

# Uma Abordagem Inteligente para Localização e Mapeamento Simultâneos de Veículos Autônomos

Gearlles V. Ferreira, Bruno J. T. Fernandes, Byron L. D. Bezerra  
Universidade de Pernambuco  
Escola Politécnica de Pernambuco  
Madalena, Recife  
{gvf, bjtf, byronleite}@ecom.poli.br

**Resumo**—No âmbito dos sistemas robóticos, técnicas para reconhecimento de objetos em Nuvem de Pontos vêm sendo estudadas nas últimas décadas conforme o avanço da tecnologia das câmeras de escaneamento tridimensional. Este trabalho apresenta uma técnica de identificação de Veículos Autônomos utilizando uma nova abordagem para reconhecimento de esferas, usado posteriormente para registro de imagens. O reconhecimento de esferas é realizado em três etapas: *sub-amostragem*, aplicação de critérios de verificação e identificação de robôs. A primeira etapa realiza uma simplificação do mapa escaneado, já a segunda tem como objetivo encontrar todos os objetos esféricos no mapa e, por fim, os robôs são encontrados a partir da validação da configuração das esferas. O registro das nuvens de pontos é realizado através de uma aplicação da Teoria de Rotação de Euler.

## I. INTRODUÇÃO

As últimas décadas foram marcadas pelo rápido crescimento da tecnologia do escaneamento 3D como, por exemplo, *lase scanners* e câmeras LIDAR (*Laser Detection and Raging*) [1]. A evolução de tais câmeras auxiliam no desenvolvimento da detecção e localização de objetos, tarefa fundamental dos principais sistemas robóticos [2], além tornar a aquisição de mapas 3D mais fácil e menos custosa [3]. Esses benefícios também afetam um problema chamado Localização e Mapeamento Simultâneos (SLAM, Simultaneous Localization and Mapping), onde veículos autônomos realizam o mapeamento de um ambiente e, ao mesmo tempo, controlam a sua localização de forma eficiente [4].

As câmeras LIDAR são dispositivos ópticos baseados nos princípios da reflexão da luz que apresentam tipicamente resultados confiáveis nas variações de profundidade [5]. Elas funcionam através do envio de múltiplos sinais de luz e da contagem do tempo de retorno do sinal refletido. Dessa forma, dado a velocidade da luz no meio de propagação e o tempo de resposta medido, pode-se obter a distância entre a lente da câmera e os outros vários pontos de um ambiente ou um objeto. As diversas distâncias podem ser mapeadas em um conjunto de vértices de um sistema de coordenadas tridimensional, chamado de Nuvem de Pontos (*Point Cloud*).

Nuvem de Pontos representam de forma precisa a região escaneada, entretanto apresentam alguns problemas. São armazenados de forma desorganizada e possuem grande

quantidade de vértices [6], o que muitas vezes torna o processamento dos dados custoso.

Utilizando a Nuvem de Pontos gerada pelas câmeras LIDAR, este trabalho tem como objetivo realizar a identificação de Veículos Autônomos utilizando o reconhecimento de esferas e o registro de imagens resultantes do SLAM. Dessa forma, o desenvolvimento do trabalho torna as atividades de mapeamento e exploração de terrenos mais independentes dos seres humanos, o que é essencial em circunstâncias de locais inabitados, *e.g.* outros planetas ou mapeamento topográfico.

### A. Descrição do problema

Ao realizar o mapeamento de uma determinada região, por motivos de relevos irregulares, formato do veículo e até situações climáticas, um Veículo Autônomo pode involuntariamente se desviar da sua rota e se perder do seu grupo. Dessa forma, para aproveitar o fato de que cada veículo pode ser equipado com um par de esferas, e com diferentes configurações, a abordagem proposta explora a capacidade de processamento dos robôs ao mesmo tempo em que realiza o mapeamento da região. Um Veículo Autônomo poderia ser identificado por outro ao realizar a varredura da região e contribuir para o controle da sua localização. Por exemplo, supondo que um robô A perdeu-se do grupo e que outro robô B esteja realizando o mapeamento da região, ao completar o escaneamento, o robô B identifica que o robô A está presente no mapa e pode contatá-lo a fim de fornecer coordenadas para que o mesmo volte para sua rota original. Entretanto, é necessária uma técnica que realize a detecção de um par de esferas com a finalidade de identificação de um Veículo Autônomo.

É válido salientar que, para fins práticos, este trabalho não leva em consideração a possibilidade da obtenção da posição atual por mecanismos de rastreamento, como Sistema de Posicionamento Global, popularmente conhecido como GPS. Isso se deve ao fato de que, em situações reais em outros planetas, tais mecanismos se tornam inválidos ou inviáveis.

## II. TÉCNICA PROPOSTA

Para solucionar o problema, é proposta uma técnica que se divide em algumas etapas, utilizando diferentes símbolos e notações apresentados na Tabela I. Na primeira etapa deve-se diminuir a quantidade de vértices presentes

Tabela I. SÍMBOLOS E NOTAÇÕES

Símbolo	Legenda
$n$	Número de vértices da Nuvem de Pontos.
$p_1$	Ponto variável.
$p_2$	Ponto a ser comparado.
$d_{p_1}$	Distância de $p_1$ ao centro.
$d_{p_2}$	Distância de $p_2$ ao centro.
$d_{p_1 p_2}$	Distância entre $p_1$ e $p_2$ .
$\delta$	Máxima distância analisada ao redor da esfera.
$\alpha$	Limiar que determina o número de pontos mais próximos ao centro.
$p$	Quantidade de pontos na região positiva da esfera.
$g$	Quantidade de pontos na região negativa da esfera.
$S_{p_2, pos}$	Soma das distâncias $p_2$ na região positiva (superior).
$S_{p_2, neg}$	Soma das distâncias $p_2$ na região negativa (inferior).
$\Delta_1$	Diferença entre a média das distâncias positivas e negativas.
$\Delta_2$	Percentual da variação entre os pontos dos hemisférios.
$t$	Uma das circunferências de secção da esfera.
$L_t^i$	Valor mínimo da distância $d_{p_1 p_2}$ média na circunferência $t$ .
$L_t^s$	Valor máximo da distância $d_{p_1 p_2}$ média na circunferência $t$ .
$n_r$	Número de pontos ao redor da esfera.
$\beta$	Limiar de $n_r$ .

na Nuvem de Pontos. Para tal, é utilizada uma técnica chamada de Voxel Grid. Essa técnica é utilizada em Nuvem de Pontos com grande número de vértices e que representam um objeto espacial, proporcionando uma redução no tempo gasto na execução do algoritmo.

Na segunda etapa, é realizado o reconhecimento das esferas presentes na Nuvem de Pontos. Seguindo alguns critérios de classificação, é realizada uma busca de circunferências concêntricas e de pontos que são supostos candidatos a centro de uma esfera.

Em seguida, é preciso identificar se as esferas são válidas, ou seja, certifica-se que há pares de esferas formados por uma esfera superior (*top sphere*) e uma outra inferior (*bottom sphere*). Essa etapa realiza o reconhecimento de um robô presente na Nuvem de Pontos.

Por fim, com o reconhecimento de esferas concluído, realiza-se o registro de mapas que representam o mesmo ambiente em diferentes ângulos, tomando como referência uma das esferas de um robô detectado. Essa tarefa é realizada pela translação e rotação da Nuvem de Pontos. A translação é realizada através do deslocamento de todos os pontos por um mesmo valor. Já a rotação é realizada pela aplicação do Teorema da Rotação de Euler [7], que afirma que toda rotação pode ser descrita utilizando três ângulos (em algumas áreas do conhecimento esses ângulos são chamados de *roll*, *pitch* e *yall*), um para cada eixo. A figura 1 apresenta uma visão geral da técnica proposta, demonstrando as etapas da mesma.

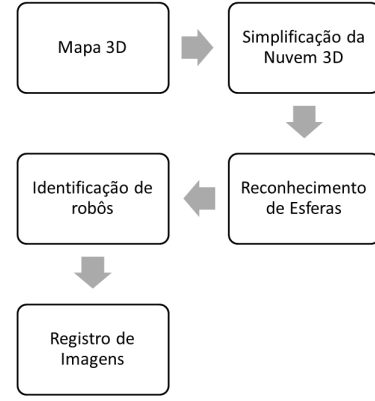


Figura 1. Visão geral da técnica proposta.

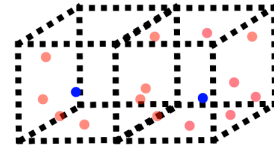


Figura 2. Ilustração do Voxel Grid. Os hexaedros representam os Voxels, os pontos vermelhos são os vértices da Nuvem de Pontos e os azuis são os centroides.

### A. Sub-amostragem

A simplificação de pontos presentes na Nuvem de Pontos utiliza a abordagem de Rede Voxelizada (*Voxel Grid*) [8]. Utilizando tal técnica, cria-se uma rede de Voxels de três dimensões na Nuvem de Pontos de entrada. Após isso, em cada Voxel, todos os pontos neles presentes são aproximados em seu centro geométrico, chamado de centroide, como observado na Figura 2.

Utilizando Voxel Grid, os centroides substituem todos os pontos dentro de um Voxel. Dessa forma, obtém-se uma Nuvem de Pontos com uma redução significativa no número de vértices e que devem representar o ambiente ou objeto original o mais próximo do possível, mesmo que algumas imprecisões ocorram [9].

### B. Reconhecimento de esferas

Depois de finalizada a etapa de sub-amostragem, a Nuvem de Pontos de entrada está adequada para ser processada pelo algoritmo de reconhecimento de esferas. Baseando-se no fato que secções horizontais de uma esfera são circunferências, como apresentado na Figura 3, a princípio é suposto que todos os vértices formam o centro de uma circunferência que pertencente à uma esfera. No decorrer do algoritmo, são aplicados diversos critérios que verificam se o vértice processado é realmente centro de uma esfera e, caso seja verificado um número fixo de circunferências concêntricas e de raio variável, uma esfera é encontrada.

Para realizar o reconhecimento de esferas, percorre-se cada vértice da Nuvem de Pontos realizando comparações com todos os outros vértices. Considerando  $p_1$  o ponto

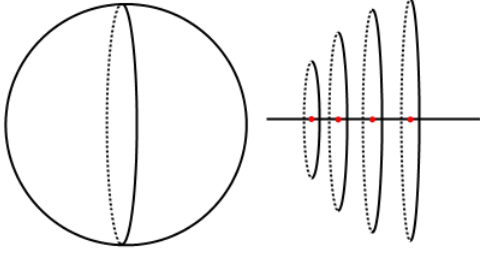


Figura 3. Circunferências concêntricas à direita representam seções verticais da esfera à esquerda.

atualmente processado e  $p_2$  o ponto a ser comparado, são calculadas três distâncias euclidianas:

- 1) Distância entre o centro e o ponto atual

$$d_{p_1} = \sqrt{p_{1_x}^2 + p_{1_y}^2 + p_{1_z}^2}; \quad (1)$$

- 2) Distância entre o centro e o ponto a ser comparado

$$d_{p_2} = \sqrt{p_{2_x}^2 + p_{2_y}^2 + p_{2_z}^2}; \quad (2)$$

- 3) Distância entre o ponto atual e o ponto a ser comparado (distância relativa)

$$d_{p_2 p_1} = \sqrt{(p_{2_x} - p_{1_x})^2 + (p_{2_y} - p_{1_y})^2 + (p_{2_z} - p_{1_z})^2}. \quad (3)$$

Após o cálculo das três distâncias, as primeiras verificações são realizadas. A ideia é filtrar os vértices pertencentes à circunferência de seção, os vértices nas proximidades da esfera ou aqueles que não fazem parte da esfera e estão próximos em um limite máximo. As seguintes verificações são realizadas.

1) *Distância relativa maior que o raio:* Caso a distância  $d_{p_1 p_2}$  (Equação 3) entre os pontos analisados for maior que o raio máximo definido, esses pontos não podem fazer parte de uma mesma esfera. O ponto é considerado um ponto ao redor se  $d_{p_1 p_2} < \delta$ , onde  $\delta$  é o limiar que determina a distância máxima externa à circunferência.

2) *Distância do ponto atual maior que o ponto comparado:* No caso  $d_{p_1} > d_{p_2}$  e a quantidade de pontos mais próximos ao centro for menor que um limiar  $\alpha$ , esse ponto analisado não é o centro da esfera por existir outro ponto que, à princípio, também faz parte da esfera e está mais próximo do centro. Por outro lado, se o número de pontos que sejam mais próximos for maior que  $\alpha$ , esse é mais um ponto que está mais perto do centro.

3) *Vértices pertencentes a um hemisfério:* Os vértices que passarem nas verificações de distâncias são considerados pertencentes a uma circunferência. Em complemento, é verificado se o vértice está em uma posição inferior ou superior da esfera, tomando como referência o plano formado pelo ponto candidato ao centro  $p_1$ , outro ponto pertencente a esfera que tenha a mesma coordenada  $y$  de  $p_1$  e o centro da Nuvem de Pontos. A soma das distâncias  $p_2$  dos pontos da região superior é representada por  $S_{p_2, pos}$ , por outro lado, na região inferior a soma é representada por  $S_{p_2, neg}$ . Tais informações são armazenadas para futuras verificações.

#### 4) Variação de pontos superiores e inferiores:

Considera-se que os vértices não formam uma esfera, mesmo que eles representem um objeto esferoide, caso a variação dos pontos da região superior e inferior (Seção II-B3) for alta. Esse critério é dividido em dois outros que serão apresentados a seguir.

- Diferença das médias

Nesse critério o cálculo da variação é realizado através da diferença  $\Delta_1$  entre a média das distâncias superiores (positivas) e inferiores (negativas). Considerando  $p$  a quantidade de pontos na região positiva e  $g$  a quantidade de pontos na região negativa,  $\Delta_1$  é dado por

$$\Delta_1 = \frac{S_{p_2, pos}}{p} - \frac{S_{p_2, neg}}{g}. \quad (4)$$

Caso o valor de  $\Delta_1$  seja elevado, desconsidera-se a possibilidade dos pontos pertencerem a uma esfera.

- Cálculo da percentagem

Através do cálculo percentual da variância  $\Delta_2$ , este critério verifica a simetria no eixo vertical do subconjunto de vértices que formam uma esfera. Primeiramente calcula-se o valor absoluto da diferença entre o número de pontos da semiesfera superior  $p$  e o número de pontos da semiesfera inferior  $g$ . Esse valor é dividido pelo número total de pontos da esfera  $p + g$ .

$$\Delta_2 = \frac{abs(p - g)}{p + g} \quad (5)$$

Caso o valor de  $\Delta_2$  seja elevado, desconsidera-se a possibilidade dos pontos pertencerem a uma esfera.

#### 5) Intervalo da distância média das circunferências:

Calcula-se a distância  $d_{p_1 p_2}$  média entre os pontos de cada circunferência  $t$  e é verificado se tal média está entre um limite inferior  $L_t^i$  e um limite superior  $L_t^s$  pré-determinados.

$$L_t^i < \frac{\sum_0^m d_{p_1 p_2}}{m} < L_t^s$$

A média é calculada dividindo a soma das distâncias  $d_{p_1 p_2}$  dos pontos pertencentes a circunferência  $t$ , pelo número total de vértices  $m$  de  $t$ . Caso o resultado não esteja no intervalo definido, não se considera esfera.

6) *Hierarquia das circunferências:* Uma esfera é formada por várias circunferências e, quanto mais distante da origem dos eixos uma circunferência estiver, maior será o número de pontos que a formam. Esse critério verifica um número mínimo de vezes que uma circunferência mais externa deve ser maior que a mais interna, como visto no Algoritmo 1.

Percorrem-se todas as circunferências encontradas verificando se a circunferência  $t$  é um número fixo de vezes maior que a circunferência  $t - 1$ . Caso positivo, o ponto atual é considerado de uma esfera.

---

**Algoritmo 1** Verificação da hierarquia das circunferências

---

```
for  $t = 1$  to  $n$  do
  if circulo[t].numVertices > circulo[t-1].numVertices *
    multiplicador[t-1] then
    esferas.push( $p_1$ ) {ponto atual}
  end if
end for
```

---

7) *Pontos ao redor*: Considera-se uma esfera caso a relação entre o número de pontos ao redor  $n_r$  e o número total de pontos  $p + g$  pertencentes a esfera não seja maior que um valor limiar  $\beta$ , dada por

$$\frac{n_r}{p + g} < \beta. \quad (6)$$

Por exemplo, de acordo com a fórmula 6,  $\beta = 0.1$  significa dizer que a soma da quantidade dos vértices localizados ao redor da esfera não deve ser maior que 10% do número total de vértices que formam a esfera.

### C. Identificação de robôs

Dado que já foram encontradas todas as esferas presentes na Nuvem de Pontos, o objetivo desta etapa é encontrar quais pares de esferas são compatíveis com as configurações válidas de um robô, e apontar qual é a esfera superior e inferior.

Para cada esfera encontrada, compara-se com todas as outras verificando as condições do que é um robô. Neste projeto, a única verificação realizada é a distância entre as esferas, que estão configuradas em uma distância entre 11 e 40 centímetros. Todavia, outros critérios poderiam ser aplicados como, por exemplo, a distância no eixo  $x$ ,  $y$  ou  $z$  e tamanho do raio mínimo e máximo.

Para encontrar as esferas inferiores e superiores, percorre-se todos os pares de esferas verificando quais delas possuem o maior componente  $y$  da coordenada do centro. Feito isso, todos os parâmetros de esferas são passados para o algoritmo de registro de imagens, que realizará a junção dos devidos mapas.

A vantagem de se utilizar esferas como referência no Registro de Imagens é que, ao ser escaneada pela câmera LIDAR, elas são vistas da mesma forma mesmo de diferentes ângulos. Dessa forma, a identificação de robô se torna mais robusta e fiel.

### D. Registro de imagens

Um Veículo Autônomo equipado de uma câmera LIDAR se move em torno de robôs com esferas que realizam a exploração ou mapeamento de um determinado ambiente. Como resultado, haverá vários mapas de um mesmo cenário, porém em diferentes ângulos. Faz-se necessário aplicar uma técnica capaz de realizar a junção desses mapas [10], chamada de Registro de Imagens (*Image Stitching*), que utiliza as esferas como ponto de referência.

Para mais de dois mapas, a técnica desenvolvida propõe realizar a junção de dois mapas e aplicar o registro

novamente no mapa resultante com os demais, e assim sucessivamente. Dessa forma, o registro de imagens é realizado várias vezes, porém apenas com dois mapas por vez.

O Registro de Imagens é realizado em duas etapas: translação e rotação. Na etapa de translação, duas nuvens de pontos são mescladas através do deslocamento de apenas um mapa, somando-se todos os vértices com o valor de diferença. Já a rotação é realizada tomando como referência a esfera inferior e utilizando os ângulos de Euler em uma matriz de rotação.

1) *Translação*: Considerando um robô  $R$ , com o centro da esfera inferior na coordenada  $R_b$  e outro robô  $T$  presente no mapa a ser transladado, com esfera inferior em  $T_b$ , deve-se calcular as diferenças entre as coordenadas dos centros.

$$\begin{aligned} d_x &= R_b.x - T_b.x \\ d_y &= R_b.y - T_b.y \\ d_z &= R_b.z - T_b.z \end{aligned} \quad (7)$$

Com as diferenças calculadas, deve-se preencher uma matriz  $4 \times 4$  que realizará a translação e multiplicar tal matriz com vértices do mapa a ser transladado, como mostrado na equação 8.

$$\begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x + d_x \\ y + d_y \\ z + d_z \\ 1 \end{bmatrix} \quad (8)$$

Essa operação descrita na equação 8 deve ser feita em todos os vértices presentes na Nuvem de Pontos, resultado no mapa transladado.

2) *Rotação*: De acordo com o Teorema de Rotação de Euler, qualquer rotação deve ser descrita usando três ângulos [11]. Se as rotações são descritas em termos das matrizes  $A_1, A_2, A_3$ , a matriz de rotação  $A$  pode ser descrita por:

$$A = A_1 A_2 A_3 \quad (9)$$

Os três ângulos que geram as três matrizes de rotação são chamados de Ângulos de Euler. A definição dos ângulos de Euler não é única e, por isso, optou-se utilizar a convenção-x [7]. De acordo com essa convenção, as rotações nos três ângulos de Euler ( $\phi, \theta, \psi$ ) podem ser:

- 1) No eixo-z, usando a matriz  $A_1$  no ângulo  $\psi$ .
- 2) No eixo-y, usando a matriz  $A_2$  no ângulo  $\theta$ ;
- 3) No eixo-x, usando a matriz  $A_3$  no ângulo  $\phi$ ;

Além disso, como apresentado por Milligan [12], as matrizes individuais que formam a matriz de rotação são dadas por

$$A_1 = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (10)$$

$$A_2 = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & \cos \theta & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix}, \quad (11)$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix}. \quad (12)$$

Aplicando as matrizes apresentadas nas Equações 12, 11, 10 na equação 9, temos a matriz  $A$  com os componentes  $a_{ij}$ :

$$\begin{aligned} a_{00} &= \cos \theta \cos \psi \\ a_{01} &= \cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi \\ a_{02} &= \sin \phi \sin \psi - \cos \phi \sin \theta \cos \psi \\ a_{10} &= -\cos \theta \sin \psi \\ a_{11} &= \cos \phi \cos \psi - \sin \phi \sin \theta \sin \psi \\ a_{12} &= \sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi \\ a_{20} &= \sin \theta \\ a_{21} &= -\sin \phi \cos \theta \\ a_{22} &= \cos \phi \cos \theta \end{aligned} \quad (13)$$

Consideramos que o robô com o LIDAR nunca está inclinado com relação ao eixo  $X$  e  $Y$ . Sendo assim, apenas a rotação no eixo- $Z$  é realizada e os ângulos  $\theta$  e  $\phi$  são nulos pelo fato de que não há rotação no eixo- $X$  e eixo- $Y$ . Dessa forma, deve-se calcular o melhor valor do ângulo  $\psi$ , que resultará na menor distância de rotação possível.

Para isso, percorre-se todo ciclo trigonométrico  $[0, 2\pi]$  tomando como base um valor de precisão. Note que esse valor de precisão é necessário porque  $\psi$  pode assumir valores contínuos entre  $[0, 2\pi]$ , dessa forma é preciso ter um valor mínimo de incremento para varrer todo o ciclo trigonométrico. Todo ângulo percorrido é utilizado para o cálculo da matriz  $A$ , que deve ser multiplicada pelo centro da esfera inferior, obtendo o novo centro rotacionado. Esse cálculo é dado por

$$A \cdot \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} x_f \\ y_f \\ z_f \end{bmatrix}. \quad (14)$$

A cada iteração deve-se armazenar o valor de  $\psi$  e a distância entre o centro antes e depois da rotação, visando a possibilidade de verificar qual valor de  $\psi$  resultará na menor distância, quando todo ciclo trigonométrico for percorrido.

Utilizando o melhor valor de  $\psi$ , aplicamos a Equação 14 novamente. Entretanto, tal equação será aplicada em todos os vértices da Nuvem de Pontos, e não apenas no centro da esfera, como foi feito no passo anterior. Após esse passo, a Nuvem de Pontos obtida já é o resultado da junção de outras duas.

### III. RESULTADOS

Os resultados apresentados nessa sessão foram realizados utilizando os arquivos PCD (*Point Cloud Data*) disponibilizados no *website*<sup>1</sup> do grupo de pesquisa do qual esse projeto faz parte. Para avaliar o desempenho, o algoritmo

<sup>1</sup>Acessível em <http://rpddi.ecomp.poli.br/aratu/scans.php>

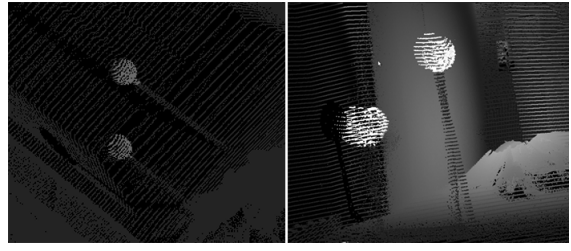


Figura 4. Detecção de esferas na Nuvem de Pontos (esquerda) e registro de imagens (direita). As esferas reconhecidas estão destacadas em branco.

foi executado variando a quantidade e a configuração das esferas na Nuvem de Pontos. A Figura 4 apresenta os resultados da técnica aplicada em uma Nuvem de Pontos.

A tabela II apresenta os valores médios dos tempos de execução medidos. Para cada Nuvem de Pontos, a tabela apresenta o número de esferas pertencentes aos robôs, o número de esferas totais, já que em algumas *Nuvem de Pontos* também existiam esferas que não pertenciam a robôs, o número de esferas reconhecidas pelo algoritmo, o número de robôs detectados e o tempo de execução em segundos.

Tabela II. TEMPO DE EXECUÇÃO - RECONHECIMENTO DE ESFERAS

Vértices	Esferas de robô	Esferas Totais	Esferas reconhecidas	Robôs encontrados	Tempo (s)
2157000	4	5	5	2	24,09
2157000	4	5	4	2	23,76
2157000	2	3	3	1	23,83
2157000	2	3	3	1	24,09
2157000	2	4	4	1	25,89
2157000	4	5	5	2	25,28
2157000	2	2	2	1	25,93
2157000	2	3	2	1	23,80

Como pode ser observado, o algoritmo apresentou uma alta taxa de reconhecimento. Entretanto, ainda houve esferas que não foram detectadas, mas nessa amostragem, não resultou em resultados insatisfatórios porque tais esferas não pertenciam aos robôs.

Como o registro de imagens é realizado utilizando-se os resultados do reconhecimento de esferas, o tempo de execução do algoritmo de *Stitching* depende do tempo da detecção de esferas em cada Nuvem de Pontos. Considerando apenas as operações do registro de imagens, o cálculo de  $\psi$  e da matriz  $A$  (Equação 9) são desprezáveis, por serem operações matemáticas de baixo custo computacional.

Tabela III. TEMPO DE EXECUÇÃO - STITCHING (MAPEAMENTO)

Vértices	Tempo (s)
4512020	62,03
215700	23,00
1555247	13,26
737810	8,19

O que torna o algoritmo custoso é o mapeamento da matriz de rotação na Nuvem de Pontos. Tal operação

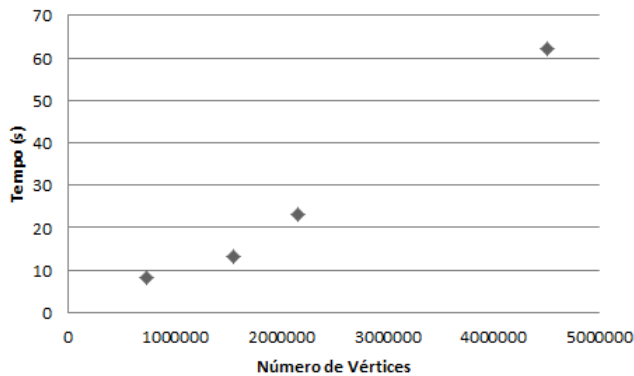


Figura 5. Gráfico que mostra a tendência linear do algoritmo de Registro de Imagens.

depende diretamente da quantidade de vértices da nuvem, apresentando um crescimento linear, como pode ser observado na Figura 5.

#### IV. CONCLUSÃO

Neste trabalho foi apresentado um estudo sobre como solucionar o problema da identificação de veículos autônomos. Tal problema foi solucionado utilizando um algoritmo de reconhecimento de esferas na Nuvem de Pontos geradas pelos escaneamentos do ambiente.

O trabalho inicia com conceitos básicos do que é uma câmera LADAR, uma breve explicação do funcionamento básico da exploração e mapeamento de terreno, e descrição do mapa *Point Cloud*. Com base nesse mapa, é realizada a simplificação de pontos, de forma a resolver os problemas iniciais da grande quantidade de vértices presentes no mapa. Utilizou-se uma técnica chamada *Voxel Grid* que se mostrou adequada por preservar a informação geométrica da Nuvem de Pontos e pelo baixo custo computacional.

No decorrer do trabalho também foi apresentado como o reconhecimento de esferas foi desenvolvido. A técnica proposta visa encontrar várias circunferências concêntricas no eixo  $y$  e o ponto central da esfera que é o ponto mais externo da esfera e mais próximo da origem dos eixos. Tal abordagem é realizada através de várias verificações, anteriormente apresentadas. As verificações utilizadas baseiam-se principalmente nas distâncias dos pontos ao centro e entre os pontos.

Além disso, ao encontrar as esferas no mapa, apresentou-se como foi realizada a identificação do robô. Como cada robô é equipado com uma configuração especial de um par de esferas, a etapa de identificação do robô visa encontrar pares de esferas que cumpram os requisitos mínimos, como distâncias mínimas entre as esferas, raio mínimo e máximo, por exemplo.

Por fim, foi demonstrado como se deu a implementação do registro de imagens. Essa etapa foi realizada através da utilização da rotação e translação do mapa. A primeira foi realizada deslocando os pontos de apenas uma Nuvem de Pontos em valores de diferença. Já na segunda, utilizou-se o Teorema de Rotação de Euler. Seguindo esse teorema,

a rotação foi feita através das rotações de cada eixo, utilizando três ângulos chamados de ângulos de Euler.

Os resultados se mostraram satisfatórios no reconhecimento de esfera. Todavia, no Registro de Imagens considera-se que os Veículos Autônomos possuem mesma coordenada-Z, o que em situação reais pode ocorrer de eles não estarem nivelados. No futuro, é pretendido melhorar nesse aspecto utilizando um acelerômetro para calcular a inclinação do robô.

#### AGRADECIMENTOS

O projeto é desenvolvido em parceria com a empresa americana SPI Team e o grupo de pesquisa de Reconhecimento de Padrões e Processamento Digital de Imagens (RPPDI) da Universidade de Pernambuco (UPE). É financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) através do programa de bolsas de Iniciação Científica.

#### REFERÊNCIAS

- [1] J. Urbach, T. Fislis, and G. Starkweather, "Laser scanning for electronic printing," *Proceedings of the IEEE*, vol. 70, no. 6, pp. 597–618, 1982.
- [2] G. Biegelbauer and M. Vincze, "Efficient 3d object detection by fitting superquadrics to range image data for robot s object manipulation," in *2007 IEEE International Conference on Robotics and Automation, ICRA 2007, 10-14 April 2007, Roma, Italy*. IEEE, 2007, pp. 1086–1091.
- [3] C. Ziyang, "Retrieval of 3d models based on spherical harmonics," in *Electrical and Control Engineering (ICECE), 2010 International Conference on*, 2010, pp. 2991–2994.
- [4] D. J. Feng and W. Wijesoma, "Improving rao-blackwellised genetic algorithmic filter slam through genetic learning," in *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, 2008, pp. 1200–1205.
- [5] M. Bansal, B. Matei, B. Southall, J. Eledath, and H. Sawhney, "A lidar streaming architecture for mobile robotics with application to 3d structure characterization," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 1803–1810.
- [6] L. Wang, B. Yuan, and Z. Miao, "3d point clouds parameterization algorithm," in *Signal Processing, 2008. ICSP 2008. 9th International Conference on*, 2008, pp. 1410–1413.
- [7] J. Kuffner, "Effective sampling and distance metrics for 3d rigid body path planning," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 4, 2004, pp. 3993–3998 Vol.4.
- [8] S.-H. Lee, S.-G. Jeong, T.-Y. Chung, and C.-S. Kim, "Real-time acquisition and representation of 3d environmental data," in *Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, 2012, pp. 1–5.
- [9] M. Jones and R. A. Satherley, "Shape representation using space filled sub-voxel distance fields," in *Shape Modeling and Applications, SMI 2001 International Conference on.*, 2001, pp. 316–325.
- [10] J. Lee, N. Kong, J. Yoo, and P. Park, "A fast image stitching algorithm in the endless hot rolling process," in *Control, Automation and Systems (ICCAS), 2011 11th International Conference on*, 2011, pp. 1264–1268.
- [11] H. Burger, "Use of euler-rotation angles for generating antenna patterns," *Antennas and Propagation Magazine, IEEE*, vol. 37, no. 2, pp. 56–63, 1995.
- [12] T. Milligan, "More applications of euler rotation angles," *Antennas and Propagation Magazine, IEEE*, vol. 41, no. 4, pp. 78–83, 1999.