



UNIVERSIDADE FEDERAL DO RIO GRANDE - FURG
CENTRO DE CIÊNCIAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
CURSO DE MESTRADO EM ENGENHARIA DE COMPUTAÇÃO

Dissertação de Mestrado

Planejamento de Próximo Melhor Ponto de Vista com Imagens Multi-domínio Baseado em Aprendizado por Reforço Profundo

Matheus Gonçalves Mateus

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal do Rio Grande - FURG, como requisito parcial para a obtenção do grau de Mestre em Engenharia de Computação

Orientador: Prof. Dr. Paulo Lilles Jorge Drews Jr.

Rio Grande, 2025

Ficha Catalográfica

M425p Mateus, Matheus Gonçalves.
Planejamento de próximo melhor ponto de vista com imagens multi-domínio baseado em aprendizado por reforço profundo / Matheus Gonçalves Mateus. – 2025.
105 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-Graduação em Computação, Rio Grande/RS, 2025.
Orientador: Dr. Paulo Lilles Jorge Drews Jr.

1. Planejamento de proximo melhor ponto de vista
2. Aprendizado por reforço profundo 3. Aprendizado contrastivo
4. Multi-domínio I. Drews Jr., Paulo Lilles Jorge II. Título.

CDU 004

Catálogo na Fonte: Bibliotecário José Paulo dos Santos CRB 10/2344

ATA DE SESSÃO DE DEFESA DE DISSERTAÇÃO DE MESTRADO

Ata nº 14/2024

Aos treze dias do mês de dezembro de dois mil e vinte e quatro, às treze horas e trinta minutos, ocorreu a sessão de Defesa de Dissertação de Mestrado de MATHEUS GONÇALVES MATEUS, que apresentou a dissertação intitulada **“Planejamento de Próximo Melhor Ponto de Vista com Imagens Multi-domínio Baseado em Aprendizado por Reforço Profundo”**, realizada sob a orientação do Prof. Dr. Paulo Lilles Jorge Drews Junior. A banca examinadora foi constituída pelos Prof. Dr. Rodrigo da Silva Guerra (FURG), Prof. Dr. Armando Alves Neto (UFMG) e Prof. Dr. Marcos Ricardo Omena Albuquerque Máximo (ITA), sob a presidência do orientador. Após a apresentação do trabalho, a banca arguiu o candidato e, a seguir, deliberou pela

- (X) aprovação da Dissertação
- () aprovação da Dissertação, sugerindo modificações no texto
- () reprovação da Dissertação

Rio Grande, 13 de dezembro de 2024.



RODRIGO DA SILVA
GUERRA:94138834087
2024.12.16 18:04:15
-03'00'

Prof. Dr. Rodrigo da Silva Guerra



Prof. Dr. Armando Alves Neto



Prof. Dr. Marcos Ricardo Omena Albuquerque Máximo



Assinado de forma digital por Paulo
Lilles Jorge Drews Junior:00124140009
Dados: 2024.12.16 18:18:27 -03'00'

Prof. Dr. Paulo Lilles Jorge Drews Junior
Orientador

À minha versão do passado, que enfrentou incertezas e desafios, e à minha versão do presente, que acreditou até o fim. Este trabalho é um marco de autossuperação e um lembrete de que sonhos se tornam realidade com esforço, resiliência e muita determinação.

AGRADECIMENTOS

Em primeiro lugar, agradeço à minha família por todo o apoio e suporte oferecido ao longo desta jornada acadêmica. Agradeço especialmente à minha mãe, Thânia, que, mesmo diante de tantas dificuldades, sempre me apoiou ao longo de todo o caminho.

Agradeço também aos meus irmãos, Rodrigo e Alexandra, e à minha cunhada, Rita. Além disso, estendo minha gratidão aos demais membros da família que aqui não foram citados, mas que, ainda assim, são fundamentais e saibam que serei eternamente grato a todos pelo incentivo e suporte que me foram dados ao longo dessa caminhada.

Agradeço ainda à minha namorada, Ingrid, por estar ao meu lado desde o início, sempre me apoiando e incentivando.

Por último, mas não menos importante, agradeço aos meus amigos, cuja presença foi de suma importância para tornar este trabalho uma realidade. Obrigado a todos pelo apoio, pela ajuda e pelo companheirismo diante de todas as adversidades que enfrentamos – adversidades que nos uniram e fizeram jus à icônica e memorável frase: “É tempo ruim o tempo inteiro!” – Charles do Bronx.

O presente trabalho foi realizado com apoio da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), Brasil. Processo nº 2024/10523-5, e pelo Programa de Recursos Humanos da Agência Nacional do Petróleo, Gás Natural e Biocombustíveis – PRH-ANP, com apoio de recursos de empresas petrolíferas, em atendimento à cláusula nº 50/2015 do contrato de R, D&I da ANP. CAPES, CNPq, FINEP e o grupo de robótica e automação Inteligente (NAUTEC - FURG) também apoiam este estudo.

“O que é bom? - Tudo o que aumenta no homem o sentimento do poder, A vontade de poder, o próprio poder. O que é mau? - Tudo o que nasce da fraqueza. O que é a felicidade? - O sentimento de que o poder cresce, de que uma resistência foi vencida. Não o contentamento, mas mais poder. Não a paz finalmente, mas a guerra.” — FRIEDRICH NIETZSCHE, ECCE HOMO

RESUMO

MATEUS, Matheus Gonçalves. **Planejamento de Próximo Melhor Ponto de Vista com Imagens Multi-domínio Baseado em Aprendizado por Reforço Profundo**. 2025. 106 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande - FURG, Rio Grande.

Em ambientes *offshore* o monitoramento de estruturas dispostas no multi-domínio ar-água é uma tarefa importante que muitas vezes depende da utilização de veículos não-tripulados para sua execução. O monitoramento é principalmente feito através de mapeamento com a reconstrução das estruturas, utilizando nuvens de pontos coletadas através de sensores como *Lidars*, sonares e câmeras embarcados ao veículo. O ambiente *offshore* é propício ao uso de um veículo híbrido aéreo-aquático não-tripulado - HUAUV, já que esse é um tipo de veículo que pode atuar livremente nos domínios ar-água, além da transição entre eles. A problemática de mapeamento propõe uma série de desafios se tratando da autonomia de veículos. Um dos principais problemas é com relação ao caminho para cobrir de forma eficiente a estrutura, sendo agravado pela problemática multi-domínio, ocasionando no problema de planejamento de próximo melhor ponto de vista - NBV. Por tanto, é proposto no presente trabalho, considerando um HUAUV, um planejador de NBV, não-baseado em modelo, capaz de atuar em multi-domínio para a cobertura de estruturas. A cobertura da estrutura é dada a partir de uma câmera estéreo, viável no ar e na água, tendo as nuvens de pontos extraídas da fusão de imagens RGB e de profundidade. O método proposto descreve um modelo de uma abordagem contrastiva panóptica de ocupação *cross-modal*, como uma representação para aprendizado por reforço profundo. Para isso, é feita a união das imagens RGB, profundidade e segmentação semântica da estrutura (2D) com nuvens de pontos (3D). O comportamento do HUAUV é definido por posições (x , y e z) e orientações, no corpo do veículo (θ) e diretamente na câmera (ϕ), empregando um comportamento com 5-DoF. Dado o funcionamento das abordagens de aprendizado utilizadas, a informação *cross-modal* é utilizada apenas durante o treinamento. O que implica em uma única entrada para o planejador, a imagem RGB da cena, caracterizando o método não-baseado em modelo, ou seja, sem conhecimento da forma da estrutura. Com o método proposto o agente foi capaz de encontrar as vistas que compõem o caminho com a melhor relação eficiência x qualidade na reconstrução, considerando a transição multi-domínio. O trabalho demonstrou resultados satisfatórios quanto à reconstrução de estruturas, destacando-se com relação à abordagem na qual foi embasada e também com relação ao estado-da-arte.

Palavras-chave: Planejamento de próximo melhor ponto de vista, aprendizado por reforço profundo, aprendizado contrastivo, multi-domínio.

ABSTRACT

MATEUS, Matheus Gonçalves. **Next-Best View Planning With Multi-Domain Images Based in Deep Reinforcement Learning**. 2025. 106 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande - FURG, Rio Grande.

In offshore environments, monitoring structures located in the air-water multi-domain is an important task that often depends on the use of unmanned vehicles for its execution. Monitoring is mainly done through mapping with the reconstruction of structures, using point clouds collected through sensors such as Lidars, sonars and cameras on board the vehicle. The offshore environment is suitable for the use of a hybrid unmanned aerial-aquatic vehicle - HUAUV, since this is a type of vehicle that can operate freely in the air-water domains, in addition to the transition between them. The mapping problem poses a series of challenges when it comes to vehicle autonomy. One of the main problems is related to the path to efficiently cover the structure, which is aggravated by the multi-domain problem, resulting in the next best viewpoint planning problem - NBV. Therefore, this work proposes a non-model-based NBV planner capable of acting in multi-domain for structure coverage, considering a HUAUV. The structure coverage is given by a stereo camera, viable in air and water, with point clouds extracted from the fusion of RGB and depth images. The proposed method describes a model of a panoptic contrastive approach to cross-modal occupancy, as a representation for deep reinforcement learning. For this, the RGB, depth and semantic segmentation images of the structure (2D) are combined with point clouds (3D). The behavior of the HUAUV is defined by positions (x , y and z) and orientations, on the vehicle body (θ) and directly on the camera (ϕ), employing a 5-DoF behavior. Given the functioning of the learning approaches used, the cross-modal information is used only during training. This implies a single input to the planner, the RGB image of the scene, characterizing the non-model-based method, i.e., without knowledge of the shape of the structure. With the proposed method, the agent was able to find the views that make up the path with the best efficiency x quality ratio in the reconstruction, considering the multi-domain transition. The work demonstrated satisfactory results regarding the reconstruction of structures, standing out in relation to the approach on which it was based and also in relation to the state-of-the-art.

Keywords: next-best view planning, deep reinforcement learning, contrastive learning, multi-domain.

LISTA DE FIGURAS

1	Aprendizado por Reforço	36
2	Políticas Determinísticas e Estocásticas.	36
3	Modelo Ator-Crítico.	40
4	Método SimCLR.	44
5	Método CURL.	47
6	Unreal Engine	51
7	Estrutura hexagonal genérica.	52
8	Eólica mono-estaca	52
9	Eólica semi-submersível.	52
10	<i>Jacket</i>	52
11	Modelo de Plataforma Semi-submersível.	53
12	Vista da estrutura com marcadores.	54
13	Vista do agente.	54
14	Diagrama do método proposto: É apresentada a relação dos dados de observação através do bloco contrastivo, sendo ele definido por três ramificações: inter-modal, multi-modal e <i>cross</i> -modal. Nesse bloco são geradas quatro funções de perda, uma para cada tipo de dado, sendo eles: imagem RGB, profundidade, segmentação semântica e nuvem de pontos. A soma dessas funções compõem a perda contrastiva que irá tornar a <i>queue</i> (<i>PixelEncoder</i>) capaz de utilizar todas essas informações para otimizar a compreensão das imagens RGB. É também é apresentada a composição das redes ator e crítico, ilustrando as ações geradas pelo agente. Essas ações referem-se à movimentação, sendo composta pelo movimento linear nos eixos x , y e z , além do movimento angular θ em torno do eixo z e o movimento angular ϕ_{gimbal} unicamente no corpo do sensor, em torno do eixo y	55
15	Avaliação de intersecção para estimativa de vistas.	57
16	Diagrama do componente contrastivo.	59
17	Ramificação inter-modal.	60
18	<i>PixelEncoder</i>	61
19	Ramificação multi-modal.	61
20	<i>ResNetEncoder</i>	62
21	Ramificação <i>Cross</i> -modal.	63
22	<i>PointNetEncoder</i>	64
23	Redes Ator e Crítico.	65

24	Estrutura geral do sistema.	71
25	Resultados de treinamento do algoritmo base para a proposta em domínios distintos.	72
26	Média das recompensas ao longo do treinamento.	74
27	Representações tri-dimensionais da estrutura de treinamento.	76
28	Representações tri-dimensionais da estrutura de treinamento.	77
29	Representações tri-dimensionais da estrutura de treinamento.	79
30	Densidade de pontos da estrutura de treinamento.	80
31	Representações tri-dimensionais da estrutura: Eólica semi-submersível	81
32	Densidade de pontos da estrutura: Eólica semi-submersível	82
33	Representações tri-dimensionais da estrutura: <i>Jacket</i>	83
34	Densidade de pontos da estrutura: <i>Jacket</i>	84
35	Representações tri-dimensionais da estrutura: Plataforma Semi-submersível.	85
36	Densidade de pontos da estrutura: Plataforma Semi-submersível.	86
37	Visualização do detalhamento da reconstrução no ambiente 4 pelo agente XPOCURL.	88
38	Recorte de reconstrução CURL.	89

LISTA DE TABELAS

1	Síntese do estado da arte para UAVs e UUVs.	35
2	Desempenho de Cobertura dos Algoritmos.	75
3	Avaliação de Recompensas e Número de Vistas.	75
4	Avaliação de Cobertura dos Algoritmos.	78

LISTA DE ABREVIATURAS E SIGLAS

AUVs	Veículos Autônomos Sub-aquáticos
CL	Aprendizado Contrastivo
CL-RRT	Loop Fechado - Árvore Aleatória de Exploração Rápida
CSQMI	Informação Mútua Quadrática de Cauchy-Schwarz
DDPG	Gradiente de Política Determinística Profunda
Deep-RL	Aprendizado por Reforço Profundo
DL	Aprendizado por Reforço
FVS	Amostragem do Ponto de Vista de Fronteira
GLNS	Pesquisa Generalizada de Grandes Vizinhanças
GTSP	Problema do Caixeiro Viajante Generalizado
HUAUV	Veículo Híbrido Aéreo-Aquático Não-tripulado
KDE	Estimativa de Densidade de Kernel
Lazy PRM	Roteiro Probabilístico Preguiçoso
LOS	Light-of-Sight
MVS	Multi-Vistas Estéreo
NBUV	Próxima Melhor Vista Subaquática
NBV	Próxima Melhor Vista
NeRF	Campos de Radiância Neural
NGS	Pesquisa Gananciosa de Grandes Vizinhanças
NN	Vizinho Mais Próximo
NP-hard	Tempo Polinomial Não-Determinístico Difícil
PDS	Amostragem de Disco de Poisson
PID	Proporcional, Derivativo e Integrativo
PRM	Roteiro Probabilístico
RAST	Árvore de Amostragem Adaptativa de Exploração Rápida
RIG	Coleta de Informação de Exploração Rápida

RRT	Árvore Aleatória de Exploração Rápida
RRT-Connect	Árvore Aleatória de Exploração Rápida Conectada
RRT*	Árvore Aleatória de Exploração Rápida Estrela
SAC	Ator Crítico Suave
SCP	Problema de Definição de Cobertura
SFM	Estrutura Para Movimento
SPM	Módulo de Predição de Superfície
SQP	Programação Quadrática Sequencial
TSDF	Função de Distância Sinalizada Truncada
TSP	Problema do Caixeiro Viajante
UAVs	Veículos Aéreos Não-Tripulados

SUMÁRIO

1	Introdução	16
1.1	Contextualização	16
1.2	Justificativa	19
1.3	Definição do Problema	19
1.4	Objetivo Geral	20
1.5	Objetivos Específicos	20
1.6	Organização do Texto	20
2	Revisão Bibliográfica	21
2.1	Planejamento de Próximo Melhor Ponto de Vista	21
2.1.1	Trabalhos Relacionados	24
2.2	Fundamentação Teórica	35
2.2.1	Aprendizado por Reforço (RL)	35
2.2.2	Aprendizado por Reforço Profundo (DRL)	41
2.2.3	Aprendizado Contrastivo (CL)	43
2.2.4	Representações Contrastivas Não-Supervisionadas para Aprendizagem por Reforço (CURL)	46
3	Descrição do Ambiente Experimental	50
3.1	Simulação	50
3.1.1	AirSim	51
3.1.2	Estruturas Alvo	52
3.2	Ambiente de Atuação	53
3.2.1	Ambiente <i>OpenAI Gym</i>	53
4	Metodologia	55
4.1	Planejador de NBV Multi-domínio	56
4.1.1	Codificação do Ambiente	56
4.1.2	Representação Contrastiva <i>Cross</i> -modal de Multi-domínio	59
4.1.3	Aprendizado Por Reforço Profundo	65
4.1.4	Inferência	67
5	Resultados	70
5.1	Configuração Experimental	70
5.2	Análise de viabilidade e comportamento	71
5.3	Análise de desempenho	73
5.3.1	Treinamento	73

5.3.2	Avaliação gradativa	75
5.3.3	Avaliação com diferentes estruturas	78
5.3.4	Discussões	87
6	Conclusão	90
	Referências	92

1 INTRODUÇÃO

1.1 Contextualização

A crescente demanda por modelos tridimensionais (3D) de alta qualidade de objetos e estruturas complexas abrange diversas áreas da indústria, entretenimento, cultura e arquitetura [1, 2, 3, 4]. Embora a computação gráfica possa atender parte dessas necessidades por meio da geração de modelos sintéticos [5, 6], a visão computacional oferece uma alternativa focada na captura da forma e das propriedades visuais de objetos reais, o que é essencial para muitas aplicações que requerem a reconstrução precisa da superfície dos objetos [7]. A automatização desse processo, muitas vezes manual e demorado, pode trazer ganhos significativos, aumentando a produtividade e a qualidade, reduzindo a necessidade de operadores especializados, minimizando o retrabalho e, consequentemente, diminuindo os custos [8, 9].

Para a automatização do processo de reconstrução, a manipulação eficiente dos sensores destaca a necessidade de um planejamento de pontos de vista - VP (do inglês, *View Planning*) [10]. As restrições dos sensores afetam diretamente as vistas definidas para cobrir ou detectar progressivamente os objetos-alvo, influenciando também na qualidade dos resultados perceptivos. Por isso, o planejador busca por um caminho ideal entre N caminhos possíveis em um espaço conhecido, visando gerar vistas que permitam ao observador avistar todas as faces visíveis do alvo [11].

Para planejar cada ponto de vista que globalmente formará um caminho ideal, o planejador considera um espaço observável conhecido, ou seja, existe o conhecimento prévio das características dimensionais do alvo. Isso garante que o horizonte de possibilidades seja finito para cada uma das vistas a serem selecionadas. Para definir cada vista são estabelecidas métricas locais e globais de avaliação, que determinam o ponto de vista que maximizará o ganho de informação para a tarefa em execução.

Ao assumir um conhecimento prévio e uma ideia global de planejamento, chega-se no conceito de próximo melhor ponto de vista - NBV (do inglês, *Next-Best View*). Este considera a seleção de cada vista, individualmente, simplificando o problema à escolha pré-definida por um número pequeno de possíveis vistas [12, 13]. Porém, em grande

parte dos problemas do mundo real, onde um modelo prévio do alvo ainda precisa ser gerado, o planejador precisa adotar um espaço observável desconhecido para a tomada de decisão. Isso implica que para cada NBV, a única informação é a vista atual. A infinidade de possíveis NBVs a serem selecionados resultam em um problema considerado não determinístico polinomial de extrema dificuldade, ou NP-hard (do inglês, *Non-deterministic Polynomial*) [14].

A formalização do problema de planejamento de NBV originou-se em meados de 1980 por Scott et al. [15]. Os autores definiram que certas características precisam ser seguidas para que um planejador seja formalmente descrito. Essas características devem ser intrínsecas ao sensor, como questões de posicionamento, quais considerações devem ser tomadas para implementação do algoritmo e como ele irá atuar sobre as propriedades do sensor.

Caracterizar o problema de planejamento de NBV foi importante para a separação direta do problema de VP. As principais diferenças entre o planejamento de NBV e VP são dadas pela saída de cada planejador e o conhecimento prévio do alvo. Enquanto no VP tem-se um cálculo direto e a saída é o caminho completo necessitando da informação pré-existente sobre o alvo, no planejamento de NBV a saída é cada NBV calculado iterativamente podendo ou não ter conhecimento prévio sobre o alvo. Este conhecimento gerou dois subgrupos para os planejadores de NBV, os baseados e não baseados em modelo. Conforme Scott et al. [15], em determinadas aplicações, o planejamento local poderia ser superior ao global, viabilizando que planejadores de NBV, em determinados casos, também fizessem uso do conhecimento prévio da estrutura.

Com a expansão dos estudos de planejamento de NBV para diversas áreas, o seu estudo em áreas como a robótica móvel tornou-se uma realidade. Sendo assim, a utilização de veículos autônomos na automatização da captura de dados para geração de modelos tridimensionais fez-se uma alternativa viável para diversas aplicações. Dessa forma, pode-se destacar a utilização de veículos aéreos não-tripulados - UAVs (do inglês, *Unmanned Aerial Vehicles*), veículos sub-aquáticos não-tripulados - UUVs (do inglês, *Unmanned Underwater Vehicles*) e até mesmo veículos híbridos aéreo-aquáticos não-tripulados - HUAUVs (do inglês, *Hybrid Unmanned Aerial-Underwater Vehicles*) em diversas aplicações como mapeamento, inspeção, fotogrametria e imageamento [16].

Com o crescimento global da indústria petrolífera, a procura por tecnologias para automatizar, otimizar e mitigar risco para trabalhadores teve o mesmo comportamento [17]. O mesmo acontece com a indústria de energia limpa, expandindo as possibilidades de disposição geográfica de parques eólicos para ambientes costeiros e *offshore*. Nesses ambientes concentra-se grande parte do crescimento e, conseqüentemente, necessidades dessas indústrias, tornando-os responsáveis pelas maiores demandas de tarefas a serem desempenhadas no ar e na água. As estruturas que se encontram nesses meios são normalmente dispostas parte dentro e parte fora da água, como acontece com plataformas

petrolíferas, navios cargueiros, tubulações de gás e de petróleo, torres eólicas entre outras estruturas industriais.

O monitoramento da saúde estrutural - SHM (do inglês, *Structural Health Monitoring*) de estruturas costeiras e *offshore*, é essencial para garantir sua segurança e integridade, já que operam em ambientes marítimos agressivos e são expostas a elementos como ondas, vento e corrosão. Esse processo consiste em avaliar, de forma contínua ou periódica, o estado das estruturas para identificar possíveis danos, desgastes e falhas antes que se tornem críticos. Isso permite um planejamento de manutenções preventivas e uma redução de riscos de acidentes [18, 19].

Atualmente, a utilização de UAVs e UUVs são fundamentais nesse monitoramento. Os UAVs são utilizados para inspecionar áreas externas de difícil acesso, como torres, hélices e laterais das plataformas, realizando voos autônomos ou controlados remotamente. Esses veículos são equipados com câmeras de alta resolução e sensores térmicos, identificando variações de temperatura e sinais de corrosão ou falhas estruturais. Por outro lado, os UUVs são usados para inspeções subaquáticas. Com câmeras e sistemas de ultrassom, esses veículos submersíveis podem alcançar áreas inacessíveis para mergulhadores, oferecendo uma análise detalhada das condições sob a água. A viabilização de propostas de HUAUVs para aplicações como essas, podem auxiliar com certas condições em ambos os meios [20, 21, 22, 23, 24].

No presente trabalho, considerando que o uso de câmeras é de extrema importância para aplicações em ambos os meios, é proposto um planejador de NBV, denominado por representações contrastivas não-supervisionadas de ocupação panóptica *cross-modal* para aprendizado por reforço - XPOCURL (do inglês, *Cross-modal Panoptic Occupancy Contrastive Representations for Reinforcement Learning*). Ele utiliza de representações contrastivas com dados de imagem RGB, profundidade, segmentação semântica e dados tri-dimensionais (nuvens de pontos) para o desenvolvimento de uma política generalista de planejamento, através de aprendizado por reforço profundo - DRL (do inglês, *Deep Reinforcement Learning*), tornando-se capaz de atuar individualmente em cada domínio e de maneira híbrida em ambos simultaneamente.

O aprendizado contrastivo - CL (do inglês, *Contrastive Learning*) é uma abordagem que traz ótimos resultados na extração de características entre os dados de entrada, através da avaliação da similaridade entre eles [25]. Inicialmente proposto para avaliação entre imagens, recebeu diversas propostas, também com ótimos resultados, combinando diferentes tipos de dados [26]. Além disso, visando o grande avanço nas aplicações de DRL, principalmente com relação a movimentação, Laskin et al. [27] propôs a abordagem de representações contrastivas não-supervisionadas para aprendizado por reforço profundo - CURL (do inglês, *Contrastive Unsupervised Representations for Reinforcement Learning*), buscando minimizar o que sabia-se sobre os problemas de convergência dos algoritmos de DRL com entradas muito grandes [28, 29] para tarefas envolvendo movimentação

e deslocamento. Com isso, surgiram diversas abordagens utilizando esse formato de aprendizado e obtendo êxito em diferentes aplicações [30, 31, 32, 33, 34, 35, 36, 37, 38].

No contexto de planejamento de NBV, a utilização do conceito CURL [27] proporciona a viabilidade de compreender diferentes características espaciais do alvo. Essa abordagem melhora a qualidade da representação dos estados no ambiente, o que é essencial para o planejamento, já que utiliza a visão como informação fundamental para exploração e cobertura. A abordagem definida pelos autores, ensina o modelo a agrupar informações semelhantes com características invariantes de interesse, como forma, textura e profundidade, ignorando variações irrelevantes, como iluminação ou pequenas oclusões. Além de distinguir aquelas que são diferentes, o que melhora a precisão e a discriminação em relação a detalhes específicos do alvo. Assim, o agente passa a compreender melhor o ambiente, resultando em um planejamento mais eficaz para definir o NBV em ambientes complexos e dinâmicos.

1.2 Justificativa

Avaliando questões de monitoramento e inspeção relacionados à SHM com o uso de veículos autônomos para a coleta de dados, que irão ser utilizados na geração de modelos tri-dimensionais para aplicações de grande impacto na indústria como *Digital Twin* [39] e *As-Built* [40], viu-se a oportunidade de explorar o desenvolvimento de um planejador de NBV multi-domínio utilizando o conceito CURL [27] para beneficiar o aprendizado de um agente através do uso de diferentes tipos de dados.

Essa abordagem foi encorajada por estudos como os de Bronstein et al. [41], onde é discutido extensivamente o efeito e determinados critérios relacionados aos tipos de dados ao se desenvolver modelos de aprendizado profundo - DL (do inglês, *Deep Learning*). Além disso, como em toda aplicação de robótica, a discrepância entre o mundo real e o ambiente de desenvolvimento é significativamente grande, e busca-se sempre a redução dessa diferença entre a simulação e o mundo real. Com esse propósito, a abordagem deste trabalho é embasada por estudos como os de Yang et al. [42], que discutem os efeitos e resultados satisfatórios do uso de dados sintéticos foto-realísticos no desenvolvimento de modelos que visam atuar em aplicações usando dados reais.

1.3 Definição do Problema

O problema formal de planejamento de NBV consiste em uma tarefa de otimização, cujo objetivo é determinar as vistas ideais para um observador em um ambiente tridimensional buscando a cobertura de um determinado alvo [15]. Seu foco principal é maximizar a quantidade de informação adquirida ou reduzir a incerteza em relação ao ambiente, considerando restrições operacionais, limitações do veículo e características específicas

do cenário.

A descrição do problema neste trabalho é dada pela formularização original do problema de planejamento de NBV, sendo ele expandido para o conceito de multi-domínio. Dessa forma, assumindo que uma nova vista a ser selecionada deve ter o campo de vista interseccionado com o campo de vista atual, independente de domínio. Assim, para efetuar o planejamento em multi-domínio é utilizada informação de imagens RGB assumindo uma câmera estéreo a bordo do veículo.

1.4 Objetivo Geral

Desenvolver um planejador de NBV (XPOCURL) através de uma política de DRL generalizada com informações panópticas e de ocupação do ambiente, buscando torná-lo capaz de usar apenas informação visual (RGB) para a efetuação da cobertura de estruturas em multi-domínio em cenários foto-realísticos simulados.

1.5 Objetivos Específicos

- Avaliar a influência contrastiva dos tipos de dados.
- Analisar o estado da arte de *encoders* em aprendizado contrastivo *multi-modal*.
- Analisar o estado da arte de *encoders* em aprendizado contrastivo *cross-modal*.
- Efetuar o planejamento de NBV no ar.
- Efetuar o planejamento de NBV na água.
- Efetuar o planejamento de NBV em ambos os domínios.

1.6 Organização do Texto

A organização do presente trabalho é dada da seguinte forma: O Capítulo 2, apresenta em primeiro lugar é descrito o embasamento para o detalhamento e definição de termos e conceitos essenciais ao planejamento de NBV. Em seguida, são apresentados os trabalhos relacionados ao problema de planejamento de NBV aplicado à cobertura para reconstrução 3D de estruturas emersas e submersas, relacionando-os com a problemática do planejamento multi-domínio e destacando os diferenciais da proposta. Por fim, é apresentado o referencial teórico utilizado para a construção da proposta. O Capítulo 3 apresenta a descrição do ambiente experimental para o desenvolvimento do trabalho. O Capítulo 4 apresenta a metodologia proposta, trazendo as principais considerações juntamente à descrição da abordagem. No Capítulo 5 os resultados são apresentados e discutidos. E por fim, a conclusão e os trabalhos futuros são abordados em Capítulo 6

2 REVISÃO BIBLIOGRÁFICA

O presente capítulo está dividido em duas grandes seções. Na primeira seção, será apresentada uma revisão referente à composição de um planejador de NBV, apresentando o embasamento teórico necessário para tal, juntamente com avaliação do estado da arte para HUAUVs, UAVs e UUVs. Dessa forma, buscou-se caracterizar um planejador relacionando-o com os conceitos da teoria da informação atrelados ao planejamento. Por fim, é apresentada uma discussão trazendo as associações entre as abordagens e relacionando-as com a proposta apresentada. Ao final da discussão é apresentado em uma tabela o resumo geral do estudo do estado da arte. Na segunda seção, será descrita a fundamentação teórica para o desenvolvimento da proposta. Nela, serão construídos os conceitos de CL, DRL e do próprio CURL [27], que foram utilizados na proposta.

2.1 Planejamento de Próximo Melhor Ponto de Vista

Para avaliar o estado da arte no escopo de multi-domínio, no contexto de UAVs, UUVs e HUAUVs, é necessário descrever previamente o conceito e os elementos necessários para a composição de um planejador, definidos por Scott et al. [15]. O conceito que define as restrições de um planejador tem diversas características, sendo as principais:

- **Especificação de qualidade do modelo:** O planejamento de visão orientado ao desempenho é baseado em uma especificação de modelo contendo requisitos de qualidade de modelo explícitos e quantificados, como precisão de medição e densidade de amostragem.
- **Algoritmo Generalizável:** A técnica deve ser capaz de ser generalizada e aplicada a uma ampla classe de sensores de alcance, sistemas de posicionamento e objetos.
- **Pontos de vista generalizados:** Além da postura do sensor, o algoritmo deve planejar parâmetros reconfiguráveis do sensor.
- **Ver sobreposição:** O algoritmo deve fornecer o grau de sobreposição de imagens necessário para integrar e registrar a informação em cada vista.

- **Robusto:** O algoritmo deve ser imune a falhas catastróficas, lidar com ruídos e artefatos do sensor e do sistema de posicionamento e exigir intervenção mínima do operador.
- **Eficiente:** O algoritmo deve ser suficientemente eficiente para ser competitivo com operadores humanos qualificados.
- **Auto-finalização:** O algoritmo deve reconhecer quando a meta foi alcançada ou quando o progresso em direção a ela estagnou.
- **Conhecimento limitado a priori:** O algoritmo deve ser eficaz com o mínimo de informação prévia do objeto, especificamente não mais do que dimensões delimitadoras aproximadas e centroide.
- **Restrições de forma:** O algoritmo deve ser eficaz com toda geometria e topologia de objeto razoável que seja compatível com os recursos de medição do sensor.
- **Restrições materiais:** O algoritmo deve ser eficaz com todas as propriedades do material do objeto compatíveis com as capacidades de medição do sensor.
- **Característica intrínsecas do sensor:** As características básicas do sensor devem ser modeladas e gerada uma matriz com os parâmetros intrínsecos.
- **Efeito de sombra:** A natureza bi-estática do sensor deve ser modelada – isto é, a separação física do laser e do detector.
- **Desempenho de medição:** O desempenho da medição do sensor deve ser modelado, incluindo a variação da precisão da medição e da densidade de amostragem dentro dos efeitos do tronco e da inclinação da superfície.
- **Postura 6-DoF (do inglês, *Degrees of Freedom*):** O algoritmo deve lidar com um sistema de posicionamento com espaço de postura irrestrito, três posições e três rotações.
- **Restrições de postura** O algoritmo deve modelar restrições nos graus de liberdade e amplitude de movimento do sistema de posicionamento.
- **Desempenho do sistema de posicionamento** O desempenho do sistema de posicionamento deve ser modelado para incluir erro de postura e tempo de reposicionamento/reconfiguração.

As características que definem os elementos necessários para um planejador de NBV, são originalmente pensadas para representações de observadores definidas por um, ou mais, sensores visuais. No contexto de robótica, essas características são adaptadas ao

funcionamento do robô e restrições do sensor acoplado à ele, já que normalmente os sensores tem seu próprio nível de abstração em termos do acesso e modificação de seus parâmetros. Tratando-se de robótica móvel, consequentemente de UAVs, UUVs e HUAUVs, essa adaptação pode ser relacionada com base nas similaridades no seu comportamento. Assim, trazendo a relação adaptada das características de: **Especificação de qualidade do modelo, Pontos de vista generalizados, Ver sobreposição, Eficiente, Auto-finalização, Conhecimento limitado a priori, Restrições de forma, Desempenho de medição, Restrições de pose e Desempenho do sistema de posicionamento.**

Além das características que devem compor um planejador de NBV, um outro conceito importante é a função de utilidade. Essa função é uma medida formal que quantifica o valor ou a informação obtida ao selecionar um determinado ponto de vista em comparação com outras possíveis vistas, sendo fundamental para orientar o planejamento avaliando e priorizando diferentes opções de vistas. A formalização da função de utilidade depende do objetivo específico da aplicação, porém comumente envolve considerações como a maximização da informação adquirida, a redução da incerteza ou a otimização de outros critérios relevantes. A formalização generalizada desse conceito supõe que, dado um conjunto de todas as possíveis vistas V e uma função de utilidade $U(v)$ para cada ponto de vista v , $U : V \rightarrow \mathbb{R}$ é uma função que mapeia pontos de vista para números reais [43, 44, 45, 46, 47]. Por tanto, o objetivo final com a formulação da função de utilidade, é encontrar a vista v^* que maximiza essa função:

$$v^* = \operatorname{argmax}_{v \in V} U(v). \quad (1)$$

Para calcular o ganho de informação, a função de utilidade faz avaliação da entropia da cena. Essa avaliação pode ser feita de maneira explícita, onde assume-se a noção de entropia da teoria da informação [48], ou internalizadas ao método abordado. Na teoria da informação, a entropia é uma medida de incerteza ou surpresa associada à uma distribuição de probabilidade. Quanto maior a entropia, maior a incerteza sobre o resultado de uma variável aleatória. A entropia $H(X)$ é uma medida da quantidade média de informação necessária para descrever o resultado de um experimento aleatório, definida por uma variável discreta X com distribuição de probabilidade $P(X)$. A quantidade de informação restante em uma determinada cena com base na probabilidade da amostra avaliada x_i , é descrita por:

$$H(X) = - \sum_i P(x_i) * \log_2(P(X)). \quad (2)$$

Sabendo que existem dois subgrupos de planejadores, baseados e não-baseados em modelo, que relacionam o conhecimento a priori de características do alvo, Scott et al. [15] aponta também a relação da função de utilidade para cada grupo. Para os métodos baseados em modelo, a função de utilidade define uma vista a partir de um conjunto de

vistas válidas geradas por um determinado critério. Já para os métodos não-baseados em modelo, as vistas são implicitamente passadas para a função de utilidade como forma de otimização tentando prever a informação que seria obtida a partir de diferentes pontos de vista. Ou então, aprendendo uma política para seleção de pontos de vista diretamente a partir de dados, sem gerar explicitamente uma lista de candidatos como é abordado por Chen et al. [49] e no presente trabalho.

2.1.1 Trabalhos Relacionados

A literatura no contexto multi-domínio, onde encontram-se os HUAUVs, traz em sua maioria trabalhos voltados ao âmbito da modelagem dinâmica e construção desse tipo de veículo [20, 21, 50, 51]. Isso é dado, devido ao fato de ainda ser uma classe muito conceitual e embrionária em termos de prototipagem, sendo esse um dos fatores limitantes para que estudos sobre navegação e planejamento de modo geral sejam realizados para HUAUVs. Portanto, propostas para essa classe de veículo comumente assumem um veículo com estabilidade mecânica e simplificação na dinâmica de movimento, capazes transicionar livremente entre os meios e navegar entre eles individualmente de maneira simulada. Majoritariamente as propostas de navegação e planejamento constantes na literatura, assumem o veículo baseado em Drews et al. [20], como também é considerado no presente trabalho.

Grando et al. [22] propuseram uma abordagem de controle contínuo baseada em DRL. Para isso, foram embarcados no veículo um *Lidar* e um sonar fazendo a transferência de aprendizado do que foi aprendido no ar com o *Lidar*, para a água com o sonar. Para o controle efetivo do veículo foram avaliadas uma abordagem determinística com o algoritmo gradiente de política determinística profunda - DDPG (do inglês, *Deep Deterministic Policy Gradient*) [52] e outra estocástica com o algoritmo ator-crítico suave - SAC (do inglês, *Soft Actor Critic*) [53]. A abordagem determinística obteve melhores resultados em ambientes mais simples e ambas tiveram bons resultados em ambientes de maior complexidade. Porém, a abordagem estocástica mostrou uma melhor política de exploração levando a resultados semelhantes ou até superiores que os da abordagem determinística. Conclui-se que ambos os agentes foram capazes de aprender as dificuldades dos ambientes, perceber a transição de meio e generalizar o suficiente para realizar a navegação sem mapas pelo ar e pela água aprendendo também a contornar obstáculos para chegar a um alvo, onde os algoritmos tradicionais falharam.

Pinheiro et al. [23] propuseram um planejador de trajetória híbrido baseado no algoritmo árvore aleatória de exploração rápida de laço fechado - CL-RRT (do inglês, *Closed Loop - Rapidly-exploring Random Tree*) [54], que consiste em uma variante do tradicional algoritmo de planejamento de caminho árvore aleatória de exploração rápida - RRT [55] (do inglês, *Rapidly-exploring Random Tree*). Para isso, foi desenvolvida uma função de custo, característica nessa abordagem de algoritmo para avaliação dos caminhos, baseada

na eficiência energética do veículo, considerando um modelo de controle proporcional derivativo integrativo - PID (do inglês, *Proportional, Derivative, Integrative*) [56] e diferentes dinâmicas entre os meios. Alcançando resultados satisfatórios na geração de caminhos válidos utilizando transições entre os meios para o deslocamento de um ponto ao outro.

de Jesus et al. [24] propuseram uma abordagem de controle contínuo baseada na CURL [27], tendo como principal objetivo lidar com problemas tidos por algoritmos de DRL para lidar com imagens em tarefas complexas como a de navegação. Para isso, assumindo o uso de uma câmera estéreo a bordo do veículo, foi efetuada a implementação do controle através da estrutura proposta por Laskin et al. [27], originalmente fazendo uso de imagens RGB e uma memória de repetição aleatória, utilizando imagens de profundidade e uma memória de repetição priorizada. Concluindo que a CURL [27] pôde desempenhar o controle do veículo em ambos os meios em ambientes com certo nível de complexidade.

Mais recentemente, Zeng et al. [57] propuseram uma abordagem de planejador de caminho e trajetória para um veículo híbrido de forma simulada, utilizando um modelo de veículo que difere-se do modelo citado anteriormente, com um controlador proposto em Lu et al. [58]. O método é voltado para missões de exploração multi-domínio utilizando uma grande quantidade de sensores que medem diferentes propriedades da água. O planejador baseia-se no ganho de informação, inspirado no planejamento de NBV, para explorar o ambiente e coletar dados relevantes evitando a colisão com obstáculos, já que a cobertura não tem um alvo em específico. O principal foco é maximizar o ganho de informação de todas as fontes. Para isso, propõem um algoritmo denominado árvore de amostragem adaptativa de exploração rápida - RAST (do inglês, *Rapidly-exploring Adaptive Sampling Tree*), onde obtiveram resultados satisfatórios no planejamento de caminho e trajetória multi-domínio com o veículo em questão.

De modo geral, a literatura de planejamento de NBV em multi-domínio considerando um único veículo, já que no presente trabalho se assume que o NBV precisa estar no campo de vista atual, carece de trabalhos nesse escopo. O que torna necessária a avaliação das abordagens tomadas em trabalhos especificamente aplicados a UAVs e UUVs, no contexto do problema de planejamento de NBV. Portanto, nas subseções seguintes serão apresentadas e discutidas as abordagens do estado da arte de planejamento de NBV para cobertura de estruturas no contexto de UAVs e UUVs.

2.1.1.1 Estado da Arte para UAVs

Começando pelas abordagens baseadas em modelo, Hepp et al. [59] propuseram uma abordagem, formulando o problema de NBV como um problema do caixeiro viajante - TSP (do inglês, *Traveling Salesman Problem*) com uma câmera RGB embarcada. O modelo é gerado através de um voo prévio produzido através de um mapa de profun-

cidade [60], criado a partir da imagem RGB, criando um OctoMap [61] baseado nesse mapa. A geração dos pontos de vista candidatos é feita com base na região de interesse definida com base no voo inicial. Duas formas de pontos de vista são geradas, a primeira é densa e próxima da região de interesse e a segunda é menos densa e nas proximidades da região de interesse. Os pontos são definidos através da relação entre os *voxels* [62] visíveis comparados ao OctoMap por meio de *raycasting*. A função de utilidade é implícita como uma otimização da sequência de pontos de vista definida através do algoritmo de múltiplas vistas estéreo - MVS [63] (do inglês, *Multi View Stereo*), gerando o modelo 3D a ser comparado através do método de estrutura a partir de movimento - SFM [64] (do inglês, *Structure From Motion*).

Voltado à objetos tri-dimensionais genéricos, Almadhoun et al. [65] apresentaram uma abordagem, formulando o problema de NBV como um TSP utilizando um *Lidar* 2D e uma câmera RGB-D embarcados. O método não gera pontos de vista candidatos de maneira explícita, portanto, a própria abordagem de seleção dos NBVs faz essa amostragem internamente. Fazendo a avaliação da simetria entre o OctoMap [61] registrado e o gerado previamente em um voo exploratório, avaliando o estado dos *voxels* [62]. Para gerar os pontos de vista foram desenvolvidos dois módulos, denominados amostragem de ponto de vista de grade adaptativa - AGVS [66] (do inglês, *Adaptive Grid Viewpoint Sampling*), responsável por gerar pontos de vista com base na informação, e amostragem do ponto de vista da fronteira - FVS [66] (do inglês, *Frontier Viewpoint Sampling*) responsável por gerar pontos de vista que alinham o sensor com o alvo. O método de amostragem do ponto de vista AGVS é inicialmente usado até que um mínimo local seja detectado com base na mudança de entropia. O componente de amostragem do ponto de vista muda para FVS, que direcionará o robô para áreas com baixa densidade. Depois de cobrir a área de baixa densidade, a amostragem do ponto de vista volta para AGVS. Para então selecionar os pontos de vista, buscam a maximização da função de utilidade, onde o principal objetivo é maximizar o ganho de informação e a densidade do modelo 3D final representado como uma PCD.

Voltado para a cobertura de estruturas urbanas, Ramesh et al. [67] propuseram uma abordagem, onde o problema é caracterizado pela formulação de uma variação do TSP denominada problema generalizado do caixeiro viajante - GTSP (do inglês, *Generalized Traveling Salesperson Problem*), com uma câmera RGB e *Lidar* embarcados. O método proposto gera os NBVs candidatos criando um grafo baseado nos *voxels* [62] visualizados no OctoMap [61] gerado. Os estados dos *voxels* (livre, ocupado e desconhecido) são computados com probabilidade de 0 estar livre, 1 estar ocupado e 0.5 significando que pode estar ocupado ou livre com igual probabilidade com base no modelo inicial. A função de utilidade, aqui implícita, possui duas partes. A primeira utiliza um solucionador denominado pesquisa generalizada em grandes vizinhanças - GLNS [68] (do inglês, *Generalized Large Neighborhood Search*) para definir o NBV. A segunda busca minimi-

zar a diferença entre a distância calculada com uma RRT da primeira aresta na solução GTSP e a distância euclidiana atual gerando um modelo 3D final representado como uma nuvem de pontos - PCD (do inglês, *Point Cloud Data*).

Também voltado a estruturas urbanas é apresentada por Shang and Shen [69] uma abordagem de planejamento de NBV caracterizada como um problema de otimização do algoritmo MVS e fundamentada como uma análise topológica utilizando uma câmera RGB embarcada. O método proposto gera os NBVs candidatos em uma elipsoide tendo um dos raios limitados pela área da malha tridimensional, que representa o modelo da estrutura, e o outro pelo campo de visão da câmera. A função de utilidade para selecionar os NBVs busca maximizar simultaneamente a métrica de qualidade de duas visualizações dentro de um subconjunto de vistas candidatas. A fidelidade do modelo é calculada como uma métrica de erro que mede a distância do vizinho mais próximo, das correspondências de pontos entre a malha tridimensional re-amostrada e a entrada. O resultado final é um modelo 3D melhorado, com base no modelo inicial e nas leituras feitas nos pontos de vista. Para gerar a trajetória utilizam em cascada três diferentes algoritmos, um solucionador LKH-TSP [70], para encontrar o caminho mais curto entre as vistas em conjunto, OBBTree [71] para detecção de colisão e em seguida é aplicado uma árvore estrela aleatória de exploração rápida - RRT* [72] (do inglês, *Rapidly-exploring Random Tree - Star*) para efetuar o trajeto entre os pontos de vista.

Agora tratando-se de abordagens não baseadas em modelo, Charrow et al. [73] propuseram um planejador de NBV genérico para estruturas de média escala, abordando um problema de otimização para seleção de ações de controle com uma câmera RGB-D embarcada. O método busca gerar caminhos, locais e globais, diretamente, ao invés de planejar cada novo NBV. Esses caminhos são baseados na cena visualizada, buscando o ganho de informação com a abordagem definida em Charrow et al. [74]. Para a seleção dos globais é utilizado o algoritmo de Dijkstra [75]. Enquanto para a seleção local é gerada uma ação de controle, denotada pela posição em x , y e z e orientação em θ . A função de utilidade que atua sobre a seleção local, dada como uma função de informação mútua quadrática de Cauchy-Schwarz - CSQMI [76] (do inglês, *Cauchy-Schwarz Quadratic Mutual Information*) que também gera a representação 3D da estrutura como uma PCD. Com as ações controle selecionadas, é aplicado o algoritmo de programação quadrática sequencial - SQP [77] (do inglês, *Sequential Quadratic Programming*) como laço de controle.

A fim de efetuar a cobertura de construções urbanas, Jing et al. [78] propuseram um planejador de NBV formulando um problema de cobertura de conjuntos - SCP (do inglês, *Set Coverage Problem*) utilizando uma câmera RGB-D embarcada. O método é definido pelos autores como semi-baseado em modelo, já que, transforma estruturas contidas em imagens, utilizando imagens de satélites e as referências de infraestrutura da planta relacionada ao mapa também de satélite, em estruturas 3D estimando a altura com base na quantidade de andares. Para aumentar a quantidade de triângulos e melhorar a representação

do modelo aplica o método *bubble mesh* [79] para a subdivisão de triângulos. Com o modelo gerado, os pontos de vista candidatos são gerados pelo método proposto anteriormente pelos autores [80], baseado na dilatação de *voxels* [81] e campos potenciais [82]. A função de utilidade definida busca maximizar a quantidade de triângulos visualizados com a relação ao modelo gerado, selecionando os NBVs através do algoritmo proposto de busca gananciosa na vizinhança - NGS (do inglês, *Neighborhood Greedy Search* para gerar um modelo 3D melhorado da estrutura.

Dhami et al. [83] propuseram um planejador de NBV, voltado para estruturas genéricas, formulando especificamente um problema de trajetória utilizando um *Lidar* multi-feixes embarcado. Para gerar os pontos de vista candidatos, primeiro utilizam um método de completção de PCD, denominado PoinTr [84] para prever a estrutura a ser reconstruída a partir dos dados coletados ao levantar voo nas primeiras iterações. Em seguida, com a estrutura predita, são calculados diversos pontos de vista com base em três métricas. As duas primeiras métricas relacionam o posicionamento dos pontos no plano (x, y) e a terceira busca relação com o elemento z da PCD. A função de utilidade para selecionar os pontos de vista, e ao mesmo tempo gerar a trajetória a ser percorrida, busca avaliar a menor distância possível analisando cada um dos pontos candidatos através do algoritmo árvore aleatória de exploração rápida conectada - RRT-Connect [85] (do inglês, *Rapidly-exploring Random Tree - Connect*) e gerando uma PCD que representa a estrutura.

Procurando também lidar com estruturas de larga escala com diferentes formatos, Hardouin et al. [86] abordam o planejamento de NBV como um problema de TSP com múltiplos UAVs utilizando câmeras estéreo embarcadas. Mostrando-se interessante para o contexto do presente trabalho por definir detalhadamente uma abordagem específica a ser implementada como planejador individualmente nos veículos. Dessa forma, para a avaliação da cobertura, informação e gerar o modelo 3D registrado da estrutura, que é gradativamente incrementado ao longo do caminho, utilizam um método definido como função de distância sinalizada truncada - TSDF [87] (do inglês, *Truncated Signed Distance Function*). Para a definição dos NBV, são gerados candidatos através de um grafo direcionado ponderado que considera a estrutura registrada. Para a seleção é considerado para cada observador um problema do caixeiro viajante assimétrico - ATSP [88] (do inglês, *Assimetric Traveling Salesperson Problem*), que unificados irão solucionar o TSP. A função de utilidade proposta busca minimizar um caminho Hamiltoniano [89] através da heurística de Lin-Kernighan [90], em função de um modelo de controle preditivo [91], para gerar a representação 3D final da estrutura. E os NBVs são selecionados por um roteiro probabilístico preguiçoso PRM [92] (do inglês, *Lazy Probabilistic Roadmap*), onde as arestas são geradas dinamicamente à medida que são necessárias durante o caminho.

Buscando também reconstruir grandes estruturas urbanas, Feng et al. [93] trazem como proposta de planejador de NBV um método que busca lidar com um ATSP, utili-

zando uma câmera estéreo embarcada. Para gerar os pontos de vista candidatos, propõem um método de completção de PCD, denominado módulo de predição de superfície - SPM (do inglês, *Surface Prediction Module*), responsável por estimar o modelo da estrutura a ser coberta e reconstruída. É utilizado na geração dos pontos uma Amostragem de Disco de *Poisson* - PDS [94] (do inglês, *Poisson Disk Sampling*) com base na estrutura estimada pelo módulo SPM. Para selecionar os pontos candidatos é feita uma análise global através do algoritmo de planejamento A* [95] e em seguida uma análise local com o algoritmo MVS. A função de utilidade é implícita, buscando otimizar o algoritmo de MVS para gerar a PCD final que representa a estrutura predita.

Guédon et al. [96] propuseram-se a reconstruir estruturas de larga escala com diferentes formatos, gerando novas complexidades com relação as outras propostas. Para isso, desenvolveram uma abordagem que lida diretamente com o problema de NBV visando o uso de apenas uma câmera RGB. A geração dos pontos de vista candidatos é dada de forma simplista, como dito pelos autores, definindo uma amostragem local aleatória, gerando os próximos pontos em torno do ponto de vista atual assumindo que o mesmo seja o melhor ponto de vista passado. Para selecionar os NBVs são definidos três módulos, o primeiro módulo transforma a imagem RGB em um mapa de profundidade para que sejam extraídas as nuvens de pontos e os próximos módulos sejam executados. O segundo módulo, é baseado em um *transformer* [97] que busca estimar o percentual atual do volume de ocupação através de uma sequência de imagens e a nuvem registrada. Por fim, o terceiro módulo, utilizando o volume de ocupação e a vista da câmera, estima o ganho de informação da cobertura de superfície [98] definindo o próximo NBV. A função de utilidade baseia-se em avaliar o ganho máximo de informação da cobertura para avaliar o NBV e gerar uma PCD que representa a estrutura a ser coberta.

Mais recentemente, Chen et al. [49] propuseram uma abordagem de DRL, denominada GenNBV, para cobertura de construções utilizando imagens RGB e de profundidade através de uma câmera estéreo. Para isso, propuseram a utilização do algoritmo PPO [99] utilizando a fusão da informação estéreo, vinda da observação do agente, com um bloco híbrido de informação temporal e geométrica unificada. Esse bloco busca integrar a informação da observação com a informação semântica da cena, que é dada através de um módulo externo que cria uma representação semântica temporal de todas as imagens RGB já observadas, a representação geométrica em *voxels* completa da cena, com dimensionamento otimizado, e uma representação temporal das posturas em que cada observação foi coletada. Nessa abordagem, a função de utilidade se relaciona diretamente com a função de recompensa do agente, que avalia a ocupação dos *voxels* comparados à representação completa da estrutura. Buscando a generalização de uma política de planejamento de NBV, foi efetuado o treinamento utilizando duas base de dados, House3K [100] e Omnibject3d [101]. Assim, com a política generalizada, o planejamento do caminho entre as vistas foi dado utilizando o algoritmo RRT [55] e o método de conjunto de níveis de

marcha rápida para frentes que avançam monotonicamente - FMM [102] (do inglês, *Fast Marching Method*).

2.1.1.2 Estado da Arte para UUVs

Analisando inicialmente abordagens baseadas em modelo, com a proposta de reconstruir a parte submersa de uma embarcação, Hollinger et al. [103] abordaram o problema como uma formulação baseada em inspeção ativa e no problema clássico de teste de hipóteses sequenciais [104] utilizando uma câmera RGB e um sonar de imageamento 3D embarcados. O modelo no qual baseiam-se é gerado através de uma navegação prévia, fazendo uma coleta manual dos dados que representaram o modelo inicial através de PDS[94]. Em seguida, geram os pontos de vista candidatos através do método de roteiro redundante baseado em amostragem proposto em [105]. Para a seleção dos NBVs, definido como um subproblema de TSP, a função de utilidade é implícita, buscando uma parametrização na visualização de cada triângulo na malha tridimensional gerada inicialmente. E dessa forma, utilizam três avaliações para definição do caminho a ser percorrido. Primeiro utiliza a heurística de *Christofides* [106] para obter uma solução inicial que, para o TSP métrico, caia dentro de três metades do custo da solução ótima. Então, é aplicada a heurística encadeada de Lin-Kernighan [107] por um período de tempo designado para reduzir a exploração e gerar o modelo 3D final. Por fim, é aplicado o algoritmo RRT [55] para interligar os NBVs selecionados evitando a colisão com a estrutura.

Sheinin and Schechner [108] propuseram a reconstrução de estruturas submersas de maneira generalista. A abordagem é formalizada como um problema de NBV geral, utilizando uma câmera RGB e uma fonte de luz embarcados. Mesmo tendo conhecimento da cena, o algoritmo não gera vistas candidatas para o par câmera-luz. A seleção de NBVs é dada diretamente pela função de utilidade, tendo a seleção local escalada para uma seleção global. A função de utilidade busca maximizar o ganho de informação com base na função que caracteriza o método próxima melhor vista subaquática - NBUV [109] (do inglês, *Next Best Underwater View*), que tem objetivo de gerar a representação 3D da estrutura. O método se baseia na análise dos triângulos que compõem a malha tridimensional do modelo inicial, utilizando a função de utilidade para prever um novo NBV com base na qualidade do NBUV.

Lewis et al. [110] propuseram a reconstrução de uma estrutura genérica através da formalização de um problema de cobertura submersa utilizando uma câmera RGB embarcada. O modelo, no qual o método baseia-se, é gerado através de uma navegação prévia que gera a região de interesse identificada a partir de dados de sonar de varredura lateral de baixa resolução, obtidos de uma varredura de alta altitude. Com base no ganho de informação relacionado ao modelo, utilizam o método proposto por Wu et al. [111] para reconstrução 3D a partir de imagens 2D. Buscando a maximização da qualidade do modelo, se utilizam da região de interesse para gerar caminhos baseados no algoritmo roteiro

probabilístico - PRM [112] (do inglês, *Probabilistic Roadmap*). A geração e seleção dos NBVs é efetuada calculando uma quantidade de intersecções para possíveis visualizações com base nos caminhos candidatos. A função de utilidade é implícita ao algoritmo, onde busca maximizar a visualização do número de faces da região de interesse com a relação ao modelo inicial para gerar o modelo 3D final da estrutura.

No contexto de abordagens não baseadas em modelo, Vidal et al. [113] buscaram reconstruir estruturas submersas com a formalização de um problema de exploração subaquática contínuo utilizando uma câmera RGB e um sonar. Para efetuar o planejamento, e também a representação 3D da estrutura, basearam-se no OctoMap [61] com a fusão de imagens RGB e a profundidade acústica. Para gerar os NBVs candidatos, basearam-se em torno da operação do vizinho mais próximo - NN [114] (do inglês, *Neighbor Nearest*). Os NBVs são gerados ao longo da superfície normal, com objetivos de explorar o ambiente via sonar ou garantir a cobertura visual da cena, dependendo da célula candidata. A seleção é feita pela função de utilidade, que equilibra distanciamento e prioriza vistas próximas, penalizando as mais profundas. O caminho é definido pelo algoritmo RRT*[72] e seguido pelo controlador LOS [115], que alinha o trajeto à estrutura.

Com o intuito de reconstruir superfícies volumétricas submersas, Zacchini et al. [116] propuseram uma abordagem formulada como um problema de exploração e ganho de informação utilizando um sonar de imageamento. Inicialmente, geram um OctoMap [61] da superfície visualizada com as imagens do sonar para avaliar a incerteza e consequentemente o ganho de informação. A geração dos pontos de vista é dada pelo algoritmo de núcleo de estimação de densidade - KDE [117] (do inglês, *Kernel Density Estimation*), onde o algoritmo foi treinado para aprender a disposição de probabilidade do OctoMap [61] e gerar vistas candidatas. Para a seleção das vistas e planejamento de caminho, a função de utilidade é maximizada pelas vistas geradas pelo KDE [117] que é atrelado ao algoritmo RRT [55]. Essa relação é definida como algoritmo de coleta de informações de exploração rápida - RIG [118] (do inglês, *Rapidly-exploring Information Gathering*), onde são geradas amostras nas regiões inexploradas do espaço de trabalho e, assim, direciona a árvore para as áreas mais promissoras para gerar o modelo 3D da estrutura. E isso se dá pelo fato de aprender *online* a distribuição das informações adquiridas, ou seja, o mapa descoberto.

2.1.1.3 Associação e Discussão do Estado da Arte

O problema formal de planejamento de NBV é uma tarefa de otimização que busca determinar as vistas ideais para um observador em um ambiente tridimensional [15], seu principal objetivo é maximizar a informação adquirida ou minimizar a incerteza sobre o ambiente. Para isso são consideradas restrições operacionais, restrições do veículo e características específicas do ambiente. A definição geral descrita por Scott et al. [15] e a escassez no estado da arte de HUAUVs no escopo de navegação e planejamento de

modo geral, ocasionou na análise do estado da arte de UAVs e UUVs individualmente elencando alguns principais pontos, baseados em Scott et al. [15], para a caracterização de um planejador de NBV. Através dessa análise, foram identificadas características, mesmo que de diferentes formas, que foram apresentadas em ambos os contextos, tendo em vista as diferentes restrições de cada classe de veículo e os possíveis sensores embarcados.

A consideração inicial para o desenvolvimento do planejador de NBV é com relação ao conhecimento da estrutura a ser coberta. Os métodos que tem o conhecimento prévio do modelo 3D da estrutura, ou geram um modelo prévio incompleto, são ambos definidos como: baseados em modelo [59, 65, 67, 69, 103, 108, 110]. Os métodos que não tem conhecimento prévio nenhum da cena ou da estrutura a ser reconstruída, são definidos como: não-baseados em modelo [73, 78, 86, 83, 93, 96, 49, 113, 116].

Os métodos que necessitam de uma navegação prévia para gerar um modelo inicial, foram aqui definidos também como métodos baseados em modelo, pois o modelo gerado com a navegação exploratória inicial se colocava como entrada nos métodos propostos. Um ponto que destacou-se, foi que a maior parte dos trabalhos no escopo de UUVs são abordagens baseadas em modelo. As outras características relacionadas, e tidas como de maior importância, são com relação à representação do modelo 3D final da estrutura, geração dos pontos de vista candidatos, seleção dos NBVs, com a definição de uma função de utilidade, e planejamento de caminho.

Considerando o sensoriamento embarcado, mesmo tendo sido considerados diferentes sensores (câmeras RGB, câmeras RGB-D, câmeras estéreo, *Lidar*, sonares e sonares de imageamento), a representação do modelo 3D final deu-se por uma nuvem de pontos [73, 65, 67, 96, 93, 83, 113] ou uma malha tridimensional [78, 59, 86, 69, 49, 108, 110, 103, 116]. Podendo-se avaliar que as abordagens com UAVs tiveram suas representações bem divididas, já as abordagens subaquáticas com UUVs tiveram quase em sua totalidade a representação da estrutura como um modelo 3D no formato de malha tridimensional. Esse é um reflexo do sensoriamento utilizado e da abordagem tomada, baseada em modelo, na maioria das propostas.

Para a geração e seleção dos pontos de vista candidatos, avaliou-se que a definição da formalização do problema é o principal fator que define as abordagens tomadas. Tendo como destaque a formalização de um problema de planejamento de NBV como uma variação do TSP [67, 65, 59, 93, 86]. Outra formalização definida como o problema de cobertura [78, 108, 110], é também relacionada em ambos os contextos. Por tanto, as abordagens que configuram um TSP buscam para geração de vistas candidatas, e consequentemente para seleção dos NBVs, heurísticas que se relacionam com escopo da teoria de grafos. As abordagens de cobertura, por outro lado, trazem uma função de utilidade baseada na dilatação de *voxels* [62] e campos potenciais. Além disso, Sheinin and Schechner [108] abordaram a maximização de primitivas geométricas visualizadas e Lewis et al. [110] buscam a maximização da visualização de faces da região de interesse.

Na definição do planejamento de caminho, é na grande maioria dos trabalhos, abordada a utilização de variantes do algoritmo de planejamento RRT [55]. Isso é dado pelo fato de a estrutura de implementação do algoritmo poder favorecer o planejamento global nos métodos propostos. Além disso, tendo em vista que a função de utilidade seleciona o NBV, assume-se com a implantação do RRT que o caminho gerado é o caminho ótimo para reconstrução da estrutura. Os que não se inserem sobre esse escopo, utilizam-se do algoritmo de MVS [59, 93] e estratégias de controle contínuo [73, 86] para efetuar o caminho ao longo da estrutura.

Ao avaliar os métodos propostos nas abordagens baseadas em modelo ao longo dos anos, percebeu-se que em sua totalidade utilizam métodos matemáticos iterativos para definir a seleção local e também para o planejamento global dos NBVs. Isso é dado pelo fato de ser mais simples a avaliação da disposição e dimensionamento da estrutura, já que é de conhecimento do planejador desde a primeira iteração. O oposto acontece nas abordagens não-baseadas em modelo, onde busca-se a redução da incerteza através de previsões.

Na água, as abordagens não-baseadas em modelo, coincidentemente as mais recentes, trazem o contexto de busca espacial com Vidal et al. [113] trazendo o algoritmo NN e mais atualmente Zacchini et al. [116] trazendo o algoritmo KDE. No ar, as propostas se iniciam com Charrow et al. [74] não trazendo nenhuma forma explícita de previsão, Jing et al. [78] utilizando os campos potências e Hardouin et al. [86] trazendo o método Lazy - PRM para a redução de incerteza.

Mais recentemente, foram apresentados métodos que utilizam algum tipo de aprendizado (DL) para reduzir a incerteza da cena. Em Dhami et al. [83], Feng et al. [93] e Guédon et al. [96] são apresentados métodos que utilizam um módulo, com propostas diferentes em cada trabalho, para completar a PCD coletada enquanto o veículo levanta voo e prever um modelo inicial da estrutura. Esse modelo, auxilia-rá no planejamento de NBV utilizando como informação a entrada de imagem RGB [96] ou estéreo (RGB e profundidade) [83, 93]. Chen et al. [49] por outro lado, propuseram uma abordagem de DRL para a generalização de uma política o algoritmo PPO que em tempo de execução usa como entrada a informação estéreo e o histórico temporal semântico e das posturas obtidas, além da representação geométrica.

A fundamentação dos métodos propostos por Dhami et al. [83], Feng et al. [93] e Guédon et al. [96], baseia-se em utilizar o modelo 3D predito da estrutura como uma métrica de referência para aplicar a proposta de planejamento de NBV. Porém, os métodos para completar nuvens de pontos utilizados, buscando prever os pontos faltantes para a reconstrução, e desenvolvidos no caso de Feng et al. [93], são treinados com uma grande quantidade de dados mas com uma pequena variedade de classes, sendo em média de 5 à 8 classes de objetos reconhecidos. Portanto, a variedade de estruturas a serem reconstruídas é escassa, tornando as abordagens pouco generalistas. A abordagem de Chen et al. [49]

difere-se das demais nesse contexto, pois a previsão é baseada na análise de incerteza da ocupação, com um modelo que estima o percentual ocupado ao longo do caminho, e não em um modelo 3D predito explicitamente.

Chen et al. [49] propuseram algo que é que amplamente discutido por Laskin et al. [27] e de Jesus et al. [24] com relação ao uso de grandes fontes de informação como entrada de algoritmos de DRL. Os resultados obtidos pela proposta são de grande relevância, porém dois principais pontos devem ser avaliados: volume de treinamento e inferência. Isso se dá pelo fato de que precisaram de um grande período de treinamento, na casa dos 32 milhões de passos, o que torna extremamente dependente de *hardwares* de topo de linha para o treinamento desse método. Além do fato da inferência necessitar de todos os dados de entrada utilizados durante o treinamento, o que pode ser ineficiente de modo geral pensando em contextos do mundo real.

No presente trabalho a abordagem é definida como não-baseada em modelo, buscando uma política generalista atuando em multi-domínio. Dessa forma, propôs-se explorar em tempo de treinamento a fusão multi-domínio de informações 2D e 3D, e em tempo de execução apenas a imagem RGB para definir as ações do agente XPOCURL. Assim, para à avaliação de cobertura, propôs-se um método baseado em particionamento de espaço binário - BSP [119] (do inglês, *Binary Space Partitioning*), sendo ele independente da geração de um modelo 3D e tornando o processo de treinamento mais eficiente e genérico. Além disso, adaptou-se o algoritmo RRT* [72] para o rastreamento visual do alvo, como é abordado por Benhimane and Malis [120] e Wang et al. [121], para garantir o contato visual ao longo de todo o caminho entre as vistas.

O presente trabalho assemelha-se em termos conceituais com a proposta de Guédon et al. [96] com relação à ocupação. No entanto, de forma geral, a proposta de Chen et al. [49] é que a mais recai sobre o escopo desse estudo, pois utiliza de diferentes tipos de dados para a generalização de uma política de DRL como um planejador de NBV. Diferenciando-se das demais abordagens por dois principais motivos: avaliação de cobertura independente da representação 3D registrada e uso de múltiplos dados apenas em tempo de treinamento. Ambos os pontos citados tornam a proposta mais eficiente, rápida e generalista para o contexto de aplicação multi-domínio.

Com a revisão apresentada ao longo da seção, alinhando-se com a proposta descrita, o uso de câmeras RGB e consequentemente o uso da imagem bruta como principal informação, mostrou-se válido na água e no ar mesmo que com finalidades diferentes. Sabendo que a noção de profundidade é essencial para reconstrução de elementos 3D, e que em termos de eficiência energética a redução de peso agregado é crucial para um veículo que busca a transição entre os meios [23], mostra-se que a utilização de uma câmera estéreo é positiva no contexto multi-domínio.

Por fim, com as abordagens de DL apresentadas pode-se perceber que a convergência e generalização dos planejadores de NBV, mostraram-se mais eficientes em termos de

diferentes estruturas que os métodos matemáticos. O que torna mais relevante a abordagem proposta de um planejador de NBV (XPOCURL), baseado na CURL [27], como é descrita nos próximos capítulos. Dessa forma, é apresentado na Tabela 1 o resumo da correlação do estado da arte para UAVs e UUVs com os principais pontos avaliados em ambos os contextos, além do posicionamento do presente trabalho com relação ao estado da arte.

Veículo	Abordagem	Sensor(es)	3D	Método	Planejamento	Autor(es)
UAV	Baseada em modelo	RGB	Malha	Grafos	MVS	Hepp et al. [59]
		<i>Lidar</i> , RGB-D	PCD	Ocupação	AGVS, FVS	Almadhoun et al. [65]
		<i>Lidar</i> , RGB	PCD	Ocupação	RRT	Ramesh et al. [67]
	Não baseada em modelo	RGB	Malha	Grafos	RRT*	Shang and Shen [69]
		RGB-D	PCD	Grafos	CSQMI	Charrow et al. [73]
		RGB-D	Malha	Ocupação	NGS	Jing et al. [78]
		Estéreo	Malha	Grafos	Lazy-PRM	Hardouin et al. [86]
		<i>Lidar</i>	PCD	Pred-NBV (DL)	RRT-Connect	Dhami et al. [83]
		Estéreo	PCD	PredRecon (DL)	MVS, A*	Feng et al. [93]
		RGB	PCD	MACARONS (DL)	<i>Decision Make</i>	Guédon et al. [96]
		Estéreo	Malha	GenNBV (DL)	RRT	Chen et al. [49]
UUV	Baseada em modelo	RGB, Sonar	Malha	Grafos	RRT	Hollinger et al. [103]
		RGB	Malha	Ocupação	NBUV	Sheinin and Schechner [108]
		RGB, Sonar	Malha	Grafos	PRM	Lewis et al. [110]
	Não baseada em modelo	RGB, Sonar	PCD e Malha	Ocupação	RRT*	Vidal et al. [113]
	em modelo	Sonar 3D	Malha	Ocupação	RIG	Zacchini et al. [116]
HUAUV	Não baseada em modelo	Estéreo	PCD	XPOCURL (DL)	RRT*	O autor

Tabela 1: Síntese do estado da arte para UAVs e UUVs.

2.2 Fundamentação Teórica

2.2.1 Aprendizado por Reforço (RL)

O conceito de aprendizado por reforço é dado de forma que um determinado agente, caracterizado por um algoritmo específico, atua em um determinado ambiente. Essa atuação é dada pela execução de uma determinada ação por parte do agente. Em contraponto, o ambiente retorna para o agente o estado futuro em que ele se encontra após exercer essa ação, juntamente com uma recompensa que determina a qualidade da ação tomada [122]. Essa relação que descreve o funcionamento do formato de aprendizado é

ilustrado na Figura 1.

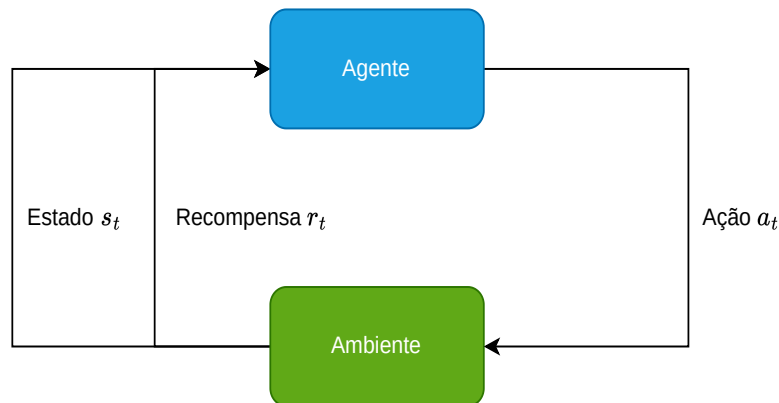


Figura 1: Aprendizado por Reforço

Fonte: O autor

Para determinar o aprendizado do agente é definido uma política que caracteriza a estratégia que o agente segue para tomar as decisões em um determinado estado, podendo ser determinística ou estocástica, como é ilustrado na Figura 2. Além da política, é definida uma função de valor que estima a utilidade de estar em um determinado estado ou tomar uma determinada ação. Pode ser a função de valor de estado (V) ou a função de valor de ação (Q). Dessa maneira, métodos em-política (do inglês, *On-policy*) tentam avaliar ou melhorar a política usada para tomar decisões. Em contraste, os métodos fora-da-política (do inglês, *Off-policy*) avaliam ou melhoram uma política diferente daquela utilizada para gerar os dados. O desenvolvimento do aprendizado do agente no ambiente divide-se em explorar novas ações para descobrir recompensas potenciais, e explorar ações já conhecidas para obter recompensas imediatas [123].

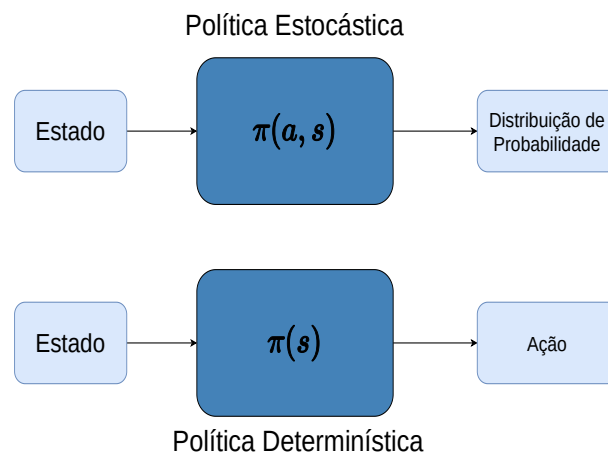


Figura 2: Políticas Determinísticas e Estocásticas.

Fonte: O autor

2.2.1.1 Processo de decisão de Markov

A descrição mais simples do aprendizado por reforço é a formulação de um Processo de Decisão de Markov - MDP [124] (do inglês, *Markov Decision Process*), onde a propriedade de Markov implica que o futuro depende apenas do presente. Assim, para cada passo de tempo, o processo está em algum estado s , e o tomador de decisão pode escolher qualquer ação a disponível. O processo responde na próxima etapa movendo-se aleatoriamente para um novo estado s' , e dando ao tomador de decisão uma recompensa correspondente $R_a(s, s')$. A probabilidade do processo passar para seu novo estado s' é influenciado pela ação escolhida, dado pela função de transição de estado $P_a(s, s')$. Dessa forma, o próximo estado s' depende do estado atual s e da ação do tomador de decisão a . Mas s e a , são independentes de todos os estados e ações anteriores, definindo que as transições de estado de um MDP satisfazem a propriedade de Markov.

A definição de um Processo de Decisão de Markov é descrita por uma quadrupla (S, A, P_a, R_a) , onde:

- S é um conjunto de estados denominado espaço de estados.
- A é um conjunto de ações denominado espaço de ação.
- $P_a(s, s') = P(s_{t+1} = s' | s_t = s, a_t = a)$ é a probabilidade de que a ação a no estado s em um tempo t , levará ao estado s' no tempo $t + 1$.
- $R_a(s, s')$ é a recompensa imediata recebida após a transição do estado s para s' , devido à ação a .

O propósito do aprendizado por reforço é orientar o agente na maximização da recompensa ao longo do tempo, guiando suas ações com base no estado atual e nas condições do ambiente. Esse direcionamento é alcançado por meio de uma função de recompensa, que estabelece a meta a ser atingida pelo agente. Conforme o agente executa ações, ele transita por distintos estados sendo recompensado de acordo com a eficácia das ações tomadas. Com a abordagem baseada no processo de Markov, o agente utiliza os seguintes passos em cada tempo t :

- O agente observa o estado atual s_t ou s .
- O agente faz a ação a_t ou a .
- O agente recebe a recompensa $r_t = R(s_t, a_t)$, sendo $R(s_t, a_t)$ a recompensa imediata após a transição de s_t com a ação a_t .
- O agente entra no próximo estado s_{t+1} ou s' .

A formalização do processo de Markov é dada através da equação de Bellman. Nessa equação, tem-se que a política aprendida para um determinado par de estado e ação $Q(s, a)$ é resultado da soma de dois fatores:

$$Q(s, a) = r(s, a) + \gamma \max_a Q(s', a), \quad (3)$$

onde o primeiro fator é a recompensa recebida através da avaliação do estado e da ação em questão, sendo determinada por uma função $r(s, a)$. O segundo fator é o produto entre o fator de desconto γ , sendo um número escalar entre zero e um, utilizado para admitir um valor de desconto para a importância da recompensa futura. E o elemento de máxima ação obtida com a política alcançada pelo estado futuro e ação tomada $\max_a Q(s', a)$.

A metodologia do processo de Markov emprega a equação de Bellman modificada, para identificar a ação mais vantajosa para um agente em um estado específico. Isso é feito considerando o desconto temporal, que penaliza estratégias desnecessárias ou que levam muito tempo para atingir o objetivo:

$$V(s_t) = \max_{a_t} (R(s_t, a_t) + \gamma V(s_{t+1})). \quad (4)$$

Nessa fórmula é representado qual é o maior valor esperado do estado, onde V é definido como o valor do estado no ambiente. Com a equação de Bellman e as probabilidades de cada ação possível, é determinada a equação do MDP, que é essencial para o aprendizado por reforço:

$$V(s_t) = \max_{a_t} (R(s_t, a_t) + \gamma \sum_{s_{t+1}} P(s_{t+1} | s_t, a_t) V(s_{t+1})). \quad (5)$$

Ao introduzir a incerteza nas ações, o processo de decisão de Markov possibilita a modelagem de um ambiente de previsão mais preciso em um contexto mais abstrato e estocástico. Isso contrasta com uma abordagem anterior, na qual todas as ações eram executadas em um ambiente determinístico, sem levar em consideração a probabilidade associada à realização de cada ação.

2.2.1.2 *Q - Learning*

Esse é um algoritmo de aprendizagem por reforço livre-de-modelo para aprender o valor de uma ação em um estado específico. Não requer um modelo de ambiente, e pode lidar com problemas com transições estocásticas e recompensas sem exigir adaptações. Tem-se que para qualquer MDP finito, o *Q - Learning* encontra uma política ótima a fim de maximizar o valor esperado da recompensa total ao longo de todas etapas sucessivas, começando pelo estado atual. O algoritmo pode identificar uma política de seleção de ação ideal para qualquer MDP finito, dado um tempo de exploração e uma política

parcialmente aleatória. Porém, ao contrário da abordagem *Markoviana* o agente avalia a qualidade de cada ação considerando o estado atual e não os futuros. A função Q , denominada função de valor, descreve a competência da ação de um agente em um determinado estado. A equação do *Q-Learning* pode ser derivada a partir da abordagem de Markov, dando origem à seguinte equação:

$$Q(s_t, a_t) = \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) [R(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) | s_t, a_t]. \quad (6)$$

Após determinar o valor de Q para uma ação específica em um estado, é crucial ter em mente que o agente provavelmente precisará explorar o ambiente repetidas vezes. Para facilitar o aprendizado por reforço, é essencial incorporar uma diferença temporal e uma taxa de aprendizado (α). Isso permite que o algoritmo aprimore seu desempenho ao longo do tempo, mesmo quando as ações não resultam em benefícios imediatos para o agente. De maneira que, podemos expressar essas relações por meio das seguintes equações:

1. A equação da diferença temporal entre os valores de Q :

$$TD_t = \Delta Q(s_t, a_t); \quad (7)$$

$$TD_t = R(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q_{t-1}(s_t, a_t). \quad (8)$$

2. A equação que atualiza o valor de Q para cada ação que passou por aquele estado:

$$Q_t(s_t, a_t) = Q_{t-1}(s_t, a_t) + \alpha TD_t(s_t, a_t). \quad (9)$$

Com a definição da Equação 9, é possível construir um agente que é capaz de aprender por meio de reforços tornando-se inteligente, e melhora seu desempenho com o passar do tempo.

2.2.1.3 Política de Gradiente

As políticas definem como os agentes escolhem as ações. Podem ser classificados em tipos fixos e não fixos. As políticas não estacionárias mudam ao longo do tempo e são úteis em contextos limitados onde a recompensa cumulativa que o agente procura otimizar é limitada a um intervalo de tempo finito. Políticas também podem ser categorizadas dado um segundo critério, sendo determinísticas ou estocásticas:

- Sendo determinística, a política é descrita por $\mu(s)$. Mapeando diretamente cada estado para uma ação específica.
- Sendo estocástica, a política é descrita por $\pi(a, s)$, onde descreve-se a distribuição de probabilidade de que a ação a possa ser escolhida no estado s .

2.2.1.4 Aprendizagem de política

Uma das categorizações de algoritmos de aprendizagem por reforço, refere-se à como diferem-se na forma como lidam com a coleta de dados e atualizações de políticas. Métodos em-política buscam avaliar e melhorar a política atualmente utilizada para tomar decisões, enquanto métodos fora-da-política avaliam e melhoram uma política diferente daquela usada para gerar dados. Dessa forma, um algoritmo em-política coleta dados usando a política atual que está sendo avaliada e aprimorada, tendo a política atualizada com base nas experiências coletadas utilizando a própria política. Por outro lado, algoritmos fora-da-política coletam dados usando uma política diferente daquela que está sendo avaliada e aprimorada. Determinando que as ações são escolhidas de acordo com uma política de exploração, enquanto a política-alvo é outra. A atualização da política é também com base na coleta de experiências, mas as ações são escolhidas de acordo com uma política de exploração diferente da política alvo.

2.2.1.5 Modelo de aprendizado Ator-Critico

Esse modelo de aprendizado, descrito no Algoritmo 1, é uma forma de representar a atualização de uma política através de redes neurais, podendo a política ser determinística ou estocástica. O agente é composto por duas redes neurais, uma responsável por estimar a política que levará ao maior acúmulo de recompensa (Ator) e outra responsável por avaliar a qualidade da política gerada (Crítico) [125]. A representação do modelo é ilustrada na Figura 3.

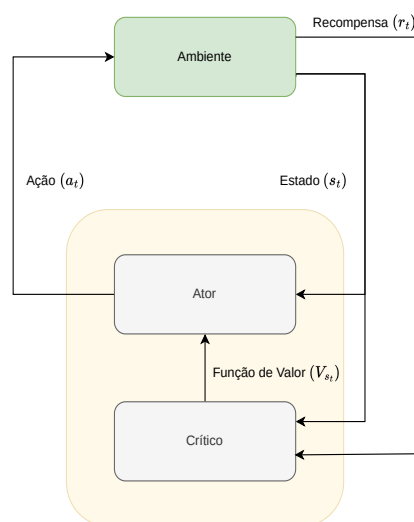


Figura 3: Modelo Ator-Crítico.

Fonte: O autor

O Ator é responsável por aprender uma política, ou seja, uma função que mapeia estados para ações. A política é representada pela função $\pi(a, s)$, onde a é a ação escolhida e s é o estado atual. O objetivo do Ator é aprender uma política que maximize a recompensa acumulada ao longo do tempo. Já o crítico é responsável por avaliar as ações tomadas pelo Ator em diferentes estados. Ele aprende uma função de valor que estima a recompensa esperada ao escolher uma ação específica em um determinado estado. O Crítico avalia as ações do Ator e fornece uma opinião sobre a qualidade dessas ações em relação ao objetivo final de maximizar a recompensa [125].

Algoritmo 1: Arquitetura Ator-Crítico

Inicializa os parâmetros: rede ator ϕ , rede crítica θ , memória de repetição M ,
número máximo de intervalos de tempo T

Inicializa os parâmetros das redes alvos $\theta' \leftarrow \theta$

para $t = 1$ **até** T **faça**

 Observa estado s e seleciona ação $a \sim \pi_\phi(\cdot|s)$

 Executa a no ambiente

 Observa próximo estado s' , e recompensa r

 Armazena (s, a, r, s') na memória M

se Atualiza rede RL **então**

 Seleciona aleatoriamente um *batch*, $B = \{(s, a, r, s')\}$ de M

 Computa a correção para o valor da ação Q:

$$y(r, s') = r + \gamma = Q_{\theta'}(s', a')$$

 Atualiza a rede crítica:

$$\nabla_{\theta} \frac{1}{|B|} \sum_{(s,a,r,s') \in B} (Q_{\theta}(s, a) - y(r, s'))^2$$

 Atualiza a política da rede ator:

$$\nabla_{\phi} \frac{1}{|B|} \sum_{s \in B} Q_{\theta_i}(s, a_{\phi})$$

 Atualiza as redes alvo:

$$\theta' \leftarrow \tau \theta + (1 - \tau) \theta$$

fim

fim

2.2.2 Aprendizado por Reforço Profundo (DRL)

A abordagem de DRL é uma adaptação do aprendizado por reforço voltado para problemas do mundo real. As possibilidades de estados, portanto, tendem ao infinito, trazendo a necessidade do uso de redes neurais para efetuar essa estimativa de estados. Assim, a abordagem é originada pelo algoritmo rede Q profunda - DQN [126] (do inglês,

Deep Q-Network), caracterizado pela implementação inicial do próprio *Q-learning*. A abordagem proposta no algoritmo foi eficaz em muitos jogos do Atari [126, 127]. E acabou atingindo um desempenho superhumano usando apenas uma entrada de imagem de pixel bruto para estimar a ação do agente. Porém, o DQN foi desenvolvido para lidar com espaços de ação discretos, tornando-se ineficaz para tarefas que exigem espaços de ação contínuos. Inclusive constatado por Peralta et al. [100] no contexto de planejamento de NBV. Assim, fazendo-se necessária a utilização de outros algoritmos para lidar com o contexto de espaço contínuo.

2.2.2.1 Algoritmo Ator-Crítico Suave (SAC)

Para descrever o algoritmo SAC é necessário previamente definir alguns pontos sobre o algoritmo DDPG, já que o SAC é uma derivação desse algoritmo especificamente. A estrutura do algoritmo DDPG é dada através de 4 redes, sendo duas delas Ator e Crítico e, respectivamente, duas redes de valor-alvo como cópias com atraso de tempo de suas redes originais que rastreiam suavemente as redes aprendidas. O Ator é responsável por mapear diretamente estados para ações através de uma política determinística $\mu(s|\theta^\mu)$, onde s representa o estado e θ^μ descreve os parâmetros da política. O Crítico representa a função de valor de ação $Q(s|\theta^Q)$, onde s representa o estado e θ^Q descreve os parâmetros do Crítico. O uso das redes de valor-alvo melhora muito a estabilidade no aprendizado, já que as equações de atualização das redes principais são dependentes dos valores calculados por elas mesmas tornando-as propensas a divergências. Também utiliza uma memória de repetição para amostrar a experiência e atualizar os parâmetros da rede neural. Com relação a exploração, considerando espaços de ação contínuos, é adicionando ruído à própria ação, podendo também ser utilizado no estado. Um dos métodos mais utilizados é o Processo Ornstein-Uhlenbeck [128] para adicionar ruído à saída, onde é correlacionado com o ruído anterior de modo a evitar que o mesmo cancele a dinâmica geral criada pelo Ator [52].

O algoritmo SAC baseia-se em uma abordagem Ator-Crítico estocástica, sendo considerado um método de fora-da-política. E que tem como formato de aprendizado do agente, a utilização de políticas passadas [53]. A principal ideia do SAC é usar funções de aproximação para aprender políticas de entropia máxima. Para isso, utiliza três redes neurais, uma função de valor Q , uma política estocástica e uma função de valor do estado. Com isso, otimiza iterativamente os objetivos definidos pelas equações de atualização. Assim, mantém uma política estocástica $\pi(a|s, \theta^\pi)$, que gera uma distribuição de probabilidade sobre as ações a . É introduzido também uma estimativa adicional do valor do estado $V(s|\theta^V)$, que é utilizado para normalizar a função Q durante as atualizações. Uma característica distintiva é a adoção da entropia máxima, uma medida da incerteza na política do agente, que é adicionada ao objetivo de otimização. A atualização da função Q ocorre através do ajuste da rede para minimizar a discrepância entre o valor previsto e o re-

torno esperado. A política estocástica também é atualizada para maximizar a combinação de recompensas esperadas e a entropia, incentivando a exploração eficiente. De maneira que, a entropia é ajustada automaticamente ao longo do treinamento para manter um nível desejado de exploração [53].

Além de visar a maximização das recompensas do sistema, o SAC também procura aumentar a entropia da política. Entropia refere-se à imprevisibilidade de uma variável, indicando maior incerteza quanto maior for a entropia. Quando uma variável aleatória possui apenas um valor, a estratégia possui entropia zero, incentivando, assim, a exploração por parte do agente. Dada a descrição de Schaul et al. [129], utiliza-se uma memória de repetição para treinar e avaliar a função de política gerada pela rede do ator. E a função de valor V e a função Q são geradas pela rede crítica. Essa memória de repetição permite salvar as experiências do agente durante o período de treinamento, permitindo que sejam utilizadas para aproximar a função Q mais qualificada do problema. Essas experiências, incluindo estados, ações, recompensas e novos estados, são armazenadas em um *buffer* de memória e amostradas aleatoriamente para o aprendizado do algoritmo. Desse modo, favorece a interrupção das correlações temporais entre episódios diferentes na fase de treinamento, mesmo com memórias curtas, e é perceptível um aumento significativo no desempenho dos algoritmos DRL fora-da-política. Independentemente do aumento nas capacidades da aplicação com DRL, o uso de memórias de repetição pode afetar o tempo de aprendizagem de um agente.

Conceitos como rede alvo e memória recorrente são essenciais para implementar algoritmos de DRL na prática. A rede alvo [130], que é uma cópia constantemente atualizada da rede base através da técnica de *Polyak-averaging* [131] para atualizar seus parâmetros. Busca estabilizar o treinamento, enquanto a memória recorrente armazena experiências passadas para promover a diversificação do conjunto de dados de treinamento e evitar correlações temporais perigosas. Esses elementos trabalham sinergicamente para aumentar a eficiência e a estabilidade da aprendizagem por reforço em ambientes complexos e dinâmicos.

2.2.3 Aprendizado Contrastivo (CL)

Nos últimos anos, diversas abordagens de aprendizado auto-supervisionado têm sido apresentadas para adquirir representações de imagens, cada uma superando suas antecessoras. Contudo, o desempenho desses métodos ainda permanece abaixo das abordagens supervisionadas. Essa dinâmica mudou com a introdução de uma nova estrutura proposta por Chen et al. [25], com a apresentação do modelo de uma estrutura simples para aprendizagem contrastiva de representações visuais - SimCLR (do inglês, *A Simple Framework for Contrastive Learning of Visual Representations*). A implantação do SimCLR [25] não apenas buscou aprimorar os métodos auto-supervisionados de última geração, mas também ultrapassou abordagens supervisionadas na classificação *ImageNet* [132] ao ex-

pandir a arquitetura utilizada.

O aprendizado contrastivo é uma abordagem que visa treinar modelos para capturar características discriminativas em dados. Essa técnica baseia-se no princípio de que instâncias semelhantes devem ser agrupadas próximas umas das outras no espaço de representação, enquanto instâncias diferentes devem ser afastadas. O objetivo principal é maximizar a similaridade entre amostras positivas e minimizar a similaridade entre amostras negativas. O método proposto por Chen et al. [25], ilustrado na Figura 4, consiste em uma rede de codificadores base $f(\cdot)$ e um *projection head* $g(\cdot)$, que busca criar uma representação mais rica e discriminativa dos dados, sendo treinados para maximizar a concordância usando uma perda contrastiva. Após a conclusão do treinamento, o *projection head* $g(\cdot)$ é descartado. Com isso, o codificador $f(\cdot)$ e a representação h , gerada pelo codificador, são os únicos utilizados para tarefas posteriores (inferência).

A execução do método é dada a partir de uma imagem de entrada x que é passada por um módulo de aumento de dados estocásticos $\mathcal{T}$, que transforma x aleatoriamente, resultando em duas visualizações correlacionadas $\tilde{x}_i$ e $\tilde{x}_j$. Essas transformações passam por um determinado *encoder* $f(\cdot)$, que extrai vetores de representação dos dados transformados, gerando duas representações h_i e h_j . Cada uma das representações, h_i e h_j , agora passam por um *projection head encoder* $g(\cdot)$ gerando duas representações não-lineares das transformações z_i e z_j .

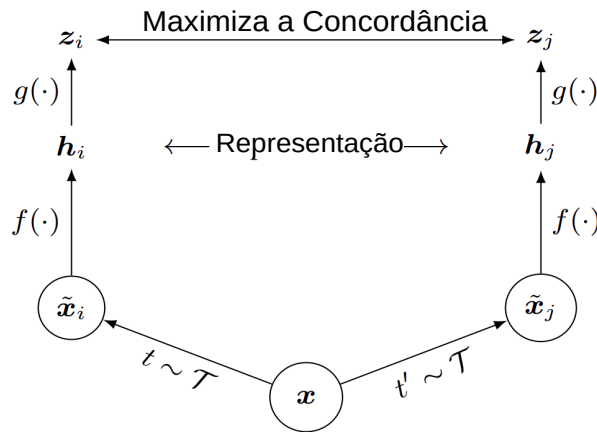


Figura 4: Método SimCLR.

Fonte: Chen et al. [25]

Sabendo que essa implementação é para cada uma das entradas, se tem que para N imagens é gerado $2N$ representações denotadas por i e j . A comparação dessas representações é feita escolhendo um pivô e esse pivô é comparado com todas as representações da outra componente, o pivô então é trocado e segue-se com a comparação. O método de comparação, como já dito, é através da similaridade entre os elementos. Essa similaridade é calculada através da similaridade dos cossenos, uma medida da similaridade

dade entre dois vetores num espaço vetorial que avalia o valor do cosseno do ângulo compreendido entre eles. Tendo a fórmula descrita como:

$$s(u, v) = \frac{u^\top v}{\|u\| \cdot \|v\|}, \quad (10)$$

onde seguindo a equação da similaridade, u e v passam a ser $\bar{z}_i$ e $\bar{z}_j$ na descrição do método. Com a avaliação da similaridade do cosseno [133], o método descreve duas funções de perda. A primeira função de perda ℓ denominada NT-Xent [134], descreve a comparação dos elementos com a dinâmica de pivôs. Já a função de perda final que atualiza $f(\cdot)$ e $g(\cdot)$, é composta por duas funções $\ell(i, j)$ e $\ell(j, i)$ que simultaneamente fazem a comparação cruzada de ramificação:

$$\ell(i, j) = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \{0, 1\}_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)}, \quad (11)$$

onde $\{0, 1\}_{[k \neq i]}$ impede a comparação com o mesmo elemento i , já que quando $k = i$ o resultado da comparação é zerado, e τ é dado como um parâmetro de temperatura. Assim, o método é então descrito pelo Algoritmo 2.

Algoritmo 2: SimCLR

Entradas: tamanho do *batch* N , constante τ , os *encoders* f e g e as transformações $\mathcal{T}$

para cada imagem $\{x_k\}_{k=1}^N$ **faça**

para todo $k \in \{1, \dots, N\}$ **faça**

 Defina duas funções de transformação $t \sim \mathcal{T}$ e $t' \sim \mathcal{T}$

 Primeira transformação

$\tilde{x}_{2k-1} = t(x_k)$

$h_{2k-1} = f(\tilde{x}_{2k-1})$ representação

$z_{2k-1} = g(h_{2k-1})$ projeção

 Segunda transformação

$\tilde{x}_{2k} = t'(x_k)$

$h_{2k} = f(\tilde{x}_{2k})$ representação

$z_{2k} = g(h_{2k})$ projeção

fim

para todo $i \in \{1, \dots, 2N\}$ e $j \in \{1, \dots, 2N\}$ **faça**

$s(i, j) = \frac{z_i^\top z_j}{\|z_i\| \cdot \|z_j\|}$ Similaridade

fim

Defina $\ell(i, j)$ **como** $\ell(i, j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \{0, 1\}_{[k \neq i]} \exp(s_{i,k}/\tau)}$

$\mathcal{L} = \frac{1}{2N} \sum_{k=1}^N [\ell(2k-1, k) + \ell(2k, 2k-1)]$

 Atualize f e g para minimizar $\mathcal{L}$

fim

retorna f e g

Com a possibilidade de utilizar diferentes fontes de informação, surge também a capacidade de utilizar diferentes tipos de dados. Esse escopo do aprendizado contrastivo é denominado de *cross-modal* [26]. O principal contexto dessa abordagem é a extração de diferentes características de um mesmo elemento. Esse método é recentemente utilizado em métodos que geram imagens a partir de texto, treinando o *encoder* contrastivo com a informação de texto e imagem aumentando a similaridade entre as informações [30, 33, 34, 38]. As abordagens propostas trazem resultados superiores a proposta anteriores do mesmo seguimento.

A exploração do aprendizado das características heterogêneas em tipos diferentes de dados ainda vai além, buscando extrair informações de diferentes fontes para determinar uma outra informação. Existem abordagens no contexto de análise de emoções, onde Mai et al. [135] propõe com uma entrada de texto, áudio e vídeo (*frames* de imagem) tem como saída a representação de uma determinada emoção em formato de texto. Outra dessas abordagens que exploram bastante a heterogeneidade dos dados é no contexto de aprendizagem semântica. Wu et al. [31] aborda o aprendizado de semântica em imagens e nuvens de pontos através da relação de texto semântico, tendo imagens, nuvens de pontos e texto classificador como entradas para o treinamento e gerando na saída uma imagem ou uma PCD segmentada. A inferência é orientada ao tipo de dado a ser segmentado.

A abordagem de combinação de imagens com nuvens de pontos para efetuar tarefas de detecção, classificação e segmentação de elementos da cena, também se destaca. A ideia dessas propostas é entender as características dos elementos cruzando a entrada de imagens 2D com nuvens de pontos 3D [32, 36, 35], e também 2D [37]. A função de perda total é o elemento chave para esses métodos, visto que, é nela que avaliação das características é feita de modo cruzado. A equação de perda final é o que caracteriza a composição do método, já que é nela que é computada a similaridade de toda a rede, tornando inviável descrever um modelo genérico que represente essas unificações.

2.2.4 Representações Contrastivas Não-Supervisionadas para Aprendizagem por Reforço (CURL)

Os algoritmos de DRL trazem consigo certas limitações para lidar com grande quantidade de dados no espaço de observação, o que remete em um espaço latente muito grande. Esse espaço latente captura características relevantes e abstratas dos dados de entrada, permitindo que o modelo aprenda representações mais significativas e compactas [136]. Referindo-se a uma representação intermediária e de dimensionalidade reduzida que é aprendida automaticamente pelos modelos durante o treinamento.

Na utilização de imagens como um componente da observação, diversas formas são descritas para lidar com o problema de espaço latente através da manipulação dos dados da observação. A avaliação do estado da arte feita por Uzkent et al. [28] e Yarats et al. [29], ilustram abordagens que em sua totalidade buscam lidar com a imagem antes de

ela fazer parte da observação. Em todas as abordagens avaliadas, inclusive nas propostas por Uz Kent et al. [28] e Yarats et al. [29], a estratégia se resume a diminuir o tamanho da imagem tentando preservar e/ou extrair suas principais características. Essa é uma abordagem válida, porém leva a uma perda de informação que pode ser significativa para o aprendizado.

O CURL [27] se coloca como uma contra-proposta a estratégias como as de Uz Kent et al. [28] e Yarats et al. [29]. Descrita como uma estratégia para combinar o aprendizado contrastivo com a aprendizagem por reforço, podendo ser utilizada com qualquer algoritmo de DRL. Seja na abordagem em-política ou fora-da-política. O algoritmo SAC, descrito anteriormente, é o método utilizado por Laskin et al. [27] para validar a abordagem. O CURL, descrito no Algoritmo 3, é apresentado como uma técnica eficiente para melhorar o desempenho de amostragem, realizando simultaneamente aprendizagem contrastiva e aprendizagem por reforço fora-da-política.

Ao contrário dos métodos de DRL baseados em pixel, o CURL [27] é capaz de aprender representações ricas de dados dimensionais através de aprendizagem não supervisionada contrastiva. Assim, tem como principal objetivo garantir que uma *query* (q), corresponda a uma *key* (k), usando pares positivos de dados modificados a partir da mesma observação $o(t)$. Essa abordagem segue o que foi apresentado em Chen et al. [25], tendo os elementos z_i e z_j representados por q e k . A representação da abordagem é ilustrada pela Figura 5, mostrando como é treinado um *encoder* visual para combinar as incorporações das versões modificadas dos dados o_q e o_k usando uma função de perda contrastiva do trabalho.

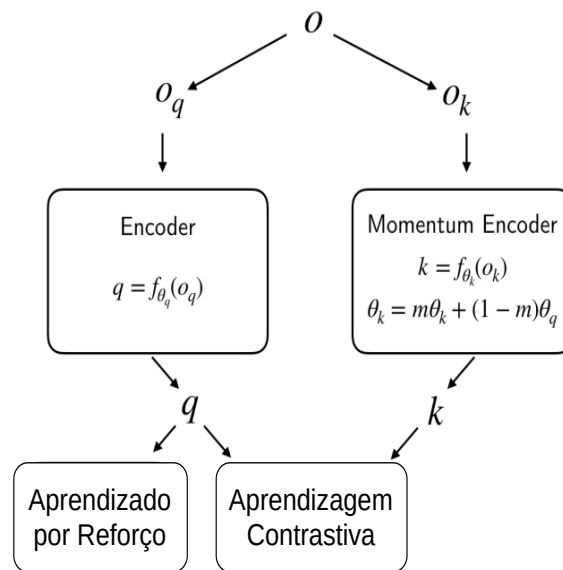


Figura 5: Método CURL.

Fonte: Laskin et al. [27]

A estrutura da abordagem descreve que o tratamento das observações da *query* o_q é dado pelo *encoder* f_{θ_q} , retornando $q = f_{\theta_q}(o_q)$. A codificação das *keys* se dá a partir do *encoder* f_{θ_k} , retornando $k = f_{\theta_k}(o_k)$. Dessa maneira, as observações são construídas a partir do *batch* (i) amostrado da memória de repetição para a atualização do DRL. A versão *Polyak-averaging* do *encoder query* é usada para atualizar θ_k , sendo $\theta_k \leftarrow \tau\theta_q + (1 - \tau)\theta_k$. A política do DRL e a função de valor são construídos no topo do codificador da *query*. Esse codificador é treinado em conjunto com os objetivos do aprendizado por reforço e aprendizagem contrastiva, aprendendo a extrair características específicas da aplicação. A equação que representa a função de perda é denominada InfoNCE [137]:

$$\mathcal{Loss}_q = \log \frac{\exp(q^\top W k_+)}{\exp(q^\top W k_+) + \sum_{i=0}^{K-1} \exp(q^\top W k_i)}, \quad (12)$$

onde W é uma matriz de parâmetros aprendidos, k_+ remete a saída positiva do *encoder* k da primeira observação recolhida do *batch* i da memória de repetição, o mesmo ocorre para k_i .

Algoritmo 3: Representações Contrastivas Não-supervisionadas para Aprendizado por Reforço (CURL)

Entradas: Parâmetros iniciais da rede ator ϕ , parâmetros da rede crítica θ_1, θ_2 , memória de repetição M , número máximo de intervalos de tempo T

Define os parâmetros das redes alvos $\theta'_1 \leftarrow \theta_1, \theta'_2 \leftarrow \theta_2$

para $t = 1$ **até** T **faça**

 Observa o estado atual $I = I_t$

 Imagem é passada pela *query* $s = f_{\theta_q}(I)$

 Observa estado s e seleciona ação $a \sim \pi_\phi(\cdot|s)$ e executa a no ambiente

 Observa próximo estado s' , e recompensa r

 Armazena (s, a, r, s', I) na memória M

se *Atualiza rede RL* **então**

 Seleciona aleatoriamente um *batch*, $B = \{(s, a, r, s')\}$ de M

 Computa a correção para o valor da ação Q :

$$y(r, s') = r + \gamma \left(\min_{i=1,2} Q_{\theta'_i}(s', a') - \alpha \log \pi_\phi(a'|s') \right)$$

 Atualiza a rede crítica:

$$\nabla_{\theta_{i=1,2}} \frac{1}{|B|} \sum_{(s,a,r,s') \in B} (Q_{\theta_i}(s, a) - y(r, s'))^2$$

 Atualiza a politica da rede ator:

$$\nabla_\phi \frac{1}{|B|} \sum_{s \in B} \left(\min_{i=1,2} Q_{\theta_i}(s, a_\phi) - \alpha \log \pi_\phi(a_\phi|s) \right)$$

 Atualiza as redes alvo:

$$\theta'_{i=1,2} \leftarrow \tau \theta'_i + (1 - \tau) \theta_i$$

fim

se *Atualiza rede contrastiva* **então**

 Seleciona aleatoriamente um *batch* de imagem I :

$$I_k \leftarrow I_q$$

$$I_q \leftarrow \text{corte aleatório } (100 \times 100 \rightarrow 84 \times 84)$$

$$I_k \leftarrow \text{corte aleatório } (100 \times 100 \rightarrow 84 \times 84)$$

$$B(I) = \{(s, a, r, s', I_q, I_k)\} \text{ de } M$$

 Computa o espaço latente *query* e *key*: $q = f_{\theta_q}(I_q)$ e $k = f_{\theta_k}(I_k)$

 Atualiza a *query*:

$$\mathcal{L}_{oss} = \log \frac{\exp(q^T k_0)}{\sum_{i \in B(d)} \exp(q^T k_i)}$$

 Atualiza a *key*:

$$f_{\theta_k} \leftarrow \tau f_{\theta_k} + (1 - \tau) f_{\theta_q}$$

fim

fim

3 DESCRIÇÃO DO AMBIENTE EXPERIMENTAL

Nesse Capítulo busca-se descrever alguns pontos importantes referentes as tecnologias utilizadas para o desenvolvimento do ambiente de simulação foto-realístico e, consequentemente, as características que foram assumidas referente à movimentação multi-domínio utilizando um HUAUV.

3.1 Simulação

Para o desenvolvimento do trabalho, viu-se a necessidade do uso de ambientes foto-realísticos para a diminuição da diferença entre a informação sintética do ambiente simulado e o mundo real [42]. Dessa forma, levando a procura de simuladores que utilizassem motores gráficos de alta qualidade que proporcionassem efeitos realísticos, principalmente com relação aos efeitos de câmera se tratando de ambientes externos e ambientes sub-aquáticos. Assim, avaliando o estado da arte em simuladores para veículos aéreos não-tripulados - UAVs [138] e veículos não-tripulados subaquáticos - UUVs [139], considerando o conceito foto-realístico do ambiente, os sensores disponíveis e a movimentação customizada em multi-domínio com o mesmo veículo, concluiu-se que os simuladores baseados na *Unreal Engine 4* (UE4), trazem maior contribuição para a problemática do trabalho, tendo em vista que a informação visual é afetada pelos efeitos de luminosidade e turbidez.

A UE4 [140] foi desenvolvida, e mantida, pela desenvolvedora de jogos *Epic Games* [141], sendo responsável pelo desenvolvimento de diversos jogos famosos no mercado atual. A qualidade foto-realística apresentada nos ambientes vem trazendo outros contextos para suas utilidades, além dos convencionais no cenário de jogos. Sua própria desenvolvedora nomeia de “modo sério” a utilização voltada ao gênero simulação. Para esse gênero a qualidade visual do cenário é algo de suma importância, o que é facilmente entregue pela UE4, como é mostrado na Figura 6. Diversos jogos já desenvolvidos se encaixam nesse gênero, sendo o futebol um dos mais famosos em suas diferentes versões. Além disso, o uso de simulação para veículos reais vem crescendo muito nos últimos anos, tendo implicações como simuladores de voo e de direção onde a similaridade com

um ambiente real é de altíssima importância. Dessa forma, encorajando o desenvolvimento de simuladores como o *AirSim* proposto pela *Microsoft*.



Figura 6: Motor gráfico *Unreal Engine*

Fonte: <https://bit.ly/3R0BfU4>

3.1.1 AirSim

O *AirSim* é um *plugin* desenvolvido para explorar a altíssima qualidade dos ambientes que a UE4 é capaz de gerar, buscando desenvolver sistemas de autonomia para UAVs e carros autônomos em ambientes foto-realísticos. Ele fornece uma interface para definir um veículo que pode ter um número arbitrário de sensores, considerando uma movimentação independente entre eles e também entre o veículo, além de um modo customizado que permite a navegação de um veículo genérico com 6-DoF independente de uma dinâmica de movimentação específica. Esse modo é denominado “*Computer Vision*”, sendo voltado para adaptação de novos veículos e coleta de dados desconsiderando restrições dinâmicas Shah et al. [142].

O modo “*Computer Vision*”, inspirado no *plugin UnrealCV* [143], se baseia na liberdade total de movimento de um observador com 6-DoF, funcionando sem atuação de forças externas ou qualquer modelo dinâmico. Esse modo foi criado com o intuito de coletar uma grande diversidade de informações, como imagens RGB, imagens semanticamente segmentadas, profundidade, normais e etc. Dessa forma, é possível assumir-se um veículo genérico que seja capaz de efetuar essa movimentação. Considerando que de maneira geral alguns modelos de veículos dentro do contexto de UAVs, AUVs e HUAUVs podem efetuar uma movimentação com 6-DoF [22, 50], além das interpretações com relação ao movimento feitas por Grando et al. [22] e de Jesus et al. [24], assumiu-

se que para o contexto abordado no presente trabalho esse seria o melhor modo à ser utilizado.

3.1.2 Estruturas Alvo

Para compor os ambientes, se utilizou estruturas do contexto da indústria de óleo e gás que compõem um ambiente *offshore*. Assim, foram criados cinco diferentes mundos. O primeiro mundo é constituído por uma primitiva geométrica hexagonal flutuante, apresentada na Figura 7, com o intuito de ser um ambiente mais simples e genérico. O segundo e o terceiro mundo são constituídos por uma torre eólica de modelos mono-estaca e semi-submersível, apresentados nas Figuras 8 e 9. O terceiro mundo é composto por uma jaqueta, nome dado à esse modelo que serve de base para torres eólicas, apresentado na Figura 10. E por fim, uma plataforma de petróleo semi-submersível, apresentada na Figura 11.

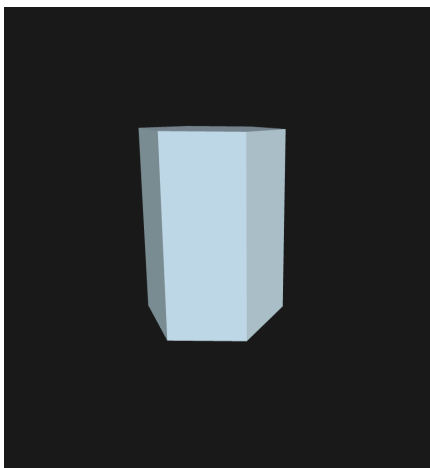


Figura 7: Estrutura hexagonal genérica.

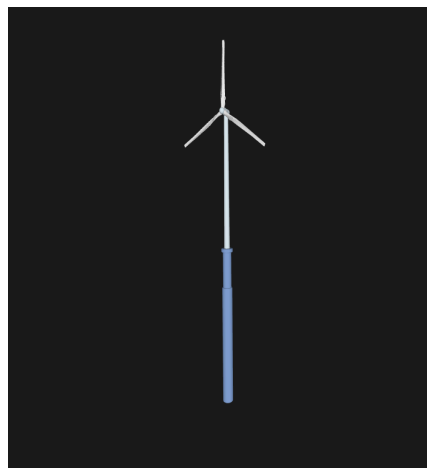


Figura 8: Eólica mono-estaca



Figura 9: Eólica semi-submersível.

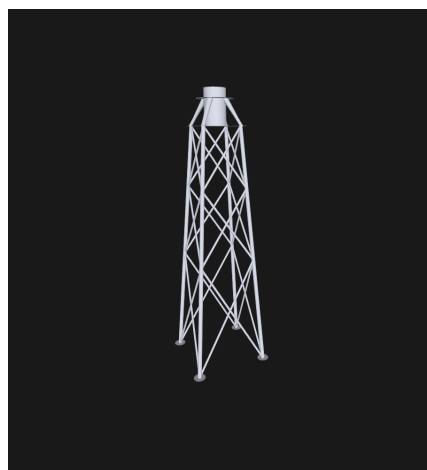


Figura 10: *Jacket*.

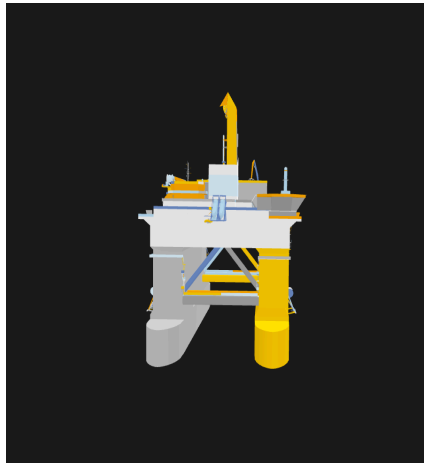


Figura 11: Modelo de Plataforma Semi-submersível.

Fonte: Denoyel et al. [144]

3.2 Ambiente de Atuação

Para a implementação de um agente baseado em DRL é necessária a construção de um ambiente, o qual irá prover a recompensa que guiará o aprendizado. Dessa forma, uma abordagem comum é a utilização dos recursos disponíveis no *OpenAI Gym* [145], pois com ele é possível codificar um ambiente em um formato que otimiza a implementação. Assim, se utilizou desse recurso para criação de ambientes customizados que irão representar a codificação de cada estrutura simulada na UE4.

3.2.1 Ambiente *OpenAI Gym*

Para a construção dos ambientes o primeiro passo foi definir a avaliação de cobertura de cada estrutura, pois essa avaliação é o elemento chave para a construção da função de recompensa. Dessa maneira, viu-se a possibilidade de utilizar marcadores dispostos sobre a superfície de cada estrutura como métrica, utilizando-os como referência visual para determinar a cobertura. E pensando na avaliação do agente de maneira individual entre os domínios, além do multi-domínio, foram utilizados diferentes marcadores dentro e fora da água. Ocasionalmente em um número distinto de marcadores em cada domínio para cada estrutura.

Para detectar os marcadores, buscou-se na imagem a existência de cada marcador, utilizando recurso semântico, e através do algoritmo BSP [119], um recurso do *Air-Sim*, obtendo-se a referência de cada marcador com relação ao observador. Com essa referência, buscando diferencia-los, é feita a comparação da referência com a posição global de cada marcador. Assim, é possível determinar se cada marcador foi visualizado individualmente. Portanto é possível determinar se todas as faces da estrutura foram visualizadas através da contagem desses marcadores.

A vantagem de utilizar marcadores para avaliação de cobertura é que eles não precisam ser explícitos, podendo-se utilizar características da própria estrutura para isso, pensando em uma generalização para redução da diferença com aplicações reais. Portanto, os marcadores foram ocultados com relação ao agente, não gerando uma tendência direta na política aprendida. As Figuras 12 e 13 ilustram a representação dos marcadores e a vista do agente com relação à estrutura.

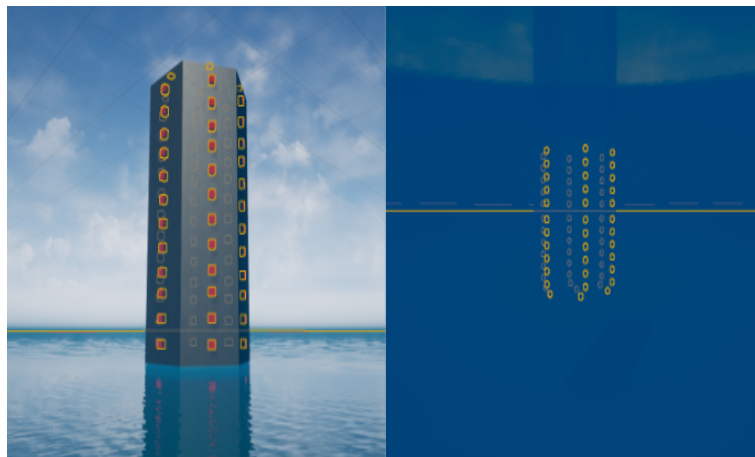


Figura 12: Vista da estrutura com marcadores.

Fonte: O autor

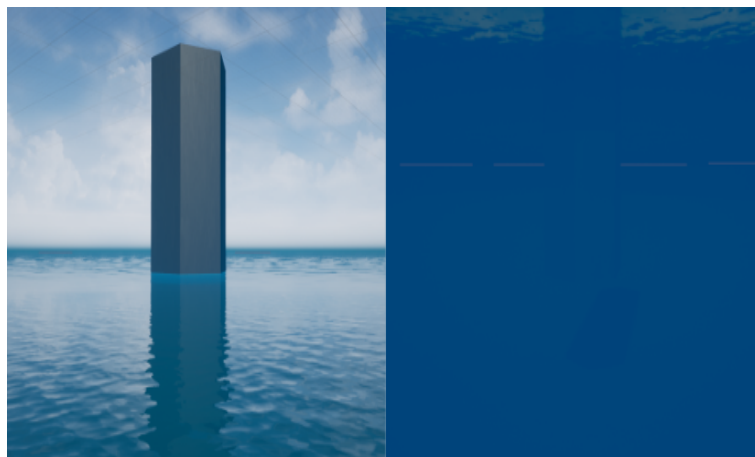


Figura 13: Vista do agente.

Fonte: O autor

4 METODOLOGIA

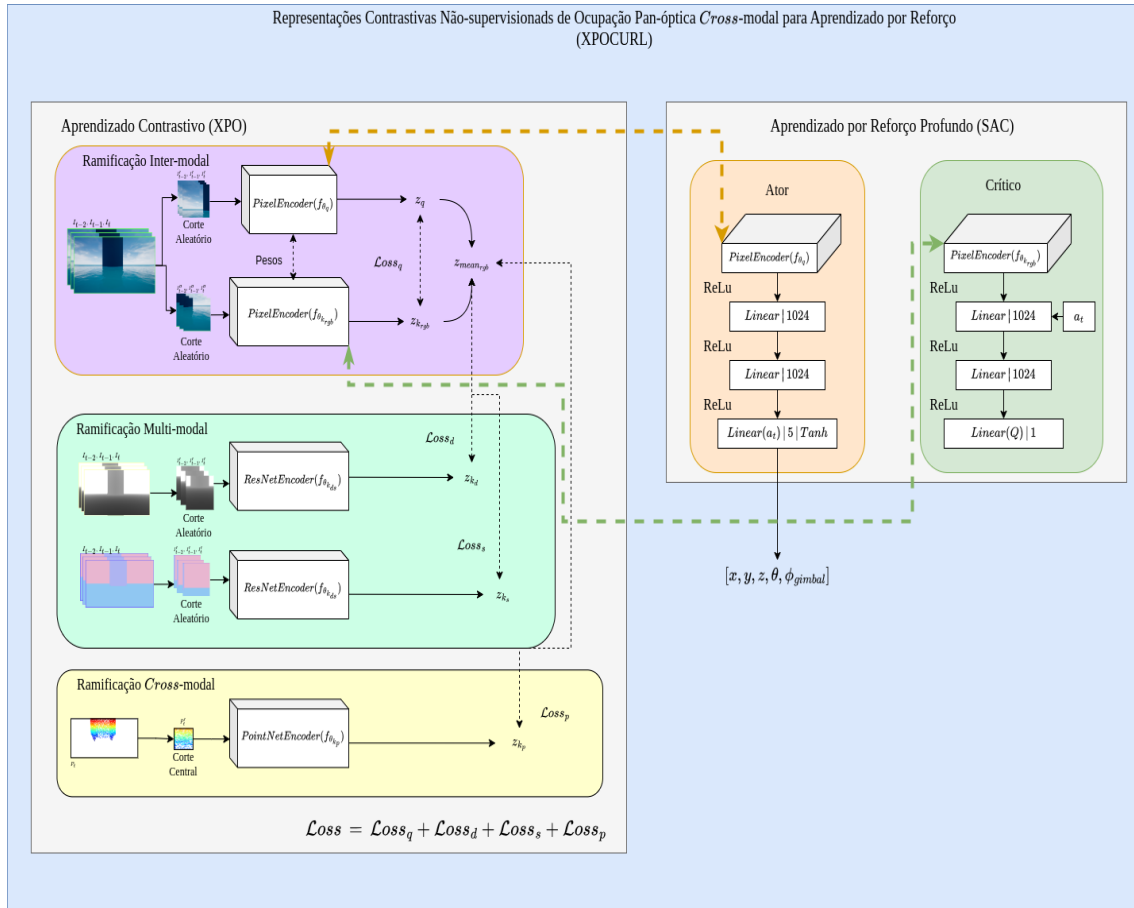


Figura 14: Diagrama do método proposto: É apresentada a relação dos dados de observação através do bloco contrastivo, sendo ele definido por três ramificações: inter-modal, multi-modal e *cross*-modal. Nesse bloco são geradas quatro funções de perda, uma para cada tipo de dado, sendo eles: imagem RGB, profundidade, segmentação semântica e nuvem de pontos. A soma dessas funções compõem a perda contrastiva que irá tornar a *queue* (*PixelEncoder*) capaz de utilizar todas essas informações para otimizar a compreensão das imagens RGB. É também é apresentada a composição das redes ator e crítico, ilustrando as ações geradas pelo agente. Essas ações referem-se à movimentação, sendo composta pelo movimento linear nos eixos x , y e z , além do movimento angular θ em torno do eixo z e o movimento angular ϕ_{gimbal} unicamente no corpo do sensor, em torno do eixo y .

No presente capítulo, será apresentada a descrição da abordagem de um planejador de NBV multi-domínio, representado pela Figura 14. Para isso, considerando a base de DRL, será descrito o detalhamento da construção dos conceitos e definições para a implementação do agente. Descrevendo também a composição de inferência e de representação 3D.

4.1 Planejador de NBV Multi-domínio

A ideia por trás da abordagem proposta é entender as características da estrutura, e do ambiente de forma geral, através de uma perspectiva panóptica e de ocupação do ambiente multi-domínio, buscando através da relação de imagens RGB, profundidade e segmentação semântica, entender a perspectiva panóptica. E através da nuvem de pontos da estrutura, busca-se generalizar um modelo capaz de entender as características tridimensionais de ocupação dessa estrutura.

Para isso é desenvolvida uma representação contrastiva que relaciona as informações panópticas e de ocupação da cena, aprendendo o espaço latente junto ao algoritmo SAC [53]. Para desenvolver o método proposto, foi analisado o modelo contrastivo apresentado por Afham et al. [32] e Higa et al. [36], levando à configuração ilustrada na Figura 16.

4.1.1 Codificação do Ambiente

4.1.1.1 Espaço de Ação

Com a estrutura de avaliação do ambiente definida, determinou-se então o comportamento do veículo e consequentemente o espaço de ação. Para isso, assumiu-se as considerações de movimentação [22, 24] e as características necessárias do observador [15]. Sabendo que o sistema de coordenadas do simulador é no formato NED (do inglês, *North, East, Down*), a caracterização das ações deu-se por 4-DoF sobre o veículo, sendo eles posições em termos dos eixos x , y e z e também orientação, com ângulos de *Euler* [146], em torno do eixo z (θ), além de 1-DoF sobre a câmera, funcionando como um *gimbal* para alterar a orientação em termos do eixo y (ϕ).

A composição do espaço de ação é baseada no mapeamento da saída do agente para um intervalo de posições e orientações relativas ao veículo. Esse mapeamento ocorre com base no princípio das vistas válidas, considerando que um próximo NBV precisa ser interseccionado ao campo de vista atual, como é ilustrado pela Figura 15. Para isso, calculou-se o ponto de intersecção máximo entre as projeções de duas vistas na vertical e horizontal baseado no conceito de geometria *epipolar* [147]. Esse método é muito utilizado na visão computacional para alinhamento entre pares estéreo, onde busca-se através de um ponto no espaço, avistado por ambas as vistas, encontrar a linha *epipolar* gerada através da intersecção mais próxima entre as vistas no plano da imagem [147].

Dessa forma, chegou-se nos intervalos entre as vistas, em metros, que melhor se adequaram para o contexto do problema. Para isso, considerou-se o veículo à uma determinada distância da estrutura colocando-se pouco acima do nível da água, como se estivesse suspenso em alguma base, com um ângulo θ no qual a frente, orientada no eixo x do corpo do veículo que é a direção e sentido da câmera, direcionado ao centro da estrutura. Percebeu-se que para as posições nos eixos x , y e z o intervalo que melhor se adequou foi de $\pm 40m$ e para as orientações, θ e ϕ_{gimbal} , foi de $\pm \pi/4 rad$. Assim, para a definição da postura, utilizou-se a posição juntamente às orientações, do veículo e da câmera separadamente, no formato de *quaternions* [148]. Esse elemento traz uma abordagem que lida com números complexos, sendo ele definido por quatro elementos q_w como parte real e q_x , q_y e q_z como parte imaginária, representando a orientação em um espaço de quatro dimensões. O objetivo é evitar o problema denominado de *Gimbal Lock*, ocasionado pelo alinhamento de dois ângulos, que ocorre na abordagem pelo método de *Euler* [146].

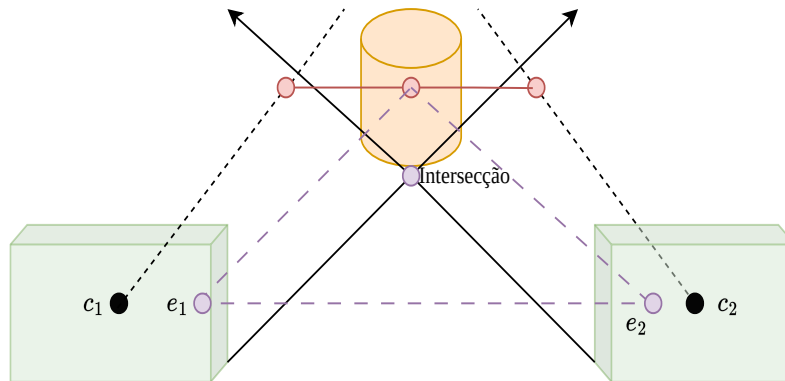


Figura 15: Avaliação de intersecção para estimativa de vistas.

4.1.1.2 Espaço de Observação

A observação, seguindo as características do CURL [27], é definida como uma lista contendo a representação temporal das últimas três vistas para cada tipo de dado. As informações visuais de imagens RGB, profundidade e semântica são transformadas para uma escala de 100×100 seguindo ainda conceitos do CURL. Para a nuvem de pontos, baseou-se em uma relação média de custo computacional e eficiência dos resultados com relação às dimensões, portanto assumiu-se que uma transformação, baseada no centro da vista, reduzindo a quantidade de pontos para 256 seria suficiente para extração de características relevantes.

4.1.1.3 Função de Recompensa

Para o cálculo da recompensa, descrita na Equação 13, foram feitas algumas considerações:

- O cálculo da recompensa baseia-se no número de marcadores alocados em cada domínio e no número de vistas necessárias para a cobertura completa.
- A não detecção de qualquer marcador implica em uma penalidade máxima e a posição do agente é reiniciada.
- A detecção de pelo menos um marcador fora de uma distância segura, definida arbitrariamente com relação à cada estrutura juntamente com as altitudes referentes a cada domínio, implica em uma redução da recompensa obtida e a posição do agente é reiniciada.

Dadas as considerações, buscou-se relacionar a recompensa com o número de marcadores e o número de vistas. Dessa forma, o principal objetivo é que o agente visualize mais cedo a maior quantidade possível de marcadores, diminuindo a incerteza ao longo das vistas planejadas. Para a construção do caminho, procura-se efetuar gradativamente a visualização dos marcadores. Portanto, um arranjo de vistas com maior aleatoriedade na visualização de novos marcadores, tende a ter uma recompensa menor que um arranjo uniforme de mesmo tamanho que reduz a incerteza ao longo do tempo, considerando que ambos não sofram penalidades.

Para o cálculo da recompensa é utilizado o número original de marcadores (o), o número faltante de marcadores a serem visualizados (xs), onde $xs = o$ inicialmente e sendo decrementado a cada novo marcador visto (nvm), o número passado de marcadores faltantes (pxs) e o passo de treino atual (cs), considerando uma distância segura (sv), que é diretamente atrelado ao número de vistas. Além disso, para a penalidade é utilizado o número máximo de passos ($maxs$), ou vistas, que compõem um episódio.

$$\begin{aligned}
 maxs &= 200 \\
 \alpha &= 1 \exp -1 \\
 \beta &= 1.5 \cdot (o/maxs) \\
 r_{prev} &= \begin{cases} (o - xs)/cs & \text{se } cs > 0, \\ (o - xs)/(cs + 1) & \text{se } cs = 0. \end{cases} \\
 r(s_t, a_t) &= \begin{cases} r_{prev} & \text{se } xs < pxs \\ \alpha \cdot r_{prev} & \text{se } xs < pxs \wedge \neg sv \\ 0 & \text{se } xs = pxs \\ -\beta & \text{se } nvm = 0 \end{cases} \quad (13)
 \end{aligned}$$

4.1.2 Representação Contrastiva *Cross*-modal de Multi-domínio

O desenvolvimento do método é dado a partir de três ramificações, como é ilustrado na Figura 16. A primeira ramificação é responsável pelo aprendizado inter-modal das representações, q e k_{rgb} , entre os *encoders* f_{θ_q} e $f_{\theta_{k_{rgb}}}$, que fazem parte do modelo original de aprendizado proposto no CURL [27]. Na segunda ramificação, ficam os componentes multi-modais de profundidade (k_d) e segmentação semântica (k_s). E por fim, na última ramificação, o componente de ocupação (k_p) com a informação da nuvem de pontos.

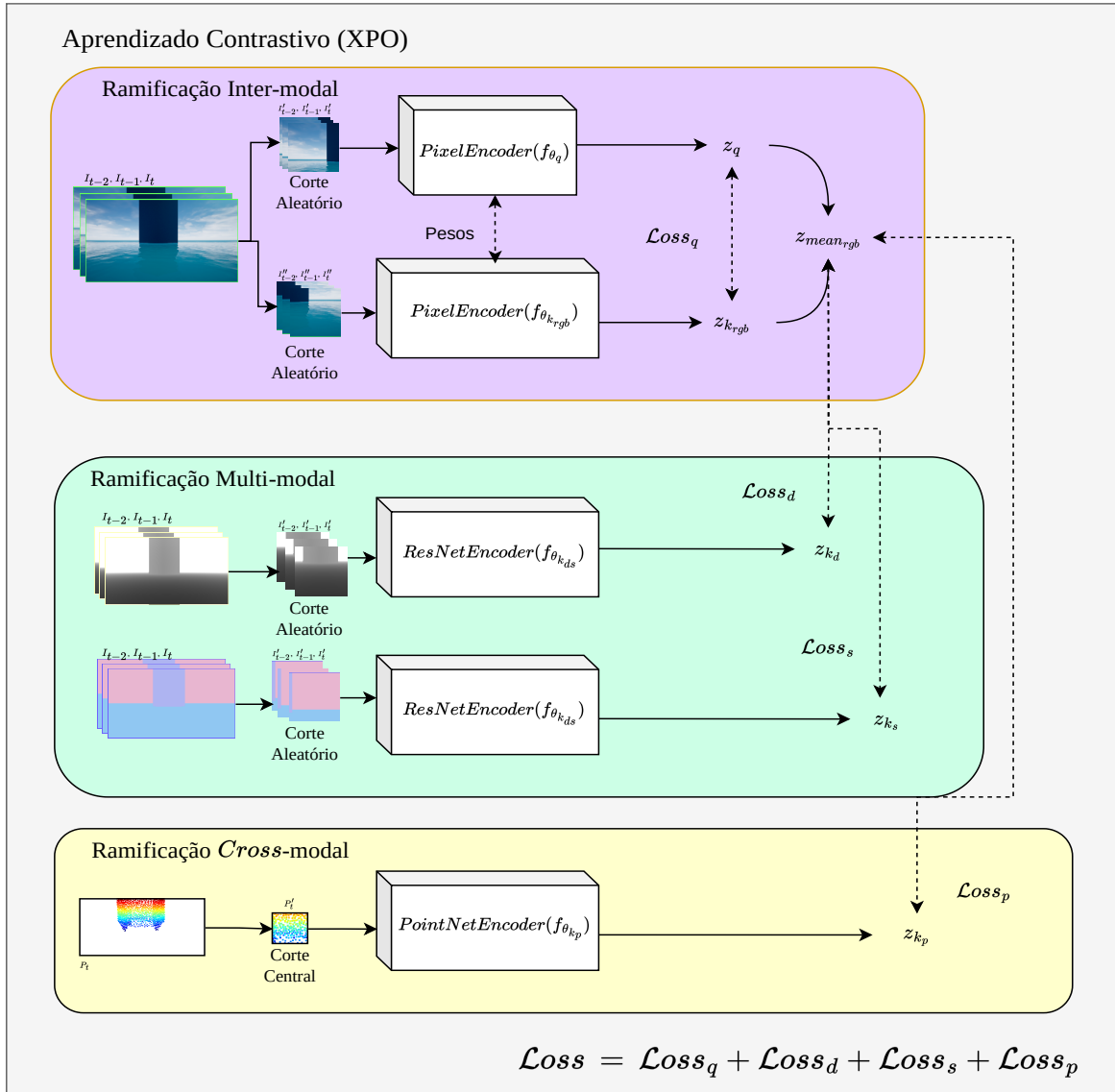


Figura 16: Diagrama do componente contrastivo.

4.1.2.1 Representação Inter-modal

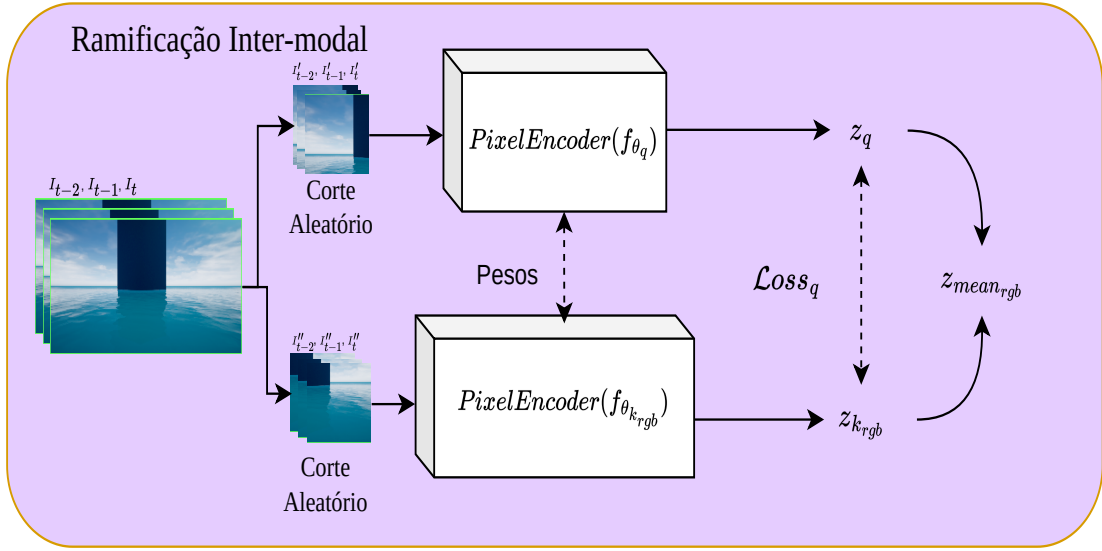


Figura 17: Ramificação inter-modal.

A implementação dessa ramificação, ilustrada na Figura 17, seguiu os mesmos parâmetros e estrutura do funcionamento apresentado em Laskin et al. [27]. Com isso, viu-se a necessidade de manter o *encoder* apresentado pelos autores, denominado *PixelEncoder* (f_{θ_q} e $f_{\theta_{k_{rgb}}}$). Esse *encoder* tem seus pesos compartilhados com as redes crítico e crítico alvo, descritas mais à frente, buscando aprender as características que melhor se adequam à aplicação. A composição das camadas do *PixelEncoder* são ilustradas na Figura 18.

Com as imagens de entrada, antes de passar pelo *PixelEncoder*, é aplicada uma mesma transformação de cortes aleatórios sobre a imagem RGB. Já que Chen et al. [25] define que o corte é uma das transformações sobre imagens que agregam maior informação ao aprendizado. Com essa transformação, são geradas uma *query* (q) e uma *key* (k_{rgb}), que resultam na função de perda parcial intra-modal ℓ_{rgb} e cruzada $\mathcal{Loss}_q$, descritas como:

$$\ell_{rgb} = \log \frac{\exp(z_q^\top W z_{k_{rgb}0})}{\exp(z_q^\top W z_{k_{rgb}0}) + \sum_{i=0}^{K-1} \exp(z_q^\top W z_{k_{rgb}i})}, \quad (14)$$

$$\mathcal{Loss}_q = \frac{1}{2N} \sum_{k=1}^N [\ell_{rgb}(z_q, z_{k_{rgb}}) + \ell_{rgb}(z_{k_{rgb}}, z_q)]. \quad (15)$$

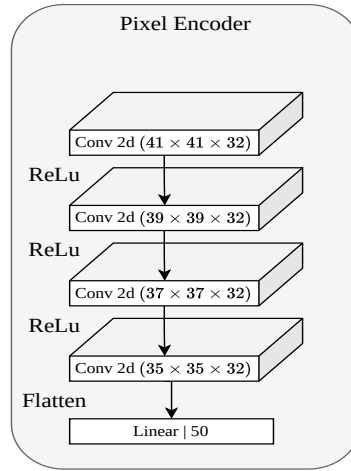


Figura 18: *PixelEncoder*.

4.1.2.2 Representação Multi-modal

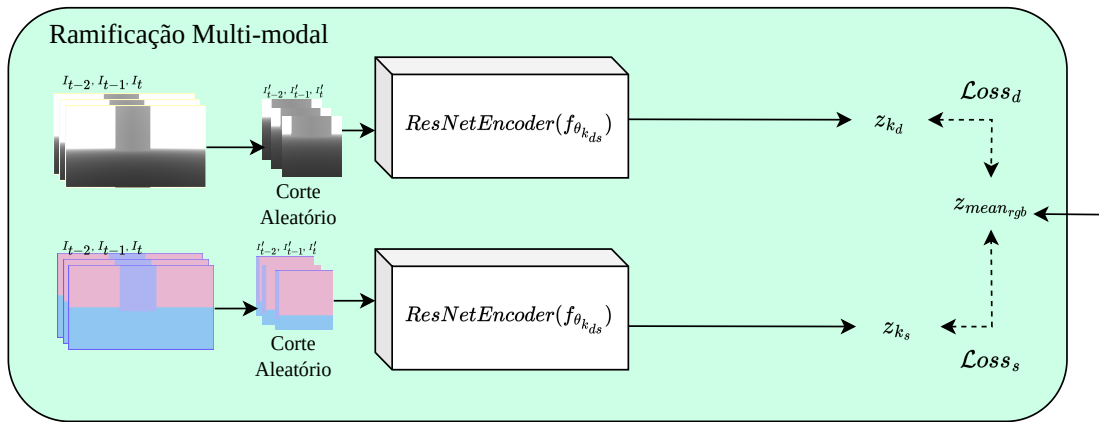


Figura 19: Ramificação multi-modal.

Pensando na escalabilidade da proposta e também no objetivo de funcionamento, procurou-se por referências de trabalhos envolvendo a geração de representações tri-dimensionais a partir de imagens para a definição do *encoder* dessa ramificação, ilustrada na Figura 19. Dessa forma, foram encontrados diversos trabalhos na literatura que abordam esse contexto. Porém, o trabalho desenvolvido por Yu et al. [149] foi o que mais se destacou.

A proposta baseia-se em Campos de Radiância Neural - NeRF [150] (do inglês, *Neural Radiance Fields*), buscando gerar o modelo 3D de um determinado objeto através da menor quantidade de vistas possível. Dessa forma, viu-se a possibilidade de utili-

zar a mesma estrutura de *encoder*, sendo ele baseado na arquitetura *ResNet34* [132]. A fundamentação de uso dada pelos autores, é que essa arquitetura de rede neural funciona muito bem como um *encoder* para extração de características que auxiliam na interpretação espacial, principalmente com poucas vistas [149]. Assim, definindo a implementação do *ResNetEncoder*, ilustrado pela Figura 20.

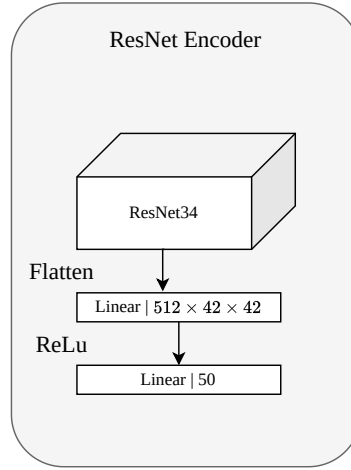


Figura 20: *ResNetEncoder*.

O *ResNetEncoder*, da mesma forma que no *PixelEncoder*, recebe as imagens já transformadas, agora de profundidade (k_d) e segmentação semântica (k_s), e codifica-as nas representações latentes z_{k_d} e z_{k_s} . Porém, a ideia agora é utilizar a estrutura *ResNet34* [132], já pré-treinada, para extrair características de profundidade e semântica com um único *ResNetEncoder* $f_{\theta_{k_{ds}}}$. A efetuação da comparação cruzada de z_{k_d} e z_{k_s} , individualmente, com os outros elementos da ramificação intra-modal, torna necessário primeiro definir o elemento $z_{mean_{rgb}}$. Ele une as informações de z_q e $z_{k_{rgb}}$ de forma ponderada, descrito como:

$$z_{mean_{rgb}} = \frac{1}{2}(z_q + z_{k_{rgb}}), \quad (16)$$

tendo agora as funções de perda parcial multi-modal ℓ_d e cruzada $\mathcal{L}_{oss_d}$ descritas pelas Equações 17 e 18, para a informação de profundidade. E também ℓ_s e $\mathcal{L}_{oss_s}$ para a informação semântica, como é descrito a seguir:

$$\ell_d = \log \frac{\exp(z_{mean_{rgb}}^\top W z_{k_d0})}{\exp(z_{mean_{rgb}}^\top W z_{k_d0}) + \sum_{i=0}^{K-1} \exp(z_{mean_{rgb}}^\top W z_{k_d i})}, \quad (17)$$

$$\mathcal{L}_{oss_d} = \frac{1}{2N} \sum_{k=1}^N [\ell_d(z_{mean_{rgb}}, z_{k_d}) + \ell_d(z_{k_d}, z_{mean_{rgb}})], \quad (18)$$

$$\ell_s = \log \frac{\exp(z_{mean_{rgb}}^\top W z_{k_s0})}{\exp(z_{mean_{rgb}}^\top W z_{k_s0}) + \sum_{i=0}^{K-1} \exp(z_{mean_{rgb}}^\top W z_{k_si})}, \quad (19)$$

$$\mathcal{Loss}_s = \frac{1}{2N} \sum_{k=1}^N [\ell_s(z_{mean_{rgb}}, z_{k_s}) + \ell_s(z_{k_s}, z_{mean_{rgb}})]. \quad (20)$$

4.1.2.3 Representação Cross-modal

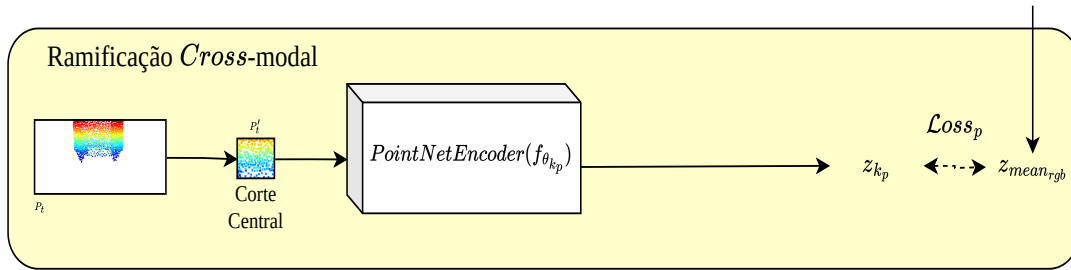


Figura 21: Ramificação Cross-modal.

Para avaliar a similaridade agora entre os elementos 2D e 3D na última ramificação, ilustrada na Figura 21, buscou-se da mesma forma que para o *ResNetEncoder*, uma referência de arquitetura que pudesse melhor se adequar à problemática. Foram avaliados trabalhos no contexto de complemento de nuvens de pontos, onde principalmente se busca entender as características tri-dimensionais do objeto a partir de poucas vistas e tentar reconstruir os pontos oclusos desconhecidos. Dessa forma, um trabalho que se destacou foi o de Groueix et al. [151] com a *AtlasNet*.

A principal característica dessa arquitetura é simplificar o processo de complemento, já que naturalmente trabalhar com nuvens de pontos é extremamente custoso do ponto de vista computacional. Portanto, utilizaram uma representação simplificada da arquitetura *PointNet* [152] para criação de um *encoder*. Dessa forma, se viu a viabilidade de adaptar o *encoder* utilizado pelos autores para efetuar a extração de características de ocupação, definindo o *PointNetEncoder* ($f_{\theta_{kp}}$) ilustrado na Figura 22.

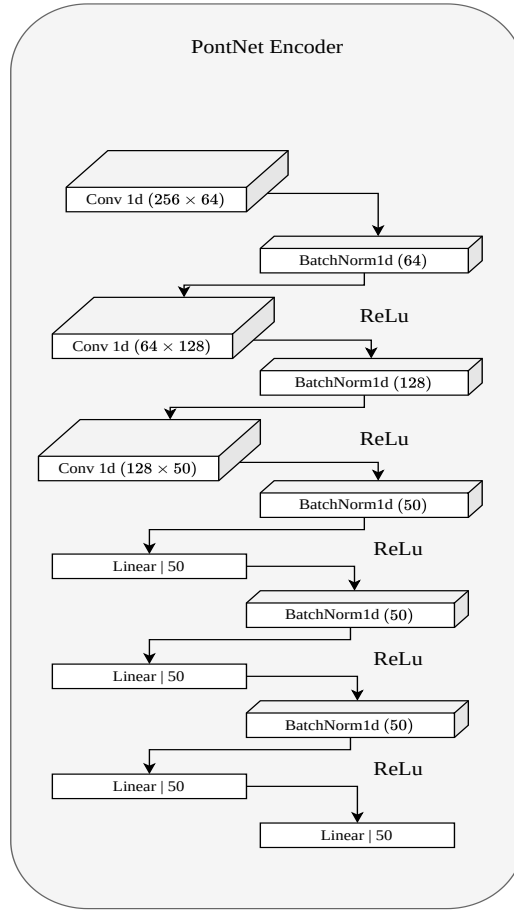


Figura 22: *PointNetEncoder*.

Visto que dados de grandes dimensões como são as nuvens de pontos demandam um grande poder computacional, a avaliação dos dados, descrita por Groueix et al. [151], mostra que uma redução no tamanho da entrada para 256 gera ainda uma extração expressiva de características da nuvem de pontos. Considerando que a problemática a ser enfrentada demanda um processamento iterativo, essa definição se encaixa perfeitamente com as necessidades da aplicação. Portanto, a nuvem reduzida (k_p), em torno do seu centro, é passada pelo *PointNetEncoder* para a geração da representação z_{k_p} . E com ela, são calculadas as funções de perda ℓ_p e $\mathcal{Loss}_p$, descritas como:

$$\ell_p = \log \frac{\exp(z_{mean_{rgb}} W z_{k_p 0})}{\exp(z_{mean_{rgb}} W z_{k_p 0}) + \sum_{i=0}^{K-1} \exp(z_{mean_{rgb}} W z_{k_p i})}, \quad (21)$$

$$\mathcal{Loss}_p = \frac{1}{2N} \sum_{k=1}^N [\ell_p(z_{mean_{rgb}}, z_{k_p i}) + \ell_p(z_{k_p i}, z_{mean_{rgb}})]. \quad (22)$$

Portanto, a função de perda total é representada pela soma de todas as perdas anteriores, sendo definida como:

$$\mathcal{Loss} = \mathcal{Loss}_q + \mathcal{Loss}_d + \mathcal{Loss}_s + \mathcal{Loss}_p. \quad (23)$$

4.1.3 Aprendizado Por Reforço Profundo

A definição do uso do algoritmo SAC [53] descrita por Laskin et al. [27], foi dada por conta da melhor capacidade de exploração que o algoritmo possui com relação aos demais. O algoritmo é projetado para lidar com espaços de ação contínuos, tornando-o uma escolha eficaz para tarefas mais complexas [27]. Sua principal ideia é usar funções de aproximação para aprender políticas de entropia máxima. Ele alterna entre avaliar e melhorar a política, buscando encontrar a função de valor V adequada e atualizar a distribuição da política. Para isso, ele utiliza três redes neurais: uma para aproximar a política (ator), outra para estimar o valor do estado V , ilustradas na Figura 23, e outra para estimar o valor de ação Q (crítica). Estas redes calculam a previsão de ação para o estado atual e geram sinais de erro de diferença temporal cada intervalo de tempo.

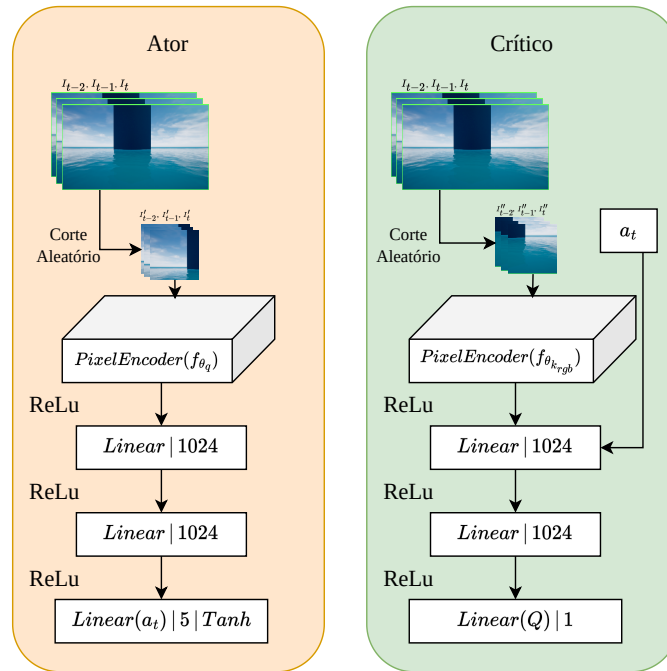


Figura 23: Redes Ator e Crítico.

Assim, com toda a estrutura contrastiva e de aprendizado por reforço profundo, é proposto o XPOCURL, descrito pelo Algoritmo 5, tendo a descrição das atualizações do aprendizado contrastivo, descrita no Algoritmo 4 para então descrever o XPOCURL.

Algoritmo 4: Atualização do Aprendizado por Contrastivo (*update - CL*)

Entrada: $encoders = \{f_{\theta_q}, f_{\theta_{k_{rgb}}}, f_{\theta_{k_{ds}}}, f_{\theta_{k_p}}\}$ e memória de repetição M

Seleciona aleatoriamente um *batch* de observação $O = \{I_q, I_{k_{rgb}}, I_{k_d}, I_{k_s}, P_{k_p}\}$:

$$I_{k_{rgb}} \leftarrow I_q$$

$$I_q \leftarrow \text{corte aleatório } (100 \times 100 \rightarrow 84 \times 84)$$

$$I_{k_{rgb}} \leftarrow \text{corte aleatório } (100 \times 100 \rightarrow 84 \times 84)$$

$$I_{k_d} \leftarrow \text{corte aleatório } (100 \times 100 \rightarrow 84 \times 84)$$

$$I_{k_s} \leftarrow \text{corte aleatório } (100 \times 100 \rightarrow 84 \times 84)$$

$$B(I) = \{(s, a, r, s', O)\} \text{ de } M$$

Computa os espaços latentes:

$$q = f_{\theta_q}(I_q), k_{rgb} = f_{\theta_{k_{rgb}}}(I_{k_{rgb}})$$

$$k_d = f_{\theta_{k_{ds}}}(I_{k_d}), k_s = f_{\theta_{k_{ds}}}(I_{k_s})$$

$$k_p = f_{\theta_{k_p}}(P_{k_p})$$

Atualiza a *query*:

$$\mathcal{Loss}_q = \log \frac{\exp(q^T k_{rgb0})}{\sum_{i \in B(O)} \exp(q^T k_{rgb i})}$$

$$q_{mean} = \frac{q + k_{rgb}}{2}$$

$$\mathcal{Loss}_d = \log \frac{\exp(q_{mean}^T k_{d0})}{\sum_{i \in B(O)} \exp(q_{mean}^T k_{di})}$$

$$\mathcal{Loss}_s = \log \frac{\exp(q_{mean}^T k_{s0})}{\sum_{i \in B(O)} \exp(q_{mean}^T k_{si})}$$

$$\mathcal{Loss}_p = \log \frac{\exp(q_{mean}^T k_{p0})}{\sum_{i \in B(O)} \exp(q_{mean}^T k_{pi})}$$

$$\mathcal{Loss} = \mathcal{Loss}_q + \mathcal{Loss}_d + \mathcal{Loss}_s + \mathcal{Loss}_p$$

Atualiza a *key*:

$$f_{\theta_{k_{rgb}}} \leftarrow \tau f_{\theta_{k_{rgb}}} + (1 - \tau) f_{\theta_q}$$

Algoritmo 5: Representações Contrastivas não-supervisionadas de Ocupação Panóptica *Cross-modal* para Aprendizado por Reforço (XPOCURL)

Entradas: Parâmetros iniciais da rede ator ϕ , parâmetros da rede crítica θ_1, θ_2 , memória de repetição M , número máximo de intervalos de tempo T

Define os parâmetros das redes alvos $\theta'_1 \leftarrow \theta_1, \theta'_2 \leftarrow \theta_2$

para $t = 1$ **até** T **faça**

 Observa o estado atual $O = \{RGB_t, D_t, S_t, P_t\}$

$I = RGB_t$

 Imagem é passada pela *query* $s = f_{\theta_q}(I)$

 Observa estado s e seleciona ação $a \sim \pi_\phi(\cdot|s)$ e executa a no ambiente

 Observa próximo estado s' , e recompensa r

 Armazena (s, a, r, s', O) na memória M

se *Atualiza SAC* **então**

 Seleciona aleatoriamente um *batch*, $B = \{(s, a, r, s')\}$ de M

 Computa a correção para o valor da ação Q:

$$y(r, s') = r + \gamma \left(\min_{i=1,2} Q_{\theta'_i}(s', a') - \alpha \log \pi_\phi(a'|s') \right)$$

 Atualiza a rede crítica:

$$\nabla_{\theta_{i=1,2}} \frac{1}{|B|} \sum_{(s,a,r,s') \in B} (Q_{\theta_i}(s, a) - y(r, s'))^2$$

 Atualiza a política da rede ator:

$$\nabla_\phi \frac{1}{|B|} \sum_{s \in B} \left(\min_{i=1,2} Q_{\theta_i}(s, a_\phi) - \alpha \log \pi_\phi(a_\phi|s) \right)$$

 Atualiza as redes alvo: $\theta'_{i=1,2} \leftarrow \tau \theta'_i + (1 - \tau) \theta_i$

fim

se *Atualiza rede contrastiva* **então**

 | *update-CL* 4

fim

fim

4.1.4 Inferência

4.1.4.1 Planejamento de caminho

Com o planejamento dos NBVs, é necessária a geração do caminho que interseccionará as vistas previstas. O método de planejamento de caminho RRT* [72], mostrou-se um método viável para o contexto do presente trabalho. Porém, por conta da natureza do planejamento, viu-se a necessidade de efetuar algumas modificações com relação à

orientação, já que originalmente o planejamento não considera diretamente a orientação ao longo do caminho. Logo, o deslocamento do veículo ao longo do trajeto ocorre, comumente, com sua frente continuamente orientada em direção à posição-alvo, tornando o caminho inapropriado para a reconstrução.

Para a adaptação, avaliou-se na literatura o conceito de planejamento com o RRT* utilizando alvos virtuais [120, 121]. A integração de alvos virtuais ao longo do trajeto, entre os pontos inicial e final, permite que o veículo mantenha uma orientação consistente ao longo do caminho. O processo começa com a definição do ponto inicial q_{inicial} e do ponto final q_{final} , além de alvos virtuais intermediários $\{q_{\text{virtual},i}\}$. Esses alvos são estrategicamente posicionados para alinhar a orientação do veículo com o objetivo final. O algoritmo então inicia a expansão de uma árvore a partir de q_{inicial} , explorando o espaço de configuração com amostras aleatórias $q_{\text{aleatório}}$. Para cada nova amostra, identifica-se o ponto mais próximo na árvore $q_{\text{próximo}}$ e tenta-se uma conexão na direção de $q_{\text{aleatório}}$.

Durante a construção da árvore, verificam-se as condições de viabilidade para cada conexão, como a ausência de colisões. Uma vez que um novo nó q_{novo} é adicionado, o RRT* reavalia conexões ao seu redor ajustando a estrutura para otimizar o custo total do trajeto, considerando como função de custo a distância entre q_{novo} e q_{inicial} . Os alvos virtuais desempenham um papel fundamental nessa etapa, pois são tratados como objetivos intermediários. O algoritmo prioriza conexões que alinhem o caminho com esses alvos, ajudando a garantir uma transição suave na direção desejada. Assim, o trajeto gerado pelo RRT* incorpora a orientação desejada do veículo, alinhando o ângulo de guinada θ com o objeto alvo ao longo do caminho.

4.1.4.2 Geração da representação tri-dimensional

Para geração das nuvens de pontos, utilizou-se o mapeamento e localização em tempo real baseado em aparência - RTAB-Map [153] (do inglês, *Real-Time Appearance-Based Mapping*), sendo esse um sistema avançado de mapeamento e localização simultânea - SLAM (do inglês, *Self-Localization And Mapping*). Projetado para ser usado em robótica móvel, sua principal função é criar mapas tridimensionais do ambiente ao redor do robô, em tempo real, enquanto acompanha sua postura dentro desse ambiente em constante mudança.

Esse sistema funciona com dados visuais provenientes de câmeras, sendo classificado como SLAM visual. Sua abordagem incremental permite a construção contínua do mapa à medida que o robô se move, integrando novas observações ao mapa existente. Uma de suas principais vantagens é sua capacidade de lidar efetivamente com desafios como fechamento de loop, onde o robô retorna a áreas previamente exploradas. A capacidade de se adaptar às mudanças nas condições de iluminação é essencial do RTAB-Map, tornando-o adequado para ambientes reais onde as condições luminosas podem variar. As circunstâncias relacionadas à iluminação são bem tratadas nos ambientes foto-realísticos

criados, levando o RTAB-Map a ser uma ótima opção para o mapeamento da estrutura.

5 RESULTADOS

Este capítulo apresenta e discute os resultados obtidos com a abordagem proposta no presente trabalho. Os dados coletados são relacionados ao comportamento do agente no ambiente de simulação, possibilitando avaliar seu desempenho na tarefa de cobertura. Portanto, são apresentados os resultados de treinamento e avaliação, cobertura, e consequentemente geração da representação tri-dimensional, juntamente com a amostragem do caminho percorrido ao longo de todas as vistas. Para isso, é inicialmente feita a avaliação da viabilidade da proposta considerando uma recompensa ainda em construção. Dessa forma, foi analisada a capacidade de convergência do algoritmo base CURL [27]. Além disso, avaliou-se também a adição contrastiva de informação, nesse caso a imagem de profundidade, através do agente denominado de representações contrastivas estéreo não-supervisionadas multi-modais para aprendizado por reforço - MSCURL (do inglês, *Multi-modal Stereo Contrastive Unsupervised Representations for Reinforcement Learning*), intermediário ao XPOCURL. Assim, é feita a avaliação dos agentes CURL, MSCURL e XPOCURL no multi-domínio (aéreo e sub-aquático), comparando-os com a contra-proposta abordada pelo GenNBV [49] nesse contexto de aplicação.

5.1 Configuração Experimental

Para a apresentação dos resultados, foram efetuados experimentos em três contextos, primeiro foi o processo de treinamento dos agentes por um período de 3000000 passos. Esse intervalo, sabendo que um episódio é composto por no máximo 200 passos, configura um número mínimo de episódios igual à 15000. Esse número é dado como mínimo, pois um intervalo é interrompido antes dos 200 passos caso a cobertura seja efetuada. Os seguintes processos são com relação à avaliação do agente. A segunda etapa foi uma avaliação recorrente efetuada à cada 100000 passos, por um período de 10 episódios. A última etapa foi a avaliação do agente em outros ambientes mais complexos com diferentes estruturas, onde efetuou-se a avaliação após o período de treinamento.

Para a obtenção dos resultados e propriamente o desenvolvimento do trabalho, foi utilizado o uma composição de *Docker* [154] *containers*, como é apresentado na Figura 24.

Foram utilizados dois *containers*. O primeiro, com um sistema operacional Ubuntu 16.04, sendo responsável por executar o ambiente de simulação na UE4. O segundo foi criado com Ubuntu 20.04, utilizando a linguagem *Python* na versão 3.11 para implementação dos *scripts* que compõem o trabalho. Para o desenvolvimento do agente, foi utilizado o *Pytorch* na versão 3.8.10 e *CUDA* na versão 12.6. Em termos de hardware, foi utilizada uma *CPU Ryzen 9*, utilizando 120GB de de memória *RAM* e uma *GPU* dedicada do modelo *RTX - 4090* com 24GB de *VRAM*.

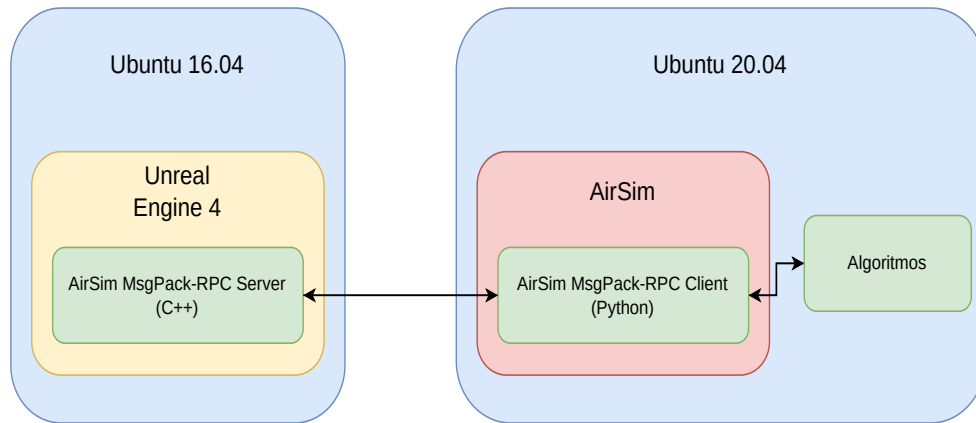


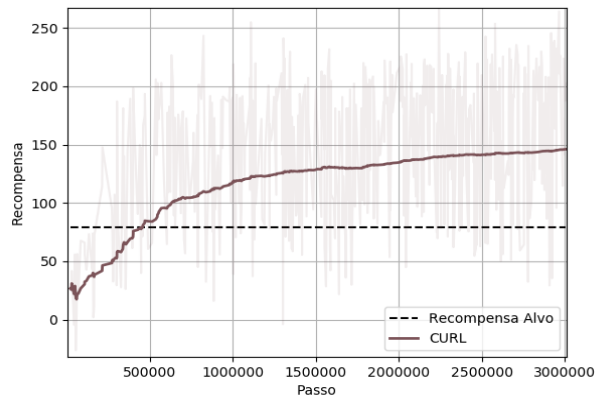
Figura 24: Estrutura geral do sistema.

5.2 Análise de viabilidade e comportamento

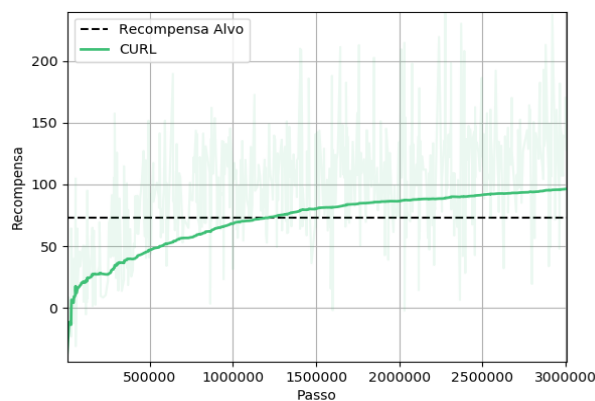
Em primeiro lugar, o objetivo foi avaliar o comportamento do agente CURL nos domínios aéreo e sub-aquático, sendo eles mais simples em um primeiro momento, e então expô-lo ao ambiente multi-domínio. Para isso, manteve-se o conceito de recompensa utilizando os marcadores como métrica de cobertura. Porém, nesse caso, utilizando uma versão mais simples da função de recompensa descrita na Equação 13 apenas para avaliar a viabilidade da proposta. Nessa versão, ainda embrionária, a checagem $xs = pxs$, que analisa vistas repetidas, ainda não havia sido adicionada à função.

Para a análise, assumiu-se que o comportamento das recompensas de um episódio devem somar um valor maior que o número de marcadores para considerar um planejamento de NBV válido. Portanto, buscou-se utilizar o ambiente com a estrutura hexagonal para uma avaliação prévia de viabilidade da abordagem base. Esse ambiente contém 79 marcadores no domínio aéreo, 73 no sub-aquático e, portanto, 152 no multi-domínio. Para a geração dos gráficos que representam o comportamento dos agentes, foi considerada a média móvel dos dados considerando as 20 próximas amostras. Assim, os gráficos ilustrados pelas Figuras 25a) e 25b) apresentam bons resultados quanto ao comportamento do agente através da recompensa média dos episódios durante o período de treinamento,

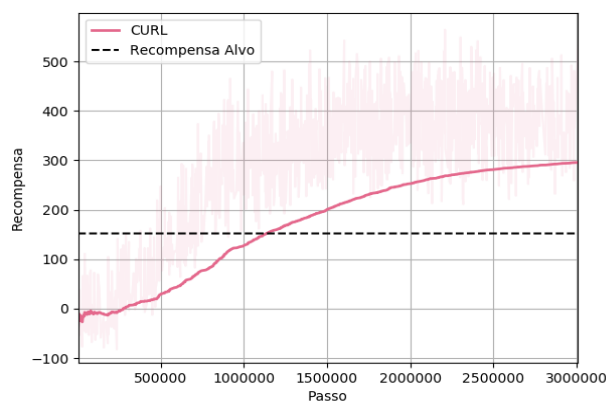
onde é possível ver que tanto no domínio aéreo quando no sub-aquático o agente consegue desempenhar um planejamento que efetua a cobertura da estrutura.



(a) Domínio Aéreo.



(b) Domínio Sub-aquático.



(c) Multi-Domínio.

Figura 25: Resultados de treinamento do algoritmo base para a proposta em domínios distintos.

Ao observar-se a viabilidade da abordagem inicial em cada domínio separadamente, buscou-se também efetuar a avaliação no contexto multi-domínio. O comportamento do agente é ilustrado na Figura 25c), onde mostrou-se, novamente, capaz de convergir mesmo com a tarefa mais complexa considerando ambos os domínios. Dessa forma, viu-se a possibilidade de efetuar as adições e adaptações propostas para as representações contrastivas que levam à caracterização do algoritmo proposto XPOCURL.

Entretanto, com os resultados obtidos, foram analisados alguns pontos com relação ao comportamento do agente voltados à cobertura. Foi possível constatar que durante o período de avaliação, as vistas definidas impossibilitaram gerar um caminho que favorecesse a criação da representação 3D com os dados visuais obtidos. Isso foi atribuído a versão simplificada da recompensa, onde a comparação entre a relação de vistas atuais e anteriores não existia. Por conta disso, vistas muito similares se tornaram uma possibilidade para o agente durante o período de treinamento, ocasionando durante a inferência em diversas posturas inválidas que não permitiram a correspondência necessária para a reconstrução da estrutura.

5.3 Análise de desempenho

Com as considerações feitas na primeira etapa, foi possível fazer as alterações necessárias que levaram à função de recompensa apresentada. Dessa forma, viu-se a possibilidade de avaliar, além da abordagem original, a adição de informação multi-modal de profundidade com a versão prévia do algoritmo proposto (MSCURL). Com isso, foi possível acrescentar informação multi-modal e também o componente *cross*-modal (XPOCURL), diretamente no contexto multi-domínio. Além disso, considerando que a abordagem em GenNBV [49] vai diretamente de encontro ao problema conceitual no escopo de DRL que o presente trabalho procura tratar, buscou-se avaliar seu desempenho no contexto de planejamento em multi-domínio considerando um ambiente *offshore*.

5.3.1 Treinamento

Durante o período de treinamento, utilizando o ambiente com a estrutura hexagonal como alvo, os quatro agentes (CURL, MSCURL, XPOCURL e GenNBV) foram avaliados além da recompensa por episódio, por duas outras métricas. A primeira é com relação ao percentual médio de cobertura ao longo do treinamento. Essa métrica busca avaliar a capacidade de cobertura considerando também vistas sem ganho de informação ao longo do planejamento. A segunda métrica, é com relação a cobertura efetuando apenas vistas que detectam novos marcadores ao longo de todo planejamento.

O gráfico ilustrado na Figura 26 demonstra o comportamento de todos os agentes no ambiente multi-domínio. A função de recompensa definida, aproxima o resultado médio das recompensas ao número de marcadores do ambiente. Dessa forma, o comportamento

das recompensas ilustra que a adição de informação multi-modal e, consequentemente, *cross*-modal trouxeram ganhos significativos à abordagem inicial com o CURL, destacando o comportamento apresentado pelo XPOCURL. Além disso, foi possível determinar que a abordagem proposta com o GenNBV não conseguiu sequer ter uma compreensão mínima do ambiente em questão, considerando as mesmas condições das abordagens baseadas em representações contrastivas. Esse comportamento condiz exatamente com o tema abordado por de Jesus et al. [24], onde descreve a dificuldade dos algoritmos de DRL de lidar com espaços de observação muito grandes ou complexos, como é o caso.

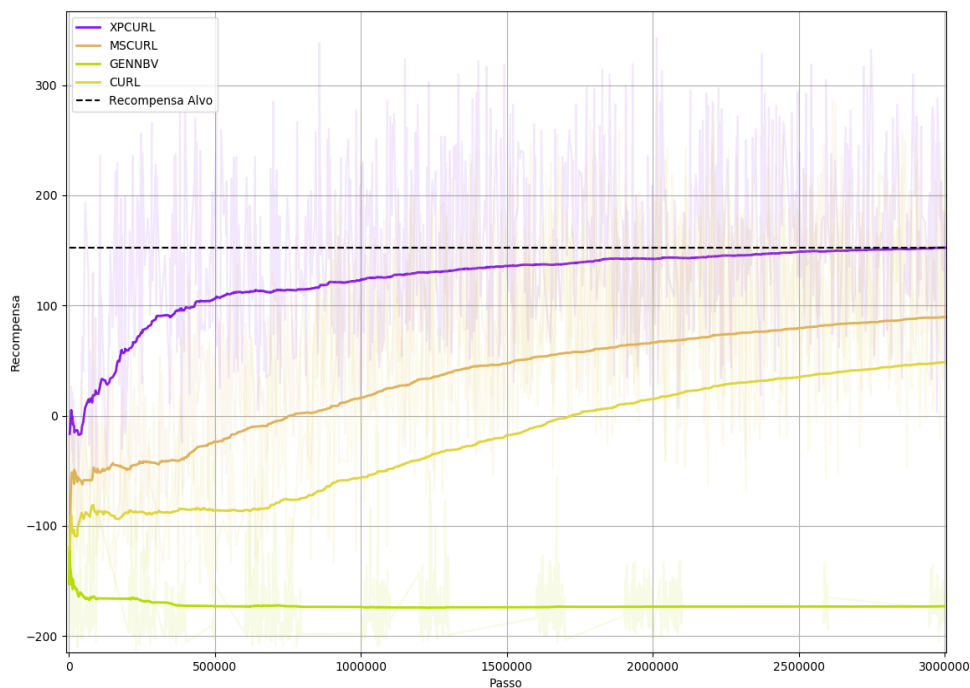


Figura 26: Média das recompensas ao longo do treinamento.

Fonte: O autor

A Tabela 2 ilustra as métricas de avaliação do comportamento atrelado às métricas de cobertura. Dessa forma, é ilustrado o percentual de efetuação da cobertura geral, considerando também as vistas sem ganho de informação, e da cobertura ótima, considerando apenas vistas com ganho de informação. É possível ver melhor o desempenho dos agentes com relação à avaliação geral de cobertura. Nessa avaliação, consegue-se analisar a convergência dos algoritmos através da média de coberturas efetuadas ao longo do período de treinamento, tendo destaque para o comportamento do XPOCURL. Contudo, o GenNBV mostra que mesmo considerando vistas sem ganho de informação, foi incapaz de efetuar qualquer cobertura completa da estrutura, ou seja, não conseguiu aprender a tarefa no

ambiente.

Algoritmo	Cobertura Geral	Cobertura Ótima
XPOCURL[O Autor]	91,48%	71,2%
MSCURL[O Autor]	68,1%	45%
CURL[Laskin et al. [27]]	59,6%	38,6%
GenNBV[Chen et al. [49]]	0%	0%

Tabela 2: Desempenho de Cobertura dos Algoritmos.

5.3.2 Avaliação gradativa

Durante o período de avaliação, ao longo do treinamento, buscou-se analisar o comportamento dos agentes a partir da média das recompensas entre os episódios desse período. Além disso, também considerou-se a média das recompensas máximas ao longo de todas as avaliações. A representação gerada pelo melhor planejamento ao longo das avaliações também foi uma das principais métricas utilizada para avaliar a performance dos algoritmos. Contudo, para efetuar a avaliação, o algoritmo GenNBV foi desconsiderado. Essa desconsideração deu-se devido ao fato de o agente não ter aprendido ao longo do treinamento, o que consequentemente o torna incapaz de efetuar a reconstrução com outras estruturas.

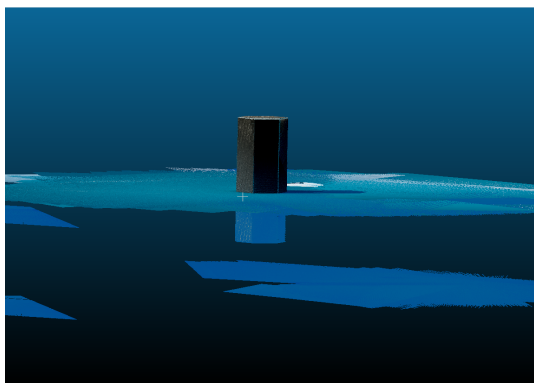
A Tabela 3 apresenta os resultados do comportamento obtido durante a avaliação a partir da média das recompensas de cada período, ilustrando o valor médio ao longo de todo o treinamento. O algoritmo XPOCURL também se destacou durante a avaliação, tendo a melhor média de recompensas ao longo dos períodos. O mesmo posicionamento aconteceu com os demais, mantendo MSCURL e CURL seguidos do não funcionamento do GenNBV.

Algoritmo	Recompensa Média	Recompensa Máxima
XPOCURL[O Autor]	154.2	184.4
MSCURL[O Autor]	85,1	126,6
CURL[Laskin et al. [27]]	61,1	101,2
GenNBV[Chen et al. [49]]	-164,2	-48,4

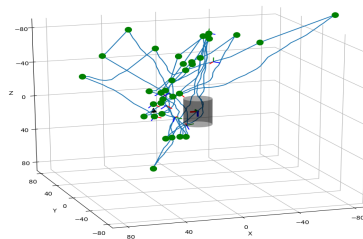
Tabela 3: Avaliação de Recompensas e Número de Vistas.

Para ilustrar o comportamento dos agentes de maneira qualitativa, são apresentados a cobertura reconstruída, a amostragem e histograma de densidade¹, além da descrição do caminho para sua geração a partir da máxima recompensa obtida, ilustradas nas Figuras 27 e 28. A principal relação que se busca avaliar com esses dados é o quão eficiente e eficaz são os caminhos gerados, nesse caso aprendidos. Essa conclusão é auxiliada com base nas análises de densidade.

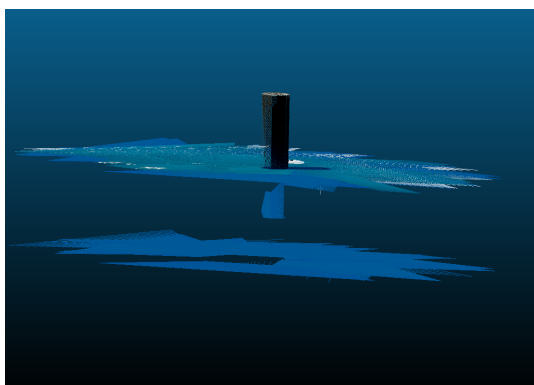
¹CloudCompare



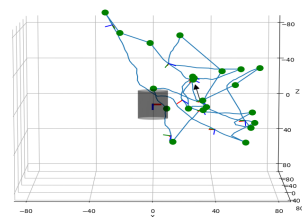
(a) Representação gerada por XPOCURL.



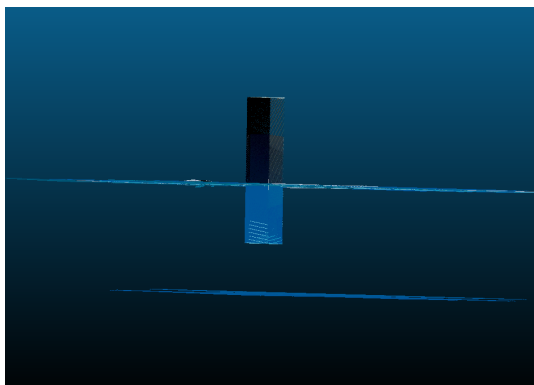
(b) Caminho gerado por XPOCURL.



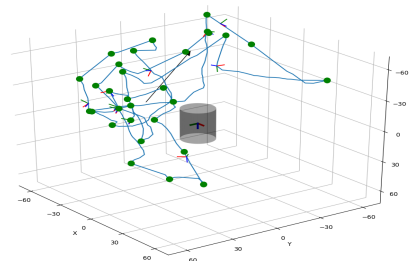
(c) Representação gerada por MSCURL.



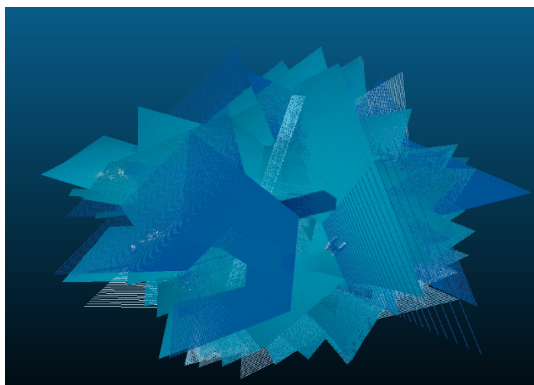
(d) Caminho gerado por MSCURL.



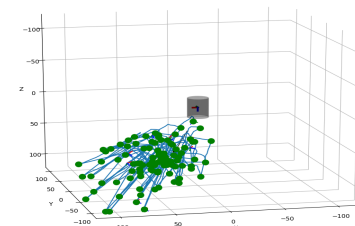
(e) Representação gerada por CURL.



(f) Caminho gerado por CURL.

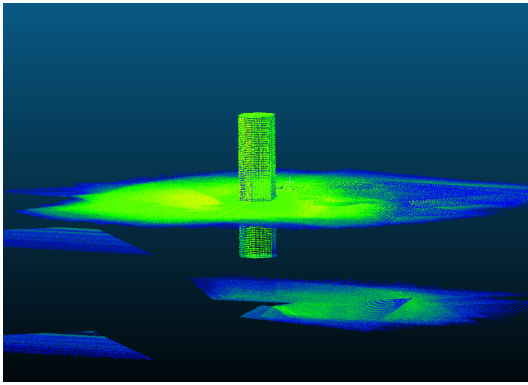


(g) Representação gerada por GenNBV.

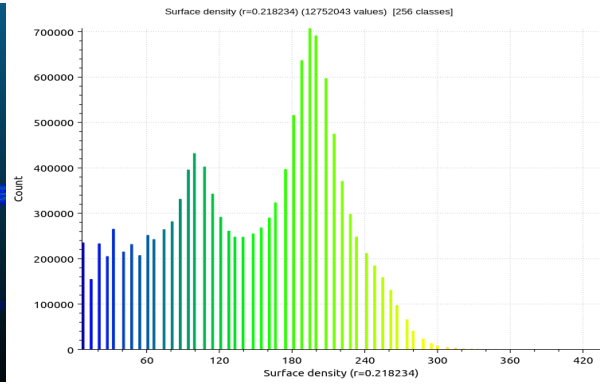


(h) Caminho gerado por GenNBV.

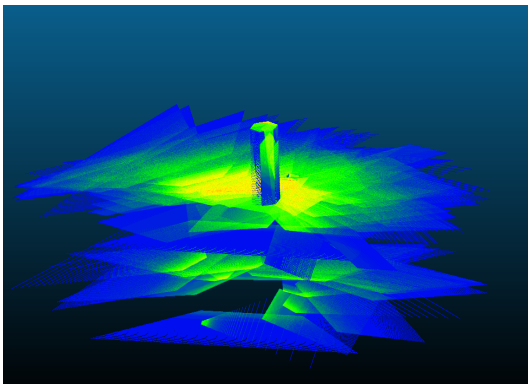
Figura 27: Representações tri-dimensionais da estrutura de treinamento.



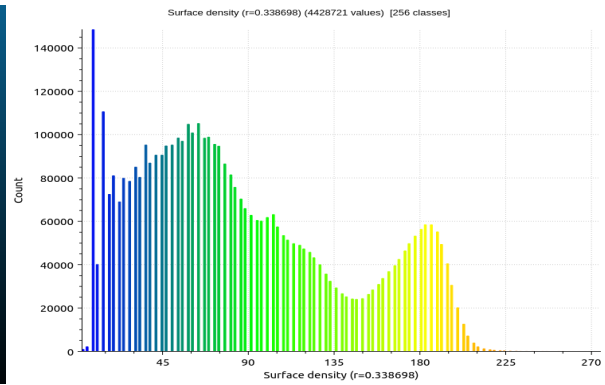
(a) Densidade gerada por XPOCURL.



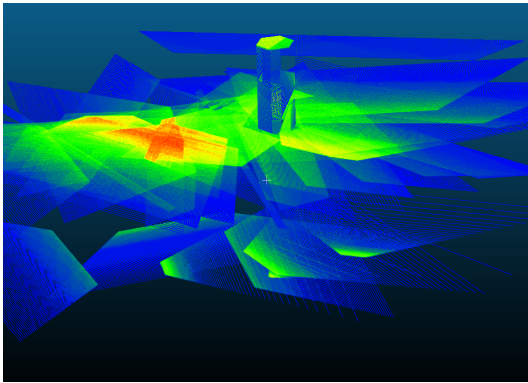
(b) Distribuição Gaussiana de densidade.



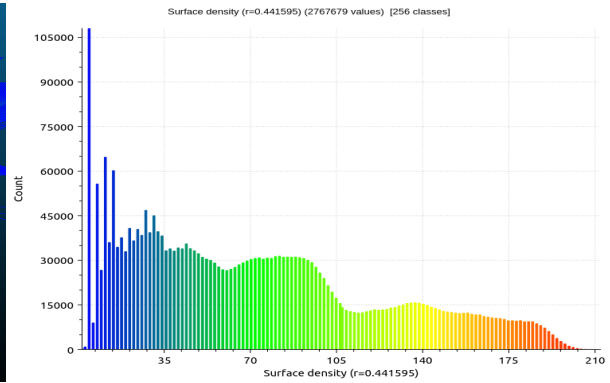
(c) Densidade gerada por MSCURL.



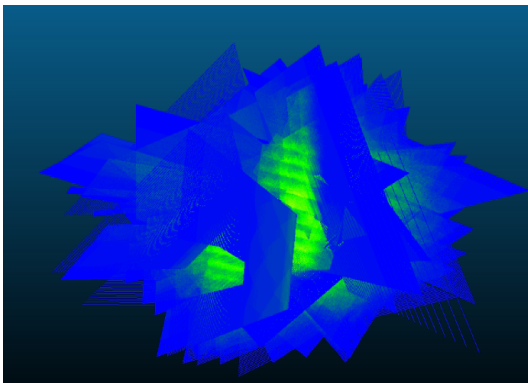
(d) Distribuição Gaussiana de densidade.



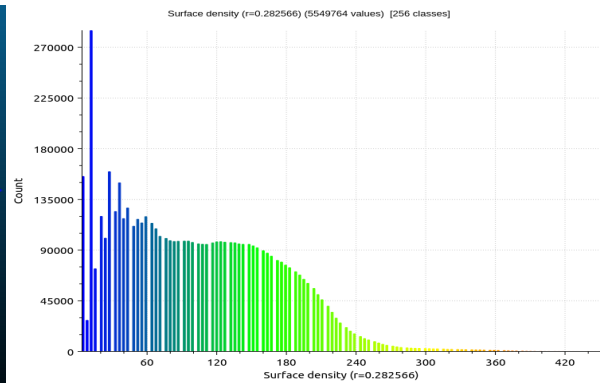
(e) Densidade gerada por CURL.



(f) Distribuição Gaussiana de densidade.



(g) Densidade gerada por GenNBV.



(h) Distribuição Gaussiana de densidade.

Figura 28: Representações tri-dimensionais da estrutura de treinamento.

Com a visualização da reconstrução e a amostragem de densidade, ilustrados na Figura 28, é possível visualizar através dos histogramas, o decaimento da distribuição dos pontos ao longo entre um agente e outro. Esse decaimento implica diretamente na qualidade do aprendizado, pois uma melhor distribuição dos pontos reconstruídos implica em um caminho mais eficaz e, conseqüentemente, implica na generalização do aprendizado.

5.3.3 Avaliação com diferentes estruturas

Com os agentes já treinados e avaliados durante o período de treinamento com a mesma estrutura, foi efetuada a avaliação para cada uma das quatro estruturas do mundo real encontradas em ambientes *offshore*. Com isso, buscou-se analisar a os resultados de cobertura, relacionando-os com a representação tri-dimensional da estrutura. A Tabela 4 ilustra a relação de cobertura efetuada por cada um dos agentes, considerando as estruturas dispostas em cada um dos 4 ambientes. Os ambientes definidos, conscientemente escalonam a complexidade na sua sequência, indo do mais simples para o mais complexo.

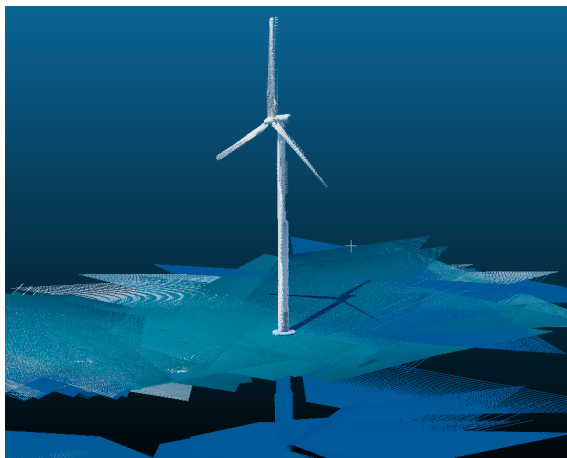
Com base nessa sequência crescente de complexidade, foi possível observar que os agentes apresentaram desempenhos variáveis dependendo das características de cada ambiente. No ambiente mais simples, os agentes obtiveram altos índices de cobertura, refletindo sua capacidade de generalização. À medida que os ambientes se tornaram mais complexos, o desempenho variou significativamente, indicando desafios como maior detalhamento e estruturas tridimensionais mais intrincadas. Esses resultados reforçam a importância de considerar a complexidade estrutural durante o treinamento e a avaliação de agentes.

Além disso, buscou-se analisar os agentes, do ponto de vista de espaço de observação, a fim de avaliar o comportamento relacionado à composição de informação. Com isso, no contexto de representações contrastivas, procurou-se efetuar essa análise relacionando-a diretamente com o desempenho de cobertura do agente. Dessa forma, avaliando a implicação nas estratégias de exploração, já que, estratégias robustas tendem a ter maior capacidade de adaptação em ambientes complexos.

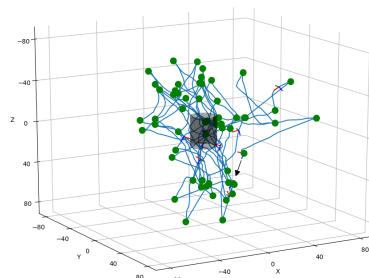
Algoritmo	Ambiente 1	Ambiente 2	Ambiente 3	Ambiente 4
XPOCURL[O Autor]	97,1%	88,2%	99,9%	55,42%
MSCURL[O Autor]	89%	85,4%	82%	33%
CURL[Laskin et al. [27]]	91,3%	49,8%	93%	0%

Tabela 4: Avaliação de Cobertura dos Algoritmos.

5.3.3.1 Ambiente 1 - Eólica mono-estaca



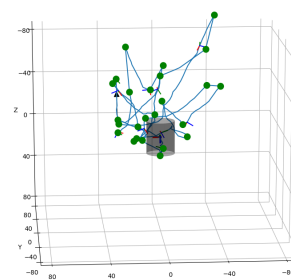
(a) Representação gerada por XPOCURL.



(b) Caminho gerado por XPOCURL.



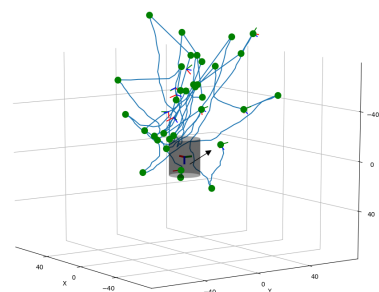
(c) Representação gerada por MSCURL.



(d) Caminho gerado por MSCURL.



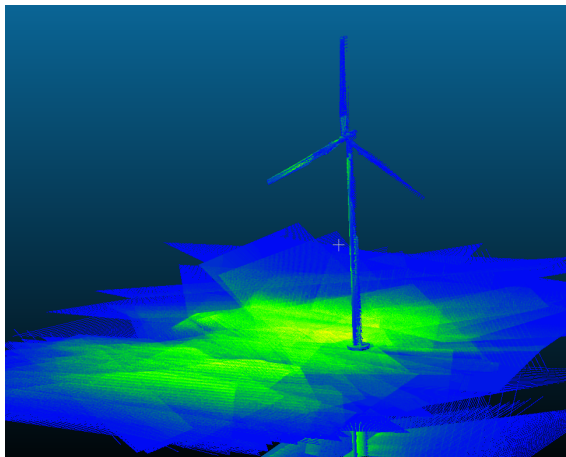
(e) Representação gerada por CURL.



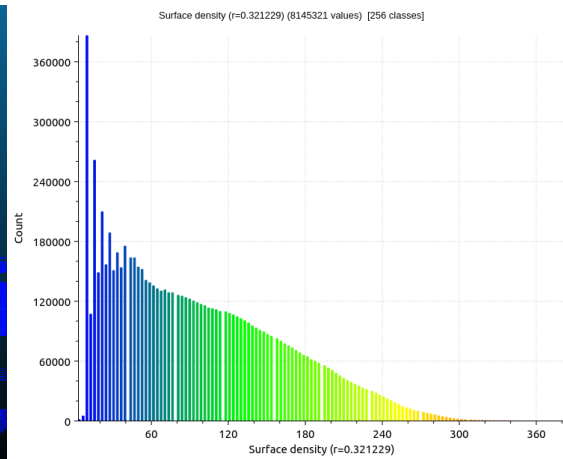
(f) Caminho gerado por CURL.

Figura 29: Representações tri-dimensionais da estrutura de treinamento.

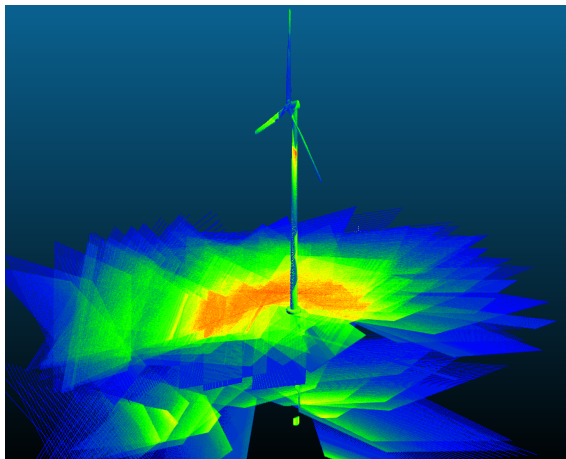
Fonte: O autor



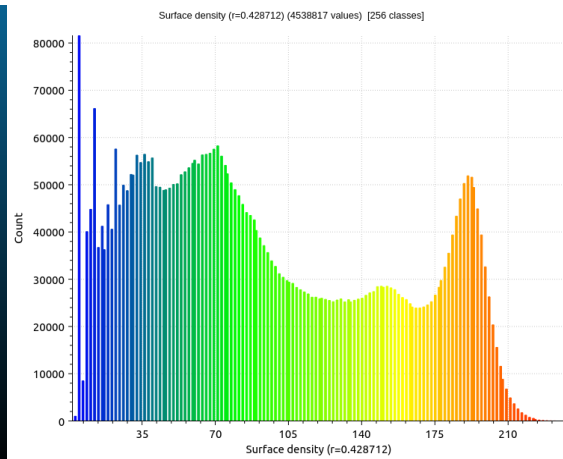
(b) Densidade gerada por XPOCURL.



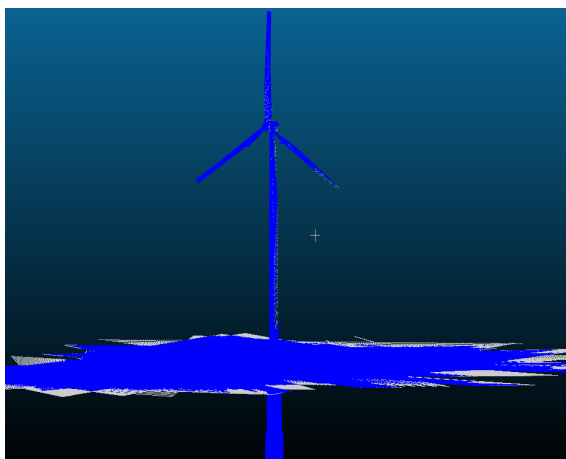
(c) Distribuição Gaussiana de densidade.



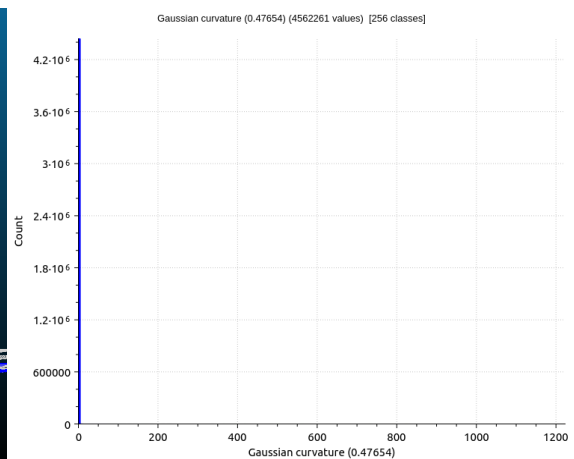
(d) Densidade gerada por MSCURL.



(e) Distribuição Gaussiana de densidade.



(f) Densidade gerada por CURL.

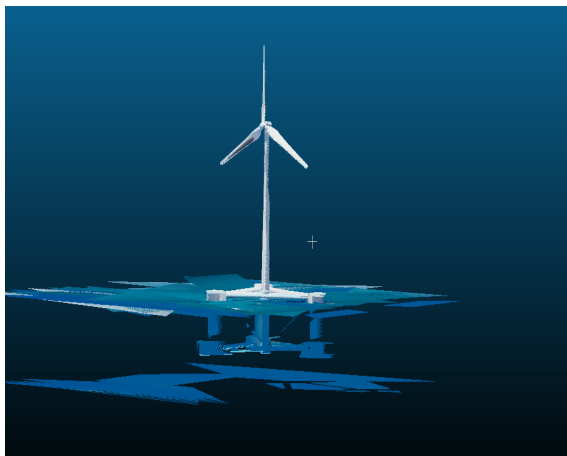


(g) Distribuição Gaussiana de densidade.

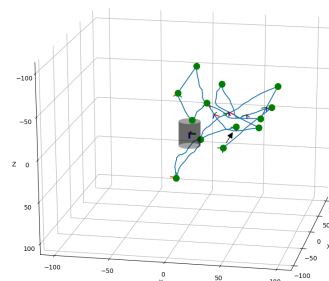
Figura 30: Densidade de pontos da estrutura de treinamento.

Fonte: O autor

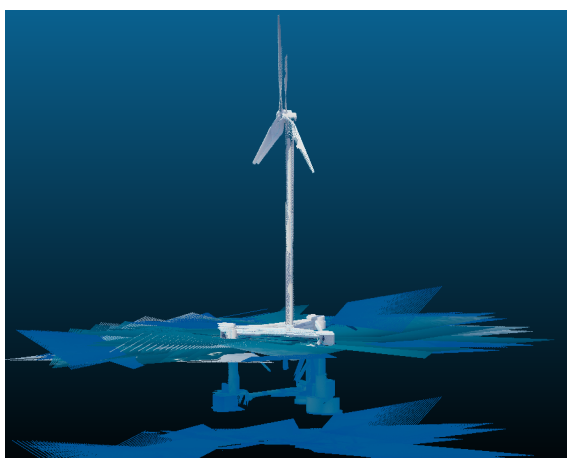
5.3.3.2 Ambiente 2 - Eólica semi-submersível



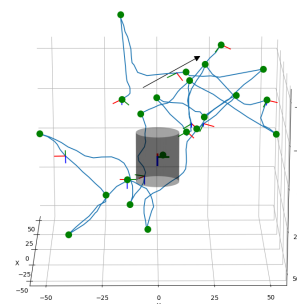
(b) Representação gerada por XPOCURL.



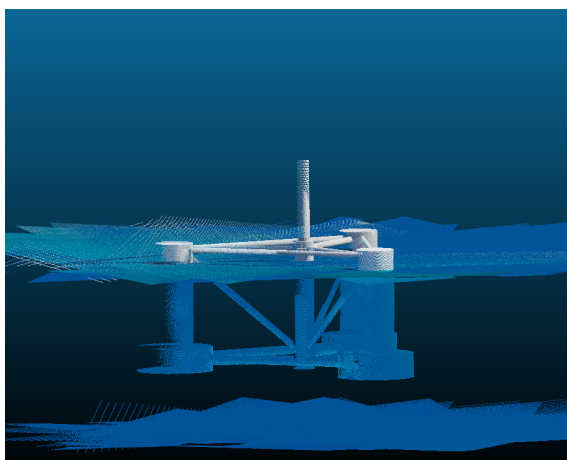
(c) Caminho gerado por XPOCURL.



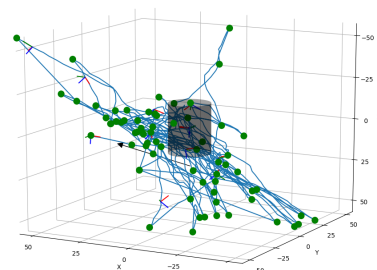
(d) Representação gerada por MSCURL.



(e) Caminho gerado por MSCURL.



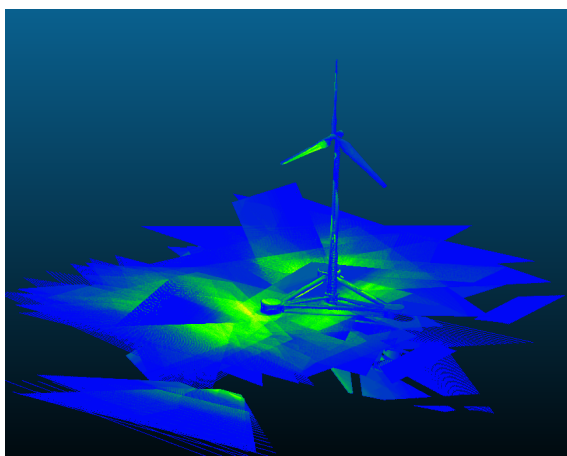
(f) Representação gerada por CURL.



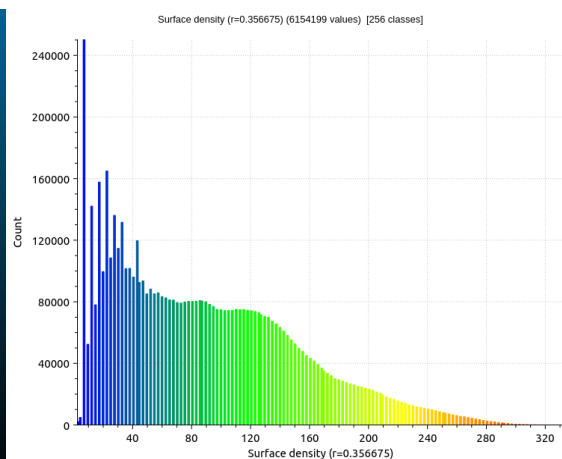
(g) Caminho gerado por CURL.

Figura 31: Representações tri-dimensionais da estrutura: Eólica semi-submersível

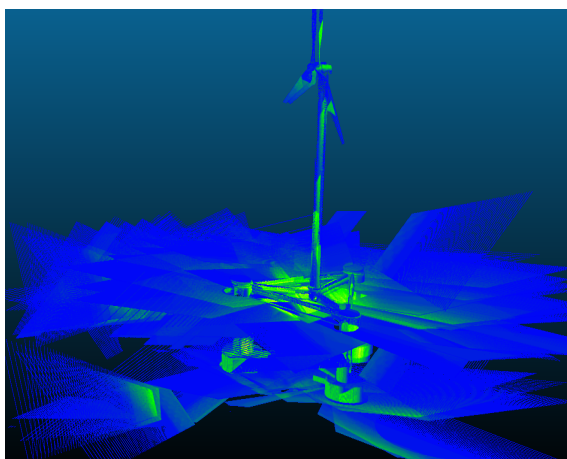
Fonte: O autor



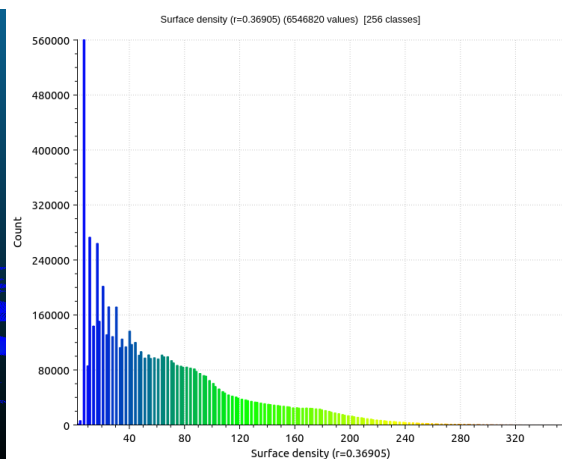
(b) Densidade gerada por XPOCURL.



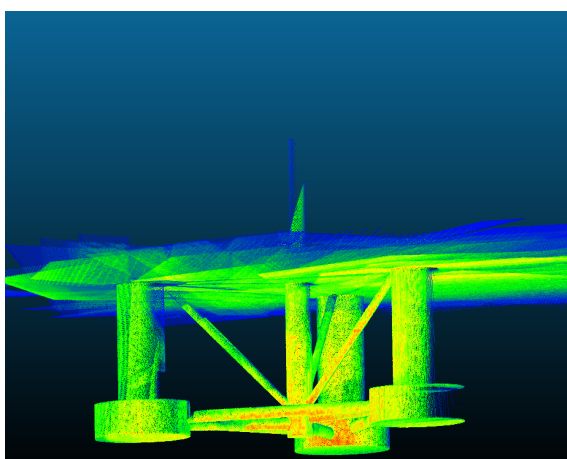
(c) Distribuição Gaussiana de densidade.



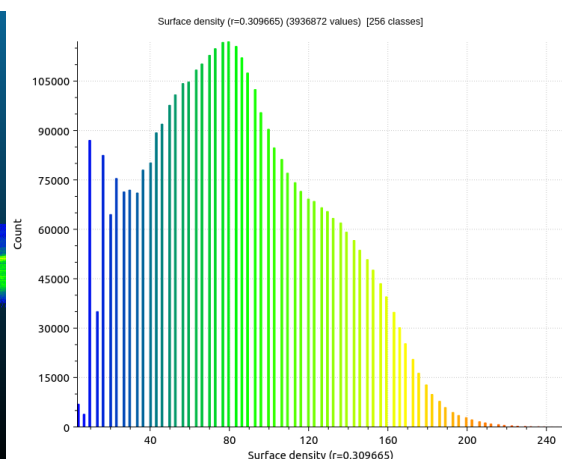
(d) Densidade gerada por MSCURL.



(e) Distribuição Gaussiana de densidade.



(f) Densidade gerada por CURL.

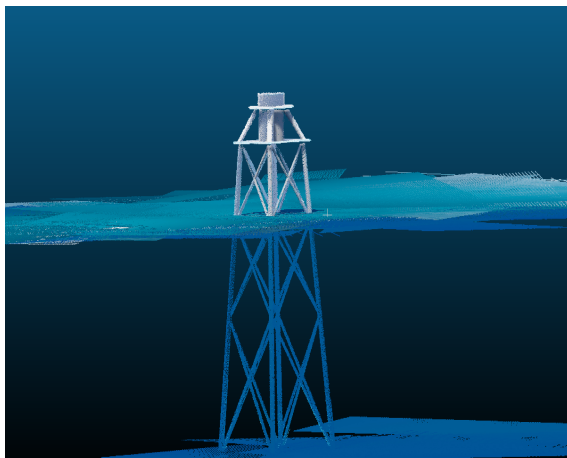


(g) Distribuição Gaussiana de densidade.

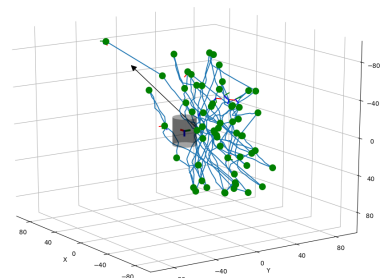
Figura 32: Densidade de pontos da estrutura: Eólica semi-submersível

Fonte: O autor

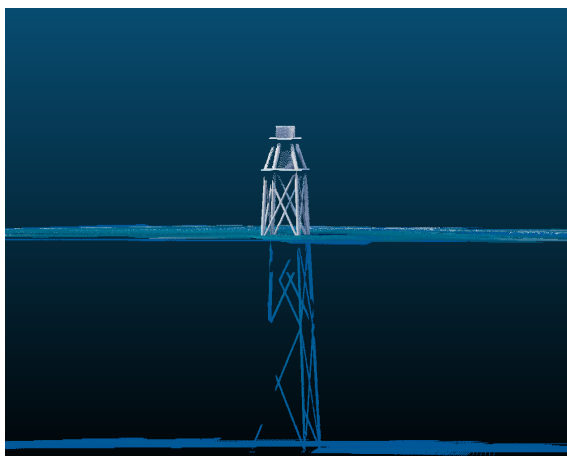
5.3.3.3 Ambiente 3 - Jacket



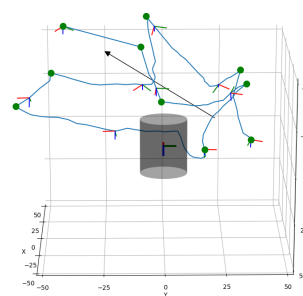
(b) Representação gerada por XPOCURL.



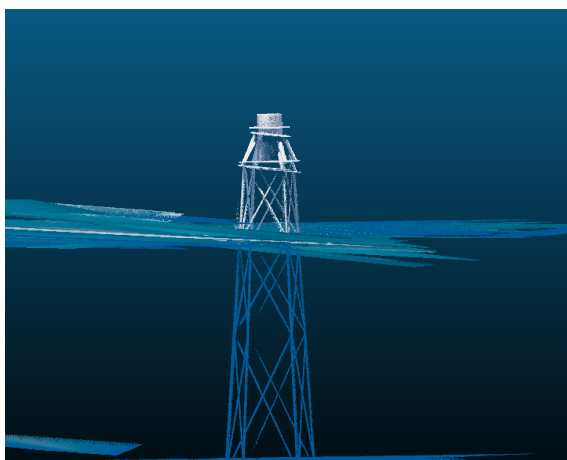
(c) Caminho gerado por XPOCURL.



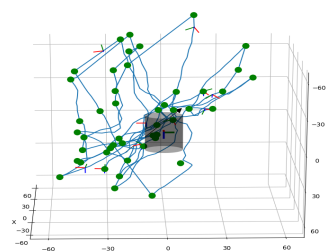
(d) Representação gerada por MSCURL.



(e) Caminho gerado por MSCURL.



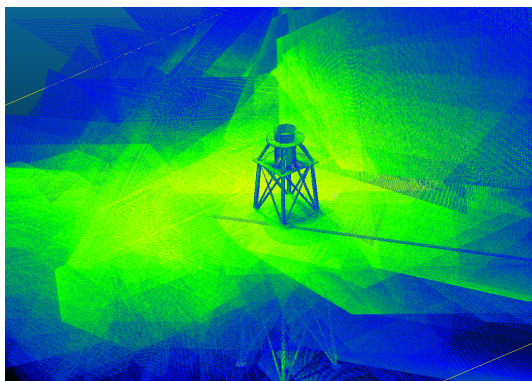
(f) Representação gerada por CURL.



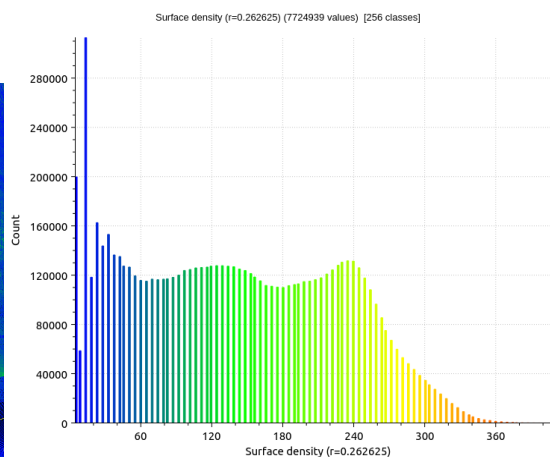
(g) Caminho gerado por CURL.

Figura 33: Representações tri-dimensionais da estrutura: *Jacket*

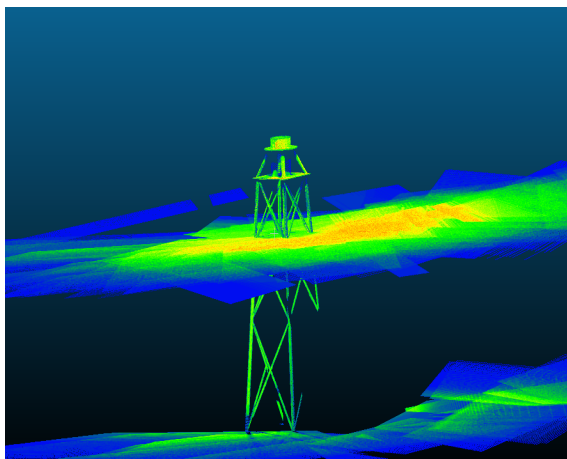
Fonte: O autor



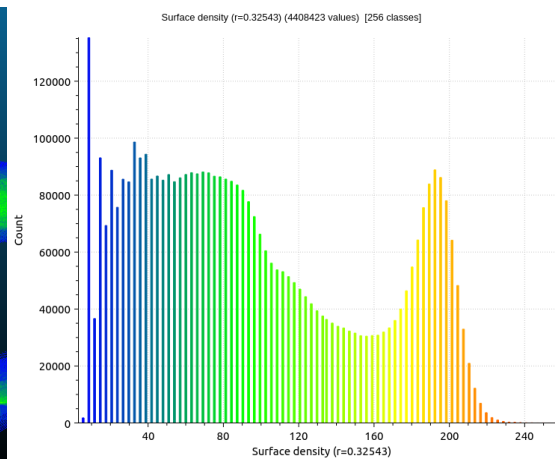
(b) Densidade gerada por XPOCURL.



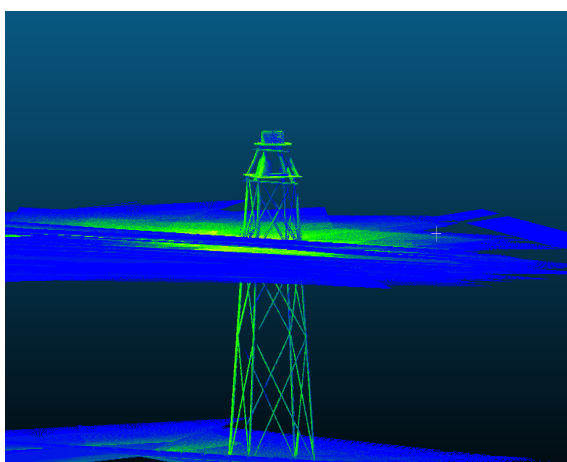
(c) Distribuição Gaussiana de densidade.



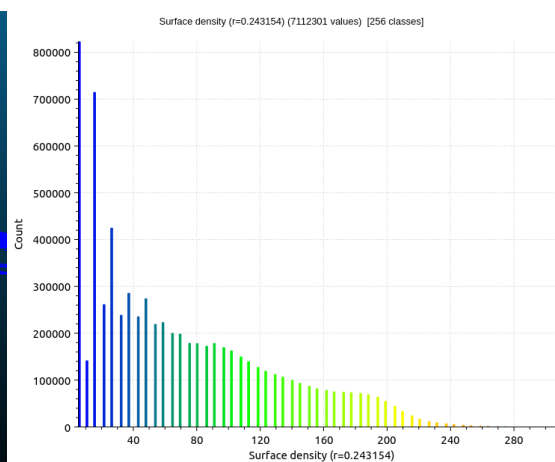
(d) Densidade gerada por MSCURL.



(e) Distribuição Gaussiana de densidade.



(f) Densidade gerada por CURL.

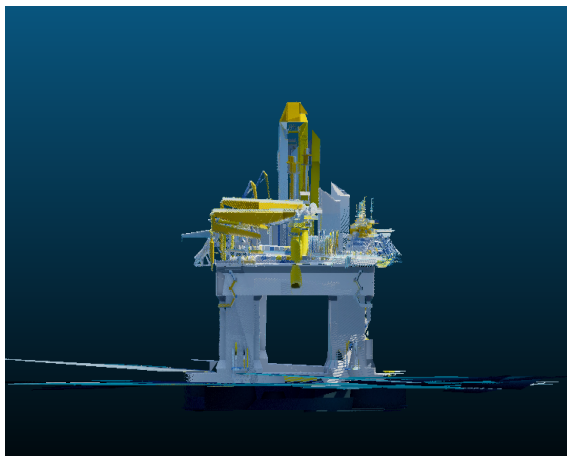


(g) Distribuição Gaussiana de densidade.

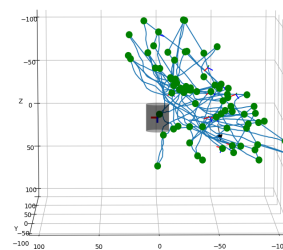
Figura 34: Densidade de pontos da estrutura: *Jacket*

Fonte: O autor

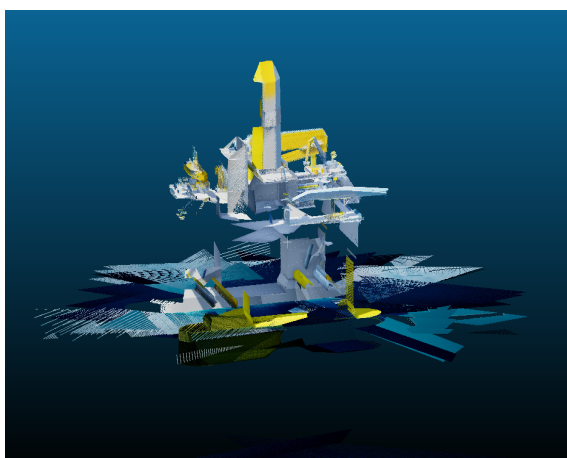
5.3.3.4 Ambiente 4 - Modelo de Plataforma Semi-submersível



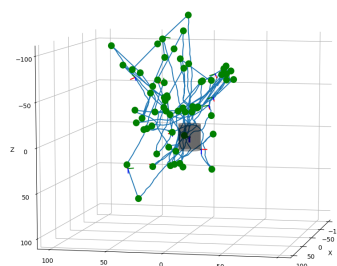
(b) Representação gerada por XPOCURL.



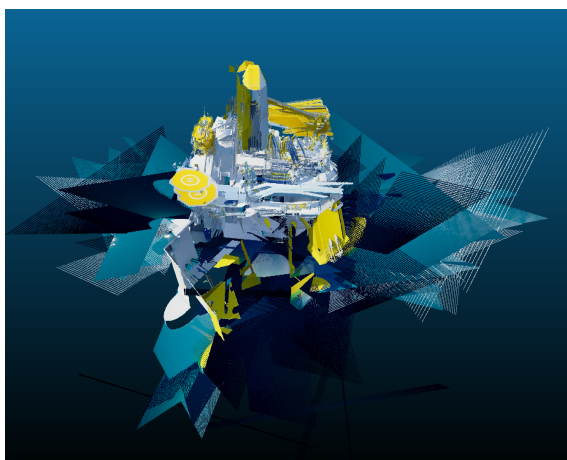
(c) Caminho gerado por XPOCURL.



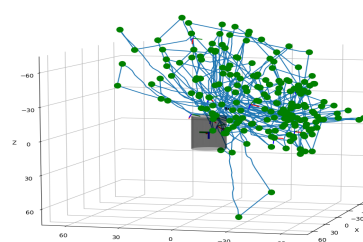
(d) Representação gerada por MSCURL.



(e) Caminho gerado por MSCURL.



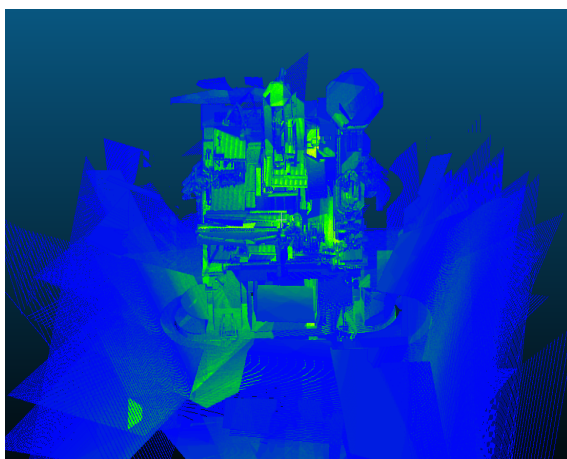
(f) Representação gerada por CURL.



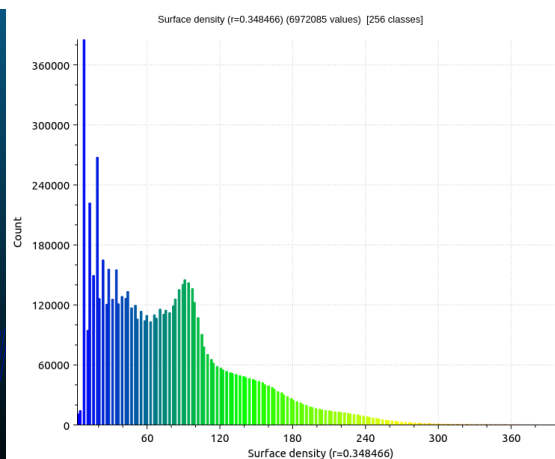
(g) Caminho gerado por CURL.

Figura 35: Representações tri-dimensionais da estrutura: Plataforma Semi-submersível.

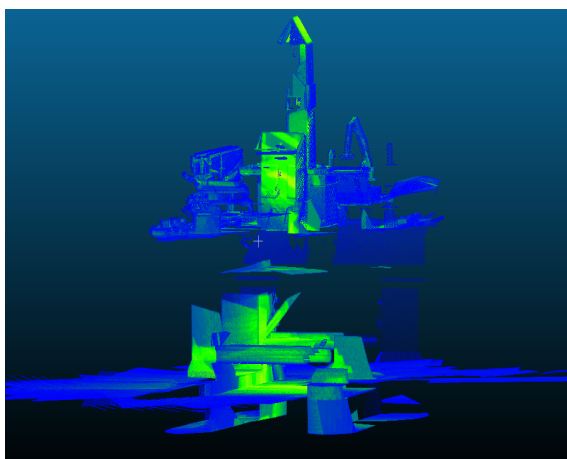
Fonte: O autor



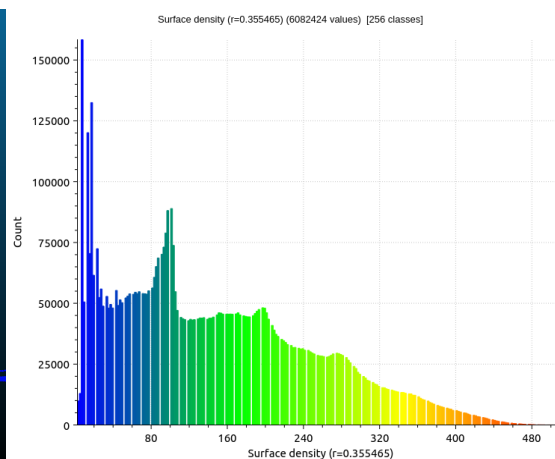
(b) Densidade gerada por XPOCURL.



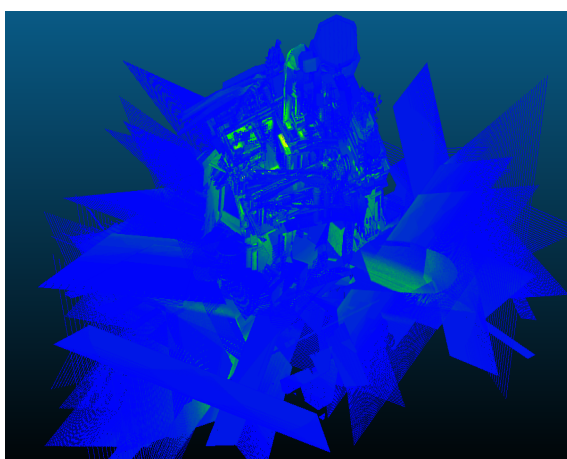
(c) Distribuição Gaussiana de densidade.



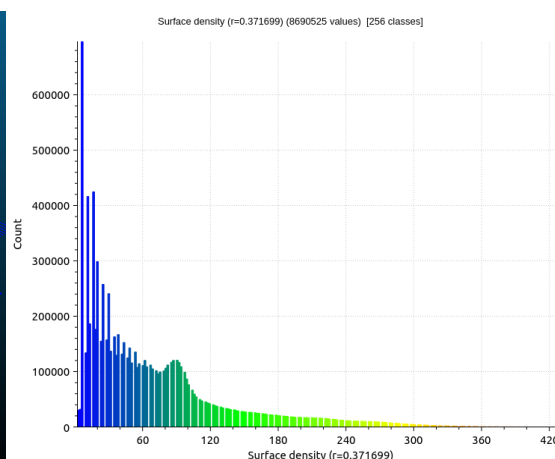
(d) Densidade gerada por MSCURL.



(e) Distribuição Gaussiana de densidade.



(f) Densidade gerada por CURL.



(g) Distribuição Gaussiana de densidade.

Figura 36: Densidade de pontos da estrutura: Plataforma Semi-submersível.

Fonte: O autor

5.3.4 Discussões

Para avaliação dos agentes, considerando o diferente número de informações utilizados por cada agente, alcançou-se tempos computacionais distintos. Avaliando o experimento em multi-domínio, o agente CURL levou aproximadamente 48h para finalização do treinamento. Com a adição da ramificação multi-domínio no bloco contrastivo através da imagem de profundidade, caracterizando o agente MSCURL, foram necessárias aproximadamente 56h para finalização, já o agente XPOCURL demandou aproximadamente 80h para completar o treino. Por fim, o agente Gen-NBV levou aproximadamente 56h para finalização do treinamento. Assim, considerando que os agentes CURL, MSCURL e XPOCURL utilizaram apenas imagem RGB durante a inferência, obteve-se aproximadamente um tempo de 35.71 milissegundos para todos os agentes. Além disso, o agente Gen-NBV, que utiliza todas as informações também durante a inferência, alcançou um tempo de aproximadamente 66.67 milissegundos.

Com relação as coberturas efetuadas, foram alcançados bons resultados de reconstrução mesmo nos ambientes mais complexos com o XPOCURL, alcançando malhas com um nível considerável de detalhamento. Para ilustrar a qualidade das reconstruções de maneira quantificada, foram construídos gráficos que mensuram a densidade dos pontos coletados ao longo do caminho. Dessa forma, foi possível avaliar as regiões que tiveram maior atenção ao longo do planejamento. Além disso, pode-se observar algumas características dos agentes com relação a generalização de comportamento.

Uma delas é com relação ao alinhamento entre as vistas geradas e as necessidades do algoritmo de reconstrução. Para a geração das vistas, pontos com relação ao planejamento como distância, cobertura e eficiência, devem ser considerados. Porém, assumindo que o agente pode errar ao longo do caminho, isso pode causar uma inconsistência na reconstrução através do RTABMAP. Isso é particularmente visível no último, e mais complexo, ambiente. E acontece devido a questões de alinhamento de características, que são dificultados por poses subsequentes com uma grande discrepância de deslocamento. Mas isso ocorre principalmente com mudanças abruptas de orientação, algo que acontece devido à união do planejamento de NBV e de caminho.

Avaliando as ações dos agentes em cada um dos ambientes, percebeu-se que a distribuição dos pontos coletados pelas vistas geradas pelo XPOCURL tenderam a criar uma região em torno da estrutura atualizada de ambos os domínios. Isso ocasionou em uma concentração de pontos distribuída de forma a conseguir reconstruir a maior parte das estruturas com detalhamento. Essa característica é visível principalmente no último ambiente, o qual não conseguiu cobrir a estrutura de forma eficaz, onde o topo da plataforma e uma de suas faces tiveram maior enfoque, como é mostrado na Figura 37.

O oposto das características do XPOCURL acontece com os demais agentes. Isso é perceptível principalmente nos ambientes 1 e 2 no qual houve uma cobertura significativa. No ambiente 1, percebeu-se que mesmo tendo conseguido cobrir quase que comple-

tamente a região da estrutura, o agente CURL gerou uma dispersão pobre de pontos, onde em 100 dos pontos houveram muito poucos pontos vizinhos, ou seja, uma baixíssima densidade e consequentemente um detalhamento do mesmo nível.

De forma geral, o comportamento dos agentes CURL e MSCURL foram antagonistas quando houve certa discrepância nos resultados de reconstrução. Essa diferença deu-se pelo foco dos agentes ao longo do planejamento, onde o CURL priorizou mais a cobertura subaquática, ilustrado nas Figuras 38a) e 38b), como acontece no ambiente 2. Já o agente MSCURL, priorizou a cobertura no domínio aéreo. Isso é principalmente demonstrado nos ambientes 1 e 3, onde a reconstrução foi muito completa no domínio aéreo.

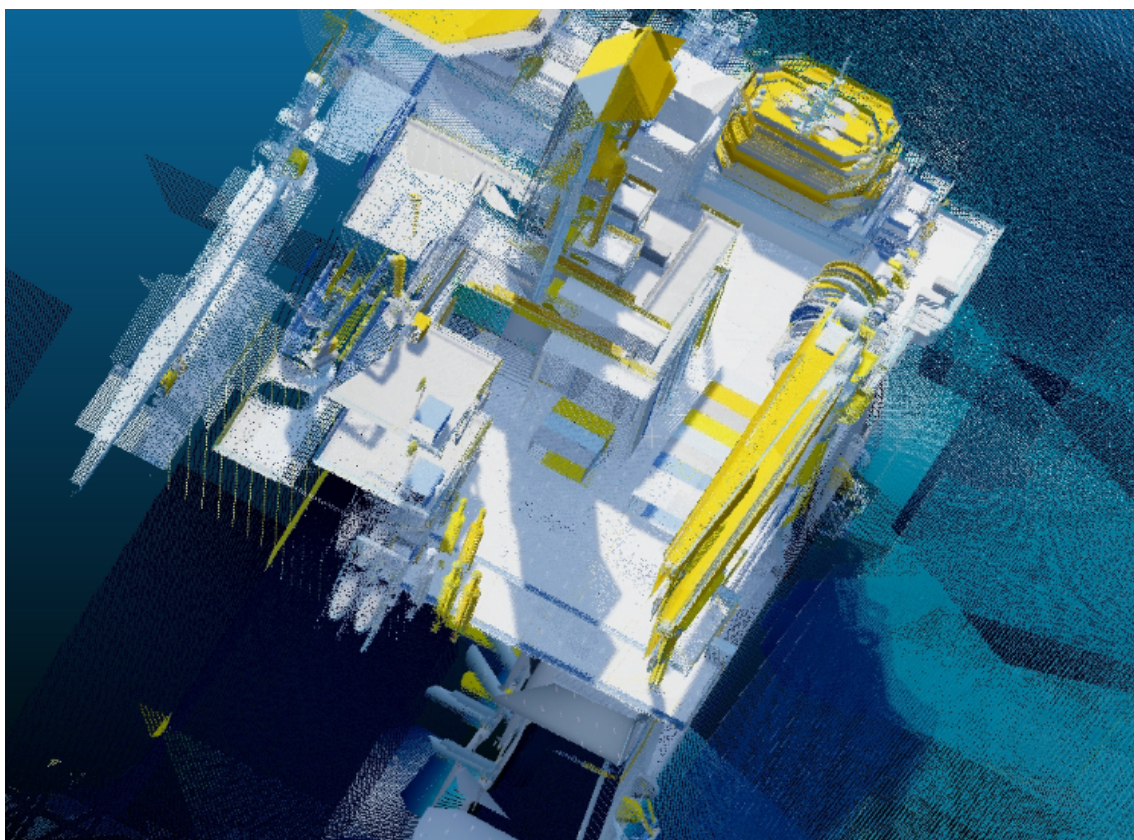
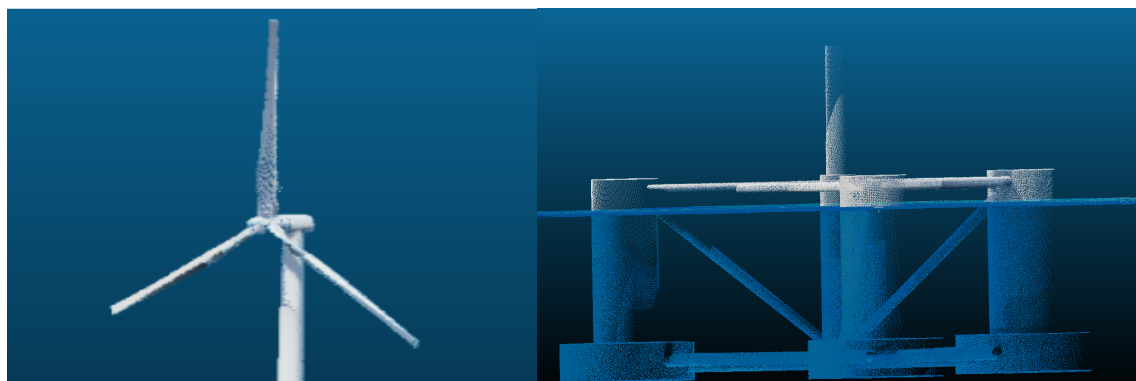


Figura 37: Visualização do detalhamento da reconstrução no ambiente 4 pelo agente XPOCURL.

O fato de o agente XPOCURL ter efetuado uma cobertura considerável em 4 dos 5 ambientes de avaliação, é um reflexo da capacidade de melhoria dos algoritmos de DRL pelas representações contrastivas. Esse ponto pode ser ainda mais enfatizado quando leva-se em consideração que o agente aprendeu a fazer transições de maneira inteligente entre dois diferentes domínios considerando o mesmo tipo de sensor e informação. Além de estar aplicado à uma tarefa de grande complexidade e considerar um ambiente que traz desafios que serão encontrados no mundo real. Do ponto de vista de HUAUVs a proposta de um planejador cujo a tomada de decisão leva em consideração o multi-domínio,

colabora para o desenvolvimento de aplicações, já visando o uso dessa classe de veículo.



(a) Densidade gerada por XPOCURL.

(b) Recorte de reconstrução MSCURL.

Figura 38: Recorte de reconstrução CURL.

6 CONCLUSÃO

Este trabalho teve como objetivo desenvolver um sistema capaz de planejar as próximas melhores vistas para a cobertura de estruturas em multi-domínio encontradas em ambientes *offshore*, considerando a movimentação apresentada por um HUAUV. As abordagens propostas e outros dois métodos foram desenvolvidos e aplicados no contexto de planejamento de NBV em multi-domínio. Essas etapas iniciais foram realizadas em um ambiente simulado e os resultados indicam que as redes propostas XPOCURL e MSCURL apresentam melhores resultados quando comparadas com outras abordagens que visam aumentar a eficácia dos métodos de aprendizado por reforço com informação complexa.

Como já foi demonstrado por de Jesus et al. [24], abordagens com representações contrastivas para aprendizado por reforço profundo são eficazes para a efetuação de tarefas complexas em ambientes multi-domínio. No entanto, neste trabalho, foi investigado a incrementação de dados de observação ricos em informação, como é o caso da imagem de profundidade, semântica e nuvem de pontos. A proposta deste trabalho é investigar o uso de imagens para o planejamento de ponto de vista e cobertura de estruturas com o uso de veículos híbridos, tendo em vista que redes de aprendizado por reforço são ineficientes em termos de amostra com observações complexas [24]. A abordagem MSCURL utiliza imagens RGB e de profundidade, aprendizado multi-modal, para aprender a política de planejamento, em seguida, utiliza apenas a informação RGB para efetuar o planejamento quando inferido. A abordagem XPOCURL utiliza imagens RGB, profundidade, semântica e nuvem de pontos, aprendizado *cross-modal*, para aprender a política e também utiliza apenas a imagens RGB quando inferido.

Os resultados apresentados no capítulo anterior mostra que o método XPOCURL é capaz de aprender de forma generalista a planejar vistas para efetuar a cobertura de estruturas em multi-domínio de maneira eficiente. A proposta mostrou-se superior às demais, demonstrando que o uso de informações heterogêneas de maneira contrastiva implica em resultados promissores para tarefas complexas. Assim, a principal contribuição deste trabalho foi desenvolver algoritmos capazes de aprender com informações visuais e nuvens de pontos, e verificou-se que o uso de representações contrastivas heterogêneas resulta

em um desempenho superior quando comparada com outras abordagens. No entanto, há uma problemática com relação ao número de vistas, o qual ainda pode ser otimizado para tornar o caminho final mais eficiente. Para continuar o desenvolvimento deste trabalho, propõe-se investigar essa problemática e explorar outras técnicas que possam auxiliar em resultados mais satisfatórios. Além disso, seria importante investigar esses algoritmos para ambientes reais, levando em consideração as diferenças entre a simulação e a realidade, e superando os desafios decorrentes da transferência de conhecimento entre esses ambientes.

A implementação dos algoritmos propostos podem ser encontradas no repositório: <https://github.com/mgmateus/xcurl>. Nesse repositório se encontra a implementação de MSCURL e XPOCURL.

REFERÊNCIAS

- [1] Zhiliang Ma and Shilong Liu. A review of 3d reconstruction techniques in civil engineering and their applications. *Advanced Engineering Informatics*, 37:163–174, 2018.
- [2] Roghieh Eskandari and Ali Motamedi. Diminished reality in architectural and environmental design: Literature review of techniques, applications, and challenges. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 38, pages 995–1001. IAARC Publications, 2021.
- [3] Wendy Flores-Fuentes, Gabriel Trujillo-Hernández, Iván Y Alba-Corpus, Julio C Rodríguez-Quinonez, Jesús E Mirada-Vega, Daniel Hernández-Balbuena, Fabian N Murrieta-Rico, and Oleg Sergiyenko. 3d spatial measurement for model reconstruction: A review. *Measurement*, 207:112321, 2023.
- [4] Su Yang, Miaole Hou, and Songnian Li. Three-dimensional point cloud semantic segmentation for cultural heritage: a comprehensive review. *Remote Sensing*, 15(3):548, 2023.
- [5] Huaquan Ying, Rafael Sacks, and Amir Degani. Synthetic image data generation using bim and computer graphics for building scene understanding. *Automation in Construction*, 154:105016, 2023.
- [6] Priya Tripathi, Seung Jae Lee, Chang Hoon Lee, and Moochul Shin. Towards 3d shape estimation from 2d particle images: A state-of-the-art review and demonstration. *KONA Powder and Particle Journal*, page 2025017, 2024.
- [7] Bing Han and Fernanda Leite. Model requirements for the development of extended reality applications in construction projects: A literature review. In *Construction Research Congress 2022*, pages 987–994, 2022.
- [8] Anshan Zhang, Jian Yang, and Feiliang Wang. Application and enabling technologies of digital twin in operation and maintenance stage of the aec industry: A literature review. *Journal of Building Engineering*, page 107859, 2023.

- [9] Jinghai Xu, Xin Shu, Peng Qiao, Shanshan Li, and Jigang Xu. Developing a digital twin model for monitoring building structural health by combining a building information model and a real-scene 3d model. *Measurement*, 217:112955, 2023.
- [10] Luís Eduardo Ramos De Carvalho and Aldo Von Wangenheim. Literature review for 3d object classification/recognition. *Pattern Anal. Appl.*, 22:1243–1292, 2017.
- [11] Sumantra Dutta Roy, Santanu Chaudhury, and Subhashis Banerjee. Active recognition through next view planning: a survey. *Pattern Recognition*, 37(3):429–446, 2004.
- [12] Uğur Murat Erdem and Stan Sclaroff. Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements. *Computer Vision and Image Understanding*, 103(3):156–169, 2006.
- [13] Aaron Mavrinac and Xiang Chen. Modeling coverage in camera networks: A survey. *International journal of computer vision*, 101:205–226, 2013.
- [14] Rui Zeng, Yuhui Wen, Wang Zhao, and Yong-Jin Liu. View planning in robot active vision: A survey of systems, algorithms, and applications. *Computational Visual Media*, 6:225–245, 2020.
- [15] William R Scott, Gerhard Roth, and Jean-François Rivest. View planning for automated three-dimensional object reconstruction and inspection. *ACM Computing Surveys (CSUR)*, 35(1):64–96, 2003.
- [16] Tarek Rakha and Alice Gorodetsky. Review of unmanned aerial system (uas) applications in the built environment: Towards automated building inspection procedures using drones. *Automation in construction*, 93:252–264, 2018.
- [17] Olakunle Elijah, Pang Ai Ling, Sharul Kamal Abdul Rahim, Tan Kim Geok, Agus Arsad, Evizal Abdul Kadir, Muslim Abdurrahman, Radzuan Junin, Augustine Agi, and Mohammad Yasin Abdulfatah. A survey on industry 4.0 for the oil and gas industry: upstream sector. *IEEE Access*, 9:144438–144468, 2021.
- [18] Ruishu F Wright, Ping Lu, Jagannath Devkota, Fei Lu, Margaret Ziomek-Moroz, and Paul R Ohodnicki Jr. Corrosion sensors for structural health monitoring of oil and natural gas infrastructure: A review. *Sensors*, 19(18):3964, 2019.
- [19] Vahid Aryai, Rouzbeh Abbassi, Nagi Abdussamie, Fatemeh Salehi, Vikram Garaniya, Mohsen Asadnia, Al-Amin Baksh, Irene Penesis, Hassan Karampour, Scott Draper, et al. Reliability of multi-purpose offshore-facilities: Present status and future direction in australia. *Process Safety and Environmental Protection*, 148: 437–461, 2021.

- [20] Paulo LJ Drews, Armando Alves Neto, and Mario FM Campos. Hybrid unmanned aerial underwater vehicle: Modeling and simulation. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4637–4642. IEEE, 2014.
- [21] Alexandre C Horn, Pedro M Pinheiro, Ricardo B Grando, Cesar B da Silva, Armando A Neto, and Paulo LJ Drews. A novel concept for hybrid unmanned aerial underwater vehicles focused on aquatic performance. In *2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE)*, pages 1–6. IEEE, 2020.
- [22] Ricardo B Grando, Junior C de Jesus, Victor A Kich, Alisson H Kolling, Nicolas P Bortoluzzi, Pedro M Pinheiro, Armando A Neto, and Paulo LJ Drews. Deep reinforcement learning for mapless navigation of a hybrid aerial underwater vehicle with medium transition. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1088–1094. IEEE, 2021.
- [23] Pedro M Pinheiro, Armando A Neto, Ricardo B Grando, César B da Silva, Vivian M Aoki, Dayana S Cardoso, Alexandre C Horn, and Paulo LJ Drews Jr. Trajectory planning for hybrid unmanned aerial underwater vehicles with smooth media transition. *Journal of Intelligent & Robotic Systems*, 104(3):46, 2022.
- [24] Junior C de Jesus, Victor A Kich, Alisson H Kolling, Ricardo B Grando, Rodrigo S Guerra, and Paulo LJ Drews. Depth-cuprl: Depth-imaged contrastive unsupervised prioritized representations in reinforcement learning for mapless navigation of unmanned aerial vehicles. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10579–10586. IEEE, 2022.
- [25] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.
- [26] Ryumei Nakada, Halil Ibrahim Gulluk, Zhun Deng, Wenlong Ji, James Zou, and Linjun Zhang. Understanding multimodal contrastive learning and incorporating unpaired data. In *International Conference on Artificial Intelligence and Statistics*, pages 4348–4380. PMLR, 2023.
- [27] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pages 5639–5650. PMLR, 2020.
- [28] Burak Uzkent, Christopher Yeh, and Stefano Ermon. Efficient object detection in large images using deep reinforcement learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1824–1833, 2020.

- [29] Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10674–10681, 2021.
- [30] Xin Yuan, Zhe Lin, Jason Kuen, Jianming Zhang, Yilin Wang, Michael Maire, Ajinkya Kale, and Baldo Faieta. Multimodal contrastive training for visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6995–7004, 2021.
- [31] Xiaoshi Wu, Hadar Averbuch-Elor, Jin Sun, and Noah Snavely. Towers of babel: Combining images, language, and 3d geometry for learning multimodal vision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 428–437, 2021.
- [32] Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9902–9912, 2022.
- [33] Basil Mustafa, Carlos Riquelme, Joan Puigcerver, Rodolphe Jenatton, and Neil Houlsby. Multimodal contrastive learning with limoe: the language-image mixture of experts. *Advances in Neural Information Processing Systems*, 35:9564–9576, 2022.
- [34] Miaoran Zhang, Marius Mosbach, David Ifeoluwa Adelani, Michael A Hedderich, and Dietrich Klakow. Mcse: Multimodal contrastive learning of sentence embeddings. *arXiv preprint arXiv:2204.10931*, 2022.
- [35] Nenglu Chen, Lei Chu, Hao Pan, Yan Lu, and Wenping Wang. Self-supervised image representation learning with geometric set consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19292–19302, 2022.
- [36] Kyota Higa, Masahiro Yamaguchi, and Toshinori Hosoi. Iccl: Self-supervised intra-and cross-modal contrastive learning with 2d-3d pairs for 3d scene understanding. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 1085–1089. IEEE, 2023.
- [37] Anas Mahmoud, Jordan SK Hu, Tianshu Kuai, Ali Harakeh, Liam Paull, and Steven L Waslander. Self-supervised image-to-point distillation via semantically to-

- lerant contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7102–7110, 2023.
- [38] Ye Wang, Bowei Jiang, Changqing Zou, and Rui Ma. Mxm-clr: A unified framework for contrastive learning of multifold cross-modal representations. *arXiv preprint arXiv:2303.10839*, 2023.
- [39] Parisa Maroufkhani, Kevin C Desouza, Robert K Perrons, and Mohammad Iranmanesh. Digital transformation in the resource and energy sectors: A systematic review. *Resources Policy*, 76:102622, 2022.
- [40] Hui Sheng Lim, Andrew Filisetti, Andreas Marouchos, Kasra Khosoussi, and Nicholas Lawrance. Applied research directions of autonomous marine systems for environmental monitoring. *OCEANS 2023-Limerick*, pages 1–10, 2023.
- [41] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. *arXiv preprint arXiv:2104.13478*, 2021.
- [42] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024.
- [43] J Irving Vasquez-Gomez, L Enrique Sucar, Rafael Murrieta-Cid, and Efrain Lopez-Damian. Volumetric next-best-view planning for 3d object reconstruction with positioning error. *International Journal of Advanced Robotic Systems*, 11(10):159, 2014.
- [44] Christian Potthast and Gaurav S Sukhatme. A probabilistic framework for next best view estimation in a cluttered environment. *Journal of Visual Communication and Image Representation*, 25(1):148–164, 2014.
- [45] Simon Kriegel, Christian Rink, Tim Bodenmüller, and Michael Suppa. Efficient next-best-scan planning for autonomous 3d surface reconstruction of unknown objects. *Journal of Real-Time Image Processing*, 10:611–631, 2015.
- [46] Stefan Isler, Reza Sabzevari, Jeffrey Delmerico, and Davide Scaramuzza. An information gain formulation for active volumetric 3d reconstruction. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3477–3484. IEEE, 2016.

- [47] Christopher Lehnert, Andrew English, Christopher McCool, Adam W Tow, and Tristan Perez. Autonomous sweet pepper harvesting for protected cropping systems. *IEEE Robotics and Automation Letters*, 2(2):872–879, 2017.
- [48] Edwin T Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.
- [49] Xiao Chen, Quanyi Li, Tai Wang, Tianfan Xue, and Jiangmiao Pang. Gennbv: Generalizable next-best-view policy for active 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16436–16445, 2024.
- [50] Siddhant Panigrahi, Vaibhav Ashok, Vijay Kumar Pediredla, Thiyagarajan Ranganathan, and Asokan Thondiyath. Mathematical modelling and control of a submersible multi-medium uav. In *OCEANS 2022-Chennai*, pages 1–9. IEEE, 2022.
- [51] Adir A Pedroso, Andressa C da Silva, Pedro M Pinheiro, and Paulo LJ Drews. Prototyping and construction of a hybrid unmanned aerial underwater vehicles. In *2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE)*, pages 61–66. IEEE, 2022.
- [52] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
- [53] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.
- [54] Yoshiaki Kuwata, Justin Teo, Gaston Fiore, Sertac Karaman, Emilio Frazzoli, and Jonathan P How. Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on control systems technology*, 17(5):1105–1118, 2009.
- [55] Steven M LaValle and James J Kuffner Jr. Randomized kinodynamic planning. *The international journal of robotics research*, 20(5):378–400, 2001.
- [56] Jingqing Han. From pid to active disturbance rejection control. *IEEE transactions on Industrial Electronics*, 56(3):900–906, 2009.
- [57] Zheng Zeng, Chengke Xiong, Xinyi Yuan, Hexiong Zhou, Yuling Bai, Yufei Jin, Di Lu, and Lian Lian. Information-driven path planning for hybrid aerial underwater vehicles. *IEEE Journal of Oceanic Engineering*, 2023.

- [58] Di Lu, Yinghao Guo, Chengke Xiong, Zheng Zeng, and Lian Lian. Takeoff and landing control of a hybrid aerial underwater vehicle on disturbed water's surface. *IEEE Journal of Oceanic Engineering*, 47(2):295–311, 2021.
- [59] Benjamin Hepp, Matthias Nießner, and Otmar Hilliges. Plan3d: Viewpoint and trajectory optimization for aerial multi-view stereo reconstruction. *ACM Transactions on Graphics (TOG)*, 38(1):1–17, 2018.
- [60] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2341–2353, 2010.
- [61] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous robots*, 34:189–206, 2013.
- [62] John Ashburner and Karl J Friston. Voxel-based morphometry—the methods. *Neuroimage*, 11(6):805–821, 2000.
- [63] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part III 14*, pages 501–518. Springer, 2016.
- [64] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [65] Randa Almadhoun, Abdullah Abduldayem, Tarek Taha, Lakmal Seneviratne, and Yahya Zweiri. Guided next best view for 3d reconstruction of large complex structures. *Remote Sensing*, 11(20):2440, 2019.
- [66] Abdullah Abduldayem, Dongming Gan, Lakmal Seneviratne, and Tarek Taha. 3d reconstruction of complex structures with online profiling and adaptive viewpoint sampling. In *Proceedings of the International Micro Air Vehicle Conference and Flight Competition, Malaga, Spain*, pages 14–16, 2017.
- [67] Poornima Ramesh, Jan-Matthis Lueckmann, Jan Boelts, Álvaro Tejero-Cantero, David S Greenberg, Pedro J Gonçalves, and Jakob H Macke. Gatsbi: Generative adversarial training for simulation-based inference. *arXiv preprint arXiv:2203.06481*, 2022.
- [68] Stephen L Smith and Frank Imeson. Glns: An effective large neighborhood search heuristic for the generalized traveling salesman problem. *Computers & Operations Research*, 87:1–19, 2017.

- [69] Zhexiong Shang and Zhigang Shen. Topology-based uav path planning for multi-view stereo 3d reconstruction of complex structures. *Complex & Intelligent Systems*, 9(1):909–926, 2023.
- [70] Keld Helsgaun. An extension of the lin-kernighan-helsgaun tsp solver for constrained traveling salesman and vehicle routing problems. *Roskilde: Roskilde University*, 12, 2017.
- [71] Stefan Gottschalk, Ming C Lin, and Dinesh Manocha. Obbtree: A hierarchical structure for rapid interference detection. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 171–180, 1996.
- [72] Jonathan D Gammell, Siddhartha S Srinivasa, and Timothy D Barfoot. Informed rrt: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In *2014 IEEE/RSJ international conference on intelligent robots and systems*, pages 2997–3004. IEEE, 2014.
- [73] Benjamin Charrow, Gregory Kahn, Sachin Patil, Sikang Liu, Ken Goldberg, Pieter Abbeel, Nathan Michael, and Vijay Kumar. Information-theoretic planning with trajectory optimization for dense 3d mapping. In *Robotics: Science and Systems*, volume 11, pages 3–12. Rome, 2015.
- [74] Benjamin Charrow, Sikang Liu, Vijay Kumar, and Nathan Michael. Information-theoretic mapping using cauchy-schwarz quadratic mutual information. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4791–4798. IEEE, 2015.
- [75] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- [76] Jose C Principe. *Information theoretic learning: Renyi’s entropy and kernel perspectives*. Springer Science & Business Media, 2010.
- [77] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.
- [78] Wei Jing, Joseph Polden, Pey Yuen Tao, Wei Lin, and Kenji Shimada. View planning for 3d shape reconstruction of buildings with unmanned aerial vehicles. In *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1–6. IEEE, 2016.
- [79] Kenji Shimada and David C Gossard. Bubble mesh: automated triangular meshing of non-manifold geometry by sphere packing. In *Proceedings of the third ACM symposium on Solid modeling and applications*, pages 409–419, 1995.

- [80] Wei Jing, Joseph Polden, Wei Lin, and Kenji Shimada. Sampling-based view planning for 3d visual coverage task with unmanned aerial vehicle. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1808–1815. IEEE, 2016.
- [81] Robert M Haralick, Stanley R Sternberg, and Xinhua Zhuang. Image analysis using mathematical morphology. *IEEE transactions on pattern analysis and machine intelligence*, (4):532–550, 1987.
- [82] Johann Borenstein and Yoram Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on systems, Man, and Cybernetics*, 19(5):1179–1187, 1989.
- [83] Harnaik Dharmi, Vishnu D Sharma, and Pratap Tokekar. Pred-nbv: Prediction-guided next-best-view for 3d object reconstruction. *arXiv preprint arXiv:2304.11465*, 2023.
- [84] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. PointR: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12498–12507, 2021.
- [85] James J Kuffner and Steven M LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 2, pages 995–1001. IEEE, 2000.
- [86] Guillaume Hardouin, Julien Moras, Fabio Morbidi, Julien Marzat, and El Mustapha Mouaddib. Next-best-view planning for surface reconstruction of large-scale 3d environments with multiple uavs. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1567–1574. IEEE, 2020.
- [87] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996.
- [88] Zehui Meng, Hailong Qin, Ziyue Chen, Xudong Chen, Hao Sun, Feng Lin, and Marcelo H Ang. A two-stage optimized next-view planning framework for 3-d unknown environment exploration, and structural reconstruction. *IEEE Robotics and Automation Letters*, 2(3):1680–1687, 2017.
- [89] Norman Biggs. *Algebraic graph theory*. Number 67. Cambridge university press, 1993.

- [90] Keld Helsgaun. An effective implementation of the lin–kernighan traveling salesman heuristic. *European journal of operational research*, 126(1):106–130, 2000.
- [91] Mina Kamel, Thomas Stastny, Kostas Alexis, and Roland Siegwart. Model predictive control for trajectory tracking of unmanned aerial vehicles using robot operating system. *Robot Operating System (ROS) The Complete Reference (Volume 2)*, pages 3–39, 2017.
- [92] Robert Bohlin and Lydia E Kavraki. Path planning using lazy prm. In *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065)*, volume 1, pages 521–528. IEEE, 2000.
- [93] Chen Feng, Haojia Li, Fei Gao, Boyu Zhou, and Shaojie Shen. Predrecon: A prediction-boosted planning framework for fast and high-quality autonomous aerial reconstruction. *arXiv preprint arXiv:2302.04488*, 2023.
- [94] Robert Bridson. Fast poisson disk sampling in arbitrary dimensions. *SIGGRAPH sketches*, 10(1):1, 2007.
- [95] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- [96] Antoine Guédon, Tom Monnier, Pascal Monasse, and Vincent Lepetit. Macarons: Mapping and coverage anticipation with rgb online self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 940–951, 2023.
- [97] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [98] Antoine Guédon, Pascal Monasse, and Vincent Lepetit. Scone: Surface coverage optimization in unknown environments by volumetric integration. *Advances in Neural Information Processing Systems*, 35:20731–20743, 2022.
- [99] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [100] Daryl Peralta, Joel Casimiro, Aldrin Michael Nilles, Justine Aletta Aguilar, Rowel Atienza, and Rhandley Cajote. Next-best view policy for 3d reconstruction. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 558–573. Springer, 2020.

- [101] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 803–814, 2023.
- [102] James A Sethian. A fast marching level set method for monotonically advancing fronts. *proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.
- [103] Geoffrey A Hollinger, Brendan Englot, Franz S Hover, Urbashi Mitra, and Gaurav S Sukhatme. Active planning for underwater inspection and the benefit of adaptivity. *The International Journal of Robotics Research*, 32(1):3–18, 2013.
- [104] Abraham Wald. Sequential tests of statistical hypotheses. In *Breakthroughs in statistics: Foundations and basic theory*, pages 256–298. Springer, 1992.
- [105] Brendan Englot and Franz Hover. Planning complex inspection tasks using redundant roadmaps. In *Robotics Research: The 15th International Symposium ISRR*, pages 327–343. Springer, 2017.
- [106] Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. 1976.
- [107] William J Cook, David L Applegate, Robert E Bixby, and Vasek Chvatal. *The traveling salesman problem: a computational study*. Princeton university press, 2011.
- [108] Mark Sheinin and Yoav Y Schechner. The next best underwater view. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3764–3773, 2016.
- [109] Kenneth H Norwich. *Information, sensation, and perception*. Academic Press, 1993.
- [110] Amy Lewis, Kolton Yager, Mitchell Keller, Bonita Galvan, Russell C Bingham, Samantha Ting, Jane Wu, Timmy Gambin, Christopher M Clark, and Zoë J Wood. Virtual planning and testing of auv paths for underwater photogrammetry. In *VISI-GRAPP (1: GRAPP)*, pages 93–101, 2020.
- [111] Jane Wu, Russell C Bingham, Samantha Ting, Kolton Yager, Zoë J Wood, Timmy Gambin, and Christopher M Clark. Multi-auv motion planning for archeological site mapping and photogrammetric reconstruction. *Journal of Field Robotics*, 36(7):1250–1269, 2019.

- [112] Lucia K Dale and Nancy M Amato. Probabilistic roadmaps-putting it all together. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 2, pages 1940–1947. IEEE, 2001.
- [113] Eduard Vidal, Narcís Palomeras, Klemen Istenič, Nuno Gracias, and Marc Carreras. Multisensor online 3d view planning for autonomous underwater exploration. *Journal of Field Robotics*, 37(6):1123–1147, 2020.
- [114] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.
- [115] Thor I Fossen, Morten Breivik, and Roger Skjetne. Line-of-sight path following of underactuated marine craft. *IFAC proceedings volumes*, 36(21):211–216, 2003.
- [116] Leonardo Zacchini, Alessandro Ridolfi, and Benedetto Allotta. Informed expansion for informative path planning via online distribution learning. *Robotics and Autonomous Systems*, 166:104449, 2023.
- [117] Berwin A Turlach et al. Bandwidth selection in kernel density estimation: a review. Technical report, Humboldt Universitaet Berlin, 1993.
- [118] Geoffrey A Hollinger and Gaurav S Sukhatme. Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research*, 33(9):1271–1287, 2014.
- [119] Bruce Naylor. Binary space partitioning trees as an alternative representation of polytopes. *Computer-Aided Design*, 22(4):250–252, 1990.
- [120] Selim Benhimane and Ezio Malis. Homography-based 2d visual tracking and servoing. *The International Journal of Robotics Research*, 26(7):661–676, 2007.
- [121] Runhua Wang, Xuebo Zhang, Yongchun Fang, and Baoquan Li. Virtual-goal-guided rrt for visual servoing of mobile robots with fov constraint. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(4):2073–2083, 2021.
- [122] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.
- [123] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G Bellemare, Joelle Pineau, et al. An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, 11(3-4):219–354, 2018.
- [124] Richard Bellman. A markovian decision process. *Journal of Mathematics and Mechanics*, pages 679–684, 1957.

- [125] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.
- [126] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
- [127] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.
- [128] George E Uhlenbeck and Leonard S Ornstein. On the theory of the brownian motion. *Physical review*, 36(5):823, 1930.
- [129] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.
- [130] Joshua Achiam. Spinning up in deep reinforcement learning, 2018.
- [131] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.
- [132] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [133] Gobinda G Chowdhury. *Introduction to modern information retrieval*. Facet publishing, 2010.
- [134] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016.
- [135] Sijie Mai, Ying Zeng, Shuangjia Zheng, and Haifeng Hu. Hybrid contrastive learning of tri-modal representation for multimodal sentiment analysis. *IEEE Transactions on Affective Computing*, 2022.
- [136] Tuomas Haarnoja, Kristian Hartikainen, Pieter Abbeel, and Sergey Levine. Latent space policies for hierarchical reinforcement learning. In *International Conference on Machine Learning*, pages 1851–1860. PMLR, 2018.
- [137] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

- [138] Cora A Dimmig, Giuseppe Silano, Kimberly McGuire, Chiara Gabellieri, Wolfgang Hönig, Joseph Moore, and Marin Kobilarov. Survey of simulators for aerial robots. *arXiv preprint arXiv:2311.02296*, 2023.
- [139] Easton Potokar, Spencer Ashford, Michael Kaess, and Joshua G Mangelson. Holocean: An underwater robotics simulator. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 3040–3046. IEEE, 2022.
- [140] Epic Games. Unreal engine. <https://www.unrealengine.com/pt-BR/>, 2004. Accessed: (2023-12-01).
- [141] Epic Games. Epic games. <https://store.epicgames.com/pt-BR/>, 2004. Accessed: (2023-12-01).
- [142] Shital Shah, Debadepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017. URL <https://arxiv.org/abs/1705.05065>.
- [143] Weichao Qiu, Fangwei Zhong, Yi Zhang, Siyuan Qiao, Zihao Xiao, Tae Soo Kim, and Yizhou Wang. Unrealcv: Virtual worlds for computer vision. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1221–1224, 2017.
- [144] Alban Denoyel, Cédric Pinson, and Pierre-Antoine Passet. Sketchfab. <https://sketchfab.com/>, 2012. Accessed: (2023-12-01).
- [145] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.
- [146] Morris Edgar Rose. *Elementary theory of angular momentum*. Courier Corporation, 1995.
- [147] Zhengyou Zhang. Determining the epipolar geometry and its uncertainty: A review. *International journal of computer vision*, 27:161–195, 1998.
- [148] William Rowan Hamilton. *Elements of quaternions*. Longmans, Green, & Company, 1866.
- [149] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021.
- [150] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

- [151] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [152] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.
- [153] Mathieu Labbé and François Michaud. Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of field robotics*, 36(2):416–446, 2019.
- [154] Docker Inc. Docker: Empowering App Development for Developers and Teams, 2024. URL <https://www.docker.com>. Acesso em: 11 out. 2024.