

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
CURSO DE ENGENHARIA DE COMPUTAÇÃO

DIEGO ALVIM STOCCHERO

**Estudo e Comparação de Algoritmos para  
Detecção de Pontos de Interesse Destinados  
à Fusão de Mapas de Grade de Ocupação  
Tridimensionais**

Monografia apresentada como requisito parcial  
para a obtenção do grau de Bacharel em  
Engenharia da Computação

Orientador: Prof. Dr. Edison Pignaton de Freitas

Porto Alegre  
2021

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões Mendes

Vice-Reitora: Prof<sup>ª</sup>. Patricia Helena Lucas Pranke

Pró-Reitor de Graduação: Prof<sup>ª</sup>. Cíntia Inês Boll

Diretora do Instituto de Informática: Prof<sup>ª</sup>. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Engenharia de Computação: Prof. Walter Fetter Lages

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*“If I have seen farther than others,  
it is because I stood on the shoulders of giants.”*

— SIR ISAAC NEWTON

## **AGRADECIMENTOS**

Agradeço ao L<sup>A</sup>T<sub>E</sub>X por não ter vírus de macro...

## RESUMO

Mapeamento usando múltiplos VANTs requer técnicas de fusão de mapas eficientes, que necessitam de métodos de detecção de pontos de interesse e de correspondência de características eficientes o suficiente para aplicações de tempo real e que funcionem com Mapas de Grade de Ocupação 3D. Este trabalho de graduação busca implementar extensões de métodos de detecção de pontos de interesse FAST, Harris e Shi-Tomasi para 3D e compará-los entre si, examinando os dados coletados para determinar as vantagens e desvantagens de cada um. A adaptação FAST3D apresentou os melhores resultados, considerando métricas de pontos de interesse relevantes detectados e tempo de execução.

**Palavras-chave:** Detecção de Pontos de Interesse Tridimensionais. Mapas de Grade de Ocupação. FAST. Harris. Shi-Tomasi.

# **Study and Comparison of Keypoint Detection Algorithms Destined for 3D Occupancy Grid Map Merging**

## **ABSTRACT**

Mapping using multiple drones require efficient map merging techniques, which require feature detection and feature matching methods efficient enough for real-time applications and work with 3D Occupancy Grid Maps. This graduation work aims to implement extensions to FAST, Harris and Shi-Tomasi feature detection methods for 3D applications and compare them, examining the collected data to determine the advantages and disadvantages of each one. FAST3D adaptation demonstrated the best results of the three in terms of relevant keypoints detected and execution time.

**Keywords:** 3D Keypoint Detection, Occupancy Grid Maps, FAST, Harris, Shi-Tomasi.

## **LISTA DE ABREVIATURAS E SIGLAS**

FAST	Features from Accelerated Segment Test
SIFT	Scale-Invariant Feature Transform
SURF	Speeded Up Robust Features
BRIEF	Binary Robust Independent Elementary Features
VANT	Veículo Aéreo Não Tripulado
SLAM	Simultaneous Localization and Mapping

## LISTA DE FIGURAS

Figura 2.1 Imagem (SHI; TOMASI, 1994) exemplo .....	15
Figura 2.2 Imagem (ROSTEN; DRUMMOND, 2006) representa o círculo de 16 pontos ao redor de p.....	16
Figura 3.1 Imagem (SIPIRAN; BUSTOS, 2011) exemplo de vizinhança ao redor de v.....	19
Figura 3.2 Imagem (MAES et al., 2010) exemplo de vizinhança.....	20
Figura 4.1 Mapas de grade de ocupação de entrada adquiridos por robôs diferentes antes do pré-processamento. ....	23
Figura 4.2 Mapas de grade de ocupação de entrada adquiridos por robôs diferentes depois do pré-processamento.....	23
Figura 5.1 Ambiente Labirinto.....	27
Figura 5.2 Mapas de grade de ocupação gerados pela simulação do ambiente Labirinto com dois VANTs. ....	28
Figura 5.3 Ambiente Livraria.....	28
Figura 5.4 Mapas de grade de ocupação gerados pela simulação do ambiente Livraria com VANTs 0, 1 e 2 em ordem. ....	29
Figura 5.5 Ambiente Hospital.....	30
Figura 5.6 Mapas de grade de ocupação gerados pela simulação do ambiente Hospital com VANTs 0, 1 e 2 em ordem. ....	30
Figura 5.7 Ambiente Pista de Corrida.....	31
Figura 5.8 Mapas de grade de ocupação gerados pela simulação do ambiente Pista de Corrida com VANTs 0 e 1 em ordem.....	31
Figura 5.9 Ambiente Casa.....	32
Figura 5.10 Mapas de grade de ocupação gerados pela simulação do ambiente Casa com VANTs 0, 1 e 2 em ordem.....	32
Figura 5.11 Ambiente Armazém.....	33
Figura 5.12 Mapas de grade de ocupação gerados pela simulação do ambiente Armazém com VANTs 0 e 1 em ordem. ....	33
Figura 6.1 Gráficos demonstrando o número de pontos de interesse detectados em cada mapa para cada algoritmo.....	36
Figura 6.2 Gráficos demonstrando o tempo para executar cada algoritmo em cada mapa.....	36
Figura 6.3 Gráficos demonstrando o número de pontos de interesse detectados para cada algoritmo conforme o número de iterações em que transformações do mapa são executadas .....	37
Figura 6.4 Pontos de Interesse Detectados para Mapa 0.....	38
Figura 6.5 Pontos de Interesse Detectados para Mapa 1.....	39
Figura 6.6 Pontos de Interesse Detectados para Mapa 2.....	39
Figura 6.7 Pontos de Interesse Detectados para Mapa 3.....	40
Figura 6.8 Pontos de Interesse Detectados para Mapa 4.....	40
Figura 6.9 Pontos de Interesse Detectados para Mapa 5.....	41
Figura 6.10 Pontos de Interesse Detectados para Mapa 6.....	41
Figura 6.11 Pontos de Interesse Detectados para Mapa 7.....	42
Figura 6.12 Pontos de Interesse Detectados para Mapa 8.....	42
Figura 6.13 Pontos de Interesse Detectados para Mapa 9.....	43



Figura 6.14 Pontos de Interesse Detectados para Mapa 10.....	43
Figura 6.15 Pontos de Interesse Detectados para Mapa 11.....	44
Figura 6.16 Pontos de Interesse Detectados para Mapa 12.....	44
Figura 6.17 Pontos de Interesse Detectados para Mapa 13.....	45
Figura 6.18 Pontos de Interesse Detectados para Mapa 14.....	45

## **LISTA DE TABELAS**

Tabela 5.1 Tabela contendo IDs dos mapas e respectivos ambientes e VANTs.....	34
---	----

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>12</b>
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>14</b>
2.1 Detectores de Pontos de Interesse .....	14
2.2 Descritores de Características .....	16
<b>3 TRABALHOS RELACIONADOS</b> .....	<b>19</b>
3.1 Harris3D .....	19
3.2 MeshSIFT .....	20
3.3 Repetibilidade de Pontos de Interesse 3D em SLAM Heterogêneo de Múltiplos Robôs .....	20
<b>4 IMPLEMENTAÇÃO</b> .....	<b>22</b>
4.1 Pré-processamento dos Mapas.....	22
4.2 Detecção, Filtragem e Orientações de Pontos de Interesse .....	23
4.2.1 FAST3D .....	24
4.2.2 HarrisGrid3D .....	25
4.2.3 Shi-Tomasi3D .....	26
4.3 Correspondências.....	26
<b>5 EXPERIMENTOS</b> .....	<b>27</b>
5.1 Ambiente Labirinto .....	27
5.2 Ambiente Livraria .....	28
5.3 Ambiente Hospital .....	29
5.4 Ambiente Pista de Corrida.....	31
5.5 Ambiente Casa .....	32
5.6 Ambiente Armazém .....	33
<b>6 RESULTADOS</b> .....	<b>35</b>
6.1 Resultados Numéricos .....	35
6.2 Resultados visuais .....	37
<b>7 CONCLUSÕES</b> .....	<b>46</b>
<b>REFERÊNCIAS</b> .....	<b>47</b>

## 1 INTRODUÇÃO

Uma das principais aplicações para Veículos Aéreos Não Tripulados (VANTs) é o mapeamento de ambientes desconhecidos, principalmente considerando o contexto de múltiplos VANTs, o que permite aumentar a escala e diminuir o tempo de completar a tarefa. Nesse contexto de múltiplos VANTs, o principal problema que surge é o de fundir os diferentes mapas gerados por cada VANT para chegar em um mapa completo. Para alcançar a fusão de mapas, é necessário primeiro achar pontos de interesse dentro da nuvem de pontos de cada mapa (Keypoint Extraction), e depois comparar esses pontos de interesses e achar as correspondências entre os mapas (Keypoint Matching), após ter obtido os descritores de características dos pontos de interesse.

Os mapas construídos pelos VANTs nesse trabalho para serem fundidos são mapas de grade de ocupação. A técnica de mapa de grade de ocupação (ELFES; MATTHIES, 1987) é uma das mais usadas para mapeamento e localização de robôs. Nela os pontos do mapa são atribuídos três tipos de estados possíveis: livre, ocupado e não explorado. Os estados são atribuídos de acordo com as entradas de sensoriamento do robô. Nesse trabalho são usados mapas tridimensionais, que possuem três tipos de rotação, enquanto mapas bidimensionais têm apenas um.

Existem várias técnicas de detecção de pontos de interesse e de descrição de características para imagem 2D que são usados para mapas de grade de ocupação 2D, mas as alternativas para 3D são mais raras, existindo algumas extensões de algoritmos 2D para 3D. Dentre as técnicas de detecção mais conhecidas estão: Harris (HARRIS; STEPHENS, 1988), Shi-Tomasi (SHI; TOMASI, 1994), SIFT (LOWE, 2004), SURF (BAY; TUYTELAARS; GOOL, 2006) e FAST (ROSTEN; DRUMMOND, 2006). Entre os descritores estão os descritores usados em SIFT e SURF, e o mais eficiente BRIEF (CALONDER; LEPETIT; FUA, 2011). De todas essas técnicas, foram encontrados extensões 3D apenas para o Harris e o SIFT.

Com todas essas informações, é relevante adaptar as técnicas para o uso em mapas 3D e comparar sua eficácia e eficiência para encontrar o melhor método para usar em fusão de mapas de grade de ocupação 3D, que é essencial para realizar uma aplicação de localização e mapeamento com múltiplos VANTs. Este trabalho está inserido dentro de um trabalho do Projeto Unmanned Systems Research Lab da Universidade Federal do Rio Grande do Sul, que propõe uma solução de fusão de mapas de grade de ocupação tridimensionais, liderado por Maik Basso.

Este trabalho está organizado conforme descrito a seguir. No capítulo 2 está descrito a base teórica necessária para atingir os objetivos propostos, detalhando conceitos como mapas de grade de ocupação, detectores de ponto de interesse 2D e descritores de características 2D. No capítulo 3 são mencionados trabalhos relacionados, que adaptam métodos 2D para 3D, ou comparam e avaliam técnicas 3D. O capítulo 4 como foi feita a implementação das adaptações de FAST, Harris e Shi-Tomasi para mapas de grade de ocupação 3D. Os capítulos 5 e 6 descrevem os experimentos realizados e os resultados obtidos, respectivamente. Finalmente, o capítulo 7 apresenta as conclusões sobre o que se espera do trabalho.

## 2 FUNDAMENTAÇÃO TEÓRICA

A fim de implementar os diferentes algoritmos de Detecção de Pontos de Interesse para uso na fusão de mapas tridimensionais, é preciso entender os princípios empregados nessas diversas alternativas, além do contexto na aplicação em que são usadas.

### 2.1 Detectores de Pontos de Interesse

O objetivo de detectar pontos de interesse é de reduzir o número de pontos para serem analisados para determinar correspondências entre os mapas, reduzindo o custo computacional. Esses pontos de interesses são características do mapa, indicando bordas (edges) e cantos (corners) dos obstáculos indicados no mapa, ignorando espaços livres e regiões interiores ocupadas. Os algoritmos podem ser dedicados a detectar determinado tipo de característica, como detectar apenas bordas ou apenas cantos. A seguir alguns detectores e seus princípios são descritos:

**Harris Corner Detection**(HARRIS; STEPHENS, 1988) é um dos algoritmos de detecção mais conhecidos e usados em aplicações de computação visual, sendo também um dos mais antigos. Os autores propuseram uma solução combinada de detecção de bordas e de cantos considerando o fato de que cantos são regiões na imagem ou no mapa com uma variação de intensidade consideravelmente grande. O algoritmo consiste basicamente de achar a diferença de intensidade para um determinado ponto  $(u,v)$  em todas as direções dentro de uma janela de tamanho determinado, usando a função de autocorrelação local de um sinal, conforme a especificação matemática abaixo:

$$E_{x,y} = \sum_{u,v} w_{u,v} [I_{x+u,y+v} - I_{u,v}]^2$$

Onde  $w_{u,v}$  é a função janela que pondera os pontos dentro dela, normalmente sendo usada uma Gaussiana. Após, é aplicado uma Expansão de Taylor na função  $E(u, v)$  para maximizar a detecção de cantos:

$$E(u, v) = \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

Onde

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

$I_x$  e  $I_y$  são derivadas imagem nas direções x e y respectivamente. Finalmente é criada uma pontuação para determinar se há um canto ou não:

$$R = \det(M) - k(\text{Tr}(M))^2$$

O algoritmo original foi proposto para duas dimensões, mas em 2011 (SIPIRAN; BUSTOS, 2011) propuseram uma extensão para trabalhar em três dimensões.

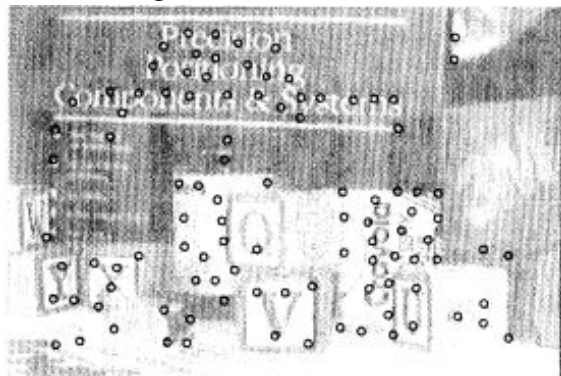
**Shi-Tomasi Corner Detection**(SHI; TOMASI, 1994) foi desenvolvido inspirado na base do Harris, alterando a função de pontuação, e obtiveram resultados melhores em relação ao antecessor. A pequena e significativa mudança que fizeram na equação está demonstrada a seguir:

$$R = \min(\lambda_1, \lambda_2)$$

Onde Lambda 1 e Lambda 2 são os autovalores da matriz M descrita no Harris Corner Detection.

O algoritmo foi proposto para duas dimensões, mais voltado para detectar características visuais. Os autores observaram que os autovalores da matriz M possuem propriedades que definem as características da janela de pontos sendo observada. Quando os dois autovalores tem valores pequenos há um indício de que tem um perfil de intensidade constante na janela, enquanto que um autovalor pequeno e um grande indicam um padrão unidirecional. Dois autovalores grandes e com pouca diferença entre si indicam um canto existente na janela, e por isso que a fórmula indica que o menor autovalor deve ser maior que o valor mínimo definido para determinar um canto. A figura 2.1 exemplifica características selecionadas pelo algoritmo Shi-Tomasi.

Figura 2.1: Imagem (SHI; TOMASI, 1994) exemplo

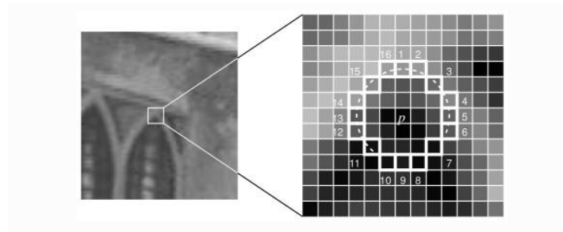


### Features from accelerated segment test (FAST)(ROSTEN; DRUMMOND, 2006)

é o método mais rápido dentre os mais conhecidos, entre eles o Harris e o Shi-Tomasi. Também foi proposto para imagens 2D. O algoritmo é descrito a seguir:

1. Um ponto  $p$  é selecionado para verificar se é de interesse ou não, sendo sua intensidade  $I_p$ .
2. Um limiar de valor apropriado  $t$  é selecionado.
3. Um círculo de 16 pontos ao redor de  $p$  é estabelecido.

Figura 2.2: Imagem (ROSTEN; DRUMMOND, 2006) representa o círculo de 16 pontos ao redor de  $p$



4. Um teste de alta velocidade é realizado para excluir não-cantos, examinando os quatro pontos nas posições 1, 9, 5 e 13 (pontos diretamente acima, abaixo, à direita e à esquerda respectivamente). Para  $p$  ser um canto, então pelo menos três desses pontos deve ser mais intenso que  $I_p + t$  ou menos intenso que  $I_p - t$ . Caso contrário, ele não pode ser um canto.
5. Caso o teste rápido passe, é verificado se existe uma sequência de pelo menos  $n$  pontos contínuos em que os pontos são mais intensos que  $I_p + t$  ou menos intensos que  $I_p - t$ , sendo  $n$  normalmente 12.
6. Como a probabilidade de haver pontos de interesse adjacentes é alta, é necessário resolver esse problema através da Supressão de Não-máximo, onde para todo ponto de interesse é atribuído uma pontuação e entre dois pontos adjacentes aquele com pontuação menor é descartado.

## 2.2 Descritores de Características

A descrição de característica tem como objetivo armazenar informações sobre a aparência ao redor de cada ponto de interesse de forma que seja invariável sob mudanças em translação, rotação e escala. O resultado normalmente é na forma de um vetor descritor para cada ponto de interesse. Esses descritores são usados para identificar ca-



racterísticas similares entre dois mapas distintos, resultando nas correspondências que são o objetivo final desse processo.

**Vetores descritores de n-dimensões** são o método de descrição mais comum, como referenciado anteriormente. Para um exemplo de um vetor de  $n = 128$  dimensões usado no método **SIFT (Scale-Invariant Feature Transform)**(LOWE, 2004), o descritor é criado da seguinte forma:

1. É estabelecida um vizinhança de 16x16 ao redor do ponto de interesse.
2. A vizinhança então é dividida em 16 sub-blocos de tamanho 4x4.
3. Para cada sub-bloco é criado um histograma de orientações de 8 binários. Gerando no total 128 valores binários. Esses valores são representados em um vetor para formar o descritor.

Esse método é muito eficaz em determinar descritores, mas não é eficiente, pelo fato de usar vetores de tamanho 128 o torna muito lento para computar, sendo inviável seu uso em aplicações de tempo real, como a de fusão de mapas usando VANTs na qual esse trabalho busca se inserir.

Para acelerar esse processo, um novo método **SURF (Speeded Up Robust Features)** (BAY; TUYTELAARS; GOOL, 2006) foi criado, usando agora um vetor de tamanho 64. Esse método usa vizinhanças de  $20s \times 20s$ , onde  $s$  é tamanho. É dividida em 4x4 sub-regiões e em cada sub-região são aplicadas transformadas wavelets na horizontal e vertical, resultando no seguinte vetor:

$$v = \left( \sum d_x, \sum d_y, \sum |d_x|, \sum |d_y| \right)$$

Apesar de melhorar consideravelmente o problema da velocidade, o fato de usar um vetor de ponto flutuante de tamanho 64 requer 256 bytes de memória, o que se torna um sério problema para milhares de descritores, ocupando muita memória e inviabilizando o uso em VANTs, que possuem memória limitada.

**BRIEF: Binary Robust Independent Elementary Features** (CALONDER; LE-PETIT; FUA, 2011) foi desenvolvido para resolver os problemas de velocidade e ocupação de memória, optando por usar strings binárias em vez de vetores de ponto flutuante. Essas strings binárias são usadas para achar correspondências com distância de Hamming. Primeiramente são selecionados  $n_d(x, y)$  pares de locais. Após isso, para o par formado por  $p$  e  $q$ , se  $I(p) < I(q)$ , então o resultado é 1, caso contrário 0. Aplicando para todos os pares é obtida um bitstring de  $n_d$  dimensões. Com essa bitstring, é possível usar Distância

de Hamming para achar as correspondências. No artigo é proposto usar  $n_d$  com valores de 128, 256 ou 512.

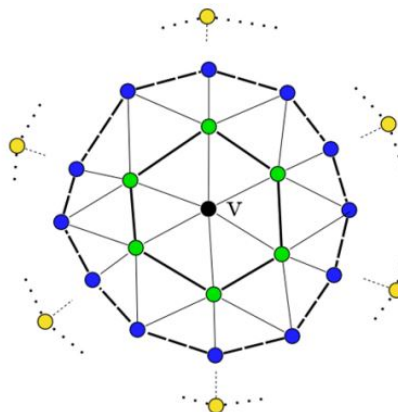
### 3 TRABALHOS RELACIONADOS

Na seção anterior foram revisitados vários algoritmos de detecção e descrição de características, e todos foram desenvolvidos considerando duas dimensões, com pontos de coordenadas  $(x,y)$ . Esse trabalho se propõe a avaliar métodos em mapas tridimensionais, com pontos de coordenadas  $(x,y,z)$ . Portanto é necessário adaptar e estender esses algoritmos para funcionar em três dimensões.

#### 3.1 Harris3D

Como mencionado anteriormente, em (SIPIRAN; BUSTOS, 2011) é proposta uma extensão ao Harris para aplicações 3D. Esse método usa os vértices de um objeto tridimensional para calcular o valor do operador Harris associado a cada vértice. Seja  $v$  um vértice a ser analisado e  $V_k(v)$  a vizinhança de  $k$  anéis ao redor de  $v$ . Então o centroide de  $V_k(v)$  é calculado e translacionado para origem do sistema de coordenadas. O plano dos pontos translacionados é então rotacionado para ficar em torno do eixo  $z$ . Finalmente as derivadas de  $v$  são obtidas e integradas junto de um função Gaussiana contínua, para então obter a matriz  $M$  relacionada a  $v$  com a qual o valor do operador Harris é obtido. A imagem 3.1 mostra a vizinhança de anéis ao redor do centroide  $v$ .

Figura 3.1: Imagem (SIPIRAN; BUSTOS, 2011) exemplo de vizinhança ao redor de  $v$

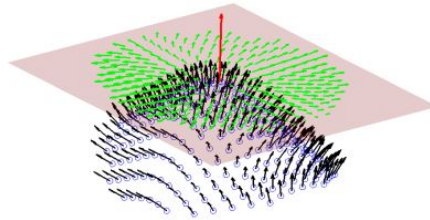


Esse método é voltado para processamento de imagens 3D, usando os vértices das imagens para cálculos, enquanto que nesse trabalho foi necessário desenvolver um método que funcione com mapas de grade de ocupação tridimensionais.

### 3.2 MeshSIFT

Proposto em (MAES et al., 2010), esse método apresenta uma adaptação do algoritmo SIFT para superfícies 3D, tendo o foco principal em reconhecimento facial 3D. A curvatura média de cada vértice em diferentes versões de mesh de entrada é contida em um espaço escalado. Então as vizinhanças de cada extremo do espaço escalado são descritas em um vetor de características contendo histogramas concatenados de índices de formas e ângulos inclinados. Finalmente as correspondências são encontradas comparando os ângulos no vetor. A figura 3.2 mostra a vizinhança do espaço escalar de um extremo com as normais e normais projetadas.

Figura 3.2: Imagem (MAES et al., 2010) exemplo de vizinhança



### 3.3 Repetibilidade de Pontos de Interesse 3D em SLAM Heterogêneo de Múltiplos Robôs

Em (BOROSON; AYANIAN, 2019) é avaliada a repetibilidade de pontos de interesse em vários algoritmos de detecção de características 3D, sendo eles Harris3D, ISS, KPQ, KPQ-SI e NARF. Esses algoritmos são separados em três categorias:

#### A. Extensões de Pontos de Interesse de Imagens 2D

Essa categoria é a que interessa para a proposta do trabalho e à qual os algoritmos descritos na seção de Fundamentação Teórica pertencem. No contexto do artigo de (BOROSON; AYANIAN, 2019), o algoritmo que se enquadra nessa categoria é o Harris3D.

#### B. Pontos de Interesse de Curvatura da Superfície

Essa é uma abordagem dedicada a aplicações 3D, onde a superfície é considerada para achar pontos de interesse, buscando orientações distintas e vizinhos com grande variação normal em todas três direções, encontrando então lugares com curvatura acentuada. No artigo citado são avaliados ISS, KPQ e KPQ-SI dessa categoria.

### *C. Pontos de Interesse de Alcance da Imagem*

Essa categoria tem uma proposta diferente: em vez de trabalhar com uma nuvem de pontos 3D, a nuvem de pontos é reduzida para uma imagem de alcance projetando a nuvem em um plano da câmara esférica localizado no ponto de vista da câmara, colorindo cada pixel por profundidade. No artigo citado o método dessa categoria avaliado é o NARF (Normal Aligned Radial Feature).

A aplicação de SLAM heterogêneo de múltiplos robôs é relacionada à aplicação em que esse trabalho se insere: mapeamento com múltiplos VANTs e o uso de detectores de pontos de interesse. Os resultados obtidos pelos autores os levaram a concluir que NARF e KPQ-SI tiveram a maior repetibilidade de pontos de interesse na mesma posição com sensores diferentes: 10%.

## 4 IMPLEMENTAÇÃO

Como mencionado anteriormente, este trabalho se dedica a comparar os algoritmos de Detecção de Pontos de Interesse para serem usados na Fusão de Mapas de Grade de Ocupação Tridimensionais. Consequentemente, é necessário entender o contexto em que a detecção de pontos de interesse acontece e as etapas da implementação da aplicação de fusão de mapas. A aplicação de fusão de mapas possui as seguintes etapas:

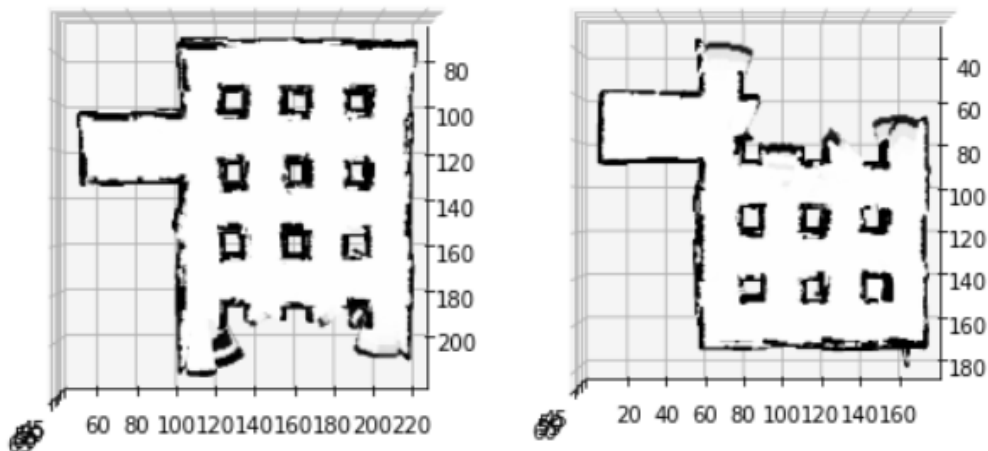
- Pré-processamento dos Mapas;
- Detecção de Pontos de Interesse;
- Filtragem de Pontos de Interesse;
- Orientações de Pontos de Interesse;
- Descrição de Características;
- Computação de Correspondências;
- Filtragem de Correspondências;
- Computação de Matriz de Transformação do Mapa e Fusão dos Mapas.

### 4.1 Pré-processamento dos Mapas

Devido à grande quantidade de pontos presentes em um mapa de grade de ocupação, antes de executar a detecção dos pontos de interesse é importante processar os dados dos mapas, gerando mapas com menos pontos e que ainda sejam descritivos do mapa original. Fazendo isso é reduzido significativamente o tempo de execução, diminuindo o custo computacional.

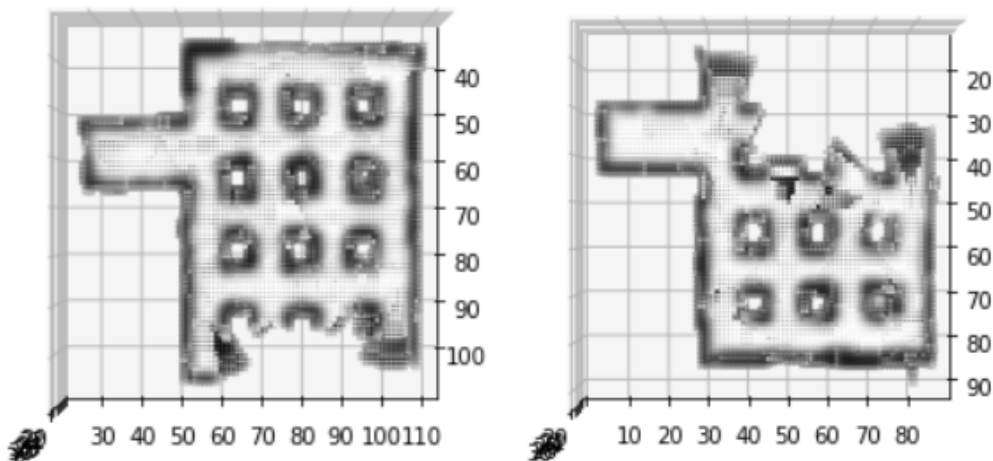
Nessa etapa são aplicados filtros de dilatação/erosão, que aumentam/diminuem a influência de um pixel à região ao seu entorno e eliminam o ruído; um filtro Gaussiano para tornar as transições de células mais fluidas e melhorar a performance nas etapas seguintes; e finalmente o *downsampling* que reduz o tamanho do mapa ao gerar uma nova representação do mapa com menos células. As figuras 4.1 e 4.2 mostram os mapas antes e depois do pré-processamento. Pode-se observar que embora tenha diminuído a definição dos mapas, é possível ainda indentificá-los, e também que sua dimensão diminuiu pela metade.

Figura 4.1: Mapas de grade de ocupação de entrada adquiridos por robôs diferentes antes do pré-processamento.



Fonte: O Autor

Figura 4.2: Mapas de grade de ocupação de entrada adquiridos por robôs diferentes depois do pré-processamento.



Fonte: O Autor

## 4.2 Detecção, Filtragem e Orientações de Pontos de Interesse

Logo após a etapa de pré-processamento e a geração dos mapas a serem comparados é feita a execução da Detecção dos Pontos de Interesse. Como visto originalmente, os algoritmos FAST, Harris e Shi-Tomasi foram originalmente desenvolvidos para mapas bidimensionais, sendo então adaptá-los para mapas tridimensionais. As implementações dessas adaptações, FAST3D, Harris3D e Shi-Tomasi3D, estão descritas nas seções 4.2.1, 4.2.2 e 4.2.3.

De forma geral, o algoritmo de detecção de pontos de interesse recebe como parâmetro um mapa de grade de ocupação tridimensional, além de outros parâmetros particulares a cada algoritmo, e retorna uma lista com todos os pontos de interesse encontrados para o determinado mapa.

Terminada a etapa de detecção, a lista de pontos de interesse retornada é filtrada. A filtragem é necessária devido à grande quantidade de pontos de interesse obtidos na etapa anterior, o que afeta o rendimento das etapas seguintes. São selecionados os pontos mais significativos para descrever um determinado entorno, através do método de Supressão de Não-Máximo (ROSTEN; DRUMMOND, 2006). Nesse método, os pontos são escolhidos conforme uma pontuação baseada no somatório das diferenças absolutas entre pontos de interesse ao redor de um determinado ponto de referência.

Concluída a filtragem, o próximo passo é calcular as orientações dos pontos restantes. As orientações auxiliam na construção de descritores de características e encontrar similaridades entre pontos equivalentes entre mapas.

#### 4.2.1 FAST3D

O primeiro algoritmo proposto desse trabalho *FAST3D* adaptado para mapas tridimensionais recebe os seguintes parâmetros de entrada: mapa de grade de ocupação tridimensional  $m$ ; número máximo de testes contínuos circulares  $n$ ; raio da máscara  $r$ ; o limiar  $l$  e o raio máximo da vizinhança  $nr$ . O algoritmo implementado segue o seguinte fluxo:

1. Com base no raio  $r$  é criada uma máscara contendo as coordenadas da células que formam um círculo ao redor da célula sendo analisada.
2. Para todas as células contidas no espaço conhecido, é calculada a média de intensidade da vizinhança  $I_p$  ao redor da célula a ser analisada, sendo a vizinhança por raio  $nr$ .
3. As intensidades de cada célula que compõe o círculo contido pela máscara do passo 1 nos planos XY, ZX e YZ são armazenadas em  $I_{xy}$ ,  $I_{zx}$  e  $I_{yz}$ .
4. Então é aplicado o teste FAST rotacionado nos três planos criados anteriormente. A rotação é considerada através de uma máscara no momento da execução da verificação cruzada, onde 4 pontos dividindo o círculo ao redor do ponto analisado são



comparados com o ponto analisado, através de

$$if(abs(I[i + k * maskStep] - Ip) > l) - > crossCheckCount + 1$$

, onde  $k$  é o número do passo,  $maskStep$  é a máscara a ser usada no passo e  $crossCheckCount$  é o contador da verificação cruzada, que é verdadeira para contador igual ou maior que 3.

5. Caso a verificação cruzada seja falsa, o próximo ponto vai ser analisado. Do contrário, a lista com os pontos contidos no círculo de raio  $r$  ao redor do ponto analisado vai ser percorrida comparando cada item da lista *circulo*:

$$if(abs(circulo[i] - Ip) > l) - > cont + 1$$

Se  $cont$  for maior ou igual a  $n$ , então o ponto analisado será adicionado a lista de pontos de interesse, e o próximo ponto é analisado.

#### 4.2.2 HarrisGrid3D

O segundo algoritmo desenvolvido *HarrisGrid3D* para adaptar Harris para mapas tridimensionais segue o mesmo fluxo do algoritmo original, sendo necessário apenas adaptar as matrizes de derivadas imagem para 3 dimensões, considerando as coordenadas  $x$ ,  $y$  e  $z$ .

1.  $I_{xx}$ ,  $I_{xy}$ ,  $I_{xz}$ ,  $I_{yy}$ ,  $I_{yz}$  e  $I_{zz}$  são computados a partir dos gradientes dos dados de entrada para  $x$ ,  $y$  e  $z$ .
2. Matriz  $M$  é calculada através de:

$$M = \sum_{x,y,z} w(x,y,z) \begin{bmatrix} I_x I_x & I_x I_y & I_x I_z \\ I_x I_y & I_y I_y & I_y I_z \\ I_x I_z & I_y I_z & I_z I_z \end{bmatrix}$$

3. A pontuação é calculada com a mesma fórmula do algoritmo original, mudando apenas a matriz  $M$ :

$$R = det(M) - k(Tr(M))^2$$

### 4.2.3 Shi-Tomasi3D

Como visto anteriormente, o algoritmo Shi-Tomasi tem a mesma base que o Harris, sendo a única mudança a fórmula de pontuação. Isso significa que a sequência de passos é a mesma descrita em 4.2.2: são computados os gradientes  $I_{xx}$ ,  $I_{xy}$ ,  $I_{xz}$ ,  $I_{yy}$ ,  $I_{yz}$  e  $I_{zz}$  dos dados de entrada; a matriz  $M$  é calculada com os gradientes. Na última etapa, a etapa de pontuação, é onde difere do HarrisGrid3D, onde os autovalores da matriz  $M$  devem ser maiores que o limiar definido como parâmetro da função, conforme a equação abaixo:

A fórmula de pontuação com a nova matriz  $M$  fica:

$$R = \min(\lambda_1, \lambda_2, \lambda_3)$$

Onde  $\lambda_3$  é o autovalor referente à coordenada  $z$  e o valor de  $R$  deve ser maior que o limiar definido.

### 4.3 Correspondências

Com o resultado de pontos de interesse filtrados e com orientações, a etapa seguinte é construir os descritores de características e então computar as correspondências. Para esse fim foi utilizado uma implementação adaptada a mapas tridimensionais do algoritmo BRIEF mencionado anteriormente. O BRIEF3D implementado faz parte do trabalho de fusão de mapas de grade de ocupação tridimensionais em que este trabalho está inserido.

Neste trabalho não é relevante aprofundar na implementação do algoritmo BRIEF3D, é apenas importante saber que as saídas da filtragem dos algoritmos de detecção anteriores, as listas dos pontos de interesse, serão as entradas para o BRIEF3D, da mesma forma que seria quando em uma aplicação bidimensional dos algoritmos.

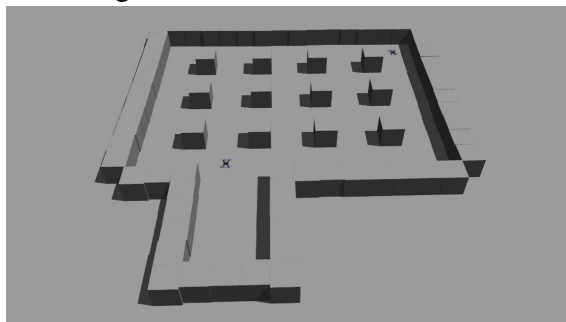
## 5 EXPERIMENTOS

Os experimentos conduzidos para comparar os algoritmos implementados FAST3D, Harris3D e Shi-Tomasi3D são baseados na diversidade de ambientes em que são executados. Seis ambientes foram escolhidos onde foram simuladas atividades de mapeamento com VANTs, gerando os 15 mapas a serem capturados os pontos de interesse. Para simulação de ambientes virtuais tridimensionais foi usado o simulador Gazebo<sup>1</sup> junto com um *Firmware PX4*<sup>2</sup> para uso de múltiplos VANTs. Os seis ambientes são descritos nas sessões a seguir.

### 5.1 Ambiente Labirinto

O ambiente Labirinto foi desenvolvido para facilitar a captura de resultados visuais, apresentando uma altura relativa baixa de 1 metro, o que facilita a visualização de topo. Além disso, o ambiente apresenta características homogêneas, possuindo obstáculos semelhantes e fáceis de serem identificados, mas com certo grau de dificuldade para diferenciar uns dos outros. A Figura 5.1 demonstra o ambiente.

Figura 5.1: Ambiente Labirinto



Fonte: O Autor

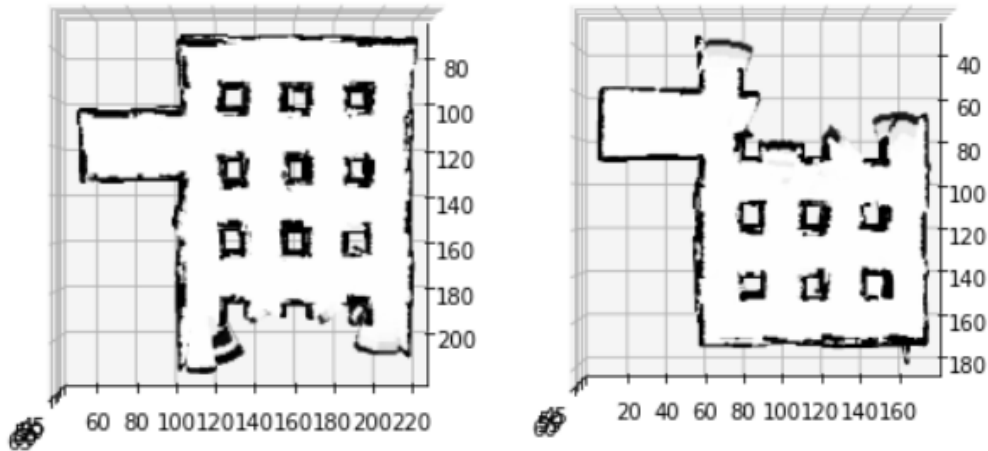
Na simulação de mapeamento desse ambiente foram utilizados dois VANTs, resultando em dois mapas, mostrados nas Figura 5.2.

---

<sup>1</sup><http://gazebosim.org/tutorials/>

<sup>2</sup><https://github.com/maikbasso/Firmware>

Figura 5.2: Mapas de grade de ocupação gerados pela simulação do ambiente Labirinto com dois VANTs.



Fonte: O Autor

## 5.2 Ambiente Livraria

O ambiente Livraria e os próximos demais ambientes foram retirados do repositório do projeto do Github AWS RoboMaker<sup>3</sup>. Esse ambiente é de tamanho médio, gerando mapas com uma quantidade considerável de pontos, e possui obstáculos bem definidos.

Figura 5.3: Ambiente Livraria

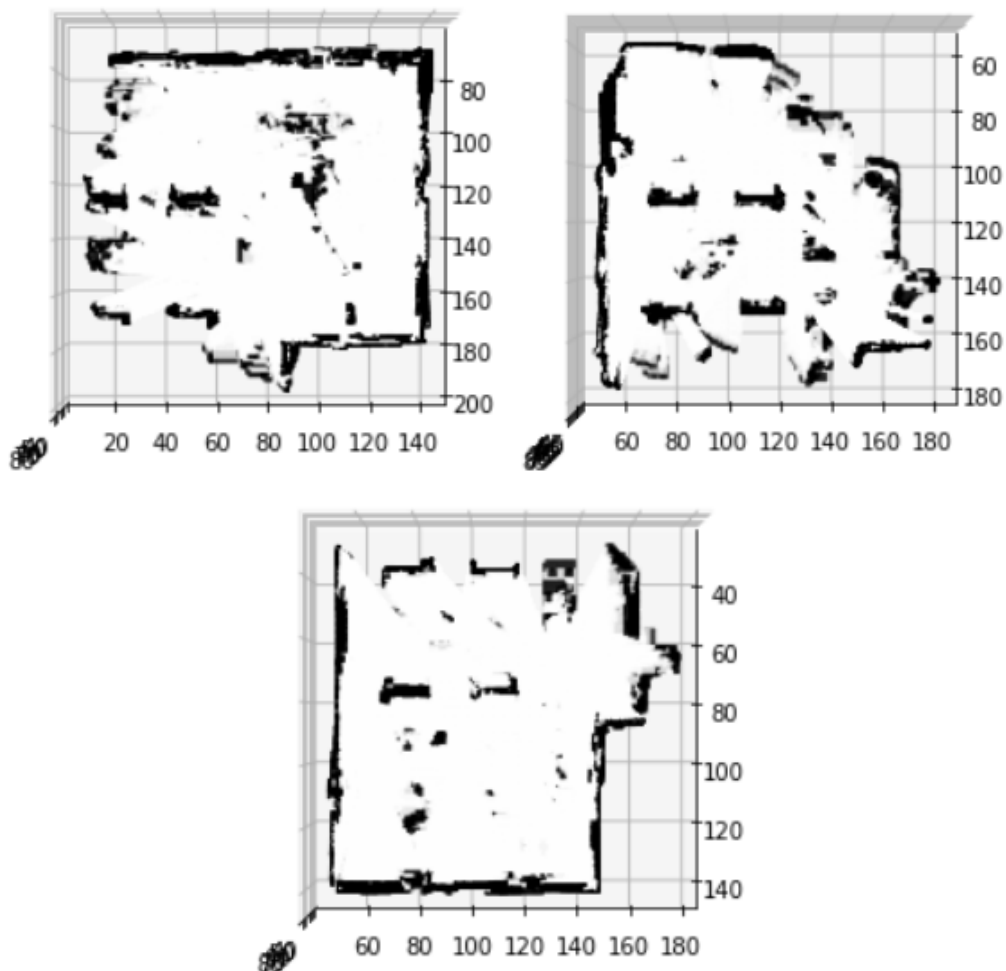


Fonte: O Autor

Na Figura 5.4 estão os mapas gerados na navegação e mapeamento executados pelos VANTs 0, 1 e 2, que serão usados como entradas para os algoritmos de detecção.

<sup>3</sup><https://github.com/aws-robotics>

Figura 5.4: Mapas de grade de ocupação gerados pela simulação do ambiente Livraria com VANTs 0, 1 e 2 em ordem.



Fonte: O Autor

### 5.3 Ambiente Hospital

O ambiente Hospital é de grande tamanho e densidade de obstáculos, gerando mapas com quantidade de pontos muito acima da maioria dos outros mapas, implicando em tempo de processamento maior. A densidade também dificulta em diferenciar pontos de interesse detectados relevantes de não relevantes.

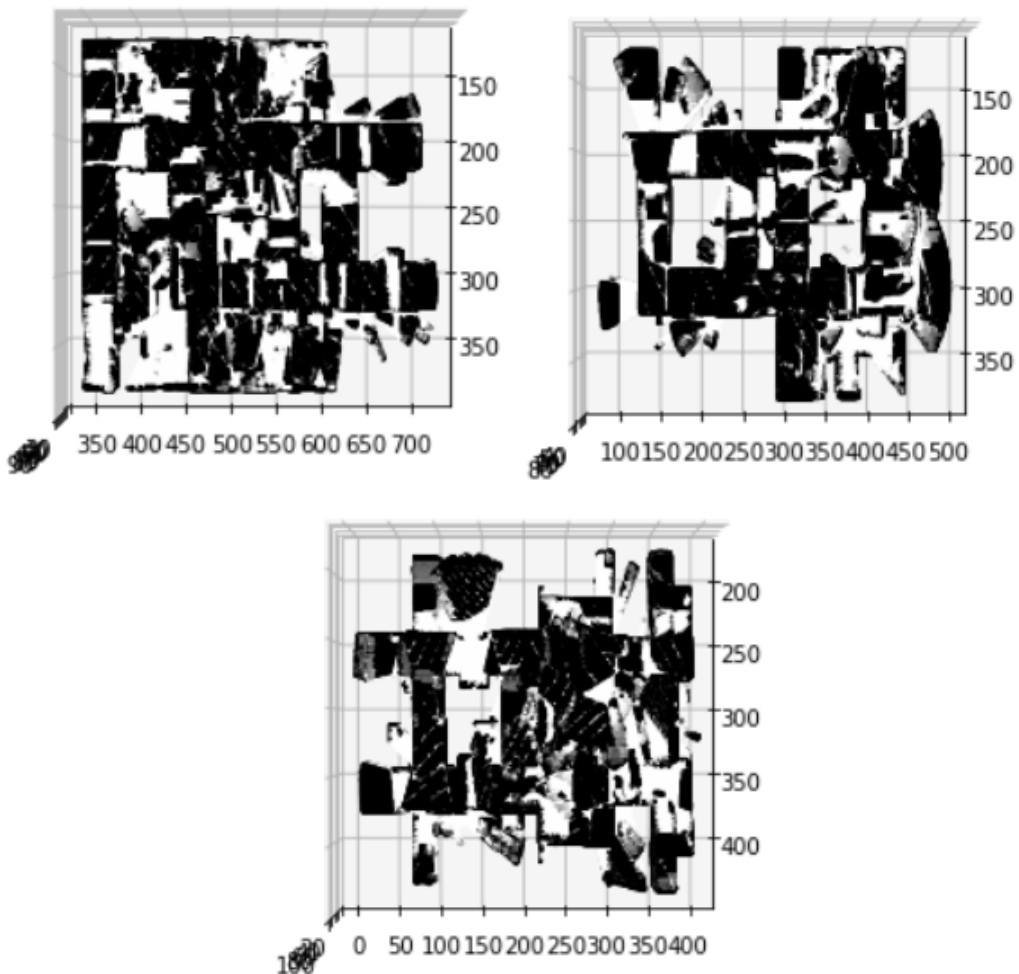
Figura 5.5: Ambiente Hospital



Fonte: O Autor

Na Figura 5.6 estão os mapas de grade de ocupação gerados na navegação e mapeamento executados pelos VANTs 0, 1 e 2, que serão usados como entradas para os algoritmos de detecção.

Figura 5.6: Mapas de grade de ocupação gerados pela simulação do ambiente Hospital com VANTs 0, 1 e 2 em ordem.

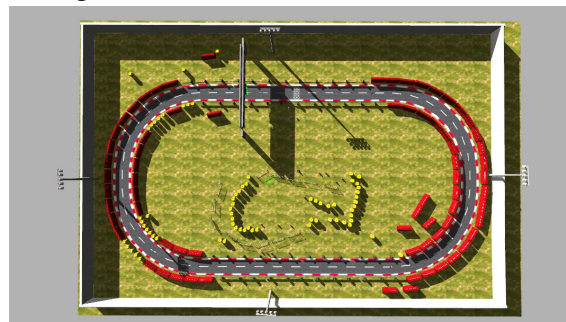


Fonte: O Autor

## 5.4 Ambiente Pista de Corrida

O ambiente Pista de Corrida é grande em escala e esparsos em obstáculos, gerando mapas com grande quantidade de pontos, mas muito espaço vazio. A grande quantidade de pontos demanda muito processamento, mas a complexidade de detectar pontos de interesse é baixa devido à pequena quantidade de obstáculos

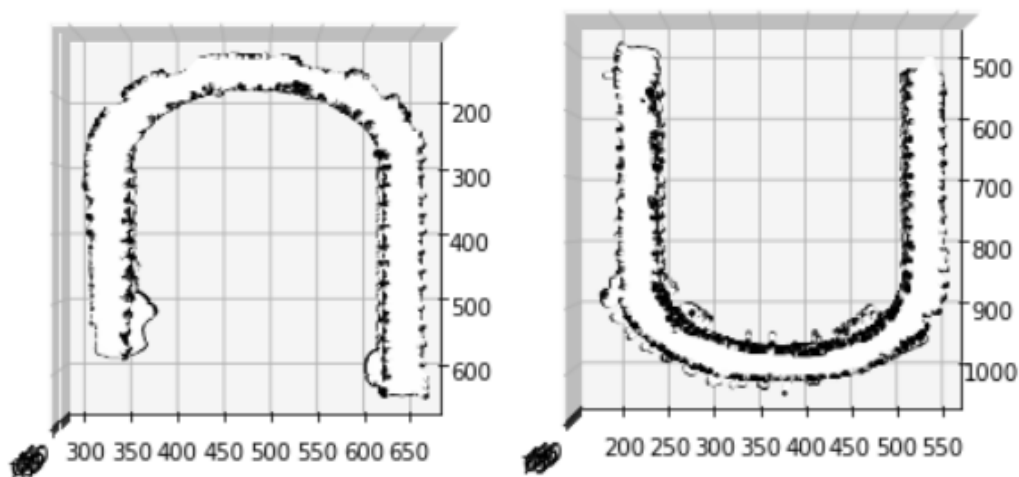
Figura 5.7: Ambiente Pista de Corrida



Fonte: O Autor

Na Figura 5.8 estão os mapas gerados na navegação e mapeamento executados pelos VANTs 0 e 1, que serão usados como entradas para os algoritmos de detecção.

Figura 5.8: Mapas de grade de ocupação gerados pela simulação do ambiente Pista de Corrida com VANTs 0 e 1 em ordem.



Fonte: O Autor

## 5.5 Ambiente Casa

O ambiente Casa é de pequena escala e grande densidade de obstáculos, gerando mapas com poucos pontos, reduzindo tempo de processamento, mas com detecção de pontos de interesse relevantes complexa.

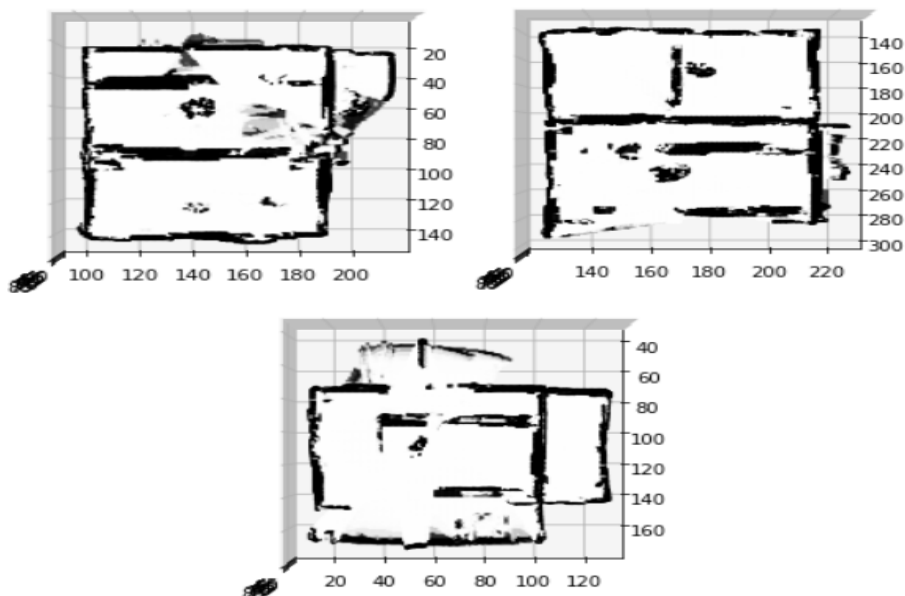
Figura 5.9: Ambiente Casa



Fonte: O Autor

Na Figura 5.10 estão os mapas gerados na navegação e mapeamento executados pelos VANTs 0, 1 e 2, que serão usados como entradas para os algoritmos de detecção.

Figura 5.10: Mapas de grade de ocupação gerados pela simulação do ambiente Casa com VANTs 0, 1 e 2 em ordem.



Fonte: O Autor



## 5.6 Ambiente Armazém

O ambiente Armazém é de pequena escala e pequena densidade de obstáculos, mas apresenta uma alta variância de altura dos obstáculos, o que aumenta a complexidade de detectar pontos de interesse relevantes, embora tenha um tempo de processamento baixo.

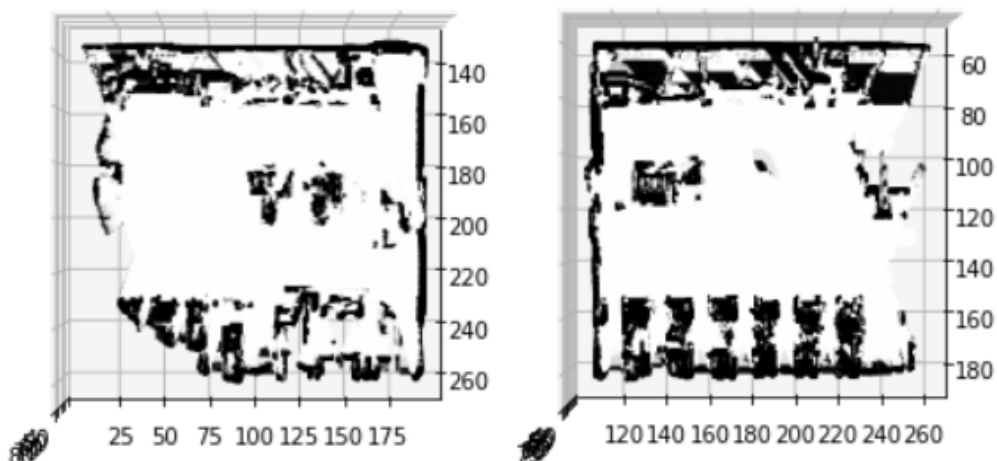
Figura 5.11: Ambiente Armazém



Fonte: O Autor

Na Figura 5.12 estão os mapas gerados na navegação e mapeamento executados pelos VANTs 0 e 1, que serão usados como entradas para os algoritmos de detecção.

Figura 5.12: Mapas de grade de ocupação gerados pela simulação do ambiente Armazém com VANTs 0 e 1 em ordem.



Fonte: O Autor

A Tabela 5.1 contém todos os mapas identificados que foram usados nos experimentos, no Capítulo 6 estão os resultados numéricos e visuais para todos os mapas. Na coluna ID estão os números que identificam os mapas; na coluna Ambiente onde cada mapa foi simulado; na coluna VANTs está o identificador do VANT que mapeou os pontos

para aquele ambiente; na coluna Resolução está a distância entre cada ponto em metros; na coluna Tamanho dos Eixos está o tamanho de cada mapa nos eixos  $x$ ,  $y$  e  $z$  em números de células.

Tabela 5.1: Tabela contendo IDs dos mapas e respectivos ambientes e VANTs

ID	Ambiente	VANT	Resolução	Tamanho dos Eixos (células)		
				x	y	z
0	livraria	0	0.1	250	250	100
1	livraria	1	0.1	190	190	100
2	livraria	2	0.1	190	190	100
3	hospital	0	0.1	730	730	100
4	hospital	1	0.1	500	500	100
5	hospital	2	0.1	620	620	120
6	labirinto	0	0.1	230	230	100
7	labirinto	1	0.1	320	320	100
8	pista de corrida	0	0.1	670	670	100
9	pista de corrida	1	0.1	1050	1050	100
10	casa	0	0.1	250	250	100
11	casa	1	0.1	310	310	100
12	casa	2	0.1	230	230	100
13	armazém	0	0.1	340	340	100
14	armazém	1	0.1	270	270	100

## 6 RESULTADOS

Após executar os experimentos descritos no Capítulo 5, são então comparados os resultados entre os algoritmos propostos FAST3D, HarrisGrid3D e Shi-Tomasi3D. Na seção 6.1 estão os resultados de experimentos onde foram coletados dados de pontos de interesse detectados, tempo de execução e estabilidade para cada método desenvolvido. Na seção 6.2 estão as demonstrações visuais das execuções de cada método para todos os mapas.

Para este trabalho não é relevante mostrar resultados de correspondências e fusão de mapas, eles são descritos como parte do trabalho de fusão de mapas de grade de ocupação tridimensionais em que este trabalho está inserido. É importante avaliar a eficácia de detectores de pontos de interesse para definir correspondências e fusão de mapas, e para isso devem ser verificada a quantidade de pontos de interesse relevantes detectados. Para uma fusão de mapas ser um sucesso, é essencial que não haja correspondências falsas depois da filtragem de correspondências, então um maior número de pontos de interesse relevantes facilita ao descritor de características a encontrar mais correspondências verdadeiras e assim permitir ao filtro retirar eventuais correspondências falsas.

### 6.1 Resultados Numéricos

Começando a análise de resultados numéricos com os dados de pontos de interesse detectados. Na Figura 6.1 estão os gráficos com número de pontos de interesse detectados antes (esquerda) e depois (direita) da filtragem de pontos. Nos gráficos é possível observar que:

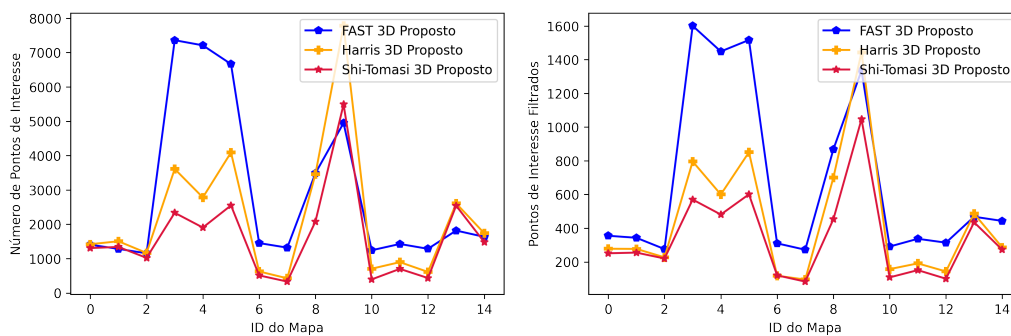
- Em média o algoritmo FAST3D detecta mais pontos de referência que os outros, principalmente considerando os pontos filtrados.
- A maior diferença de pontos detectados foi entre os mapas de id 3, 4 e 5, correspondendo aos mapas do ambiente Hospital, o ambiente mais complexo entre todos.
- O algoritmo FAST3D apenas foi superado, considerando pontos de interesse detectados, em 3 ambientes antes da filtragem: mapas 0,1 e 2 (Livreria); mapas 8 e 9 (Pista de Corrida); e mapas 13 e 14 (Armazém). E depois da filtragem FAST3D os superou ou empatou, o que indica que detectou pontos mais relevantes que os outros.

Na Figura 6.2 estão contidos 2 gráficos: o da esquerda descreve o tempo para executar cada algoritmo e gerar a lista de pontos de interesse para cada mapa, enquanto na direita é descrito o tempo para executar o mesmo algoritmo de filtragem de pontos para as entradas geradas em cada mapa pelos algoritmos especificados.

Pode se observar no gráfico da esquerda que FAST3D é processado em tempo igual ou inferior aos outros algoritmos para todos os mapas, sendo que HarrisGrid3D e o Shi-Tomasi tiveram um desempenho muito pior para os mapas 8 e 9 (Pista de Corrida), por causa do número menor de cantos nos obstáculos do mapa.

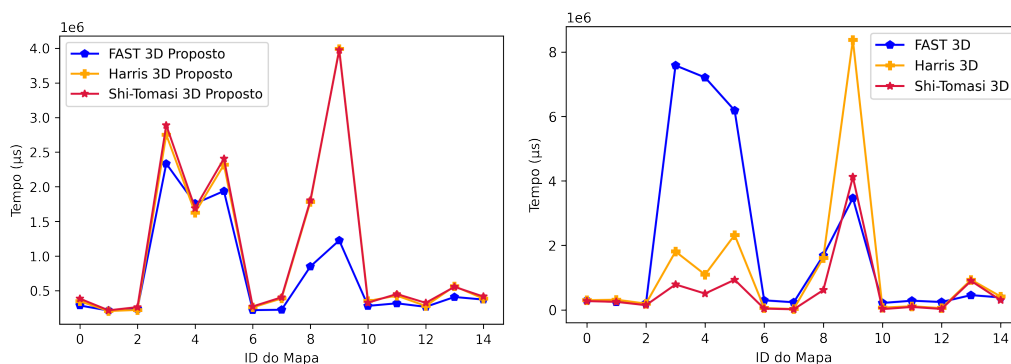
Enquanto no gráfico da direita o tempo é diretamente proporcional ao número de pontos detectados na etapa anterior. Consequentemente, houve grandes discrepâncias nos mapas 3,4 e 5 (Hospital); e 8 e 9 (Pista de Corrida), por serem os mapas de maior escala e com maior diferença de pontos detectados entre os algoritmos.

Figura 6.1: Gráficos demonstrando o número de pontos de interesse detectados em cada mapa para cada algoritmo



Fonte: O Autor

Figura 6.2: Gráficos demonstrando o tempo para executar cada algoritmo em cada mapa



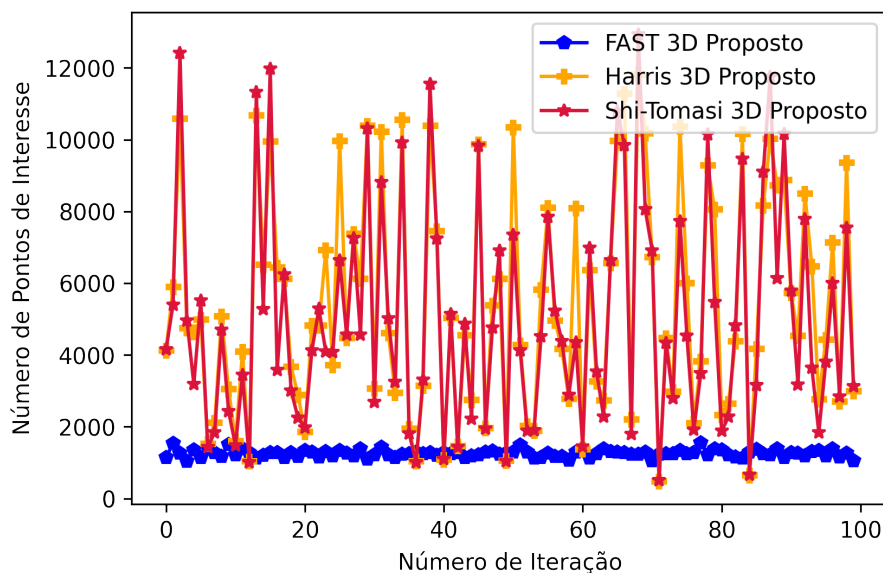
Fonte: O Autor

Para testar a estabilidade dos algoritmos propostos, submetendo os mesmos mapas a 100 transformações randômicas nos 6 graus de liberdade. Para cada transformação fo-

ram detectados os pontos de interesse correspondentes. Os resultados constam na Figura 6.3.

Com base no gráfico, pode-se observar que o algoritmo FAST3D se manteve estável no número de pontos de interesse detectados ao longo das transformações, enquanto os outros algoritmos demonstram variações abruptas entre transformações. Isso evidencia que os algoritmos Harris3D e Shi-Tomasi3D são mais suscetíveis a variações, o que pode causar problemas nas etapas seguintes.

Figura 6.3: Gráficos demonstrando o número de pontos de interesse detectados para cada algoritmo conforme o número de iterações em que transformações do mapa são executadas



Fonte: O Autor

## 6.2 Resultados visuais

Nessa seção estão presentes os resultados visuais da execução dos algoritmos com os mapas de entrada descritos no capítulo anterior. É importante também notar que os mapas apresentam uma visibilidade com difícil por serem visualizados com ponto de vista de topo de um mapa tridimensional, causando distorções em visualização bidimensional.

Nas Figuras 6.7, 6.8 e 6.9 estão os resultados dos mapas gerados no ambiente Hospital, que é o cenário mais complexo entre os presentes nos experimentos conduzidos. Observa-se que os mapas de pontos de interesse gerados pelo FAST3D são os que melhor representam o relevo, representando perfeitamente as paredes do ambiente. Enquanto o HarrisGrid3D gera os mapas com mais ruído, e o Shi-Tomasi é o que pior representa o

relevo, com número baixo de paredes representadas.

Tomando as Figuras 6.11 e 6.12 como exemplo, observa-se que os mapas produzidos por Harris3D e Shi-Tomasi descrevem menos as bordas dos objetos e focam mais nos cantos, especialmente o Shi-Tomasi. Enquanto o FAST3D detecta tanto bordas quanto cantos.

Em suma, observando as Figuras 6.4 a 6.18, verifica-se que o algoritmo FAST3D detectou pontos de interesse que melhor descrevem o relevo de obstáculos, com mapas mais detalhados e com menos ruído.

Figura 6.4: Pontos de Interesse Detectados para Mapa 0

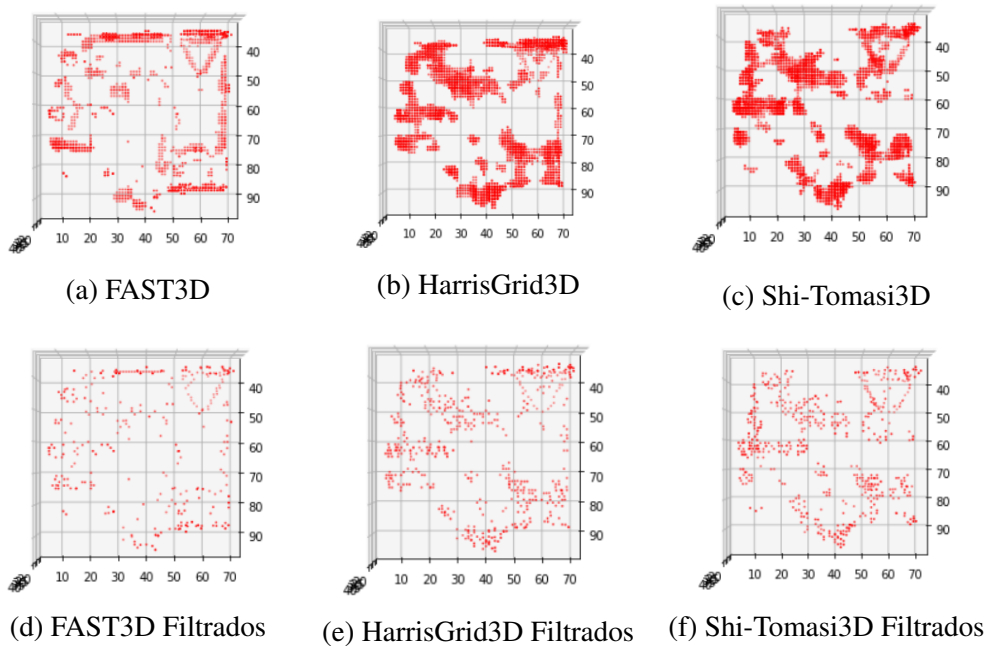


Figura 6.5: Pontos de Interesse Detectados para Mapa 1

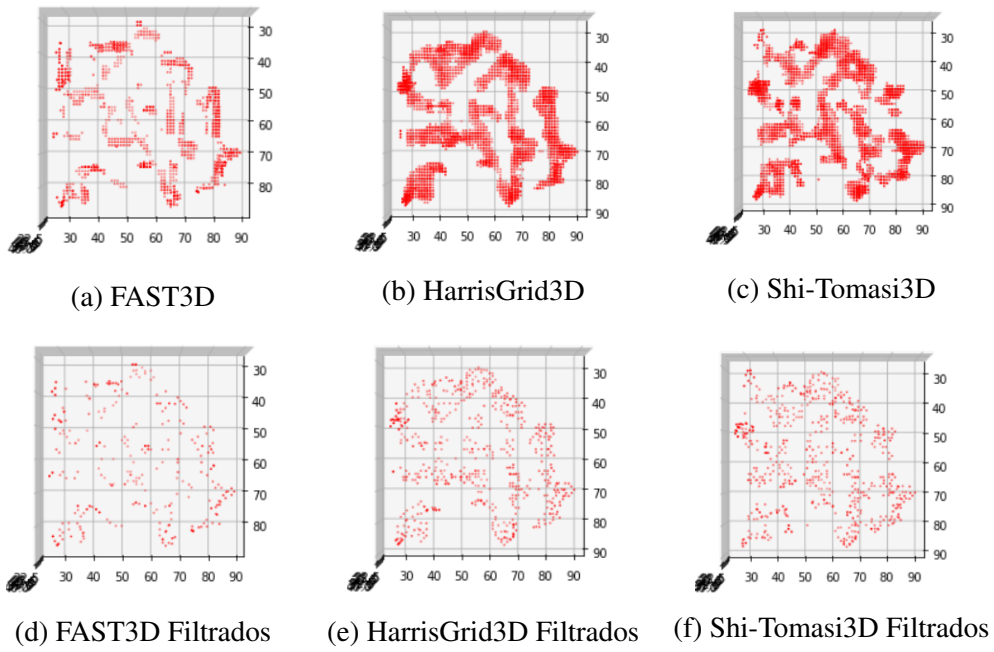


Figura 6.6: Pontos de Interesse Detectados para Mapa 2

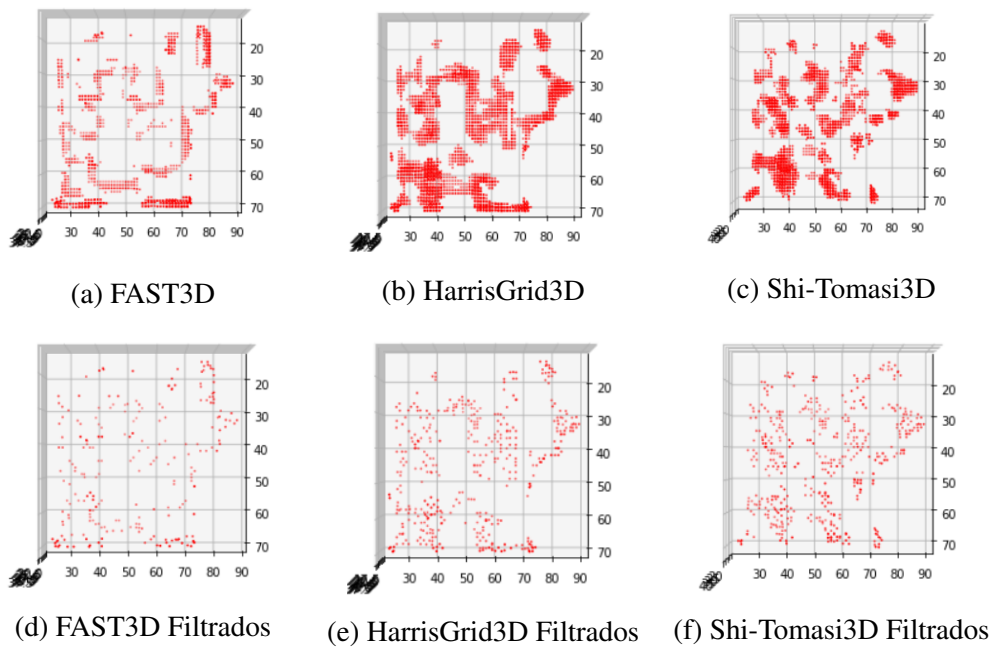
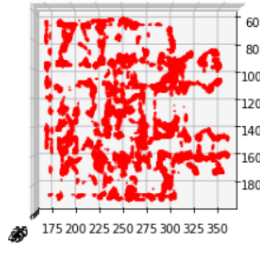


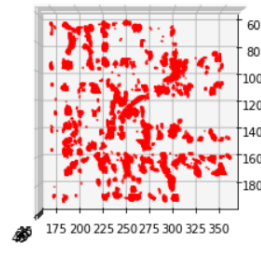
Figura 6.7: Pontos de Interesse Detectados para Mapa 3



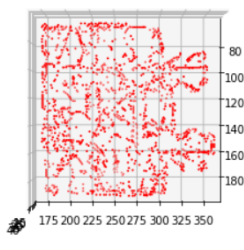
(a) FAST3D



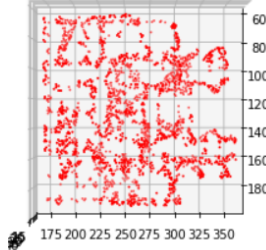
(b) HarrisGrid3D



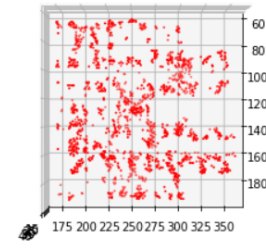
(c) Shi-Tomasi3D



(d) FAST3D Filtrados

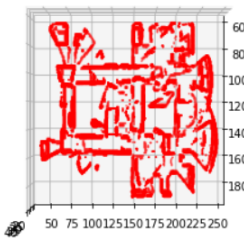


(e) HarrisGrid3D Filtrados

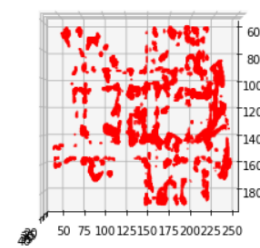


(f) Shi-Tomasi3D Filtrados

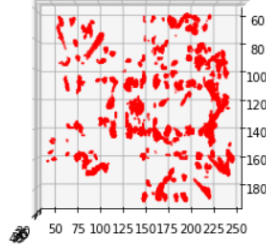
Figura 6.8: Pontos de Interesse Detectados para Mapa 4



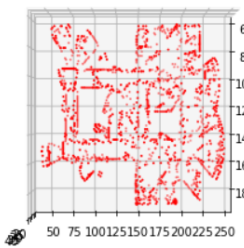
(a) FAST3D



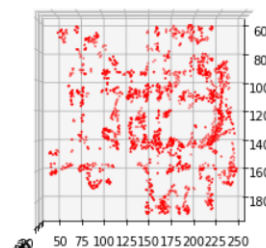
(b) Harris3D



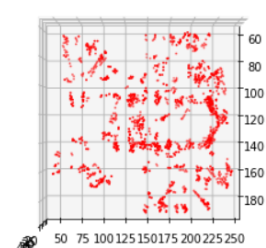
(c) Shi-Tomasi3D



(d) FAST3D Filtrados



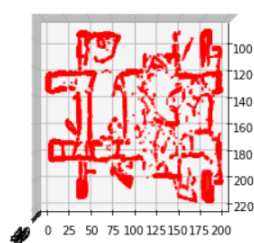
(e) HarrisGrid3D Filtrados



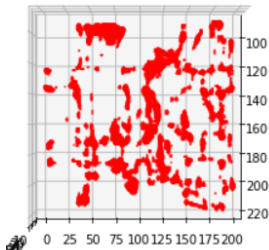
(f) Shi-Tomasi3D Filtrados



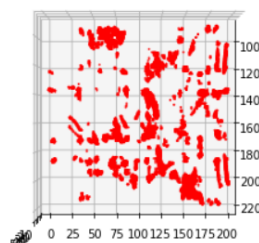
Figura 6.9: Pontos de Interesse Detectados para Mapa 5



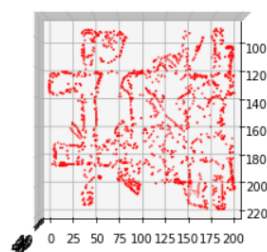
(a) FAST3D



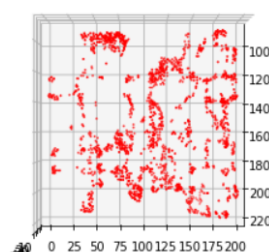
(b) HarrisGrid3D



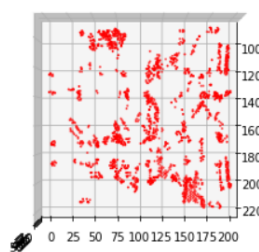
(c) Shi-Tomasi3D



(d) FAST3D Filtrados

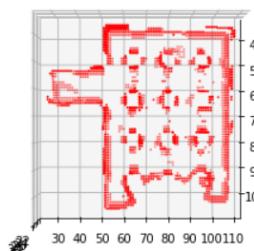


(e) HarrisGrid3D Filtrados

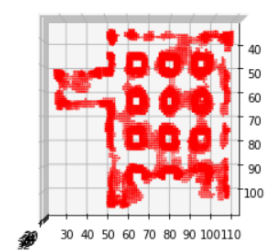


(f) Shi-Tomasi3D Filtrados

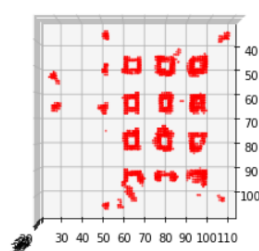
Figura 6.10: Pontos de Interesse Detectados para Mapa 6



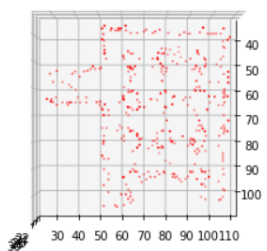
(a) FAST3D



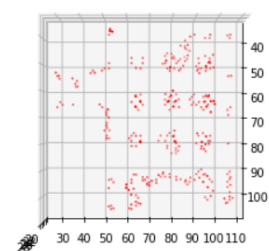
(b) HarrisGrid3D



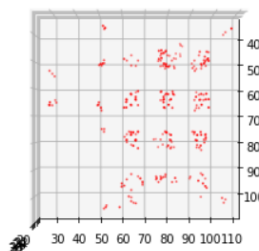
(c) Shi-Tomasi3D



(d) FAST3D Filtrados

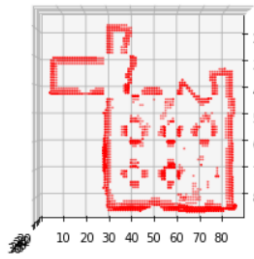


(e) HarrisGrid3D Filtrados

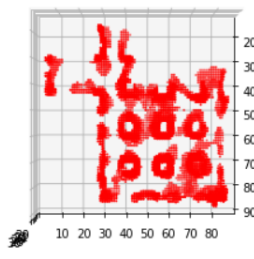


(f) Shi-Tomasi3D Filtrados

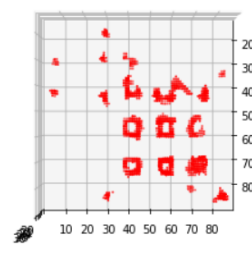
Figura 6.11: Pontos de Interesse Detectados para Mapa 7



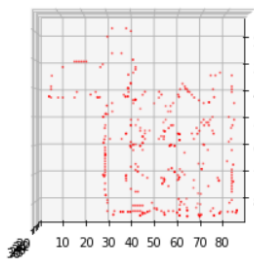
(a) FAST3D



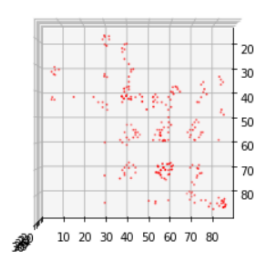
(b) HarrisGrid3D



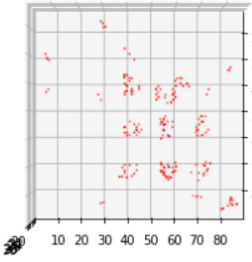
(c) Shi-Tomasi3D



(d) FAST3D Filtrados

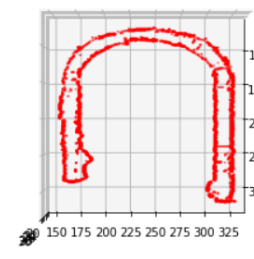


(e) HarrisGrid3D Filtrados

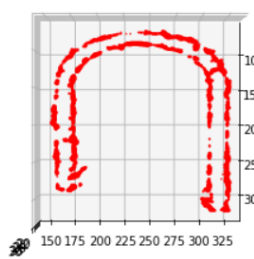


(f) Shi-Tomasi3D Filtrados

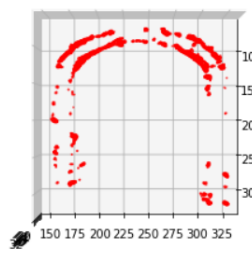
Figura 6.12: Pontos de Interesse Detectados para Mapa 8



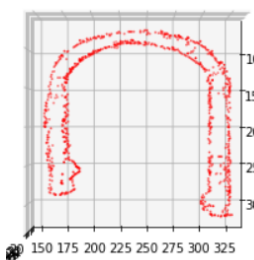
(a) FAST3D



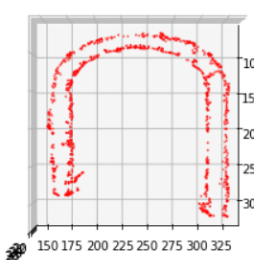
(b) HarrisGrid3D



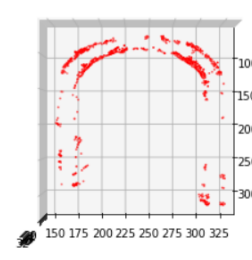
(c) Shi-Tomasi3D



(d) FAST3D Filtrados

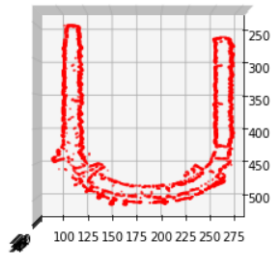


(e) HarrisGrid3D Filtrados

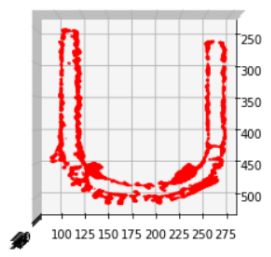


(f) Shi-Tomasi3D Filtrados

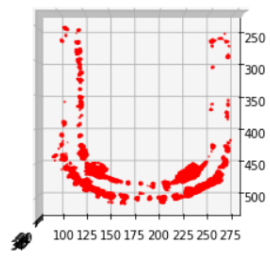
Figura 6.13: Pontos de Interesse Detectados para Mapa 9



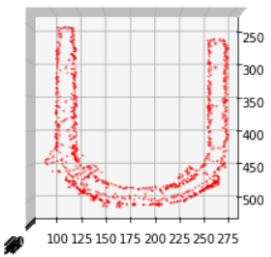
(a) FAST3D



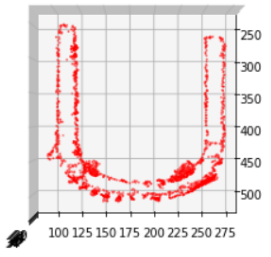
(b) HarrisGrid3D



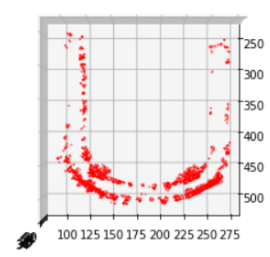
(c) Shi-Tomasi3D



(d) FAST3D Filtrados

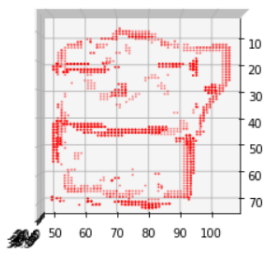


(e) HarrisGrid3D Filtrados

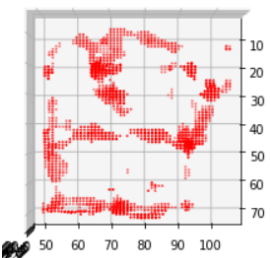


(f) Shi-Tomasi3D Filtrados

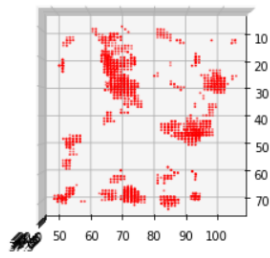
Figura 6.14: Pontos de Interesse Detectados para Mapa 10



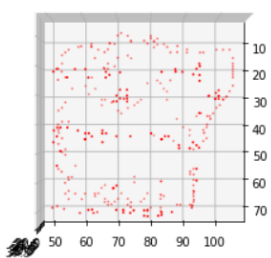
(a) FAST3D



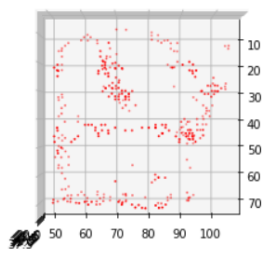
(b) HarrisGrid3D



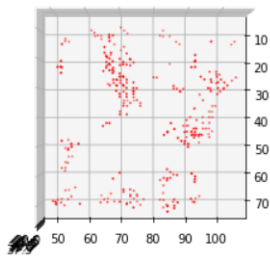
(c) Shi-Tomasi3D



(d) FAST3D Filtrados

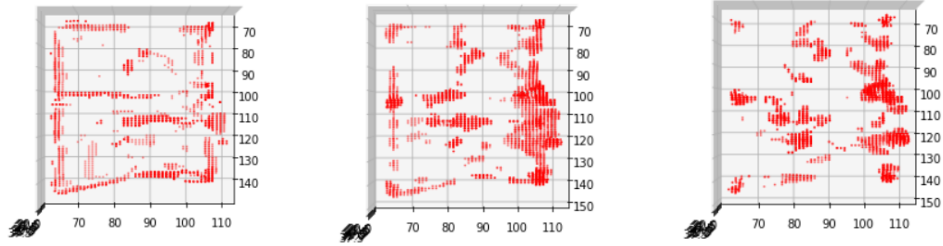


(e) HarrisGrid3D Filtrados



(f) Shi-Tomasi3D Filtrados

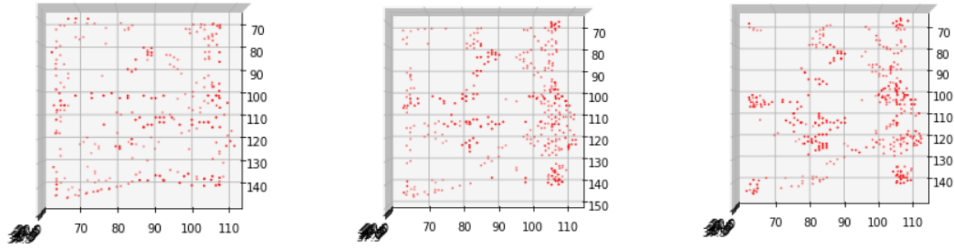
Figura 6.15: Pontos de Interesse Detectados para Mapa 11



(a) FAST3D

(b) HarrisGrid3D

(c) Shi-Tomasi3D

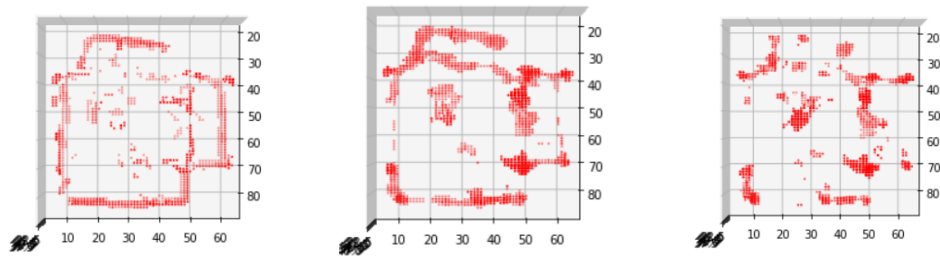


(d) FAST3D Filtrados

(e) HarrisGrid3D Filtrados

(f) Shi-Tomasi3D Filtrados

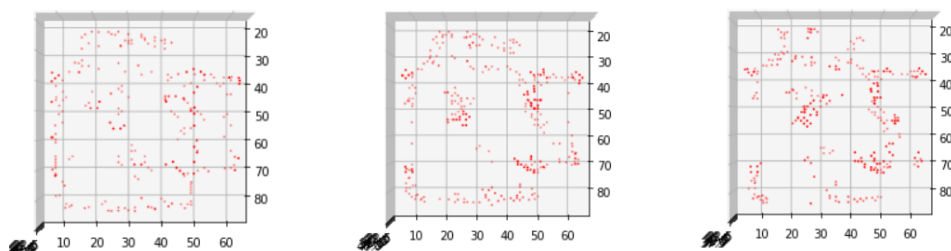
Figura 6.16: Pontos de Interesse Detectados para Mapa 12



(a) FAST3D

(b) HarrisGrid3D

(c) Shi-Tomasi3D



(d) FAST3D Filtrados

(e) HarrisGrid3D Filtrados

(f) Shi-Tomasi3D Filtrados

Figura 6.17: Pontos de Interesse Detectados para Mapa 13

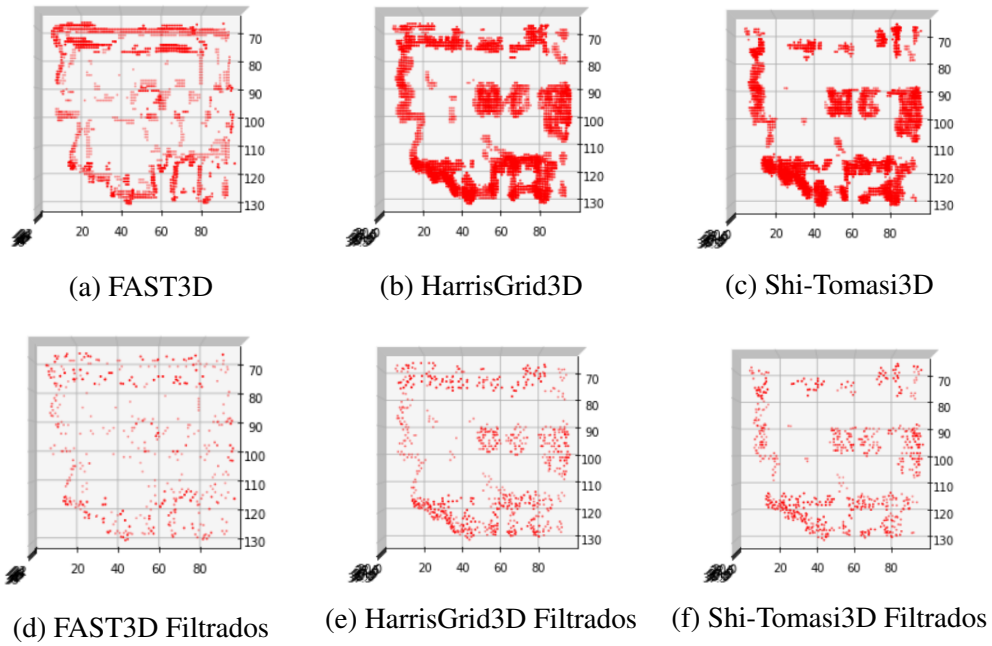
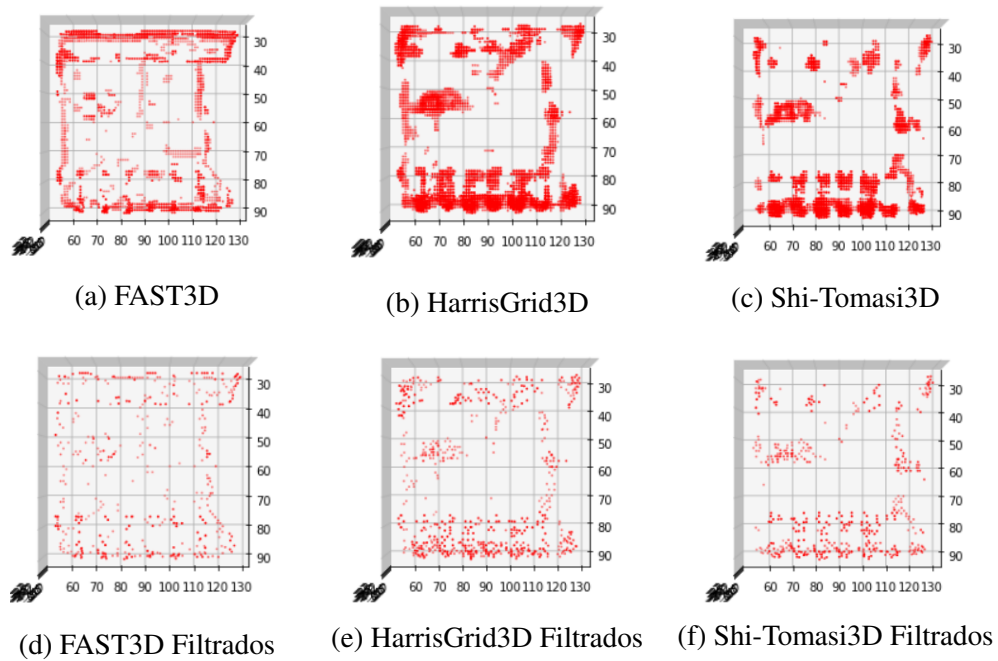


Figura 6.18: Pontos de Interesse Detectados para Mapa 14



## 7 CONCLUSÕES

Baseado nos resultados obtidos no Capítulo 6, é possível afirmar que o algoritmo FAST3D proposto superou os outros algoritmos propostos HarrisGrid3D e Shi-Tomasi, tanto em efetividade quanto performance.

Em questão de performance, apresentou tempos de execução inferiores aos demais e com mais pontos detectados. Enquanto que quando considerando a aplicação de fusão de mapas, ele possui vantagens sobre os demais, por ter maior estabilidade quanto a variações nos mapas; e por detectar tanto bordas quanto cantos de obstáculos, aumentando as chances de uma fusão correta. Enquanto a falta de descrição de bordas e a variância de pontos detectados dos algoritmos HarrisGrid3D e Shi-Tomasi3D atrapalha a fusão de mapas.

Como trabalhos futuros, existe a possibilidade de usar técnicas de aprendizado para otimizar a escolha de parâmetros dos algoritmos de detecção de pontos de interesse, considerando que a variância de valores de parâmetros conforme os mapas de entrada é grande. Esses parâmetros otimizados facilitariam a executar os algoritmos em aplicações de tempo real com VANTs.

Em suma, este trabalho propõe métodos inovadores para aplicações de mapas de grade de ocupação tridimensionais, onde não há muitos trabalhos atualmente que desenvolveram técnicas de detecção de pontos de interesse para mapas de grade de ocupação tridimensionais como neste trabalho, destacando principalmente os resultados obtidos para FAST3D implementado.

## REFERÊNCIAS

BAY, H.; TUYTELAARS, T.; GOOL, L. V. Surf: Speeded up robust features. In: . [S.l.: s.n.], 2006. v. 3951, p. 404–417. ISBN 978-3-540-33832-1.

BOROSON, E. R.; AYANIAN, N. 3d keypoint repeatability for heterogeneous multi-robot slam. In: **2019 International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2019. p. 6337–6343.

CALONDER, M.; LEPETIT, V.; FUA, P. Brief: Binary robust independent elementary features. 12 2011.

ELFES, A.; MATTHIES, L. Sensor integration for robot navigation: Combining sonar and stereo range data in a grid-based representataion. **26th IEEE Conference on Decision and Control**, v. 26, p. 1802 – 1807, 1987.

HARRIS, C.; STEPHENS, M. A combined corner and edge detector. In: **In Proc. of Fourth Alvey Vision Conference**. [S.l.: s.n.], 1988. p. 147–151.

LOWE, D. Distinctive image features from scale-invariant keypoints. **International Journal of Computer Vision**, v. 60, p. 91–110, 11 2004.

MAES, C. et al. Feature detection on 3d face surfaces for pose normalisation and recognition. In: **2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)**. [S.l.: s.n.], 2010. p. 1–6.

ROSTEN, E.; DRUMMOND, T. Machine learning for high-speed corner detection. In: . [S.l.: s.n.], 2006. v. 3951. ISBN 978-3-540-33832-1.

SHI, J.; TOMASI. Good features to track. In: **1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 1994. p. 593–600.

SIPIRAN, I.; BUSTOS, B. Harris 3d: A robust extension of the harris operator for interest point detection on 3d meshes. **The Visual Computer**, v. 27, p. 963–976, 11 2011.