, a ação, o monitoramento, a observação e a aprendizagem como as cinco principais funções dos robôs que impactam no processo de tomada de decisão, elencando as características e obstáculos dos métodos utilizados na execução de cada uma dessas funções. Ainda, listaram oito itens que determinam essa tomada de decisão: dinamicidade do ambiente, tipo de tarefa, detalhes da missão, rapidez com que o ambiente varia, quão assistido é o ambiente, custo e pontos críticos do sistema, interação e cooperação com outras pessoas e/ou robôs, nível de autonomia do robô. Concluíram que a tomada de decisão de um robô não está limitada apenas ao planejamento de sua missão; a ação do robô é uma reação ao contexto do ambiente e da tarefa a ser executada; é essencial ele determinar uma hierarquia das tarefas a serem realizadas; o monitoramento da missão deve relacionar o que foi planejado e o que está sendo executado; observar não é só processar sinais, mas requer ações; selecionar o robô adequado é um evidente gargalo no processo de tomada de decisões.

Ayawli et al. (2018) identificam as principais limitações e obstáculos presentes nos métodos de planejamento de rotas de robôs móveis. Os autores apresentam que planejar uma rota e tomar decisões são funções indispensáveis e complexas em robôs autônomos. Os métodos considerados na análise dizem respeito a robôs móveis terrestres, sendo classificados em três grupos: computacionais inspirados na natureza (estratégias baseadas em algum fenômeno natural), convencionais (estratégias que não são baseadas em algum fenômeno natural) e híbridos (estratégias que envolvem dois ou mais métodos de rota). Os autores concluem que para garantir um planejamento de rota eficiente e seguro, evitando obstáculo, é necessário utilizar um método híbrido, que envolve técnicas visuais, dilatação morfológica, diagrama de Voronoi, neuro-fuzzy, A* (BING et al., 2011) e algoritmos de suavização de caminho.

Mota et al. (2018) propuseram um sistema de localização e navegação para robôs móveis utilizando Redes de Petri e RFID. A proposta é modelar o mapa de um ambiente por meio de uma Rede de Petri e o robô deve se deslocar de um ponto a outro deste mapa a partir do disparo das transições da rede. Inicialmente é dado um trajeto, a ser cumprido por um robô, representado pela matriz de incidência de uma Rede de Petri. A rede representa o mapa global de um ambiente estruturado. Os autores desenvolveram um sistema de localização utilizando RFID e o robô executou o trajeto determinado a partir da dinâmica da rede de Petri. Os testes foram realizados

em quatro trajetos distintos e em todos o robô cumpriu a missão de seguir o mapa.

Gutiérrez e López (2019) tem por objetivo gerar um mapa bidimensional de um ambiente. Para alcançar o objetivo proposto, usaram um robô do tipo lagarta equipado com LIDAR e câmera estéreo ZED (Câmera Estéreo Binocular), que executou tarefas de localização e mapeamento simultâneos. A trajetória do robô foi monitorada em tempo real pelo MATLAB.

Os testes foram executados em um ambiente indoor estruturado e em condições controladas. Considerou-se os dados obtidos pelo LIDAR e pela câmera ZED. As distâncias entre objetos estimadas pelo sistema foram comparadas com a distância entre os objetos no ambiente real. Os resultados obtidos tiveram erro percentual máximo de 1% para o LIDAR e 4% para a câmera ZED.

Patle et al. (2019) apresentam as principais técnicas de navegação de robôs móveis encontradas na literatura. Os autores dividem essas técnicas em dois grupos: clássicas, como decomposição celular e seguindo um roteiro e reativas, como lógica fuzzy e redes neurais. As clássicas estão relacionadas a técnicas que usam gráficos, enquanto as reativas utilizam algoritmos com maior e mais rápida capacidade de aprendizagem e adaptação comparando-a com as clássicas. As técnicas reativas são mais robustas que as clássicas e geram resultados melhores em qualquer tipo de terreno por apresentarem maior capacidade de se adaptar em ambientes desconhecidos. Concluem ainda qual técnica de navegação é mais adequada para as principais aplicações de robôs móveis. Destacam que a maioria dos artigos faz análises baseadas em simulações, com apenas um robô e em ambientes estáticos.

Bavle et al. (2020) desenvolveram um *framework* SLAM com baixo custo computacional funcionando em tempo real de forma rápida e eficiente. O método utilizado pelos autores é considerado inovador por combinar visão de baixo custo/odometria visual inercial (VO/VIO) com informações geométricas extraídas de objetos identificados pelo robô e que não eram conhecidos previamente.

Os testes foram realizados em condições ambientais distintas em um robô móvel aéreo e os dados provenientes dos sensores foram processados em tempo real pela unidade central de processamento do robô. Os resultados foram comparados com o desempenho de outros algoritmos SLAM presentes na literatura a partir de um conjunto de dados RGB-D padrão e os autores concluíram que o método proposto é rápido, robusto e de baixo custo computacional.

He et al. (2020) desenvolveram um algoritmo de reconstrução de um modelo tridimensional de um ambiente dinâmico. Os autores enfatizam que o algoritmo pode ser utilizado em diferentes campos da indústria. Utilizaram sensor LIDAR (Mapeamento por detecção de Luz) para obtenção de dados do ambiente. Ainda, como o IMU (Unidade de Medição Inercial) possui alta frequência de atualização e alta precisão em suas medições, utilizaram também este sensor para resolver os problemas de posicionamento do LIDAR. O algoritmo foi validado a partir da reconstrução do ambiente interno de um moinho. Os resultados obtidos foram comparados com o algoritmo de LOAM (Odometria e Mapeamento em tempo real usando LIDAR) (SHAN; ENGLLOT, 2018). O algoritmo proposto obteve resultados mais precisos e exigiu custo computacional reduzido.

Shit (2020) identifica e analisa os principais problemas e limitações de uma localização precisa de robôs autônomos. O autor destaca que o problema da navegação de robôs é complexo e a estratégia comumente utilizada na literatura de dividir o sistema do robô em módulos (percepção, mapeamento, controle, planejamento e tomada de decisão) não é totalmente eficaz, pois esta abordagem desconsidera a capacidade de compreender a dinamicidade do ambiente explorado. Assim, propõe que sejam incluídas ações biológicas no processo de navegação como

memorização, inferência e atualização da situação, sobretudo em ambientes dinâmicos. Destaca que localizar-se com eficiência e precisão é primordial para um mapeamento e navegação eficazes do ambiente explorado. Assim, conclui que, apesar dos inúmeros métodos já apresentados na literatura, são necessárias mais pesquisas para se desenvolver um sistema de localização precisa. Autonomia de um robô móvel possui níveis diretamente associados ao nível de eficácia do método de localização utilizado. Defende que para obter precisão na localização implica em utilizar sensores cooperativamente e não um especificamente, que, por sua vez, é impactada pelo problema de comunicação entre os sensores em tempo real e com eficiência.

Gobhinath et al. (2021) utilizam ROS (Sistema Operacional Robótico) e LIDAR para desenvolver um sistema de localização e mapeamento simultâneos. É proposto um robô móvel equipado com Arduino, LIDAR e uma Raspberry com ROS que se comunica com um notebook via bluetooth. O notebook recebe as coordenadas dos obstáculos do ambiente enviadas pelo robô e processa essas informações utilizando a *toolbox* do ROS no MATLAB. Por fim, um mapa 2D do ambiente é gerado. Os autores destacam que o sistema desenvolvido é robusto e concluem que os resultados foram satisfatórios, considerando o custo computacional utilizado para captar obstáculos em um raio de até 8 metros. O sistema conseguiu gerar um mapa 2D do ambiente explorado.

Zhao et al. (2021) propõem um *framework* com localização e mapeamento simultâneo de robôs móveis para ambientes cujos cenários variam constantemente como shoppings e supermercados. Os autores destacam que as principais vantagens do seu algoritmo são a execução em tempo real e pode ser utilizado por longos períodos, tendo em vista sua capacidade de adaptar constantemente o mapa do ambiente conforme as variações que surgem no dia a dia. O *framework* sugerido obteve resultados eficientes e foi validado. Os testes foram realizados durante um mês, utilizando um robô móvel em um supermercado. O principal gargalo deste método é o custo computacional, principalmente a quantidade de memória necessária para usar o algoritmo em um ambiente dinâmico durante um longo período. Entretanto, a solução desse problema foi desenvolvida a partir da criação de sub-mapas, atualizando-se, assim, no mapa principal somente a parte do ambiente que sofreu mudanças.

Ding et al. (2022) discutem as habilidades necessárias a um robô móvel autônomo para executar suas missões de forma otimizada. Elencam a capacidade de adaptação a diferentes ambientes, obter comunicação estável (robótica cooperativa) e controle de movimento robusto como sendo os principais desafios enfrentados por robôs móveis autônomos atualmente. Destacam que para se adaptar o robô deve conhecer o ambiente e isso ocorre por meio de um módulo de percepção eficiente. Com relação à comunicação, quando vários robôs trabalham em um mesmo ambiente é fundamental que o meio de comunicação entre eles seja estável e compartilhe o máximo de informações úteis possíveis entre eles. Por fim, o controle do movimento para a realização das tarefas previamente estabelecidas precisa dos módulos de percepção e de comunicação. A análise realizada ocorre a partir de quatro aspectos dos robôs móveis autônomos: aplicação, técnicas de percepção, comunicação usada entre os robôs e mecanismos de controle. Os autores destacam ainda que outra contribuição de seu trabalho é que eles apontam o caminho para o desenvolvimento de trabalhos futuros sobre este tema, o qual é a convergência dos aspectos supracitados.

É válido destacar a relevância do tema abordado nesta pesquisa ao observar a coluna “país” da Tabela 1, na qual se mostra que pesquisas sobre robótica móvel são desenvolvidas em diversos países do globo terrestre, tanto no ocidente como no oriente. Em todos os artigos citados, os autores corroboram que o problema da navegação de robôs móveis é complexo e que ainda não foi encontrada uma solução genérica para qualquer tipo de ambiente. A maioria dos

trabalhos publicados apresentam resultados teóricos, baseados em simulações e em condições controladas.

Tabela 1 – Estado da arte sobre Robótica Móvel associada a Rede de Petri e Visão Computacional.

Autor	País	Situação Problema	Métricas Avaliativas	Conclusões
Kotb e ABI-CHAR (2017)	Kwait	Formalizar uma estrutura para fluxo de trabalho usando algoritmo SLAM	Algoritmos probabilísticos	Foi desenvolvido um framework que executa SLAM probabilístico e aprende as ações de navegação necessárias para alcançar um ponto de chegada previamente determinado
Ingrand e Ghallab (2017)	França	Elencar as características e obstáculos das principais funções executadas durante a tomada de decisão de robôs autônomos	-	A tomada de decisão de um robô não está limitada apenas ao planejamento de sua missão; a ação do robô é uma reação ao contexto do ambiente e da tarefa a ser executada; é essencial ele determinar uma hierarquia das tarefas a serem realizadas; o monitoramento da missão deve relacionar o que foi planejado e o que está sendo executado; observar não é só processar sinais, mas requer ações; selecionar o robô adequado é um evidente gargalo no processo de tomada de decisões
Ayawli et al (2018)	China/ Gana	Identificar as principais limitações e obstáculos presentes nos métodos de planejamento de rotas de robôs móveis	Eficiência do planejamento de rotas	Os métodos considerados na análise dizem respeito a robôs móveis terrestres e os autores concluem que para garantir um planejamento de rota eficiente e seguro, evitando obstáculos, é necessário utilizar um método híbrido, que envolve técnicas visuais, dilatação morfológica, diagrama de Voronoi, neuro-fuzzy, A* e algoritmos de suavização de caminho
Mota et al (2018)	Brasil	Desenvolver sistema de localização e navegação de robô móvel	Movimentos do robô, Correspondência entre número de lugares e transições na RP e número de cômodos e cartões RFID no ambiente, respectivamente	Os autores desenvolveram um sistema de localização utilizando RFID e o robô executou o trajeto determinado a partir da dinâmica da rede de Petri. Os testes foram realizados em quatro trajetos distintos e em todos o robô cumpriu a missão de seguir o mapa
Gutiérrez e López (2019)	Colômbia	Gerar mapa bidimensional de um ambiente	Erro percentual das medições	Os testes foram executados em um ambiente indoor estruturado e as distâncias entre objetos estimadas pelo sistema foram comparadas com a distância entre os objetos no ambiente real, obtendo erro percentual máximo de 1 e 4% para LIDAR e câmera ZED, respectivamente
Patle et al (2019)	Índia/ Iraque	Apresentar as principais técnicas de navegação de robôs móveis.	Tipo de obstáculo, técnicas de navegação e tipo de aplicação	As abordagens reativas são mais robustas que as clássicas e geram resultados melhores em qualquer tipo de terreno, pois apresentam maior capacidade de se adaptar em ambientes desconhecidos. Apresentam ainda qual técnica de navegação é mais adequada para as principais aplicações de robôs móveis. Destacam que a maioria dos artigos fazem análises baseadas em simulações, com apenas um robô e em ambientes estáticos.
Bavle et al (2020)	Espanha/ EUA	Desenvolver framework SLAM com baixo custo computacional que funcione em tempo real	ATE (do inglês, the average trajectory error) e VIO (do inglês, Visual Inertial Odometry)	Os testes foram realizados em um robô móvel aéreo, que processava os dados <i>onboard</i> em tempo real. Os resultados foram comparados com o desempenho de outros algoritmos SLAM presentes na literatura a partir de um conjunto de dados RGB-D padrão.
He et al (2020)	China	Desenvolver um algoritmo de reconstrução de um modelo tridimensional de um ambiente dinâmico	Acurácia e resolução angular	Os resultados obtidos foram comparados com o algoritmo LOAM. O algoritmo proposto obteve resultados mais precisos e exigiu custo computacional reduzido

(continuação)

Autor	País	Situação Problema	Métricas Avaliativas	Conclusões
Shit (2020)	Índia	Identificar e analisar os principais problemas e limitações de uma localização precisa de robôs autônomos	Complexidade computacional, ruído dos sensores e <i>delays</i> de comunicação	A autonomia de um robô móvel possui níveis que estão diretamente associados ao nível de eficácia do método de localização utilizado
Gobhinath et al (2021)	Índia	Utilizar ROS e LIDAR para desenvolver um sistema de localização e mapeamento simultâneos	Acurácia	O sistema conseguiu gerar um mapa 2D do ambiente explorado, identificando com eficiência obstáculos num raio de até 8m
Zhao et al (2021)	República Tcheca	Propor um framework de "longa vida" com localização e mapeamento simultâneo de robôs móveis para ambientes cujo cenários variam constantemente	MCR (do inglês, map change rate), CRI (do inglês, Correctness Rate of Initialization), MRCL (do inglês, Mileage Rate of Correct Localization) e AMS (do inglês, Average Matching Score)	O framework desenvolvido obteve resultados eficientes e foi validado. Os testes foram realizados utilizando um robô móvel em um supermercado e este funcionou eficientemente durante o período analisado (um mês)
Ding et al (2022)	China/Japão	Discutir habilidades necessárias a um robô móvel autônomo para executar suas missões de forma otimizada	Adaptabilidade geométrica, desvio de obstáculos e número de falhas	A análise realizada ocorre a partir de quatro aspectos dos robôs móveis autônomos: aplicação, técnicas de percepção, comunicação usada entre os robôs e mecanismos de controle. Os autores destacam ainda que outra contribuição de seu trabalho é que eles apontam que a convergência dos aspectos supracitados é o caminho para o desenvolvimento de trabalhos futuros sobre robôs móveis autônomos

Fonte: Elaboração Própria.

2.5 Considerações Finais do Capítulo

A Robótica Móvel tem evoluído ao longo dos anos e suas aplicações são diversas, desde atividades perigosas aos seres humanos a atividades de entretenimento. Entretanto, o problema da localização, mapeamento e navegação de robôs persiste, sobretudo em ambiente internos. Nesse sentido, podem ser observadas na literatura diferentes técnicas para atender o problema supracitado, mas não há unanimidade quanto sua solução.

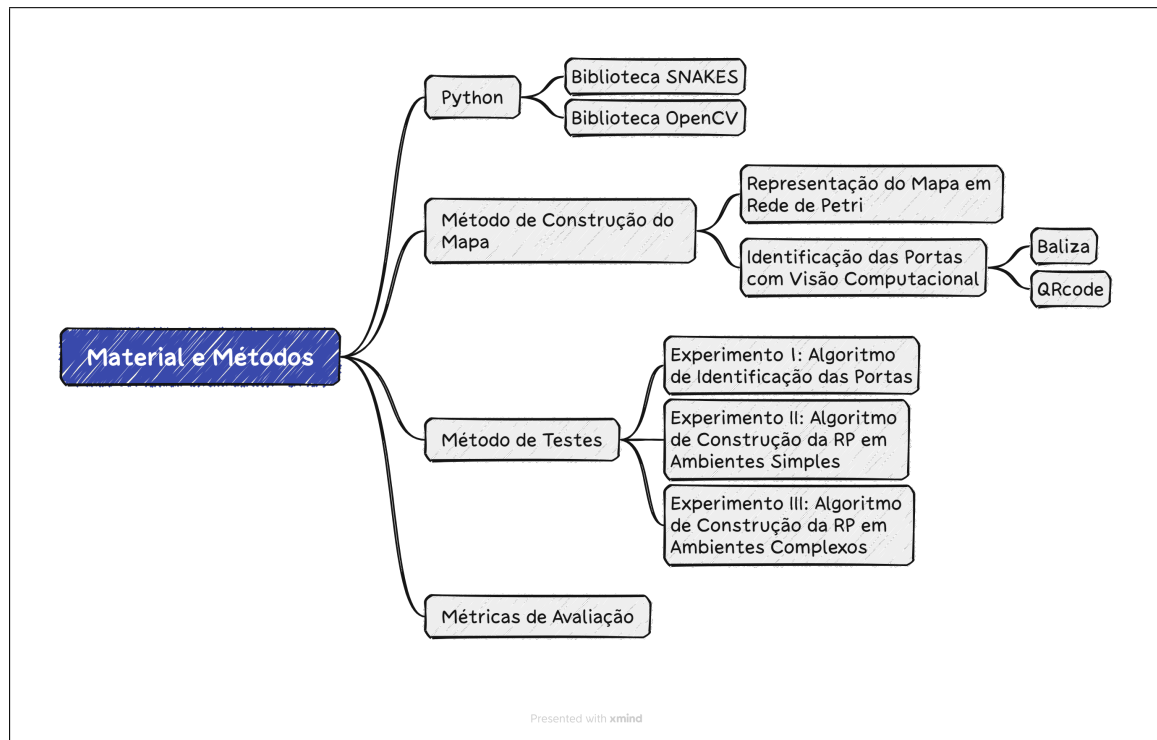
Além da Robótica Móvel, esse capítulo faz uma abordagem geral sobre a técnica de modelagem de sistemas a eventos discretos conhecida como redes de Petri, bem como apresenta suas definições gráfica e matemática e suas aplicações, inclusive em robótica. Nota-se que esta técnica, em geral, é utilizada para acompanhamento de processos e não execução deles.

É realizada, ainda, uma explanação sobre visão computacional, suas aplicações em diferentes seguimentos e percebe-se ênfase em suas aplicações em sistemas de localização de robôs móveis, em que é utilizada com frequência e eficiência para esta finalidade. Por fim, é apresentado o estado da arte dessas temáticas e suas relações com a robótica.

3 MATERIAL E MÉTODOS

Nesta seção é apresentada uma descrição das tecnologias utilizadas para se alcançar o objetivo deste trabalho, bem como a metodologia utilizada na condução dos testes e as métricas avaliadas no processo de SLAM proposto. O material e métodos abordados nesta seção estão representados na Figura 8 e são discutidos ao longo deste capítulo.

Figura 8 – Material e Métodos.



Fonte: Elaboração Própria.

3.1 Material

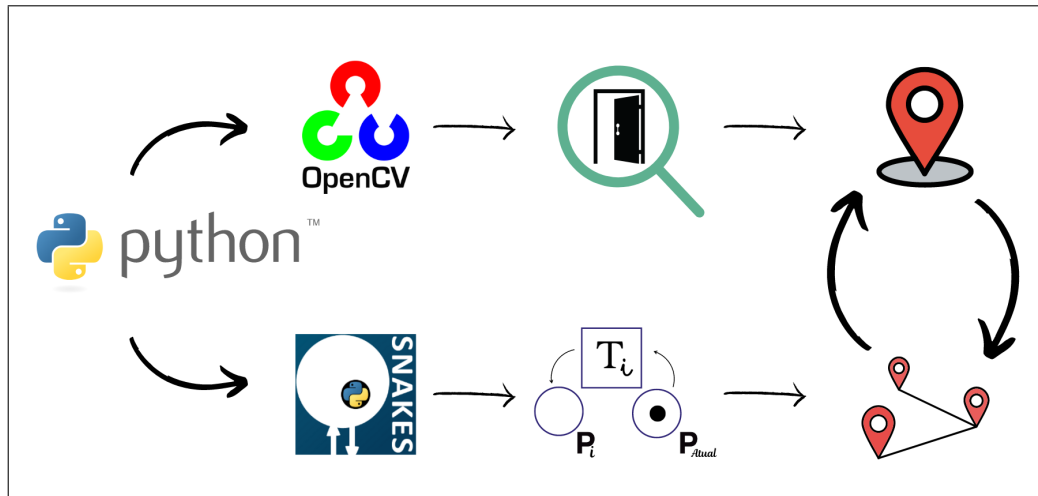
Na Figura 9 está representado o material utilizado para o desenvolvimento desta pesquisa. A programação do algoritmo para localização e mapeamento simultâneo foi desenvolvida em Python com o uso das bibliotecas SNAKES e OpenCV na IDLE Python 3.8.

3.1.1 Biblioteca SNAKES

SNAKES é uma biblioteca utilizada para desenvolvimento e manipulação de Redes de Petri. Inicialmente a SNAKES era utilizada através da programação em Python, mas hoje pode ser utilizada também com outras linguagens de programação. Possui compatibilidade com diferentes *plugins*, tornando-a uma ferramenta poderosa e abrangente. Destaca-se o *plugin gv*, pois este possibilita desenhar graficamente redes de Petri utilizando o GraphViz (POMMEREAU, 2015).

Conforme descreve Franck Pommereau, da University of Evry/Paris-Saclay, França, autor desta biblioteca, “o principal objetivo do SNAKES é ser uma biblioteca geral de redes de

Figura 9 – Material.



Fonte: Elaboração Própria.

Petri, capaz de lidar com a maioria dos modelos de redes de Petri e fornecer ao pesquisador uma ferramenta para prototipação rápida de novas ideias” (POMMEREAU, 2010).

3.1.2 Biblioteca OpenCV

OpenCV é uma biblioteca multiplataforma desenvolvida pela Intel, em 2000, utilizada para manipulação e tratamento de imagens em projetos que utilizam visão computacional e processamento de imagens.

Foi desenvolvida em linguagem C e pode ser executada tanto em sistemas operacionais de computadores (Windows, Linux, Android, MacOS, FreeBSD, OpenBSD) como dispositivos móveis (Android, Maemo, iOS).

3.2 Método de Construção da Rede de Petri

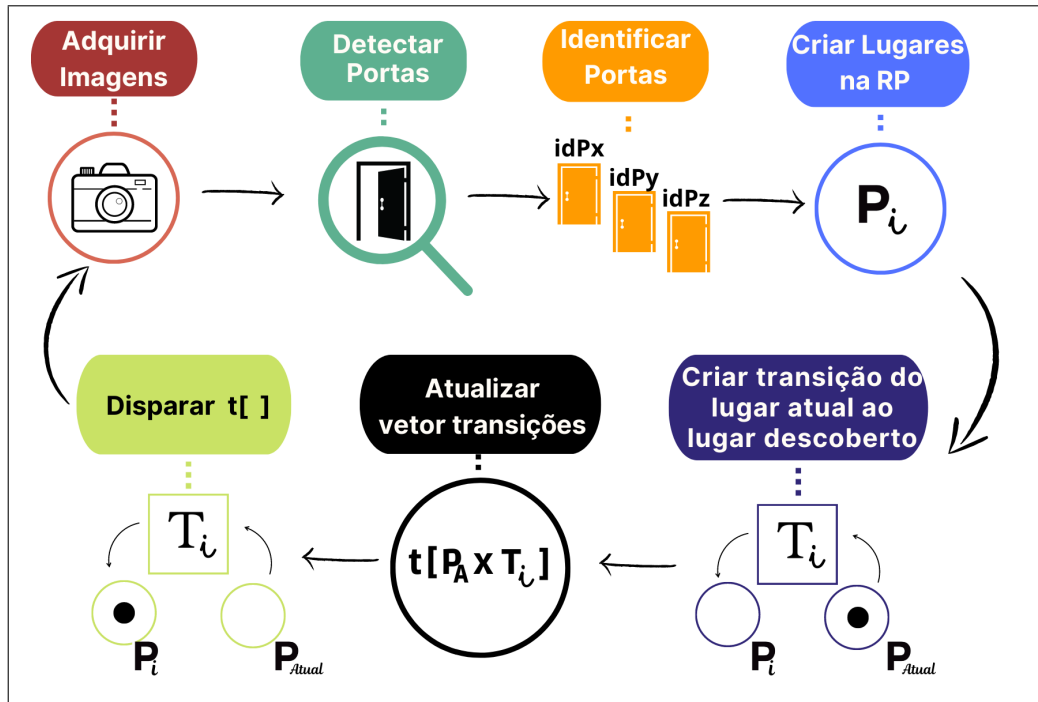
Na Figura 10 são mostradas as etapas de construção da RP que representa o mapa do ambiente explorado.

Inicialmente, as imagens são adquiridas e o algoritmo procura portas na cena representada pelas imagens supracitadas. Quando uma porta é detectada, é atribuída a ela uma identificação (id) e em seguida é criado um lugar na rede, caso a porta encontrada seja nova, ou seja, caso a porta não tenha sido identificada anteriormente. Por conseguinte, uma transição é criada na rede, conectando o lugar P_{atual} ao novo lugar encontrado (P_i). Os lugares equivalem aos cômodos do ambiente explorado e as transições equivalem às portas que conectam esses cômodos.

O vetor $t[]$ que armazena as transições é atualizado, de modo que é acrescentada em $t[]$ a transição T_i . Em seguida, a nova transição é disparada, representando avanço da ficha para o lugar P_i . O processo então é reiniciado. A ficha representa um robô móvel que se desloca entre os cômodos do ambiente explorado.

A seguir são descritas cada etapa do algoritmo de localização e mapeamento (ALM) utilizado desde a identificação das portas do ambiente, para localização do robô, até a construção do mapa.

Figura 10 – Etapas do processo de mapeamento do ambiente por RP.



Fonte: Elaboração Própria.

Essa metodologia de mapeamento está representada no fluxograma da Figura 11 e no Algoritmo 1. A descrição do fluxograma em questão é realizada a seguir, com as seguintes considerações: ao lado de cada porta a uma baliza para identificar a existência de uma porta; as palavras cômodo e porta se referem ao ambiente explorado e as palavras lugar e transição se referem a RP que representa o mapa topológico do ambiente explorado; disparar uma transição significa, para o ambiente real, atravessar a porta que conecta um cômodo atual a um cômodo de destino.

O ALM representado pelo fluxograma da Figura 11 é composto por sete etapas. No início do algoritmo é criado o lugar inicial p_0 e variáveis são declaradas.

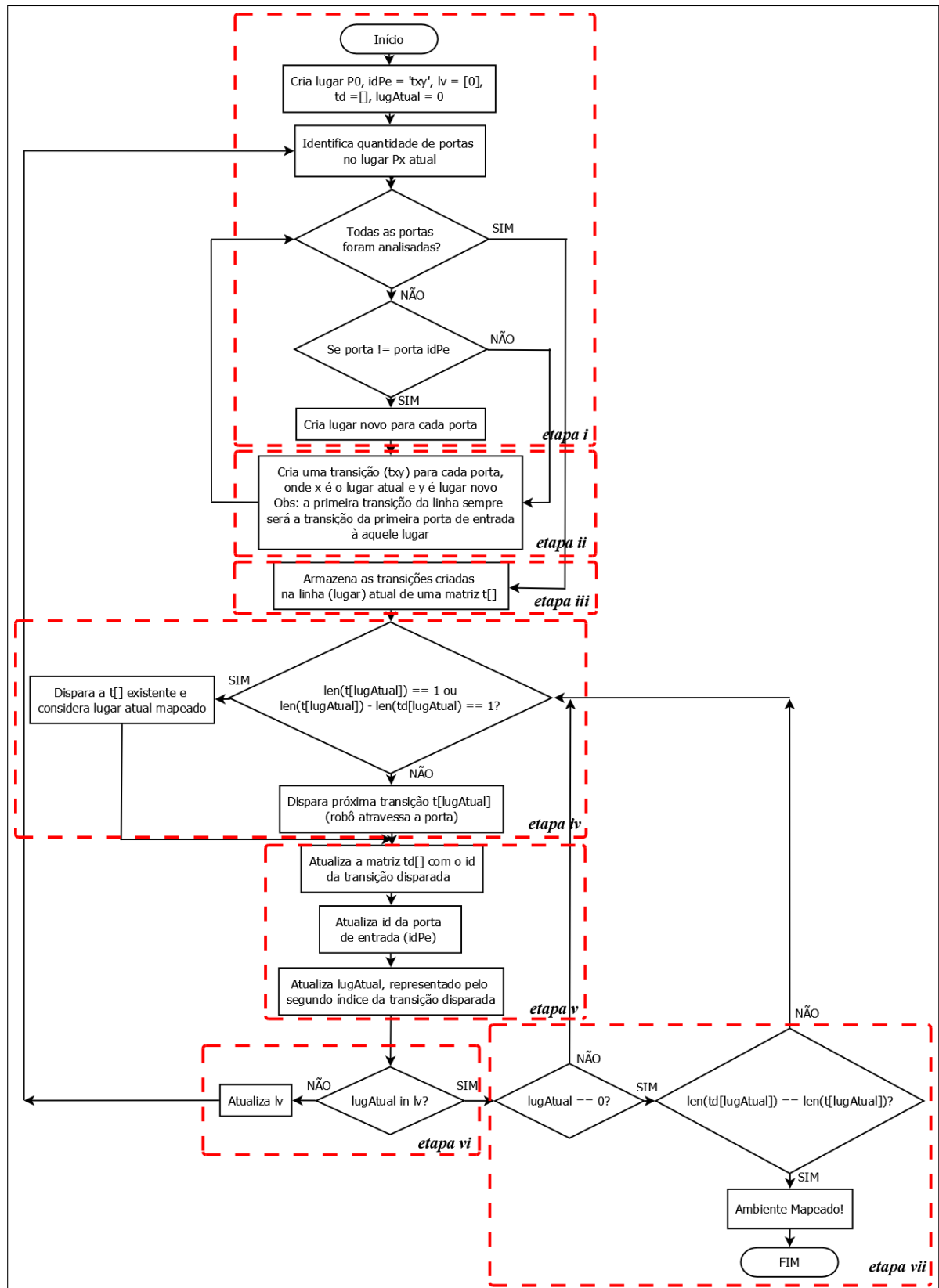
A etapa i marca o início do loop do algoritmo. Nesta etapa, deve-se informar ao algoritmo, a quantidade e a identificação das portas do cômodo. Essa informação pode ser disponibilizada manualmente via *IDLE Shell Python* quando solicitada pelo algoritmo, ou imagens do ambiente podem ser capturadas por uma câmera para o algoritmo detectar as portas e atribuir uma identificação a cada porta encontrada. Caso a porta seja diferente da porta de entrada ao cômodo atual, cria-se um novo lugar e segue-se para a etapa ii ; caso não, segue-se para a etapa ii .

Na etapa ii , cria-se uma transição para cada conexão entre o cômodo atual e os outros cômodos conectados a ele.

Na etapa iii , as transições criadas são armazenadas na linha atual de uma matriz $t[]$. Note-se que o índice da linha atual de $t[]$ é equivalente ao índice do lugar atual.

Na etapa iv , a transição da linha atual é disparada e segue-se para a próxima etapa. Note-se a condição de disparo a seguir descrita: se existir mais de uma transição na linha atual da matriz $t[]$, disparar a primeira transição ainda não disparada a partir da segunda transição existente na linha atual, mas se não, ou seja, se tiver apenas uma transição na linha atual, então

Figura 11 – Metodologia utilizada no mapeamento do ambiente.



Fonte: Elaboração Própria.

```

Entrada      : Inicialização das matrizes e do lugar inicial na Rede de Petri
Saída       : Mapa do ambiente representado por uma Rede de Petri


$p \leftarrow \text{cria\_matriz}(\text{var.idLugar}, \text{var.qtdPortas}, 0);$   

 $td \leftarrow \text{cria\_matriz}(\text{var.idLugar}, \text{var.qtdPortas}, 0);$   

 $rp.\text{add\_place}(\text{Place}("p\%" \text{var.idLugar}, 1));$   

while Verdadeiro do  

     $p[] \leftarrow \text{idPortas};$   

    for  $i$  to  $\text{var.qtdPortas}$  do  

        if  $p[i] \neq \text{var.idPe}$  then  

             $rp.\text{add\_place}(\text{Place}("p\%" \text{var.idLugar},));$   

             $rp.\text{add\_transition}(\text{Transition}("t\%\%" \text{var.idLugarA}, \text{var.idLugar}));$   

             $t[\text{var.idLugarA}].\text{insert}(i, 't\%' \text{var.idLugarA}, \text{var.idLugar});$   

        end  

        else  

             $rp.\text{add\_transition}(\text{Transition}("t\%\%" \text{var.idLugarA}, \text{lugAnt}[\text{var.idLugarA}]));$   

             $t[\text{var.idLugarA}].\text{insert}(0, 't\%' \text{var.idLugarA}, \text{lugAnt}[\text{var.idLugarA}]);$   

        end  

    end  

    while  $\text{var.idLugarA}$  in  $lv$  do  

        if  $\text{len}(t[\text{var.idLugarA}]) == 1$  or  $\text{len}(t[\text{var.idLugarA}]) - \text{len}(td[\text{var.idLugarA}]) == 1$  then  

             $rp.\text{transition\_fire}("\%" t[\text{var.idLugarA}][0]);$   

        end  

        else  

            for  $i$  to  $\text{len}(t[\text{var.idLugarA}])$  do  

                if  $i \neq 0$  then  

                    if  $t[\text{var.idLugarA}][i]$  in  $td[\text{var.idLugarA}]$  then  

                        continue;  

                    end  

                    else  

                         $\text{var.c} \leftarrow t[\text{var.idLugarA}][i];$   

                         $rp.\text{transition\_fire}("\%" t[\text{var.idLugarA}][i]);$   

                        break;  

                    end  

                end  

            end  

            end  

             $td[\text{var.idLugarA}].\text{append}(\text{var.c});$   

             $\text{var.idPe} \leftarrow \text{str}("t\%" \text{var.c}[2], \text{var.c}[1]);$   

             $\text{var.idLugarA} \leftarrow \text{int}(\text{var.c}[2]);$   

            if  $\text{var.idLugarA}$  in  $lv$  then  

                if  $\text{var.idLugarA} == 0$  then  

                    if  $\text{len}(td[\text{var.idLugarA}]) == \text{len}(t[\text{var.idLugarA}])$  then  

                         $rp.\text{draw}(\text{"Mapa.png"});$   

                        break;  

                    end  

                end  

            end  

            end  

            else  

                 $lv.\text{append}(\text{var.idLugarA});$   

                break;  

            end  

        end  

    end  

end


```

Algoritmo 1: Pseudocódigo do ALM.

dispara-se a transição existente e considera-se cômodo mapeado¹.

Na etapa *v*, a transição disparada é armazenada em $td[]$, seguindo a regra de que o índice da linha atual de $td[]$ é equivalente ao índice do lugar atual. Nessa etapa são atualizados também o índice da porta de entrada ($idPe$) no lugar atual e o índice do lugar atual.

Na etapa *vi*, se entrando em um cômodo novo, ou seja, se o índice do lugar atual não estiver registrado no vetor lugar visitado ($lv[]$), segue-se para a etapa *i*; se sim, segue-se para a etapa *vii*.

Na etapa *vii*, caso o cômodo atual seja aquele representado pelo lugar p_0 , verifica-se se todas as transições da linha 0 de $t[]$ foram disparadas. Assim, se essas duas condições tenham sido satisfeitas, diz-se que o mapeamento do ambiente está concluído. Entretanto, se o índice do lugar atual for diferente de 0 ou se $td[lugAtual]$ for diferente de $t[lugAtual]$, o algoritmo segue para o início da etapa *iv*.

3.2.1 Identificação das Portas no Ambiente

O processo de identificação de portas em um ambiente por meio de métodos de visão computacional pode ser feito, por exemplo, com o auxílio de balizas, QRcode e/ou reconhecimento de padrões. A técnica de reconhecimento de padrões foi dispensada, tendo em vista seu alto custo computacional. Nesta perspectiva, foram realizados testes com QRcode e com uma baliza.

A aquisição das imagens de cada cômodo do ambiente foi realizada por meio de um *smartphone Samsung Galaxy J5 Pro* equipado com uma câmera de 13 megapixels. As imagens obtidas foram utilizadas como dados de entrada para o algoritmo de identificação das portas (AIP).

Assim, foi anexada ao lado da porta uma baliza no formato de círculo de 10 cm de diâmetro na cor vermelha, bem como um QRcode com comprimento e largura de 10 cm, como está representado na Figura 12. Esses acessórios foram anexados a aproximadamente 1,5 m de altura em relação ao solo e as imagens foram adquiridas, nesta mesma altura, por meio da câmera posicionada a uma distância fixa de 1,2 m em diferentes ângulos (30°, 45°, 60° e 90° em relação à superfície da parede em que está a porta) para validar a robustez do AIP, ou seja, para se analisar os dados de saída do algoritmo a partir de imagens obtidas em diferentes ângulos. A seguir são descritas as etapas do algoritmo de ambos os métodos.

Com relação à baliza, foi inicialmente realizado um pré-processamento nas imagens, que foram redimensionadas para 640x480 pixels. Em seguida, a imagem RGB foi convertida em HSV. Essa transformação de escala de cores possui o objetivo de separar a informação acromática e cromática (SCHNEIDER; BELLON; ARAKI, 2003) e, assim, facilitar a detecção das balizas vermelhas contidas ao lado de cada porta. A Figura 13(a) exhibe a imagem na escala HSV.

Após o processo anterior, é aplicado um filtro vermelho para detectar a ocorrência de pixels vermelhos da baliza e a dilatação, que manipula a morfologia da imagem e implica diretamente na etapa de segmentação, conforme representado na Figura 13(b).

Em seguida foi aplicada a binarização para separar as regiões de interesse de análise na imagem (em vermelho) e o restante dos elementos da imagem (em preto), conforme exhibe a Figura 13(c).

¹ Cômodo mapeado significa que todas as possibilidades de trajeto que poderiam ser originadas desse cômodo já foram mapeadas e o algoritmo não aborda mais o lugar representado por esse cômodo até o fim do mapeamento.

Figura 12 – Porta com baliza e QRcode.

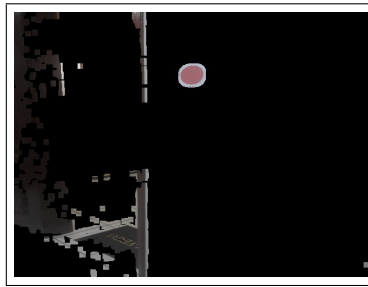


Fonte: Elaboração Própria.

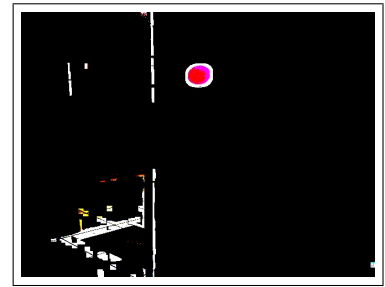
Figura 13 – Imagem 640x480 pixels (a) na escala HSV, (b) pós processo de dilatação e com filtro vermelho aplicado e (c) pós processo de binarização.



(a)



(b)



(c)

Fonte: Elaboração Própria.

Aplicou-se, ainda, uma operação de abertura da imagem, erosão seguida de dilatação, para melhorar a visualização do objeto segmentado. Por fim, o algoritmo varre a imagem buscando os pixels vermelhos que identificam a porta.

Em relação ao QRcode, o processo de identificação das portas é simplificado. A partir da linha de comando `cv2.QRCodeDetector()` o algoritmo retorna o valor contido no QRcode encontrado na imagem, que, por sua vez, representa a identificação da porta encontrada.

Independente do método utilizado, com as portas encontradas, torna-se possível a construção da RP. Os cômodos do ambiente são lugares da RP, as portas são transições e a ficha é o robô móvel, conforme é apresentado no Capítulo 4.

3.3 Método de Testes

Os testes são conduzidos por meio de 3 experimentos. No experimento I, são avaliados dois métodos de identificação das portas. Neste caso, ambos utilizando técnicas de visão computacional para identificação de QRcode e de balizas coloridas. No experimento II, é avaliado o algoritmo de construção da RP para modelagem do mapa de um ambiente considerado simples. No experimento III, é avaliado o algoritmo de construção da RP para modelagem do mapa de dois ambientes considerados complexos.

Nos testes para identificação das portas (experimento I) são consideradas dez imagens para cada ângulo de captura das imagens em relação à superfície da parede onde está a porta. Ao lado de cada porta tem-se uma baliza e um QRcode que fornecem os dados de entrada, os quais são a identificação (id) de cada uma delas, para o algoritmo, que, por sua vez, armazena esses dados em um vetor. O vetor supracitado pode ser utilizado no processo de mapeamento para a criação de cada transição da RP.

Identificar as portas é uma etapa fundamental no processo de mapeamento, tendo em vista que estão relacionadas à criação dos lugares da RP e das transições que entram e saem de cada lugar.

Para os testes de geração do mapa topológico (experimentos II e III), é necessário que o algoritmo detecte o id de cada porta do ambiente via técnicas de visão computacional, a partir das imagens fornecidas ou, manualmente, o usuário pode fornecer ao ALM, sempre que este solicita, via tela do computador, a quantidade de portas e a identificação (id) de cada porta contida no cômodo analisado. Os dados *quantidade de portas* e *id* são fornecidos por meio da *IDLE Shell Python* obedecendo, respectivamente, à seguinte estrutura: k , sendo $k = 1, 2, \dots, k_n$, em que n é o número de portas do ambiente e t_{xy} , em que t representa uma transição que liga dois lugares da RP, x é o índice que representa o lugar de entrada e y é o índice que representa o lugar de saída.

Para cada cômodo seguem-se os seguintes passos: o sistema² é deslocado em direção à primeira porta T_{x1} encontrada no cômodo P_x ; passa por ela e chega ao cômodo P_y . Após o novo cômodo ser mapeado, o sistema volta para P_x e é deslocado para a porta T_{x2} e assim sucessivamente até que todo o ambiente seja mapeado.

Os testes de geração do mapa são realizados em dois tipos de ambientes: simples e complexos. Ambiente considerado simples é aquele cujos cômodos são conectados sequencialmente, ou seja, em que cada cômodo possui no máximo duas portas, que conectam este a um ou dois outros cômodos. Enquanto ambiente considerado complexo é aquele ambiente em que pelo menos um cômodo possui mais que duas portas, que conectam este a, no mínimo, outros três cômodos.

3.4 Métricas de Avaliação

A representação de mapas por meio de redes de Petri é avaliada neste trabalho sob dois aspectos: analogia entre o sistema real e a rede de Petri, por meio das métricas de avaliação a seguir listadas, bem como observância das propriedades comportamentais da RP.

- Uma RP é construída?
- O modelo gráfico da RP representa um mapa topológico do ambiente real?

² Neste caso o sistema é representado por uma pessoa com uma câmera, mas poderia ser representado por um robô móvel equipado com uma câmera.

- A quantidade de transições da RP é equivalente à quantidade de portas no ambiente real na proporção 2:1?
- A quantidade de lugares da RP é igual à quantidade de cômodos no ambiente real?
- As transições da RP diferem portas de entrada e de saída?
- As transições que ligam um lugar de saída a um lugar de entrada ratificam que as portas, representadas pelas transições, conectam o cômodo de saída e o cômodo de entrada do ambiente real?

As análises conduzidas para cada métrica de avaliação e para as propriedades comportamentais das redes de Petri estão descritas no Capítulo 4.

3.5 Considerações Finais do Capítulo

O sistema de localização e mapeamento simultâneos em ambientes *indoor* foi desenvolvido a partir da condução de 3 experimentos, que estão relacionados aos algoritmos de identificação de portas e de geração do mapa destes ambientes. Foram utilizados um *smartphone Samsung Galaxy J5 Pro*, bem como as bibliotecas *SNAKES* e *OpenCV*, respectivamente, para a construção da RP e processamento das imagens. Os algoritmos foram desenvolvidos em *Python*.

No algoritmo de identificação de portas, foram realizados testes com QRcodes e balizas ao lado de cada porta do ambiente a fim de identificá-las e de encontrar a relação entre cada porta e cada cômodo do ambiente que elas conectam. Enquanto no algoritmo de geração do mapa, os testes foram conduzidos em ambientes considerados simples e complexos. Ainda, são propostas métricas de avaliação para validar o algoritmo de geração do mapa.

4 RESULTADOS E ANÁLISE DAS MÉTRICAS DE AVALIAÇÃO

Esta seção contém os resultados dos testes de identificação das portas, assim como apresenta os resultados para os testes de geração do mapa topológico, representado por RP, para ambientes simples e complexos. Ainda, analisa e discute cada uma das métricas de avaliação apresentadas no Capítulo 3.

É importante destacar que como o objetivo deste trabalho é propor um método de localização e mapeamento simultâneos para robôs móveis autônomos em ambientes internos, nos testes executados em ambientes simples e complexos, é necessário que o algoritmo tenha como dados de entrada as imagens de cada cômodo para detectar as portas e suas respectivas identificações ou que o algoritmo solicite ao usuário para informar a quantidade e o id de cada porta existente no cômodo. A partir dessas informações, o algoritmo imprime na tela do computador a transição que deve ser disparada, ou seja, o id de qual a próxima porta que o sistema deve atravessar. O processo se repete até que o algoritmo imprima na tela do computador que todo o ambiente foi mapeado e gere a imagem da rede de Petri criada.

Nestes testes, o sistema supracitado é representado por uma pessoa com uma câmera para fins de validação do algoritmo, mas destaca-se que o mesmo sistema poderia ser representado por um robô móvel equipado com uma câmera.

Nesta perspectiva, independente do método utilizado para a identificação das portas, a principal discussão em questão e a principal contribuição desta pesquisa é sobre a representação do ambiente por meio de uma RP. A partir da consideração exposta, em caráter de comparação, os resultados apresentados para a identificação das portas utilizando balizas e QRcode foram conduzidos e também são discutidos neste capítulo.

4.1 Experimento I: Algoritmo de Identificação das Portas

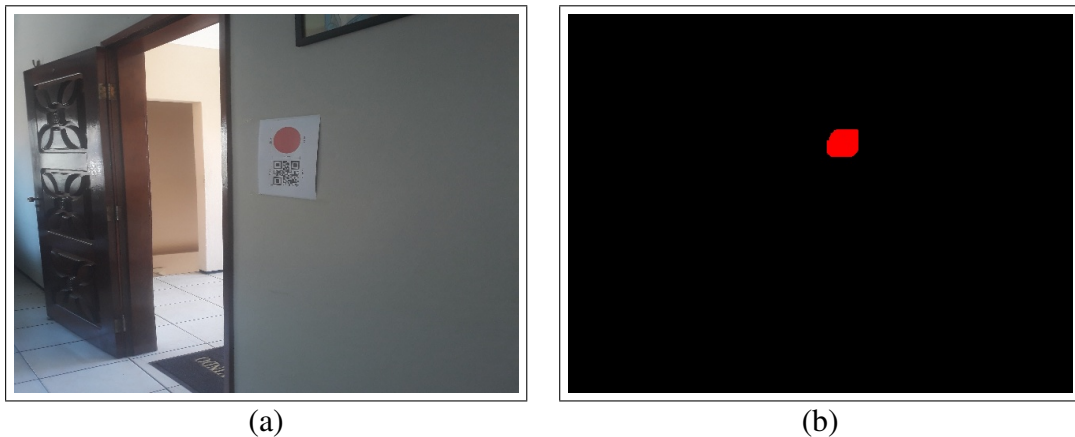
A identificação das portas de cada cômodo é fundamental para a construção do mapa do ambiente explorado, ao se utilizar o algoritmo proposto nesta pesquisa. São utilizadas marcações nas portas para que o algoritmo, a partir das técnicas de visão computacional, reconheça as portas existentes no ambiente.

Com relação aos testes com o uso da baliza, na Figura 14(a) representa-se a imagem original e na Figura 14(b) representa-se a mesma imagem após ser segmentada. De posse desta segunda imagem, um algoritmo varre a imagem pixel a pixel e busca pela baliza. Com isso, a primeira parte da validação é concluída, o algoritmo consegue encontrar as portas do ambiente, tendo em vista que cada porta possui uma baliza.

O algoritmo consegue reconhecer as portas em todos os ângulos analisados, indicando que o método de identificação de portas com o auxílio de balizas do tipo supracitado é robusto e apresenta resultados satisfatórios para imagens obtidas a partir de um ângulo de 30° a 150° em relação à superfície da porta, onde está anexada a baliza e distantes 1,5 m em relação ao solo.

Além disso, utilizando as balizas, que podem ser de diferentes tamanhos e formatos, torna-se possível identificar se cada porta encontrada é uma porta já conhecida, ou se é uma nova, por ser possível atribuir um número de identificação conforme as dimensões e/ou formato (circular, triangular, retangular entre outros) da baliza. Essa informação é fundamental, tendo em vista que cada porta representa uma transição da RP, que por sua vez é construída para

Figura 14 – Imagem do ambiente explorado (a) na escala RGB e (b) segmentada.



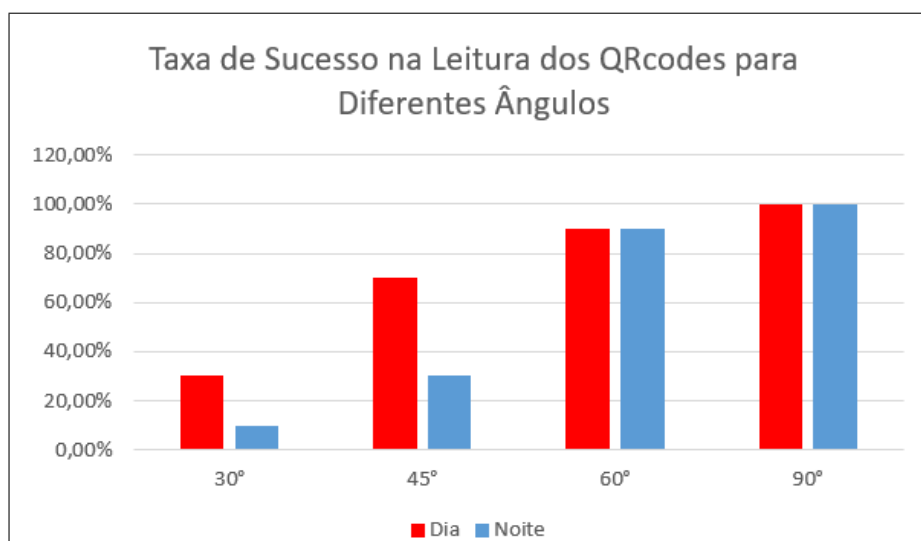
Fonte: Elaboração Própria.

representar o mapa topológico do ambiente.

É válido destacar também que o sentido de deslocamento é considerado. Em outras palavras, se o sistema sair do cômodo A em direção ao cômodo B por uma porta X, a porta X representa uma transição t_{AB} na RP. Analogamente, se o sistema sair do cômodo B em direção ao cômodo A pela mesma porta X, esta representa uma nova transição t_{BA} .

Com relação aos testes com o uso do QRcode, os resultados estão compilados na Figura 15. As imagens são adquiridas para os ângulos, em relação à superfície da parede da porta, de 30°, 45°, 60° e 90° tanto no período da manhã como no período da noite. Em cada posição angular são captadas 10 imagens, tendo sido registrada a taxa de sucesso para leitura correta do QRcode para cada ângulo.

Figura 15 – Taxa de sucesso na leitura de QRcodes em diferentes ângulos.



Fonte: Elaboração Própria.

No período da manhã, a iluminação das imagens obtidas sofre influência exclusivamente da luz natural provinda do sol que está presente no ambiente, ao passo que nos testes conduzidos

no período da noite, a iluminação das imagens obtidas sofrem influência exclusivamente da iluminação artificial por lâmpadas de LED presentes no ambiente.

Nota-se que a taxa de sucesso na detecção dos QRcodes é maior no período diurno que no período noturno. Ainda, esta taxa de sucesso é maior quanto mais próximo o ângulo de captura das imagens, em relação à parede que contém a porta, for de 90°.

Após a análise dos resultados, pode-se concluir que o método de identificação de portas utilizando balizas representadas por imagens é melhor que o método utilizando QRcode, tendo em vista que independente do período do dia e do ângulo testado, em todos os testes o algoritmo encontrou a baliza, o que nem sempre ocorreu com os testes conduzidos com os QRcodes. Entretanto, este último método possui uma característica relevante que é o fato de cada QRcode armazenar dados, neste caso o id de cada porta. Por outro lado, para identificar o id de uma porta utilizando as balizas supracitadas, seria necessário utilizar balizas com diferentes características (cor, tamanho, formato) e atribuir a cada conjunto de características um valor que seria o id da porta representada por ela.

4.2 Experimento II: Algoritmo de Geração do Mapa de Ambientes Simples

Após a execução dos testes de identificação das portas, foram realizados testes de geração do mapa em ambiente simples, a fim de se validar as técnicas propostas.

Na Figura 16(a) é representada a planta baixa¹ de um ambiente, bem como as portas que conectam os cômodos. Na Figura 16(b) é exibida a rede de Petri que representa o mapa topológico da planta baixa supracitada. As vantagens de se representar o mapa de um ambiente por meio de uma RP estão relacionadas à praticidade e ao baixo custo computacional utilizado para manipulação da rede, cujo processamento requerido está relacionado a operações matemáticas com matrizes. Isso se destaca em relação a outros métodos de mapeamento que utilizam RFID, QRcode, algoritmos probabilísticos e reconhecimento de padrões, por serem métodos complexos, limitados, seja pelo *hardware* ou condições ambientais, e, sobretudo o último, demanda alto custo computacional.

Quanto as desvantagens, tem-se a ausência de medições precisas, haja vista que esta é uma característica dos mapas topológicos. Além disso, o mapa gerado é em duas dimensões e a principal informação fornecida é a conectividade entre os cômodos do ambiente, enquanto o uso de técnicas de mapeamento com LIDAR, por exemplo, permite a construção de mapas tridimensionais.

4.3 Experimento III: Algoritmo de Geração do Mapa de Ambientes Complexos

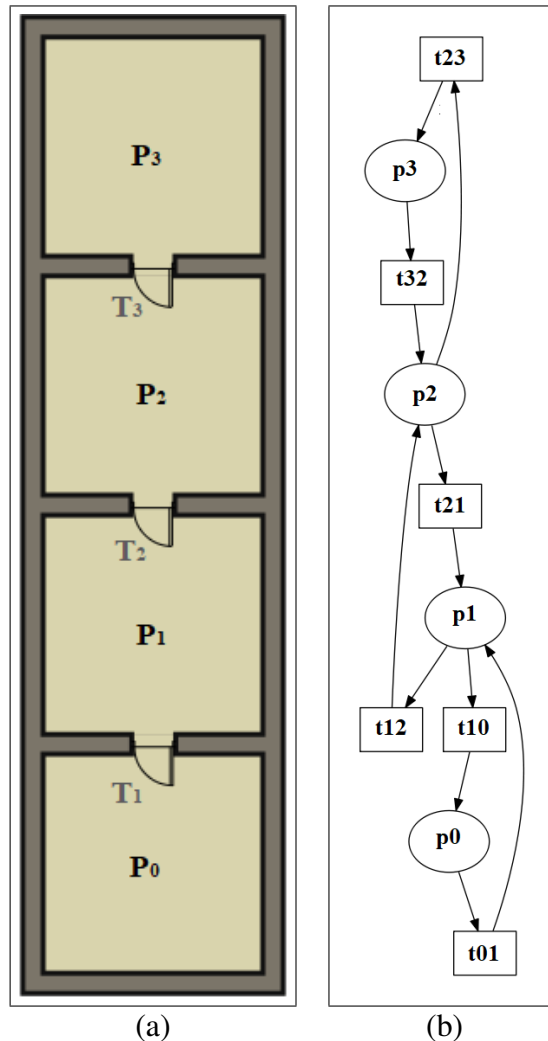
Após a efetivação dos testes em ambientes simples, foi realizada a etapa final de validação do algoritmo em quatro modelos de ambiente complexo.

Na Figura 17(a) é representada a planta baixa de um ambiente, modelo 1, constituído por cômodos que possuem conexão com mais de dois lugares e as portas que conectam os cômodos. Na Figura 17(b) é exibida a rede de Petri que representa o mapa topológico do ambiente.

Na Figura 18(a) é representada a planta baixa de um ambiente, modelo 2, constituído por cômodos que possuem conexão com mais de dois lugares, porém com *layout* diferente do modelo 1. Na Figura 18(b) é exibida a rede de Petri que representa o mapa topológico do ambiente.

¹ Planta baixa desenhada no site <https://floorplancreator.net/>

Figura 16 – Representação do ambiente simples mapeado em (a) Planta baixa e (b) sua respectiva rede de Petri.



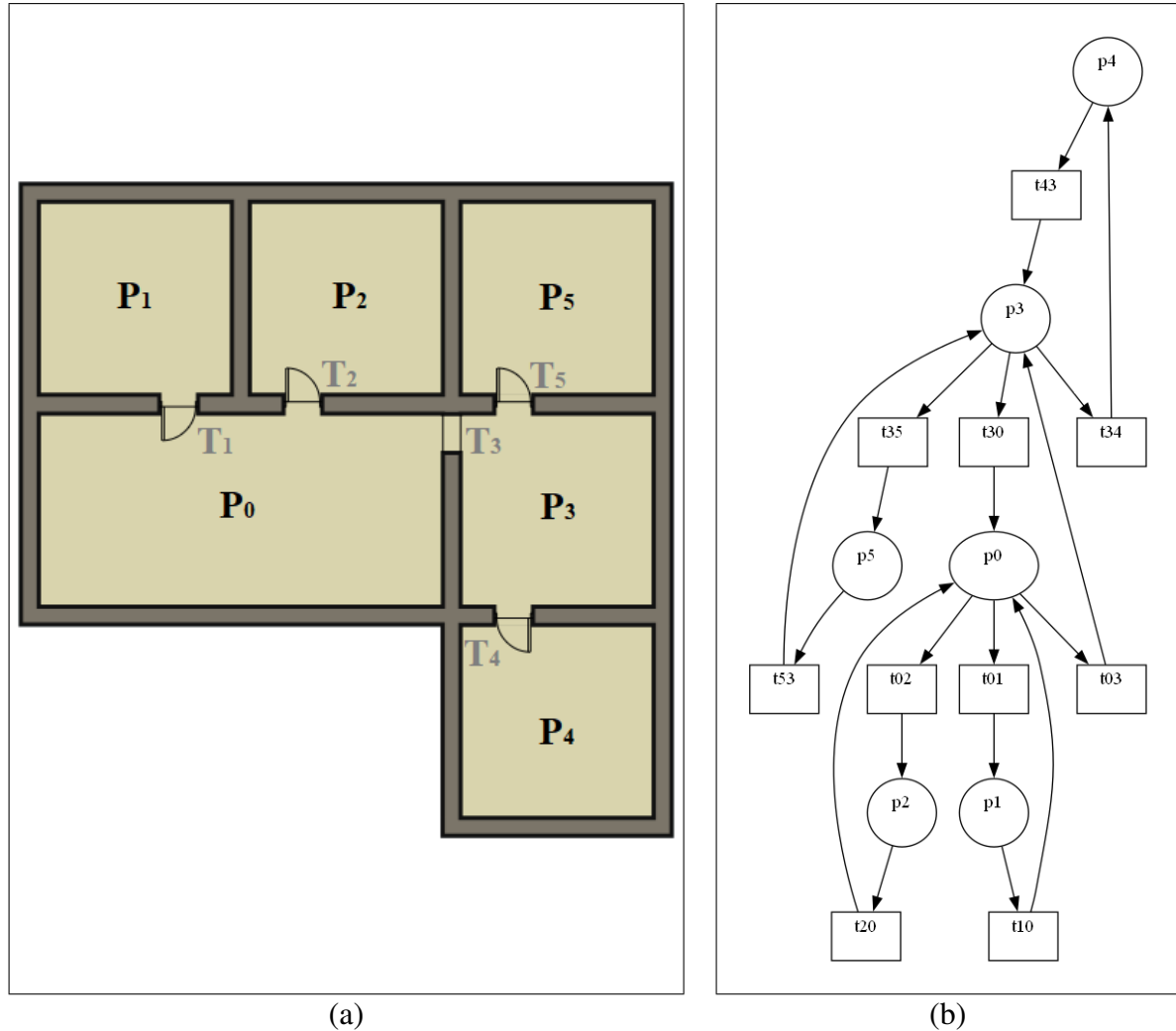
Fonte: Elaboração Própria.

Na Figura 19(a) é representada a planta baixa de um ambiente, modelo 3, que é diferente dos demais modelos pelo fato de ser constituído pelo cômodo P_1 , que possibilita a conexão com um cômodo previsto anteriormente. Por exemplo, considerando que, no início do mapeamento, a rota escolhida é sair de P_0 e ir para P_2 , quando o sistema chega em P_2 , ele encontra duas portas, sendo uma a porta pela qual ele entrou vindo de P_0 e sendo outra a porta que dá acesso ao cômodo P_1 , que já havia sido previsto na RP quando o sistema estava em P_0 . Nos modelos 1 e 2, sempre que o sistema acessa um cômodo, cada porta encontrada, exceto a porta pelo qual ele entrou, leva a um lugar ainda não previsto na RP. Na Figura 19(b) é exibida a rede de Petri que representa o mapa topológico do ambiente.

Na Figura 20(a) é representada a planta baixa de um ambiente, modelo 4, constituído pelo cômodo P_1 que possui conexão com um outro cômodo P_2 por meio de mais de uma porta. Na Figura 20(b) é exibida a rede de Petri que representa o mapa topológico do ambiente.

É importante destacar que todos os modelos de ambiente foram mapeados corretamente, tendo em vista que as redes de Petri que os representam mostram com precisão a conectividade entre os cômodos do ambiente e as respectivas portas que os conectam.

Figura 17 – Representação do modelo 1 de um ambiente complexo em (a) Planta baixa e (b) sua respectiva rede de Petri.



Fonte: Elaboração Própria.

Independente de qual cômodo do ambiente o mapeamento se inicie, o algoritmo sempre retorna uma RP que corresponde ao ambiente explorado. Neste caso, ocorre apenas alteração das nomenclaturas associadas a cada cômodo, tendo em vista que sempre o cômodo de partida é representado pelo lugar P_0 e o índice de cada lugar na rede é incrementado a partir de P_0 à medida que são explorados.

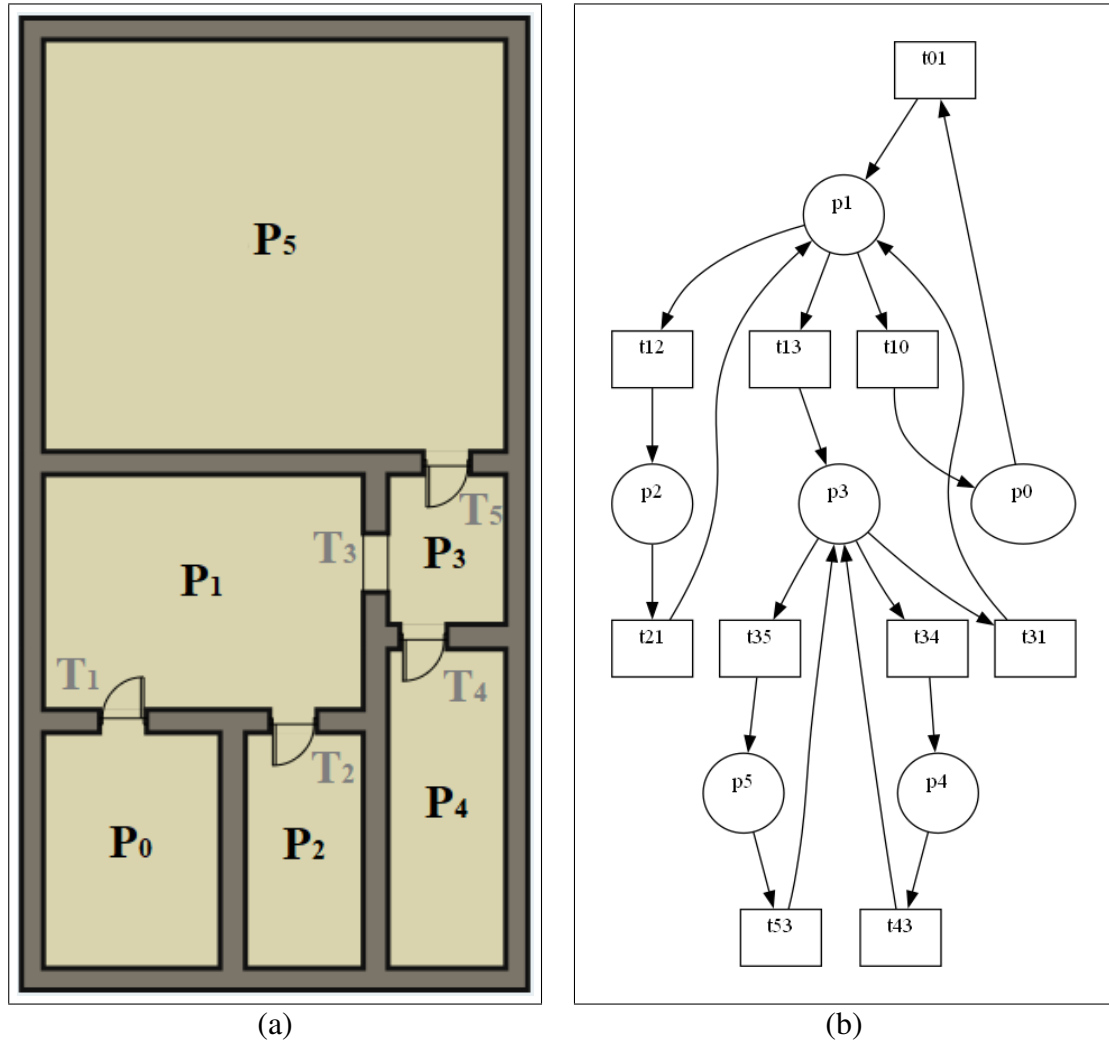
Ainda, cada lugar da RP pode ser alcançado partindo-se de um lugar qualquer da rede, pois os grafos das RP são fortemente conexos para representação do mapa topológico de um ambiente por redes de Petri.

4.4 Discussões

A análise dos resultados apresentada nesta seção está estruturada sob três aspectos: métricas de avaliação, comparação do método proposto nesta pesquisa com os métodos citados no Estado da Arte (Capítulo 2) e principais propriedades comportamentais das redes de Petri.

As tecnologias empregadas nesta pesquisa estão relacionadas com o desenvolvimento de

Figura 18 – Representação do modelo 2 de um ambiente complexo em (a) Planta baixa e (b) sua respectiva rede de Petri.

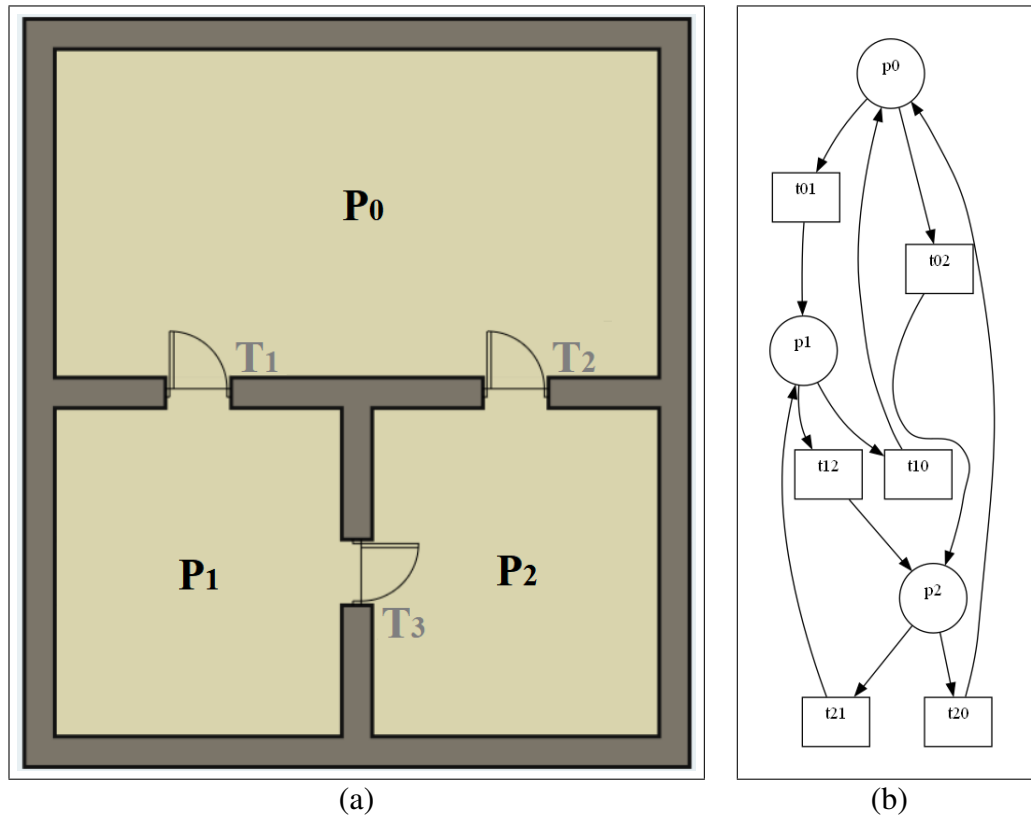


Fonte: Elaboração Própria.

um algoritmo de localização e mapeamento simultâneos para robôs móveis. Para localização, foi usada a câmera de um *smartphone Samsung Galaxy J5 Pro* e algoritmo de visão computacional. As imagens capturadas pela câmera são processadas e o resultado é a identificação das portas que existem na cena. Com relação ao mapeamento, a partir da identificação das portas, o algoritmo modela o ambiente por meio de uma rede de Petri. Assim, a partir dos resultados dos testes de identificação das portas e de geração do mapa, pode-se concluir que as técnicas propostas no objetivo deste trabalho são viáveis e eficientes.

É importante destacar que, em vez do método de captura de imagens supracitado, pode-se utilizar outras alternativas como uma *picamera* conectada a uma *raspberry* (SAHU et al., 2019; GHILDIYAL et al., 2019), câmeras omnidirecionais (JAYASURIYA; RANASINGHE; DISSANAYAKE, 2020) e robôs móveis equipados com estas ou com outros tipos de sensores (GUTIÉRREZ; LÓPEZ, 2019; BAVLE et al., 2020; ZHAO et al., 2021) para detecção das balizas. Inclusive pode-se, caso seja necessário um sistema mais robusto, utilizar um algoritmo de reconhecimento de padrões para identificar as portas sem necessidade de balizas (ZHOU; TUZEL, 2018; AFIF et al., 2019; CZERNIAWSKI; LEITE, 2020) e embarcar esse sistema em

Figura 19 – Representação do modelo 3 de um ambiente complexo em (a) Planta baixa e (b) sua respectiva rede de Petri.



Fonte: Elaboração Própria.

um AGV.

Quanto à identificação das portas, estas foram encontradas corretamente. Portanto, uma RP foi construída para cada ambiente testado. O modelo gráfico da RP representa um mapa topológico do ambiente real. Isso pode ser observado ao comparar-se as Figuras 16(a) e 16(b). O ambiente possui 4 cômodos (P_0 , P_1 , P_2 , P_3), bem como a RP possui 4 lugares (p_0 , p_1 , p_2 , p_3).

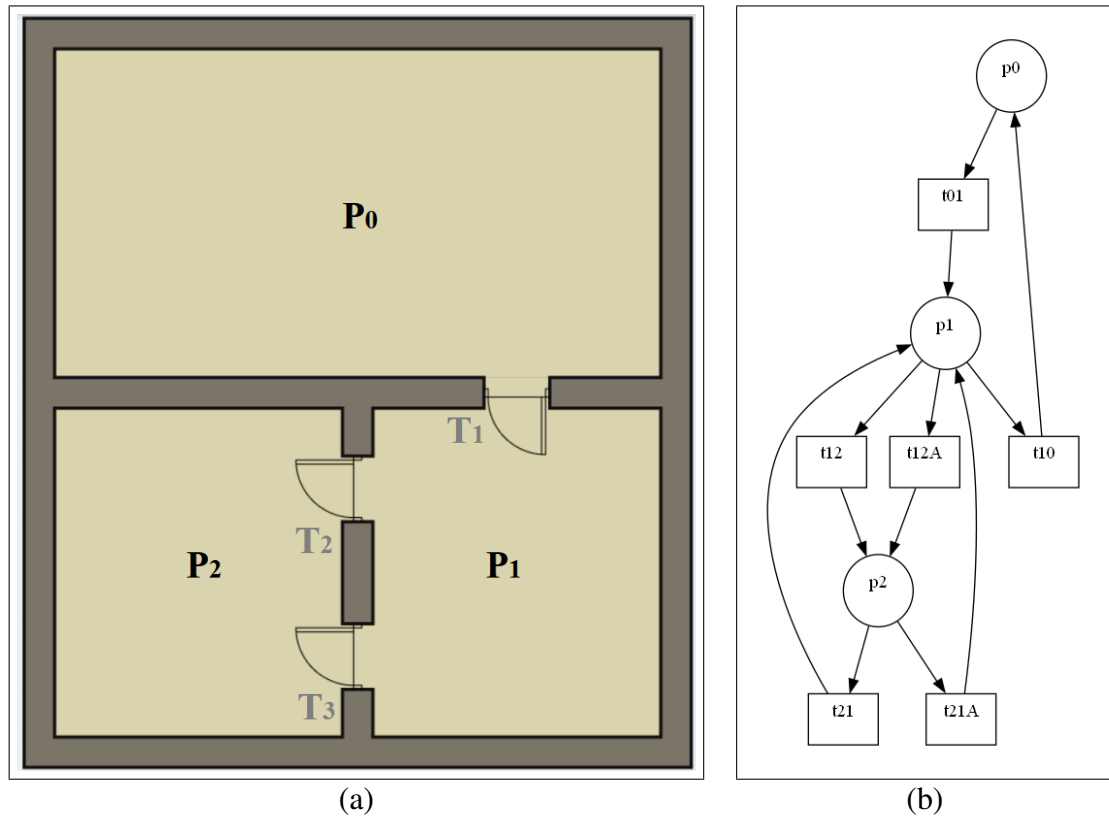
Ainda sobre o processo de identificação das portas, o algoritmo de geração do mapa permite que o usuário digite via *IDLE Shell Python*, sempre que solicitado, a quantidade de portas e o id de cada porta do ambiente.

Identificar as portas de cada cômodo do ambiente é indispensável para a construção da RP, que representa o mapa topológico da região explorada, pois cada porta está associada a um par de transições da rede.

As transições da RP diferem portas de entrada e de saída. Analisando a Figura 16(a), o ato de sair do cômodo P_2 em direção ao cômodo P_3 , implica atravessar a porta T_3 . Essa ação é coerente com a RP, tendo em vista que levar uma ficha de p_2 para p_3 implica em disparar a transição t_{23} . Assim como, o ato de sair do cômodo P_3 em direção ao cômodo P_2 , também implica em atravessar a porta T_3 , sendo essa ação ratificada pela RP, pois levar uma ficha de p_3 para p_2 implica em disparar a transição t_{32} . Portanto, a quantidade de transições da RP é equivalente à quantidade de portas no ambiente real na proporção 2:1, ou seja, cada porta do ambiente real equivale a duas transições na RP.

Diante do exposto, pode-se afirmar que a taxa de acerto, para representação dos mapas de

Figura 20 – Representação do modelo 4 de um ambiente complexo em (a) Planta baixa e (b) sua respectiva rede de Petri.



Fonte: Elaboração Própria.

ambientes considerados simples (AS) ou complexos (AC1, AC2, AC3 e AC4) por RP, é 100%, tendo em vista que todas as métricas avaliativas são atendidas, conforme Tabela 2.

Portanto, sabendo-se que mapas topológicos simplificam o processo de mapeamento de ambientes, assim como têm por principal objetivo expor a conectividade entre regiões e considerando a taxa de acerto supracitada, pode-se determinar que as RP são uma alternativa viável e confiável para a modelagem do mapa de ambientes internos.

A RP que representa o ambiente navegado pelo robô é desenvolvida em linguagem de programação Python, sendo utilizada a biblioteca OpenCV para o tratamento das imagens e a biblioteca SNAKES para a construção e simulação da RP. A biblioteca OpenCV foi escolhida devido a sua comprovada eficiência e uso na vasta literatura já publicada (BRADSKI; KAEHLER, 2008).

Quanto à biblioteca utilizada pelo algoritmo da construção da RP, foram encontradas duas opções: SNAKES (POMMEREAU, 2010), (POMMEREAU, 2015) e PNet (CHAY; GOH; LING, 2016). A SNAKES foi escolhida considerando a vasta disponibilidade de material na literatura, por possuir *plugins*, que dentre outras funções, permitem a visualização gráfica da RP, ferramentas de construção, análise e edição da rede. Já a PNet apresenta a principal vantagem de ser mais intuitiva e ser recomendada para iniciantes em programação Python. Entretanto, há poucos artigos na literatura sobre a PNet, o que torna esta ferramenta limitada para a finalidade proposta neste trabalho.

Para fins de validação da técnica proposta, é importante realizar também uma comparação

entre o mapeamento por RP e demais técnicas abordadas na literatura. A Tabela 3 contém informações para fins de análise e comparação entre os métodos propostos nesta pesquisa e os trabalhos citados no Estado da Arte (Capítulo 2). Dentre os principais métodos de geração de mapa de ambientes internos citados na literatura estão mapas baseados na extração de recursos, geometria, grade de ocupação e aprendizado de máquina.

O mapeamento baseado em recursos do ambiente consiste em extrair dados característicos e/ou significativos da área explorada a fim de se estabelecer associações e permitir a um robô se localizar (MOTA et al., 2018; BAVLE et al., 2020; ZHAO et al., 2021). Nesse caso, são utilizadas câmeras para extrair esses dados. Esse tipo de recurso também pode ser usado para fornecer os dados de entrada do ALM proposto neste trabalho, tendo em vista que a técnica proposta considerou o uso de balizas para identificação das portas do ambiente.

Quanto aos mapas baseados na geometria da área explorada, são utilizados sensores como sonares e LIDAR para coletar dados relacionados à estrutura e ao formato dos objetos que constituem o ambiente (GUTIÉRREZ; LÓPEZ, 2019; HE et al., 2020; GOBHINATH et al., 2021). O uso de LIDAR exige maior tempo e poder de processamento dos robôs quando comparado ao método proposto, além de maior custo financeiro.

A técnica de mapeamento baseada em grades de ocupação consiste basicamente em dividir o ambiente em porções menores (células) e cada porção é classificada como livre ou ocupada. Para a efetivação dessa técnica são comumente utilizados métodos probabilísticos como filtro de partículas, filtro Bayes e filtro de Kalman (KOTB; ABI-CHAR, 2017). Apesar de ser uma técnica utilizada há décadas, apresenta como principal desvantagem, em relação à técnica proposta, o fato de que no mapeamento por RP os recursos são facilmente reconhecidos e geralmente mais distintos.

O mapeamento baseado em aprendizagem de máquina é uma das técnicas de construção de mapas explorada com maior frequência nos últimos anos. Essa modelagem consiste em ensinar o robô a reconhecer padrões no ambiente navegado e a consequente tomada de decisões, considerando os dados de treinamento (MOSAVI; VARKONYI-KOCZY, 2016; PATLE et al., 2019; XIAO et al., 2022). Apresenta-se como uma das alternativas mais eficientes na geração de mapas, entretanto, suas maiores desvantagens são o tempo, principalmente quando se utiliza aprendizagem por reforço (tentativa e erro) e alto custo computacional para o processamento de dados, sobretudo quando se utilizam redes neurais artificiais.

Diante do exposto, nota-se que a técnica de mapeamento de ambientes por RP apresenta-se com potencial inovação e originalidade. Quando comparada a outras técnicas de mapeamento, nesse método são observadas vantagens como baixo custo computacional e financeiro, praticidade, rapidez e acurácia.

Com relação às propriedades comportamentais das redes de Petri, que representam o mapa topológico de um ambiente e estão ilustradas nas Figuras 16b, 17b e 18b, é realizada uma análise dos modelos a partir das principais propriedades elencadas por Dias (2007).

A RP, enquanto mapa de um ambiente, deve obedecer à propriedade da vivacidade, pois é necessário existir uma sequência de disparos de transições que leve uma marcação inicial M_0 a uma marcação alcançável M_k , de modo que para cada marcação sempre haja pelo menos uma transição habilitada. Ainda, as redes precisam ser reversíveis, ou seja, deve existir uma sequência de disparos que permita chegar a uma marcação inicial M_0 a partir de qualquer marcação alcançável M_k . Assim, é fundamental que a RP seja viva e reversível para garantir que um robô consiga sair de um cômodo qualquer e chegar a qualquer outro cômodo do ambiente mapeado.

Tabela 3 – Comparação entre o método proposto nesta pesquisa e demais métodos para localização e mapeamento.

Autores	Tipo de Pesquisa	Abordagem Proposta	Pontos Fortes	Limitações	Diferenciais do Método Proposto neste Trabalho
Kotb e ABI-CHAR (2017)	Pesquisa aplicada	SLAM probabilístico e redes de Petri	Combina comportamento de navegação e mapeamento em uma rede de fluxo de trabalho estendida	O SLAM depende de dados de odometria e a proposta está em início de desenvolvimento	Diferente da Visão Computacional, a odometria acumula erros de posicionamento
Ingrand e Ghallab (2017)	Artigo de revisão	Deliberação para robôs autônomos	Apresentam uma visão global das funções de deliberação em robótica e discutem o estado da arte nesta área	Foco da pesquisa em robôs autônomos	—
Ayawli et al (2018)	Artigo de revisão	Estratégias de planejamento de rota de robôs móveis	Analisa e discutem diversos métodos de planejamento de rota usados por robôs	Dentre os módulos de um AGV, abordam o subsistema de planejamento de rotas de um robô móvel	—
Mota et al (2018)	Pesquisa aplicada	RFID para localização e redes de Petri como mapa	Utiliza rede de Petri para representar a conectividade das regiões do ambiente	Para o robô se localizar precisa encontrar e ler tags RFID dispostas em pontos específicos	Localização usando câmera permite o robô se localizar em qualquer instante
Gutiérrez e López (2019)	Pesquisa aplicada	LIDAR e câmera estéreo	Resolve um dos problemas mais comuns da robótica na área de SLAM, que é o controle do movimento e rastreamento do robô	A visualização do mapa é feita pelo MATLAB	O uso do software MATLAB exige alto custo computacional e limita o uso em projetos embarcados
Patle et al (2019)	Artigo de revisão	Estratégias de planejamento de rota de robôs móveis	Análise considera várias condições ambientais e identifica lacunas no estado da arte	Dentre os módulos de um AGV, abordam o subsistema de planejamento de rotas de um robô móvel	—
Bavle et al (2020)	Pesquisa aplicada	SLAM visual planar semântico	Apresentam um algoritmo SLAM semântico visual rápido, robusto e leve	Testes conduzidos em robô móvel do tipo aéreo e utilizam giroscópio	Diferente da Visão Computacional, giroscópios acumulam erros de medição
He et al (2020)	Pesquisa aplicada	SLAM usando LIDAR	Teste em ambiente industrial real, melhor precisão e custo computacional reduzido	Comparação dos resultados com o algoritmo de odometria e mapeamento baseado em LIDAR	A construção e interpretação de mapas utilizando redes de Petri é eficiente para representar a conectividade de regiões e mais simples que o método proposto
Shit (2020)	Artigo de revisão	Técnicas de localização precisas para robôs móveis	Enumera e discute as principais limitações para uma localização precisa: complexidade computacional, ruído do sensor e atrasos na comunicação	Dentre os módulos de um AGV, aborda o subsistema de localização de um robô móvel	—
Gobhinath et al (2021)	Pesquisa aplicada	SLAM usando LIDAR e ROS	Custo benefício financeiro e operacional	Fragilidade estrutural, instabilidade do protótipo e utiliza MATLAB	O mapeamento por RP dispensa o uso de hardware auxiliares como computador equipado com MATLAB
Zhao et al (2021)	Pesquisa aplicada	SLAM em ambientes dinâmicos	Período de testes superior a um mês em um ambiente real e dinâmico	Testes conduzidos em um único local	Testes conduzidos em múltiplos locais explicitam a robustez do sistema
Ding et al (2022)	Artigo de revisão	Técnicas de percepção, comunicação e controle de robôs móveis autônomos	Discussões amplas sobre quatro perspectivas, em vez de uma, como se observa com frequência na literatura	Análise conduzida sobre quatro tópicos: aplicações, tecnologias de percepção, comunicações de robôs e mecanismos de controle do movimento	—

Fonte: Elaboração Própria.

A exploração de um ambiente por um robô implica que este percorre diversos trajetos. Assim, o conjunto de todos os trajetos que um robô pode realizar para sair de um cômodo e chegar a outro é equivalente ao grafo de alcançabilidade de uma RP.

É indispensável, também, para a modelagem de mapas por redes de Petri, a observância da propriedade limitação, tendo em vista que o número de fichas em cada lugar é finito. Neste caso, como a quantidade de possíveis robôs navegando por este ambiente é finita, pode-se garantir que todas as RP que representam o mapa de um ambiente são limitadas.

Por fim, tem-se a propriedade de persistência, que determina que havendo mais de uma transição habilitada em uma marcação M_k , o disparo de uma transição não desabilita as demais transições. No caso das RPs avaliadas para o mapeamento de ambientes, as redes não são persistentes por possuírem conflito efetivo, ou seja, para cômodos que possuem duas ou mais portas, quando o robô atravessar uma porta, necessariamente ele não pode atravessar outras até que retorne ao cômodo em questão. A observância, ou não, da propriedade de persistência não implica em erro no processo de construção do mapa.

5 CONCLUSÃO E TRABALHOS FUTUROS

O principal objetivo deste trabalho é propor um método de localização e mapeamento simultâneos para robôs móveis autônomos em ambientes internos. O mapa do ambiente é representado por uma rede de Petri e a localização do robô é realizada por visão computacional.

Para a realização do objetivo proposto, os objetivos específicos foram atingidos, sendo desenvolvido o algoritmo de identificação das portas (AIP) em um ambiente interno e o algoritmo de construção da rede de Petri (ALM). Os algoritmos foram implementados e avaliados por meio de três experimentos.

A partir dos resultados apresentados e discutidos ao longo desta pesquisa e considerando as métricas de avaliação utilizadas, conclui-se que é possível se localizar utilizando visão computacional e mapear um ambiente explorado utilizando rede de Petri.

Quanto às contribuições deste trabalho, destacam-se a originalidade e o potencial de inovação da técnica de mapeamento por meio de redes de Petri, bem como são propostas seis métricas avaliativas que podem ser utilizadas para validar trabalhos posteriores que usem o método de mapeamento apresentado.

Nesse sentido, considerando as contribuições mencionadas, este estudo assume um papel de relevância no campo da Robótica Móvel. Apresenta-se um novo método de modelagem de mapas de ambientes que se destaca por seu baixo custo computacional, ao mesmo tempo em que demonstra alta eficiência e intuitividade. Essa abordagem inovadora oferece uma solução promissora para a área, proporcionando uma alternativa acessível e de fácil utilização na criação de mapas de ambientes.

Em um trabalho futuro, em vez de o sistema de captura de imagens ser posicionado manualmente, pode-se utilizar um robô explorador, equipado com câmera e programado com os algoritmos AIP e ALM, que execute uma varredura do ambiente de forma autônoma.

Ainda, dando continuidade a esta pesquisa, como neste trabalho não é contemplado o módulo de planejamento de rotas, pode-se fornecer ao robô o mapa de um ambiente representado por sua respectiva RP, determinar uma missão de navegação e fornecer ao robô uma sequência de disparos de transições que determina o trajeto que ele deve executar. A ideia é utilizar a dinâmica da rede para conduzir o robô de um ponto de partida a um ponto de chegada.

Sugere-se, também, como trabalho futuro utilizar esta técnica para modelagem de ambiente *indoor* navegado por um conjunto de robôs móveis, que poderiam trabalhar cooperativamente em ambiente industrial, por exemplo. Neste último caso, é fundamental se observar as propriedades comportamentais da RP, de modo a garantir que não existam conflitos nas rotas e tarefas a serem executadas pelos robôs.

REFERÊNCIAS

- AFIF, M. et al. Indoor image recognition and classification via deep convolutional neural network. In: **Smart Innovation, Systems and Technologies**. [S.l.]: Springer International Publishing, 2019. p. 364–371.
- AGUIAR, A. S. et al. Localization and mapping for robots in agriculture and forestry: A survey. *MDPI AG*, v. 9, n. 4, p. 97, 2020.
- AJAYKUMAR, G.; STEELE, M.; HUANG, C.-M. A survey on end-user robot programming. **ACM Computing Surveys**, Association for Computing Machinery (ACM), v. 54, n. 8, p. 1–36, nov 2022.
- ALI, M. H. et al. Vision-based robot manipulator for industrial applications. **Procedia Computer Science**, Elsevier BV, v. 133, p. 205–212, 2018.
- ALLAUDDIN, M. S. et al. Development of a surveillance system for forest fire detection and monitoring using drones. In: **IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium**. [S.l.]: IEEE, 2019.
- ANDERSON, M.; MCKEE, M.; MOSSIALOS, E. Covid-19 exposes weaknesses in european response to outbreaks. **BMJ**, BMJ, p. m1075, mar 2020.
- AYAWLI, B. B. K. et al. An overview of nature-inspired, conventional, and hybrid methods of autonomous vehicle path planning. **Journal of Advanced Transportation**, Hindawi Limited, v. 2018, p. 1–27, jul 2018.
- AZIM, M. et al. Six-wheeled wireless bomb defusal rover. In: **Advances in Automation, Signal Processing, Instrumentation, and Control**. [S.l.]: Springer Singapore, 2021. p. 1021–1033.
- BANAS, W. et al. The modular design of robotic workcells in a flexible production line. **IOP Conference Series: Materials Science and Engineering**, IOP Publishing, v. 95, p. 012099, nov 2015.
- BARTOŠ, M. et al. An overview of robot applications in automotive industry. **Transportation Research Procedia**, Elsevier BV, v. 55, p. 837–844, 2021.
- BAVLE, H. et al. Vps-slam: Visual planar semantic slam for aerial robotic systems. **IEEE Access**, v. 8, p. 60704–60718, 2020.
- BESSA, J. A. et al. Global location of mobile robots using artificial neural networks in omnidirectional images. *Institute of Electrical and Electronics Engineers (IEEE)*, v. 13, n. 10, p. 3405–3414, 2015.
- BIETRESATO, M. et al. A tracked mobile robotic lab for monitoring the plants volume and health. In: **2016 12th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)**. [S.l.]: IEEE, 2016.
- BING, H. et al. A route planning method based on improved artificial potential field algorithm. In: **2011 IEEE 3rd International Conference on Communication Software and Networks**. [S.l.: s.n.], 2011. p. 550–554.

- BRADSKI, G.; KAEHLER, A. **Learning OpenCV: Computer Vision with the OpenCV Library**. [S.l.]: O'Reilly, 2008. ISBN 0596516134.
- CARDOSO, J.; VALETTE, R. **Redes de Petri**. 1. ed. [S.l.: s.n.], 1997. v. 1. ISBN 8532800955.
- CHAKRAVARTULA, S. S. N. et al. Computer vision-based smart monitoring and control system for food drying: A study on carrot slices. Elsevier BV, v. 206, p. 107654, 2023.
- CHAVES, D. et al. Integration of CNN into a robotic architecture to build semantic maps of indoor environments. In: **Advances in Computational Intelligence**. [S.l.]: Springer International Publishing, 2019. p. 313–324.
- CHAY, Z. E.; GOH, B. F.; LING, M. Pnet: A python library for petri net modeling and simulation. **Advances in Computer Science : an International Journal**, v. 5, n. 4, p. 24–30, 2016. ISSN 2322-5157. Disponível em: <<http://www.acsij.org/acsij/article/view/527>>.
- CHEN, X.; ZHANG, Q.; SUN, Y. Non-kinematic calibration of industrial robots using a rigid–flexible coupling error model and a full pose measurement method. Elsevier BV, v. 57, p. 46–58, 2019.
- CONSTÂNCIO, A. S.; CARVALHO, D. R.; TSUNODA, D. F. Aplicações de visão computacional na saúde: revisão de literatura incrementada com técnicas de processamento de linguagem natural. Research, Society and Development, v. 11, n. 10, p. e218111032942, 2022.
- CZERNIAWSKI, T.; LEITE, F. Automated digital modeling of existing buildings: A review of visual object recognition methods. Elsevier BV, v. 113, p. 103131, 2020.
- DAN, H.-C. et al. An improved computation method for asphalt pavement texture depth based on multiocular vision 3d reconstruction technology. Elsevier BV, v. 321, p. 126427, 2022.
- DENIGRES, A. Z. Z.; KOSE, E. L. Análise e aplicação da rede de petri no sistema de saúde brasileiro. **Mecatrone**, Universidade de Sao Paulo, Agencia USP de Gestao da Informacao Academica (AGUIA), v. 3, n. 1, dec 2018.
- DIAS, G. L. **Ferramentas para a Integração de Redes de Petri e VHDL na Síntese de Sistemas Digitais**. mathesis, 2007.
- DING, N. et al. A comprehensive review on automatic mobile robots: Applications, perception, communication and control. **Journal of Circuits, Systems and Computers**, World Scientific Pub Co Pte Ltd, v. 31, n. 08, mar 2022.
- DOPPLER, K. et al. Future indoor network with a sixth sense: Requirements, challenges and enabling technologies. **Pervasive and Mobile Computing**, Elsevier BV, v. 83, p. 101571, jul 2022.
- DU, Y.; ZHOU, Y.; HU, H. Dynamic assignment of roles and users for business processes under security requirements. Institute of Electrical and Electronics Engineers (IEEE), p. 1–11, 2023.
- EL-SAYED, M. E. et al. Computer vision for package tracking on omnidirectional wheeled conveyor: Case study. Elsevier BV, v. 116, p. 105438, 2022.
- ENGEMANN, H. et al. OMNIVIL—an autonomous mobile manipulator for flexible production. **Sensors**, MDPI AG, v. 20, n. 24, p. 7249, dec 2020.

EYIOKUR, F. I. et al. A survey on computer vision based human analysis in the COVID-19 era. Elsevier BV, v. 130, p. 104610, 2023.

FAROOQ, M. U.; EIZAD, A.; BAE, H.-K. Power solutions for autonomous mobile robots: A survey. Elsevier BV, v. 159, p. 104285, 2023.

FENERICK, J. A.; VOLANTE, C. R. EVOLUÇÃO DAS INDÚSTRIAS, OS BENEFÍCIOS DA AUTOMAÇÃO e AS PERSPECTIVAS DO MERCADO DA ROBÓTICA NO BRASIL e NO MUNDO. Interface Tecnológica, v. 17, n. 1, p. 734–745, 2020.

FENG, X.; LIU, Q.; WANG, Z. AUV modeling and analysis using a colored object-oriented petri net. In: **First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06)**. [S.l.]: IEEE, 2006.

FENG, Z. et al. An overview of collaborative robotic manipulation in multi-robot systems. **Annual Reviews in Control**, Elsevier BV, v. 49, p. 113–127, 2020.

FORSYTH, D.; PONCE, J. **Computer Vision: A Modern Approach. (Second edition)**. Prentice Hall, 2011. 792 p. Disponível em: <<https://hal.inria.fr/hal-01063327>>.

FRAGAPANE, G. et al. Autonomous mobile robots in sterile instrument logistics: an evaluation of the material handling system for a strategic fit framework. **Production Planning & Control**, Informa UK Limited, p. 1–15, feb 2021.

FREITAS, B. I. de; BASILIO, J. C. Diagnose de falhas em tempo real de sistemas a eventos discretos modelados por redes de petri rotuladas utilizando redes de petri rotuladas com prioridades. In: **Procedings do XV Simpósio Brasileiro de Automação Inteligente**. [S.l.]: SBA Sociedade Brasileira de Automática, 2021.

GHILDIYAL, S. et al. Control of wheeled mobile robot using centroid algorithm. In: **2019 Innovations in Power and Advanced Computing Technologies (i-PACT)**. [S.l.]: IEEE, 2019.

GOBHINATH, S. et al. Simultaneous localization and mapping [slam] of robotic operating system for mobile robots. In: **2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)**. [S.l.: s.n.], 2021. v. 1, p. 577–580.

GONG, M. et al. Fault diagnosis of motor based on mutative scale back propagation net evolving fuzzy petri nets. In: **2017 Chinese Automation Congress (CAC)**. [S.l.]: IEEE, 2017.

GONZALEZ, R.; WOODS, R. **Processamento Digital de Imagens**. 3. ed. [S.l.]: Pearson, 2009. ISBN 9788576054016.

GOTO, T. Surgeon-supporting robots. In: **Introduction to Robotics in Minimally Invasive Neurosurgery**. [S.l.]: Springer International Publishing, 2022. p. 101–114.

GUL, F. et al. Novel implementation of multi-robot space exploration utilizing coordinated multi-robot exploration and frequency modified whale optimization algorithm. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 9, p. 22774–22787, 2021.

GUNARDI, Y.; HANAFI, D. Petri net modeling for mobile robot motion in sharp turning cases. In: **2022 International Seminar on Intelligent Technology and Its Applications (ISITIA)**. [S.l.]: IEEE, 2022.

GUNARDI, Y. et al. Mathematics base for mobile robot navigation using mirror petri net method. IOP Publishing, v. 1230, n. 1, p. 012026, 2019.

GUO, M.-H. et al. Attention mechanisms in computer vision: A survey. Springer Science and Business Media LLC, v. 8, n. 3, p. 331–368, 2022.

GUPTA, S.; FATIMA, Z.; KUMAWAT, S. Study of the bioenergetics to identify the novel pathways as a drug target against mycobacterium tuberculosis using petri net. **Biosystems**, v. 209, p. 104509, 2021. ISSN 0303-2647. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0303264721001532>>.

GUTIÉRREZ, E. L. Á.; LÓPEZ, F. R. J. Global map generation and SLAM using LiDAR and stereo camera for tracking motion of mobile robot. **ITECKNE**, Universidad Santo Tomas, v. 16, n. 2, p. 58–70, dec 2019.

HE, X. et al. Lidar guided stereo simultaneous localization and mapping (slam) for indoor three-dimensional reconstruction. In: **2020 Chinese Automation Congress (CAC)**. [S.l.: s.n.], 2020. p. 7238–7242.

HOLSINGER, F. C.; BIRKELAND, A. C.; TOPF, M. C. Precision head and neck surgery: robotics and surgical vision technology. Ovid Technologies (Wolters Kluwer Health), v. 29, n. 2, p. 161–167, 2021.

HUTMACHER, F. Why is there so much more research on vision than on any other sensory modality? *Frontiers Media SA*, v. 10, 2019.

INGRAND, F.; GHALLAB, M. Deliberation for autonomous robots: A survey. **Artificial Intelligence**, v. 247, p. 10–44, 2017. ISSN 0004-3702. Special Issue on AI and Robotics. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0004370214001350>>.

JAYASURIYA, M.; RANASINGHE, R.; DISSANAYAKE, G. Active perception for outdoor localisation with an omnidirectional camera. In: **2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.]: IEEE, 2020.

KALINOV, I. et al. High-precision UAV localization system for landing on a mobile collaborative robot based on an IR marker pattern recognition. In: **2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)**. [S.l.]: IEEE, 2019.

KAVITHA, K. et al. Identifying legitimate user in ddos attack using petri net. **Materials Today: Proceedings**, 2022. ISSN 2214-7853. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2214785322036938>>.

KHATIB, E. I. A.; JARADAT, M. A. K.; ABDEL-HAFEZ, M. F. Low-cost reduced navigation system for mobile robot in indoor/outdoor environments. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 8, p. 25014–25026, 2020.

KLANCAR, G. et al. **Wheeled Mobile Robotics**: From fundamentals towards autonomous systems. [S.l.]: Elsevier Science Technology Books, 2017. 502 p. ISBN 9780128042045.

KOTB, Y.; ABI-CHAR, P. A theoretical cooperative work-flow net-based framework for odometric and probabilistic slam. In: **2017 9th IEEE-GCC Conference and Exhibition (GCCCE)**. [S.l.: s.n.], 2017. p. 1–6.

KUMAR, V. et al. Computer vision based object grasping 6dof robotic arm using picamera. In: **2018 4th International Conference on Control, Automation and Robotics (ICCAR)**. [S.l.: s.n.], 2018. p. 111–115.

- LEE, I. Service robots: A systematic literature review. **Electronics**, MDPI AG, v. 10, n. 21, p. 2658, oct 2021.
- LEONARD, J.; DURRANT-WHYTE, H. Simultaneous map building and localization for an autonomous mobile robot. In: **Proceedings IROS '91:IEEE/RSJ International Workshop on Intelligent Robots and Systems '91**. [S.l.]: IEEE, 1991.
- LEONARD, J. J.; BAHR, A. Autonomous underwater vehicle navigation. In: **Springer Handbook of Ocean Engineering**. [S.l.]: Springer International Publishing, 2016. p. 341–358.
- LI, A. Q. Exploration and mapping with groups of robots: Recent trends. **Springer Science and Business Media LLC**, v. 1, n. 4, p. 227–237, 2020.
- LI, C.; SUN, X.; CAI, J. Intelligent mobile drone system based on real-time object detection. **Computers, Materials and Continua (Tech Science Press)**, v. 1, n. 1, p. 1–8, 2019.
- LI, S. et al. Autonomous drone race: A computationally efficient vision-based navigation and control strategy. **Elsevier BV**, v. 133, p. 103621, 2020.
- _____. Robust diagnosability analysis using basis reachability graph. **Institute of Electrical and Electronics Engineers (IEEE)**, v. 11, p. 9751–9762, 2023.
- LI, Y.; YU, M.; HU, Y. Modeling and analysis of classroom teaching evaluation platform based on colored petri nets. In: **2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)**. [S.l.: s.n.], 2019. p. 442–445.
- LIANG, C.-J.; KAMAT, V. R.; MENASSA, C. C. Teaching robots to perform quasi-repetitive construction tasks through human demonstration. **Elsevier BV**, v. 120, p. 103370, 2020.
- LIU, Y. et al. Unmanned airboat technology and applications in environment and agriculture. **Elsevier BV**, v. 197, p. 106920, 2022.
- LLUVIA, I.; LAZKANO, E.; ANSUATEGI, A. Active mapping and robot exploration: A survey. **MDPI AG**, v. 21, n. 7, p. 2445, 2021.
- LU, N. et al. Research on fault diagnosis simulation of hydraulic system of cnc machine tool based on fuzzy petri net. In: **2019 4th International Conference on Power and Renewable Energy (ICPRE)**. [S.l.: s.n.], 2019. p. 201–205.
- LÄSSIG MARKUS LORENZ, E. S. I. W. R.; BUCHNER, T. Robotics outlook 2030: How intelligence and mobility will shape the future. 2021. Disponível em: <<https://www.bcg.com/pt-br/publications/2021/how-intelligence-and-mobility-will-shape-the-future-of-the-robotics-industry>>.
- MA, S.; LI, Y.; PENG, Y. Spectroscopy and computer vision techniques for noninvasive analysis of legumes: A review. **Elsevier BV**, v. 206, p. 107695, 2023.
- MONIRUZZAMAN, M. et al. Teleoperation methods and enhancement techniques for mobile robots: A comprehensive survey. **Robotics and Autonomous Systems**, Elsevier BV, v. 150, p. 103973, apr 2022.
- MORENO, F. M.; HUSSEIN, A.; GARCIA, F. Landmark placement optimization for accurate localization in autonomous vehicles. In: **2021 IEEE International Intelligent Transportation Systems Conference (ITSC)**. [S.l.]: IEEE, 2021.

MOSAVI, A.; VARKONYI-KOCZY, A. R. Integration of machine learning and optimization for robot learning. In: **Advances in Intelligent Systems and Computing**. [S.l.]: Springer International Publishing, 2016. p. 349–355.

MOTA, F. A. X. D. et al. Localization and navigation for autonomous mobile robots using petri nets in indoor environments. **IEEE Access**, v. 6, p. 31665–31676, 2018.

MURATA, T. State equation, controllability, and maximal matchings of petri nets. *Institute of Electrical and Electronics Engineers (IEEE)*, v. 22, n. 3, p. 412–416, 1977.

_____. Petri nets: Properties, analysis and applications. **Proceedings of the IEEE**, Institute of Electrical and Electronics Engineers (IEEE), v. 77, n. 4, p. 541–580, apr 1989.

MUŠIČ, G. Petri net based solution supervision and local search for job shop scheduling. **IFAC-PapersOnLine**, v. 54, n. 1, p. 665–670, 2021. ISSN 2405-8963. 17th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2021. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2405896321009472>>.

NAGATANI, K. et al. Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots. **Journal of Field Robotics**, Wiley, v. 30, n. 1, p. 44–63, oct 2012.

NEERINCX, A.; HIWAT, T.; GRAAF, M. de. Social robot for health check and entertainment in waiting room: Child's engagement and parent's involvement. In: **Adjunct Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization**. [S.l.]: ACM, 2021.

NEJAD, M. G. et al. Process sequencing for a pick-and-place robot in a real-life flexible robotic cell. **The International Journal of Advanced Manufacturing Technology**, Springer Science and Business Media LLC, v. 103, n. 9-12, p. 3613–3627, may 2019.

NILOY, M. A. K. et al. Critical design and control issues of indoor autonomous mobile robots: A review. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 9, p. 35338–35370, 2021.

OLIVEIRA, E. M. de et al. A computer vision system for coffee beans classification based on computational intelligence techniques. Elsevier BV, v. 171, p. 22–27, 2016.

ONITA, D. et al. Quality control in porcelain industry based on computer vision techniques. In: **2018 International Young Engineers Forum (YEF-ECE)**. [S.l.: s.n.], 2018. p. 79–84.

OSTAFEW, C. J. et al. Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking. Wiley, v. 33, n. 1, p. 133–152, 2015.

PATLE, B. et al. A review: On path planning strategies for navigation of mobile robot. **Defence Technology**, v. 15, n. 4, p. 582–606, 2019. ISSN 2214-9147. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2214914718305130>>.

PEREIRA, S. B. Robótica: Recorde de instalações e espaço garantido para robôs e cobots. 2020. Disponível em: <<https://revista-automacao.com/market-overview/34213-robotica-recorde-de-instalacoes-e-espaço-garantido-para-robôs-e-cobots>>.

PETRI, C. A. **Kommunikation mit Automaten**. 1962. Disponível em: <<http://edoc.sub.uni-hamburg.de/informatik/volltexte/2011/160/>>.

- POMMEREAU, F. Quickly prototyping petri nets tools with snakes. In: . [S.l.]: ACM, 2010.
- _____. SNAKES: A flexible high-level petri nets library (tool paper). In: **Application and Theory of Petri Nets and Concurrency**. [S.l.]: Springer International Publishing, 2015. p. 254–265.
- PRADOS, C. et al. ROMERIN: Organismo robótico escalador basado en patas modulares con ventosas activas. Universitat Politecnica de Valencia, v. 20, n. 2, p. 175–186, 2022.
- QIN, Q. et al. Development of ROS-based laser navigation forklift AGV system. In: **Advances in Mechanical Design**. [S.l.]: Springer Singapore, 2022. p. 2135–2145.
- RAWASHDEH, N.; ABU-ALRUB, N. Gripper control design and simulation for OpenROV submarine robot. MDPI AG, v. 10, n. 10, p. 252, 2021.
- REHMAN, B. U. et al. Towards a multi-legged mobile manipulator. In: **2016 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.]: IEEE, 2016.
- REN, C.; DING, Y.; MA, S. A structure-improved extended state observer based control with application to an omnidirectional mobile robot. **ISA Transactions**, Elsevier BV, v. 101, p. 335–345, jun 2020.
- ROMERO ROSELI APARECIDA F.; PRESTES, E. O. F.; WOLF, D. **Robotica Movei**. [S.l.]: LTC - GRUPO GEN, 2014. ISBN 9788521623038.
- RUBIO, F.; VALERO, F.; LLOPIS-ALBERT, C. A review of mobile robots: Concepts, methods, theoretical framework, and applications. SAGE Publications, v. 16, n. 2, p. 172988141983959, 2019.
- SAHU, B. K. et al. Development of hardware setup of an autonomous robotic vehicle based on computer vision using raspberry pi. In: **2019 Innovations in Power and Advanced Computing Technologies (i-PACT)**. [S.l.]: IEEE, 2019.
- SANTOS, P. G. de et al. Field robots for intelligent farms—inhering features from industry. **Agronomy**, MDPI AG, v. 10, n. 11, p. 1638, oct 2020.
- SARKER, S. et al. Robotics and artificial intelligence in healthcare during COVID-19 pandemic: A systematic review. Elsevier BV, v. 146, p. 103902, 2021.
- SCHNEIDER, M. J.; BELLON, O. R.; ARAKI, H. Experimentos em fusão de imagens de alta resolução. **Bulletin of Geodetic Sciences**, v. 9, n. 1, 2003. ISSN 1982-2170. Disponível em: <<https://revistas.ufpr.br/bcg/article/view/1428>>.
- SCHWALL, M. et al. **Waymo Public Road Safety Performance Data**. [S.l.]: arXiv, 2020.
- SHAN, T.; ENGLLOT, B. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In: **2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.: s.n.], 2018. p. 4758–4765.
- SHIT, R. C. Precise localization for achieving next-generation autonomous navigation: State-of-the-art, taxonomy and future prospects. **Computer Communications**, v. 160, p. 351–374, 2020. ISSN 0140-3664. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0140366420303935>>.

SIEGWART, R.; NOURBAKHSI, I. R. **Introduction to Autonomous Mobile Robots (Intelligent Robotics and Autonomous Agents)**. [S.l.]: The MIT Press, 2004. 335 p. ISBN 9780262195027.

SIMON, D. F.; TEIXEIRA, M.; da Costa, J. P. Availability estimation in photovoltaic generation systems using timed petri net simulation models. **International Journal of Electrical Power & Energy Systems**, v. 137, p. 106897, 2022. ISSN 0142-0615. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S014206152100137X>>.

SINGH, A.; PATIL, D.; OMKAR, S. Eye in the sky: Real-time drone surveillance system (DSS) for violent individuals identification using ScatterNet hybrid deep learning network. In: **2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)**. [S.l.]: IEEE, 2018.

SMITH, R.; CUCCO, E.; FAIRBAIRN, C. Robotic development for the nuclear environment: Challenges and strategy. **MDPI AG**, v. 9, n. 4, p. 94, 2020.

SUN, Y. A comparative study on the monte carlo localization and the odometry localization. In: **2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA)**. [S.l.]: IEEE, 2022.

SUN, Z. et al. Fuzzy adaptive recursive terminal sliding mode control for an agricultural omnidirectional mobile robot. **Elsevier BV**, v. 105, p. 108529, 2023.

TAHERI, H.; ZHAO, C. X. Omnidirectional mobile robots, mechanisms and navigation approaches. **Elsevier BV**, v. 153, p. 103958, 2020.

TANG, G.; WEBB, P.; THROWER, J. The development and evaluation of robot light skin: A novel robot signalling system to improve communication in industrial human–robot collaboration. **Elsevier BV**, v. 56, p. 85–94, 2019.

TARDÓS, J. D. et al. Robust mapping and localization in indoor environments using sonar data. **SAGE Publications**, v. 21, n. 4, p. 311–330, 2002.

THRUN, S. Robotic mapping: A survey. **School of Computer Science**, feb 2002.

_____. Springer Science and Business Media LLC, v. 15, n. 2, p. 111–127, 2003.

THRUN, S.; BURGARD, W.; FOX, D. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In: **Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)**. [S.l.]: IEEE, 2000.

TSOLAKIS, N. et al. Towards AI driven environmental sustainability: an application of automated logistics in container port terminals. **Informa UK Limited**, v. 60, n. 14, p. 4508–4528, 2021.

URREA, C.; KERN, J.; ALVARADO, J. Design and evaluation of a new fuzzy control algorithm applied to a manipulator robot. **Applied Sciences**, MDPI AG, v. 10, n. 21, p. 7482, oct 2020.

WEI, Z. et al. A research on metric learning in computer vision and pattern recognition. In: **2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)**. [S.l.: s.n.], 2018. p. 254–259.

XIAO, X. et al. Motion planning and control for mobile robot navigation using machine learning: a survey. Springer Science and Business Media LLC, v. 46, n. 5, p. 569–597, 2022.

YANG, G.-Z. et al. Combating COVID-19—the role of robotics in managing public health and infectious diseases. **Science Robotics**, American Association for the Advancement of Science (AAAS), v. 5, n. 40, mar 2020.

YANG, P. et al. A new information code for robot localization. IOP Publishing, v. 1518, n. 1, p. 012004, 2020.

YASUDA, G. Hierarchical and distributed control of robotic manufacturing processes based on petri nets. In: **2008 IEEE International Conference on Automation and Logistics**. [S.l.]: IEEE, 2008.

YIN, L.; FANG, H.; SHAO, H. Design and implementation of petri net for brain-computer interface system. In: **2019 Chinese Automation Congress (CAC)**. [S.l.: s.n.], 2019. p. 5810–5814.

YU, W. et al. Modeling and analysis of medical resource allocation based on timed colored petri net. **Future Generation Computer Systems**, v. 111, p. 368–374, 2020. ISSN 0167-739X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0167739X19334028>>.

ZGHAIR, N. A. K.; AL-ARAJI, A. S. A one decade survey of autonomous mobile robot systems. **International Journal of Electrical and Computer Engineering (IJECE)**, Institute of Advanced Engineering and Science, v. 11, n. 6, p. 4891, dec 2021.

ZHANG, C.; LU, Y. Study on artificial intelligence: The state of the art and future prospects. Elsevier BV, v. 23, p. 100224, 2021.

ZHANG, H. ye; LIN, W. ming; CHEN, A. xia. Path planning for the mobile robot: A review. MDPI AG, v. 10, n. 10, p. 450, 2018.

ZHANG, X. et al. Large-scale 3d printing by a team of mobile robots. Elsevier BV, v. 95, p. 98–106, 2018.

ZHAO, M. et al. A general framework for lifelong localization and mapping in changing environment. In: **2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.: s.n.], 2021. p. 3305–3312.

ZHONG, W.; ZHAO, J.; HU, H. Beacon-based firing control for authorization security in workflows. Institute of Electrical and Electronics Engineers (IEEE), p. 1–16, 2023.

ZHOU, J.; RENIERS, G. Petri-net based cooperation modeling and time analysis of emergency response in the context of domino effect prevention in process industries. **Reliability Engineering & System Safety**, v. 223, p. 108505, 2022. ISSN 0951-8320. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S095183202200165X>>.

ZHOU, Y.; TUZEL, O. Voxelnet: End-to-end learning for point cloud based 3d object detection. In: **2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 4490–4499.

ZHU, J. et al. Availability model for data center networks with dynamic migration and multiple traffic flows. Institute of Electrical and Electronics Engineers (IEEE), p. 1–1, 2023.

ZHU, X. A petri-net modeling approach for airport apron traffic dynamics. In: **2017 36th Chinese Control Conference (CCC)**. [S.l.]: IEEE, 2017.

APÊNDICE A – ALGORITMOS

Os algoritmos para identificação das portas (AIP) e localização e mapeamento simultâneos (ALM), além de um vídeo tutorial sobre o processo de geração do mapa topológico de um ambiente via ALM podem ser acessados em: <https://github.com/FcoAlanMota/Dissertacao.git>