Preensão robótica seletiva em 6D utilizando Redes Neurais Convolucionais

Caio Cristiano Barros Viturino * **André Gustavo Scolari Conceição** *

* *LaR - Laboratório de Robótica, Departamento de Engenharia Elétrica e de Computacao, Universidade Federal da Bahia, BA, (e-mails: engcaiobarros@gmail.com e andre.gustavo@ufba.br).*

Abstract: This paper proposes a visual system to generate multiple and selective 6D grasps using a Convolutional Neural Network (CNN) called GraspNet and a modified version of a CNN entitled Single Shot Multibox Detector (SSD). GraspNet is a robotic grasping technique, based on Variational Autoencoders, to generate grasps from the point cloud of the object and the end effector. It is capable of generating several grasps for a single object, enabling the exploitation of viable kinematic solutions to avoid collisions with other objects in the robot's workplace. However, this technique does not allow the selection of specific objects to grasp. To mitigate this problem, a modified version of the SSD was applied for the detection and selection of objects. The performance analysis of the proposed system is presented through simulation results in ROS/Gazebo with parts of complex geometry.

Resumo: Este artigo propõe um sistema visual de geração preensões múltiplas, seletivas e em 6D, utilizando uma Rede Neural Convolucional (CNN) denominada de *GraspNet* e uma versão modificada da CNN intitulada *Single Shot Multibox Detector* (SSD). A *GraspNet* é uma técnica de preensão robótica, baseada nos Autoencoders Variacionais, para gerar preensões a partir da nuvem de pontos do objeto e do efetuador final. Ela é capaz de conceber diversas preensões para um único objeto, possibilitando a exploração das soluções cinemáticas viáveis para evitar colisões com outros objetos na área de trabalho do robô. No entanto, essa técnica não permite ao manipulador efetuar preensões em objetos de forma seletiva. Para mitigar este problema, uma versão modificada da SSD foi adotada para a detecção e seleção de objetos para a posterior preensão. A análise de desempenho do sistema proposto é apresentada através de resultados de simulação no ROS/Gazebo com peças de complexidade geométrica.

Keywords: Robotic, grasping, computer vision, manipulation, robot grasping detection

Palavras-chaves: Robótica, preensão, visão computacional, manipulação, detecção de preensão robótica.

1. INTRODUÇÃO

A preensão robótica possui alta complexidade em ambientes não estruturados, como casas ou armazéns, em que os objetos a serem manipulados dispõem de geometrias diversas e heterogêneas. A natureza pluridisciplinar da robótica tornou-se o alicerce do seu desenvolvimento exponencial nos últimos anos. Com a cooperação da Inteligência Artificial e Visão Computacional, o progresso tecnológico observado na área de robôs autônomos, sejam eles móveis ou manipuladores, foi notoriamente relevante, excedendo os métodos clássicos, baseados em abordagens analíticas e empíricas, em relação a performance e robustez (Morrison et al., 2018; Mousavian et al., 2019).

No sentido de gerar preensões em ambientes não estruturados – não reconhecidos previamente - os robôs devem analisar, de forma independente e contínua, os dados inerentes ao espaço de trabalho, a fim de permitir medidas e ações decisórias a eventos fortuitos (Danielczuk et al., 2019). Outrossim, a essência da preensão robótica integra a percepção do ambiente; dessa forma, processamentos computacionais eficientes tornam-se requisitos para os algoritmos pertencentes à referida conjunção (Lenz et al., 2015). Técnicas de preensão robótica baseadas em aprendizagem profunda têm revelado uma alta capacidade e eficiência na solução dos problemas supracitados (Morrison et al., 2019; Mousavian et al., 2019).

Robôs manipuladores podem ser programados manualmente para execução de tarefas detalhadas no algoritmo de controle. Essa abordagem é denominada de analítica ou geométrica, comumente designada de características planejadas a mão (do inglês: *hand-designing features*). Essa metodologia foi copiosamente utilizada em pesquisas passadas (Maitin-Shepard et al., 2010; Kragic and Christensen, 2003).

A despeito dos resultados satisfatórios em ambientes conhecidos, os métodos analíticos aplicados à preensão robótica não são praticáveis em ambientes dinâmicos e não estruturados (Kober and Peters, 2010). Métodos analíticos consideram que a cinemática, dinâmica, geometria e posição do objeto, além dos seus contatos com o efetuador final são inteiramente conhecidos, portanto, pouco aplicá-

veis à prática devido a sua complexidade de modelagem (Morrison et al., 2018).

Contrário aos métodos analíticos, a imprescindibilidade de um modelo analítico completo é suprimida parcial ou completamente por métodos empíricos (Lenz et al., 2015). Esses métodos utilizam aprendizado de máquina para conceber modelos que associem dados rotulados por humanos aos dados de sensores RGB+D.

No que tange as técnicas baseadas em aprendizagem profunda, como as Redes Neurais Convolucionais (CNN), há um aporte considerável na preensão robótica aplicada a objetos desconhecidos. Por meio dessas, é possível extrair importantes características de objetos que auxiliem na geração de preensões, superando a performance e robustez alcançadas por meio de modelos empíricos e analíticos (Morrison et al., 2018; Mahler et al., 2017; Lenz et al., 2015; Johns et al., 2016).

Não obstante os desempenhos de técnicas recentes baseadas em CNN apontados em Ribeiro and Grassi (2019), Viereck et al. (2017), Mahler et al. (2017), Johns et al. (2016), Levine et al. (2016), Pinto and Gupta (2016) e Lenz et al. (2015), essas não ostentaram resultados satisfatórios em sistemas de malha fechada e de tempo real, em virtude dos seus altos custos computacionais e da limitação em não considerar as melhores preensões possíveis entre as geradas, dado um conjunto de objetos desconhecidos e desordenados. A solução do problema da eficiência computacional foi proposta por Morrison et al. (2018), apesar de considerar apenas preensões planares e únicas, ou seja, uma preensão factível por objeto. Mousavian et al. (2019) alvitraram uma técnica geradora de múltiplas preensões por objeto em seis dimensões. Desse modo, há a possibilidade de originar diferentes soluções cinemáticas para o robô que desconsiderem possíveis colisões com outros objetos no ambiente.

Tratando-se da quarta revolução industrial, existe uma alta demanda por manipuladores robóticos flexíveis e versáteis, capazes de lidar com novos ambientes e produtos (Costa et al., 2020). Nesse aspecto, a manufatura aditiva tem demonstrado um interesse crescente por esses sistemas, a qual regularmente lida com um número crescente de novas peças (Arrais et al., 2019). Diante do anteposto, necessita-se de um método de preensões múltiplas, não planares e seletivas - quando é possível selecionar um objeto específico do ambiente -, com alta eficiência computacional e capaz de agarrar objetos complexos, desconhecidos e desordenados.

Em consequência disso, este trabalho estende os trabalhos anteriores dos autores Viturino et al. (2020), Lemos et al. (2019) e de Oliveira et al. (2020). A performance das redes conhecidas como *Faster R-CNN*, SSD300 e SSD512, com base ResNet50 (He et al., 2016) foram avaliadas por Lemos et al. (2019) na detecção de objetos produzidos por impressora 3D (manufatura aditiva). de Oliveira et al. (2020) desenvolveram um sistema de preensão robótica baseado em primitivas geométricas para estimar a posição e orientação do objeto e Viturino et al. (2020) integraram uma rede de detecção de objetos, baseando-se no trabalho de Lemos et al. (2019), a uma CNN capaz de gerar preensões robóticas seletivas, planares e únicas de alta performance computacional.

No presente trabalho, busca-se integrar uma rede de detecção de objetos, conhecida como *Single Shot Multibox Detector* (SSD) (Liu et al., 2016) ao método de geração de preensões múltiplas e em 6D, denominado de GraspNet (Mousavian et al., 2019), para que seja possível realizar preensões seletivas. A investigação de desempenho do método proposto é exposta através de resultados de simulação no ROS/Gazebo com peças de complexidade geométrica.

O restante do artigo está organizado da seguinte forma: serão discutidos os trabalhos relacionados na Seção 2. A definição do problema será tratada na Seção 3. As redes SSD e GraspNet serão apresentadas, respectivamente, nas Seções 4 e 5. O método de preensão proposto será descrito na Seção 6. Os protocolos experimentais serão detalhados na Seção 7. Os resultados serão evidenciados na Seção 8. Por fim, a Seção 9 trará a conclusão deste trabalho.

2. TRABALHOS RELACIONADOS

Quando aplicadas à preensão robótica, as CNNs necessitam de dados profusos de treinamento, os quais poderiam requerer meses de observação e treinamento utilizando robôs reais. Em virtude disso, autores têm utilizado simulações de *software* para obtê-los (Mousavian et al., 2019; Mahler et al., 2017; Johns et al., 2016) ou banco de dados elaborados por meio de objetos e câmeras reais, com preensões predeterminadas e rotuladas manualmente (Morrison et al., 2018, 2019).

No que se refere à modalidade dos dados aplicados na inferência e treinamento de técnicas de preensão robótica, certos métodos utilizam somente imagens de profundidade (Morrison et al., 2019, 2018; Mahler et al., 2017), nuvem de pontos (Mousavian et al., 2019) ou a união de imagens de cores e profundidade (Mahler et al., 2017).

Preensões geradas podem ser únicas (Morrison et al., 2018; Johns et al., 2016) ou múltiplas (Mousavian et al., 2019; Gualtieri et al., 2016). Esta, referente à concepção de mais de uma preensão factível por objeto; aquela, às preensões que convergem para uma única solução viável por objeto. Preensões múltiplas ainda são alvos de poucas investigações devido à necessidade de exploração do espaço tridimensional composto pelo objeto e seu alto custo computacional relacionado, se comparado às técnicas de preensão única. Entretanto, elas são de fundamental importância, dado que nem toda preensão gerada é cinematicamente viável ou está em rota de colisão com outros objetos no ambiente (Mousavian et al., 2019). Portanto, torna-se imprescindível a obtenção de preensões múltiplas robustas, dado o mesmo objeto no ambiente.

A representação espacial da preensão robótica pode ser determinada de duas formas: planar (Morrison et al., 2018) e em 6D (Mousavian et al., 2019). Essa, refere-se à possibilidade de exploração completa do espaço tridimensional Euclidiano; aquela, à limitação da orientação da preensão, de forma que o efetuador final deve aproximar-se do objeto ortogonalmente ao plano composto por esse.

O trabalho de Levine et al. (2016) foi um dos primeiros a associar técnicas de aprendizagem profunda à preensão robótica de objetos desordenados em malha fechada. No entanto, A rede neural utilizada possui um grande número de camadas, cerca de um milhão de parâmetros, resultando

em frequências experimentais que variam entre 2 e 5 Hz. Viereck et al. (2017) supera a limitação diminuindo o número de camadas da rede neural, alcançando uma frequência de processamento de 5 Hz, não sendo ainda adequada para aplicações em malha fechada. Morrison et al. (2018) desenvolveram uma técnica de preensão robótica denominada *Generative Grasping Convolutional Neural Network* (GG-CNN), uma CNN de apenas 6 camadas e 62 mil parâmetros. A GG-CNN foi capaz de gerar uma preensão à aproximadamente 50 Hz e sua taxa de sucesso foi de 83% nos objetos citados, superando métodos anteriores no quesito performance e robustez. No entanto, apenas uma preensão planar factível é gerada por imagem.

Mousavian et al. (2019) sugeriram um método de preensão robótica denominado de *Graspnet*, baseado nos *Autoencoders Variacionais* (Kingma and Welling, 2013). Através desse método, diversas preensões 6D factíveis são geradas a partir de nuvens de pontos. A diversidade de preensões favorece a sua execução em ambientes complexos, aumentando a viabilidade cinemática. No entanto, não é possível selecionar um objeto específico no espaço de trabalho do robô.

3. DEFINIÇÃO DO PROBLEMA

3.1 Premissas

No presente trabalho, o sistema proposto deve gerar preensões seletivas múltiplas e em 6D, utilizando um sensor RGB+D. É admitido que os objetos são posicionados em uma superfície plana e a preensão é realizada por um efetuador final de dedos paralelos usando uma câmera posicionada no punho do robô. Todos os parâmetros internos da câmera são conhecidos. A preensão é definida pela posição e orientação do efetuador final de geometria conhecida no eixo de coordenadas da câmera.

3.2 Definições

Imagem RGB. $\boldsymbol{C} = \mathbb{R}^{H\times W\times 3}$ expressa uma imagem de cores RGB, em que todos os objetos no espaço de trabalho são considerados.

Imagem de profundidade. $\boldsymbol{I} = \mathbb{R}^{H\times W}$ representa uma imagem de profundidade 2.5D, em que todos os objetos no espaço de trabalho são considerados.

Nuvem de pontos filtrada. $\boldsymbol{N_f}$ denota a nuvem de pontos do objeto selecionado.

Conjunto de preensões. $\tilde{\boldsymbol{G}} = (\tilde{\boldsymbol{P_f}}, \tilde{\boldsymbol{O_f}}) = ((\tilde{x}, \tilde{y}, \tilde{z}), (\tilde{R}, \tilde{P}, \tilde{Y}))$ descreve um conjunto de preensões em seis dimensões, onde $\tilde{R}$, $\tilde{P}$ e $\tilde{Y}$ representam, respectivamente, os eixos de rotação rolagem (do inglês: *roll*), arfagem (do inglês: *pitch*) e guinada (do inglês: *yaw*).

Preensão filtrada. $\boldsymbol{G_f} = (\boldsymbol{P_f}, \boldsymbol{O_f}) = ((x, y, z), (R, P, Y))$ evidencia a posição $\boldsymbol{P_f}$ e orientação $\boldsymbol{O_f}$ da preensão escolhida dentre o conjunto de preensões geradas $\tilde{\boldsymbol{G}}$ no eixo de coordenadas da câmera.

Preensão no eixo de coordenadas do robô. $\boldsymbol{G_{fb}} = (\boldsymbol{P_{fb}}, \boldsymbol{O_{fb}}) = ((x, y, z), (R, P, Y))$ denota a posição $\boldsymbol{P_{fb}}$ e orientação $\boldsymbol{O_{fb}}$ da preensão selecionada no eixo de coordenadas do robô.

Posição e orientação atual do robô. $\boldsymbol{G_a} = (\boldsymbol{P_a}, \boldsymbol{O_a}) = ((x, y, z), (R, P, Y))$ descreve a posição $\boldsymbol{P_a}$ e orientação $\boldsymbol{O_a}$ atual do efetuador final do manipulador robótico.

4. SINGLE SHOT MULTIBOX DETECTOR

A presente seção apresenta de forma sucinta o funcionamento da Single Shot Multibox Detector (SSD). Para maiores detalhes o trabalho de Liu et al. (2016) pode ser consultado.

A SSD é uma CNN capaz de identificar múltiplos objetos em uma imagem RGB. Duas variantes da SSD foram apresentadas em Liu et al. (2016): a SSD300 e a SSD512, as quais foram treinadas a partir de imagens de respectivamente $300 \times 300\,px$ e $512 \times 512\,px$. A SSD gera, para cada objeto reconhecido, as coordenadas de caixas delimitadoras que englobam os objetos, identificadores de classes e valores de confiança. Segundo o autor supracitado, a SSD supera a performance de trabalhos anteriores em termos de processamento.

No modelo original da SSD, as camadas iniciais são partes de uma rede projetada e treinada para a identificação de objetos, dispensando as camadas utilizadas para a classificação dela. A este grupo de camadas é dado o nome de rede base. Através da SSD, há a possibilidade de adotar redes bases distintas. A parte restante da rede, denominada de auxiliar, é composta por camadas convolucionais que decrescem em tamanho, de forma progressiva.

Neste trabalho, foram avaliadas as performances dos modelos SSD300, com base VGG16 (Simonyan and Zisserman, 2014) e a SSD512, com base ResNet50 (He et al., 2016) e VGG16, na detecção dos modelos 3D dos objetos mostrados na Figura 1. Esses objetos foram propostos por Mahler et al. (2017) para testes de preensão robótica devido a sua complexidade geométrica. As redes mencionadas foram pré-treinadas com a base de dados COCO (Lin et al., 2014) e VOC (Everingham et al., 2010).

A Tabela 1 mostra uma comparação das redes testadas para as quatro classes de objetos presentes na Figura 1. Percebe-se que a SSD512 com a rede de base ResNet50, pré-treinada com o dataset COCO obteve o maior mAP (do inglês: *mean Average Precision*) para cada IoU (do inglês: *Intersection over Union*) de 0.5, 0.75 e na média de 0.5 a 0.95, indicando maior robustez na detecção. Portanto, devido ao seu desempenho, a rede mencionada foi utilizada nos experimentos de preensão para detecção de objetos.

Tabela 1. Comparativo do mAP das redes SSD300 e SSD512 com diferentes IoUs.

Network	mAP@IoU			Precisão média por classe			
	0.5	0.75	0.5:0.95	Bar clamp	Vase	Part 1	Part 3
SSD300-VGG16-VOC	93	86.4	67.7	99.62	90.91	81.64	90.91
SSD512-ResNet50-COCO	**97.7**	**92.2**	**73.5**	100	100	90.91	99.86
SSD512-ResNet50-VOC	96.5	87.5	69	100	99.93	90.91	90.91
SSD512-VGG16-VOC	94.1	87	69.5	99.62	99.48	81.82	90.91
SSD512-VGG16-COCO	95.4	89.6	70.7	100	90.76	100	90.83

5. GRASPNET

Esta seção descreve o funcionamento da *GraspNet* de forma breve. Embora seja suficiente para a compreensão

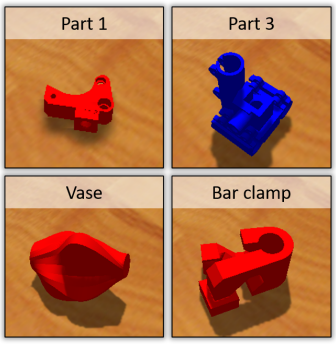

Figura 1. Objetos utilizados no treinamento da SSD512-ResNet50. Estes objetos foram propostos por Mahler et al. (2017) para testes de preensão robótica devido a sua complexidade geométrica.

do método proposto, para maiores detalhes, consulte Mousavian et al. (2019).

A GraspNet é uma CNN de geração de preensões múltiplas e em 6D, baseada em *Autoencoders* Variacionais (Kingma and Welling, 2013), modelos generativos profundos desenvolvidos com base nos *Autoencoders* clássicos. Essa rede consiste em um codificador e decodificador que mapeiam os dados de entrada para um espaço latente de dimensionalidade reduzida. O codificador e o decodificador utilizam a arquitetura da rede *PointNet++* (Qi et al., 2017) para extrair características espaciais de cada ponto da nuvem de pontos do objeto e do efetuador final do robô para cada preensão gerada, seja ela bem-sucedida ou não.

Essa rede possui dois módulos: gerador e avaliador. O módulo gerador baseia-se em diferentes amostras de um espaço latente e da nuvem de pontos parcial do objeto e da garra para produzir diferentes preensões e o módulo avaliador aceita ou rejeita as preensões com base na sua probabilidade de sucesso. A GraspNet foi treinada a partir de preensões obtidas no *software* FleX (Vicent et al., 2016). Das $10,8$ milhões de preensões amostradas, 2 milhões foram preensões bem-sucedidas. Durante o treinamento, foi realizada uma minimização da função de custo de reconstrução das preensões dada por

$$\mathcal{L}(g, \hat{g}) = \frac{1}{n} \sum \|\mathcal{T}(g; p) - \mathcal{T}(\hat{g}; p)\|_1 \quad (1)$$

Sendo $\mathcal{T}(\cdot)$ a representação da posição e orientação das preensões *ground-truths* g e das preensões geradas pelo decodificador $\hat{g}$.

Considerando o espaço latente z, a nuvem de pontos do objeto N_f e a preensão gerada $\hat{g}$, o codificador é responsável por mapear cada par $(N_f, \hat{g})$ para um espaço latente z e o decodificador reconstrói a preensão $\hat{g}$ através de z. Para garantir uma distribuição normal do espaço latente, a divergência de Kullback-Leibler é utilizada entre a saída do codificador $Q(\cdot|\cdot)$ e uma distribuição normal $\mathcal{N}(0, I)$, tal que a função custo é dada por

$$\mathcal{L}_{\text{vae}} = \sum_{z \sim Q, g \sim \tilde{G}} \mathcal{L}(\hat{g}, g) - \alpha \mathcal{D}_{KL}[Q(z \mid N_f, g), \mathcal{N}(0, I)] \quad (2)$$

O módulo avaliador da *GraspNet* associa uma probabilidade de sucesso a cada preensão, observando a nuvem de pontos parcial do objeto N_f tal que $P(S \mid g, N_f)$. Ele foi treinado utilizando preensões bem-sucedidas e falhas. A função custo aplicada no treinamento desse módulo é dada por

$$\mathcal{L}_{\text{avaliador}} = -(y \log(S) + (1 - y) \log(1 - S)) \quad (3)$$

sendo S a probabilidade de sucesso inferida da preensão e y o rótulo binário representando o *ground-truth*.

Além disso, há um processo de refinamento de preensões, a qual aplica um deslocamento na preensão, dado por Δg, para aumentar as suas chances de sucesso, de modo que

$$P(s = 1 \mid g + \Delta g) > P(s = 1 \mid g) \quad (4)$$

Durante a inferência da preensão, o codificador Q é removido e os valores latentes z são amostrados de $\mathcal{N}(0, I)$. É importante salientar que não foi realizado fine-tuning na GraspNet.

6. PROCESSO DE DETECÇÃO DE PREENSÕES

Primeiramente, no processo de detecção de preensões, um objeto deve ser selecionado no espaço de trabalho do robô de forma manual ou aleatória. Depois, os estágios seguintes são executados em cascata, como mostra a Figura 2.

No estágio 1, a imagem $\boldsymbol{C}$ é obtida do modelo virtual da câmera *Intel Realsense* no *Gazebo*.

No estágio 2, é executada uma tentativa de reconhecimento do objeto solicitado, utilizando a imagem $\boldsymbol{C}$ como entrada para a SSD, que retorna uma caixa delimitadora indicada no estágio 3.

No estágio 4, a caixa delimitadora gerada pela SSD na imagem $\boldsymbol{C}$ é copiada para a imagem de profundidade $\boldsymbol{I}$.

No estágio 5, a região descrita pela caixa delimitadora na imagem de profundidade $\boldsymbol{I}$ é transformada em uma nuvem de pontos $\boldsymbol{N_f}$ através do método conhecido como *back projection*, utilizando os parâmetros intrínsecos da câmera. O resultado é mostrado no estágio 6.

No estágio 7, a *GraspNet* recebe a nuvem de pontos filtrada $\boldsymbol{N_f}$ e retorna um conjunto de preensões $\tilde{\boldsymbol{G}}$.

No estágio 8, a partir de $\tilde{\boldsymbol{G}}$ é possível selecionar $\boldsymbol{G_f}$ que possua o maior *score* gerado pela *GraspNet*. No entanto foi observado que nem sempre uma preensão bem-sucedida possui essa característica. Diante disso, foi desenvolvida uma heurística para seleção de $\boldsymbol{G_f}$, considerando as preensões $\tilde{\boldsymbol{G}}$ que possuem um *score* acima de 70% e que $\boldsymbol{O_f}$ seja a mais próxima de $\boldsymbol{O_a}$.

$$\tilde{\Phi} = |O_a - \tilde{O}_f| \quad (5)$$

sendo $\tilde{\Phi}$ um vetor de diferenças entre O_a e $\tilde{O}_f$.

A preensão G_f é obtida de forma que

$$i_d = \max(i \cdot (1 - \text{sign}(\tilde{\Phi}_i - min(\tilde{\Phi})))); \;\; i = 0 \ldots p \quad (6)$$

sendo i_d o índice do menor Φ e p a quantidade de preensões geradas. Portanto,

$$\boldsymbol{G_f} = \tilde{G}(i_d) \quad (7)$$

No estágio 9, $\boldsymbol{G_{fb}}$ é obtida através de uma sequência de transformações homogêneas conhecidas, como em

$$\boldsymbol{G_{fb}} = t_{RC}(\boldsymbol{G_f}) \quad (8)$$

onde t_{RC} representa a transformação homogênea do sistema de coordenadas da câmera em relação a base do robô. A preensão final é mostrada no estágio 10.

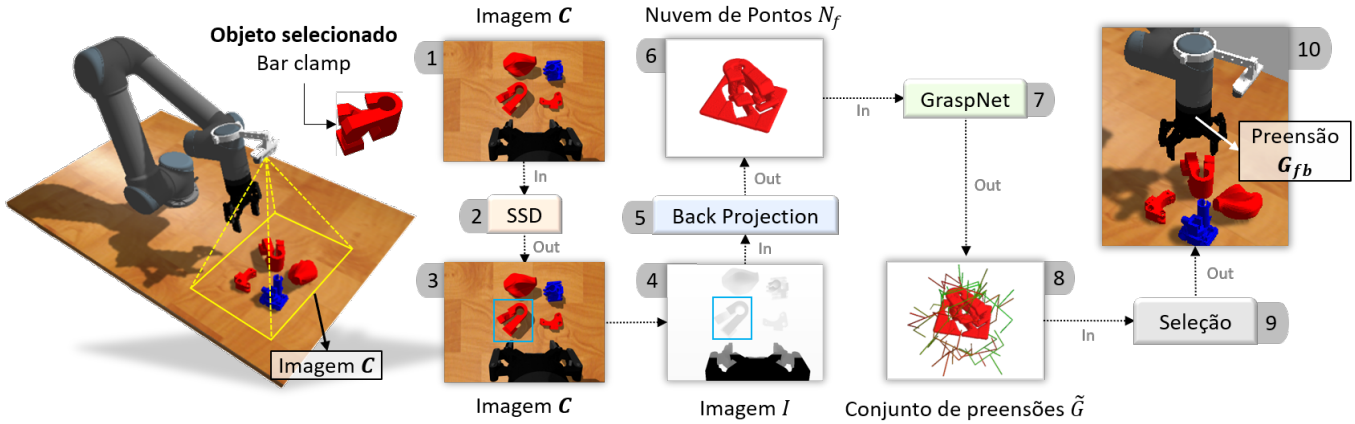

Figura 2. Processo de preensões múltiplas e em 6D, utilizando a rede neural Graspnet e a rede de detecção de múltiplos objetos denominada de SSD, para realização de uma preensão seletiva. As imagens retratam uma simulação realizada no Gazebo, utilizando o modelo virtual da câmera Intel Realsense, o manipulador robótico UR5 e o efetuador final Robotiq 2F-85.

7. SETUP EXPERIMENTAL

Neste trabalho, os experimentos foram realizados no simulador robótico Gazebo (Koenig and Howard, 2004) integrado ao *Robot Operating System* ROS (Quigley et al., 2009) versão *Kinetic*. Foi empregado um modelo virtual do robô UR5 da *Universal Robots* (Robots, 2019), equipado com o efetuador final 2F-85 da *ROBOTIQ* e com a câmera Intel Realsense D435. A distância entre a câmera e o ponto central entre os dedos do efetuador final é de, aproximadamente, 23 cm. Foi utilizado um *notebook* com o processador Intel Core i7 8550U, a placa de vídeo NVIDIA GeForce MX150 e o sistema operacional Ubuntu 16.04.

Nas simulações, as preensões foram realizadas em malha aberta e com múltiplos objetos. Após cada tentativa de preensão, os objetos são reorganizados no espaço de trabalho do robô aleatoriamente. As classes dos objetos empregados estão indicadas na Figura 1. É estabelecido que a tentativa de preensão é bem-sucedida quando um objeto de interesse é identificado e levantado 20 cm da superfície da mesa. Para obter os ângulos das juntas do robô, dadas a posição e a orientação da preensão, utilizou-se o TRAC-IK (Beeson and Ames, 2015), uma biblioteca *open-source* otimizada para solucionar problemas de cinemática inversa. O PyTorch foi utilizado para a aplicação da GraspNet e o GluonCV foi utilizado para implementar a SSD512-ResNet50

8. RESULTADOS

Foram realizadas 30 tentativas de preensão por objeto e após cada tentativa, esses foram reposicionados aleatoriamente no espaço. A Figura 3 revela algumas preensões bem-sucedidas e falhas da *GraspNet*.

Verifica-se que a *GraspNet* falha ocasionalmente ao tentar realizar uma preensão 6D em objetos pequenos, como o *Part 1*, evento facilmente observado na Figura 4. Da mesma forma, a geometria curvada do *Vase* provoca o deslizamento da peça, caso os pontos de pressão entre os dedos do efetuador final e a peça não gerem atrito suficiente para mantê-la em suspensão. Das 120 preensões realizadas, $58,3\%$ obtiveram sucesso e $29,2\%$ falharam.

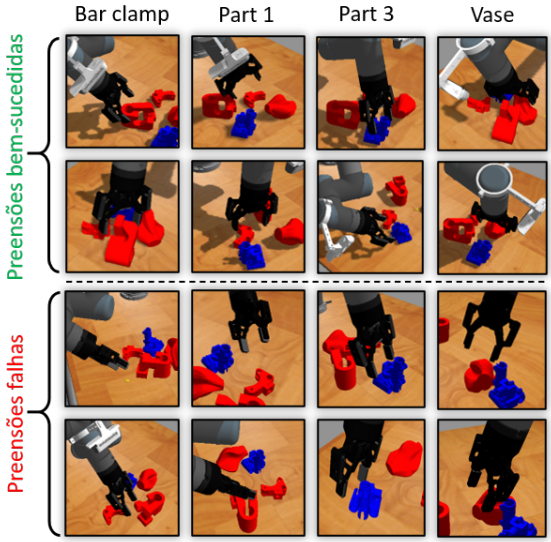

Figura 3. Tentativas de preensões bem-sucedidas e falhas realizadas nos objetos *Bar clamp, Part 1, Part 2* e *Vase* pela rede de geração de preensão *GraspNet*

Dessas tentativas, as detecções falharam em $12,5\%$. A taxa de acertos é representada como uma fração do número total de tentativas de preensão.

Para investigar a repetibilidade das preensões, foram determinadas aleatoriamente e mantidas duas posições quaisquer para cada peça da Figura 1, uma posição F e outra S. Foram executadas dez tentativas de preensão para cada posição F e S de cada objeto. A Figura 5 sugere que a *GraspNet* produz diferentes preensões para a mesma nuvem de pontos $\boldsymbol{N_f}$ do objeto. A Tabela 2 indica que a repetibilidade das preensões da *GraspNet* é desigual, de modo que a repetibilidade média foi de $77,5\%$. Por conseguinte, há $77,5\%$ de chances de gerar a mesma preensão consecutivamente, seja ela bem-sucedida ou falha. A repetibilidade encontrada é aceitável, considerando a robustez limitada da simulação em relação aos cálculos de atrito entre a peça e o efetuador final.

A Figura 6 exibe as amostras do tempo de processamento de dez preensões, levando em conta objetos em posições di-

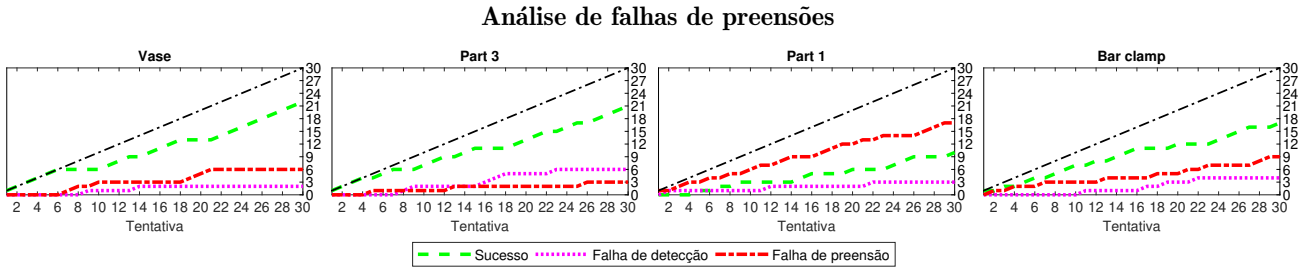

Figura 4. Preensões realizadas pela GraspNet, utilizando a SSD512-ResNet50-COCO para selecionar um objeto no espaço de trabalho do robô. Os objetos utilizados nos experimentos foram o *Vase, Part3, Part1* e *Bar clamp*. A cada tentativa de preensão, os objetos foram reorganizados de forma aleatória.

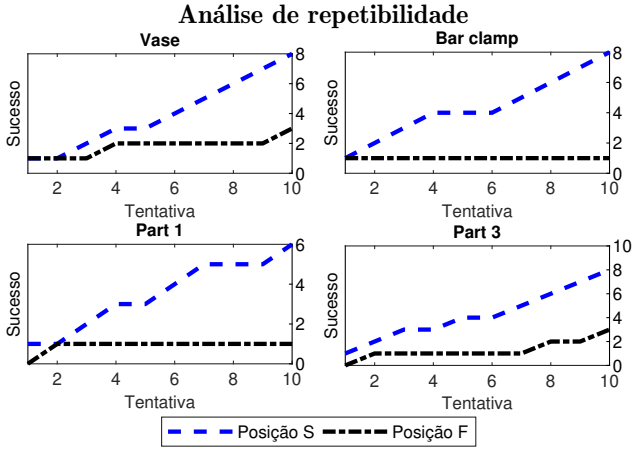

Figura 5. Preensões realizadas pela GraspNet, utilizando a SSD512-ResNet50-COCO, nos objetos da Figura 1. Para a análise de repetibilidade, a cada tentativa de preensão, os objetos são mantidos na mesma posição e orientação.

Tabela 2. Repetibilidade das preensões realizadas na Figura 5 para as posições F e S do *Vase, Bar clamp, Part 1* e *Part 3*.

	Repetibilidade de preensão			
	Vase	**Bar clamp**	**Part 1**	**Part 3**
Posição S	80%	80%	60%	80%
Posição F	70%	90%	90%	70%

ferentes. Cada inferência da SSD512-ResNet50 leva 50 ms com o *hardware* apresentado. É importante destacar que o código foi predominantemente escrito em Python e o *hardware* utilizado foi inferior ao encontrado em Mousavian et al. (2019). Apesar do tempo de processamento considerado da *GraspNet*, diversas preensões são geradas por objeto, característica importante para ampliar a viabilidade cinemática, considerando possíveis colisões com o ambiente de trabalho do robô.

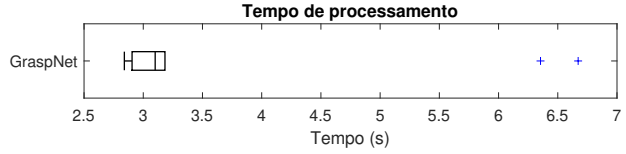

Figura 6. Tempo de processamento da *Graspnet* utilizando o *hardware* citado na Seção 7.

Enfatiza-se que os dados apresentados pertencem às análises preliminares praticadas em simulações computacionais. Evitou-se realizar testes presenciais no Laboratório de Robótica (LaR) da Universidade Federal da Bahia (UFBA) devido a pandemia do COVID-19. Esses serão realizados posteriormente.

9. CONCLUSÃO

Neste artigo, foi proposto um sistema de geração preensões seletivas, múltiplas e em 6D, que utiliza um sensor RGB+D, através da integração da *GraspNet* e uma versão modificada da arquitetura SSD, intitulada de SSD512-ResNet50-COCO. Considerando os objetos de geometrias diversas propostos por Mahler et al. (2017), a *GraspNet* obteve resultados melhores em peças maiores, como o *Vase, Part 3* e *Bar clamp*, dado que preensões 6D em peças pequenas resultam em colisões com a superfície sob elas. Vale notar que a *GraspNet* não foi treinada com objetos de geometrias complexas.

É fundamental ressaltar que a *GraspNet* utiliza somente a nuvem de pontos do objeto para gerar múltiplas preensões factíveis. A diversidade das preensões geradas possibilita a exploração de diversas soluções cinemáticas viáveis para eliminar aquelas que estão em colisão com outros objetos no espaço de trabalho do robô. Trabalhos posteriores serão conduzidos objetivando-se a melhoria da *GraspNet* na preensão de objetos de geometrias complexas, além da implementação da técnica proposta em ambientes reais.

AGRADECIMENTOS

Este estudo recebeu apoio financeiro do CNPQ termo de outorga numero 311029/2020-5. Os autores agradecem à FAPESB (Fundação de Amparo à Pesquisa do Estado da Bahia) pelo apoio financeiro. Este estudo foi financiado também pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

REFERÊNCIAS

Arrais, R., Veiga, G., Ribeiro, T.T., Oliveira, D., Fernandes, R., Conceição, A.G.S., and Farias, P. (2019). Application of the open scalable production system to machine tending of additive manufacturing operations by a mobile manipulator. In *EPIA Conference on Artificial Intelligence*, 345–356. Springer.

Beeson, P. and Ames, B. (2015). Trac-ik: An open-source library for improved solving of generic inverse kinematics. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 928–935. IEEE.

Costa, F.S., Nassar, S.M., Gusmeroli, S., Schultz, R., Conceição, A.G., Xavier, M., Hessel, F., and Dantas, M.A. (2020). Fasten iiot: An open real-time platform for vertical, horizontal and end-to-end integration. *Sensors*, 20(19), 5499.

Danielczuk, M., Kurenkov, A., Balakrishna, A., Matl, M., Wang, D., Martín-Martín, R., Garg, A., Savarese, S., and Goldberg, K. (2019). Mechanical search: Multi-step retrieval of a target object occluded by clutter. In *2019 International Conference on Robotics and Automation (ICRA)*, 1614–1621. IEEE.

de Oliveira, D.M., Lemos, C.B., and Conceição, A.G. (2020). Sistema de preensão robótica utilizando redes neurais convolucionais e primitivas geométricas. *Anais da Sociedade Brasileira de Automática*, 2(1). doi:10.48011/asba.v2i1.1097.

Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2), 303–338.

Gualtieri, M., Ten Pas, A., Saenko, K., and Platt, R. (2016). High precision grasp pose detection in dense clutter. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 598–605. IEEE.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Johns, E., Leutenegger, S., and Davison, A.J. (2016). Deep learning a grasp function for grasping under gripper pose uncertainty. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4461–4468. IEEE.

Kingma, D.P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kober, J. and Peters, J. (2010). Imitation and reinforcement learning. *IEEE Robotics & Automation Magazine*, 17(2), 55–62.

Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 3, 2149–2154. IEEE.

Kragic, D. and Christensen, H.I. (2003). Robust visual servoing. *The international journal of robotics research*, 22(10-11), 923–939.

Lemos, C., Farias, P., Simas Filho, E., and Scolari Conceicao, A. (2019). Convolutional neural network based object detection for additive manufacturing. In *2019 19th International Conference on Advanced Robotics (ICAR)*, 420–425. doi:10.1109/ICAR46387.2019.8981618.

Lenz, I., Lee, H., and Saxena, A. (2015). Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5), 705–724.

Levine, S., Pastor, P., Krizhevsky, A., and Quillen, D. (2016). Learning hand-eye coordination for robotic grasping with large-scale data collection. In *International Symposium on Experimental Robotics*, 173–184. Springer.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C.L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., and Berg, A.C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision*, 21–37. Springer.

Mahler, J., Liang, J., Niyaz, S., Laskey, M., Doan, R., Liu, X., Ojea, J.A., and Goldberg, K. (2017). Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv preprint arXiv:1703.09312*.

Maitin-Shepard, J., Cusumano-Towner, M., Lei, J., and Abbeel, P. (2010). Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *2010 IEEE International Conference on Robotics and Automation*, 2308–2315. IEEE.

Morrison, D., Corke, P., and Leitner, J. (2018). Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. *arXiv preprint arXiv:1804.05172*.

Morrison, D., Corke, P., and Leitner, J. (2019). Multi-view picking: Next-best-view reaching for improved grasping in clutter. In *2019 International Conference on Robotics and Automation (ICRA)*, 8762–8768. IEEE.

Mousavian, A., Eppner, C., and Fox, D. (2019). 6-dof graspnet: Variational grasp generation for object manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2901–2910.

Pinto, L. and Gupta, A. (2016). Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *2016 IEEE international conference on robotics and automation (ICRA)*, 3406–3413. IEEE.

Qi, C.R., Yi, L., Su, H., and Guibas, L.J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A.Y. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, 5. Kobe, Japan.

Ribeiro, E.G. and Grassi, V. (2019). Fast convolutional neural network for real-time robotic grasp detection. In *International Conference on Advanced Robotics (ICAR)*.

Robots, U. (2019). Ur5 collaborative robot arm | flexible and lightweight robot arm. `https://www.universal-robots.com/products/ur5-robot/`. Accessed: 2019-10-15.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Vicent, J., Sabater, N., Tenjo, C., Acarreta, J.R., Manzano, M., Rivera, J.P., Jurado, P., Franco, R., Alonso, L., Verrelst, J., et al. (2016). Flex end-to-end mission performance simulator. *IEEE Transactions on Geoscience and Remote Sensing*, 54(7), 4215–4223.

Viereck, U., Pas, A.t., Saenko, K., and Platt, R. (2017). Learning a visuomotor controller for real world robotic grasping using simulated depth images. *arXiv preprint arXiv:1706.04652*.

Viturino, C.C.B., de Lima Santana Filho, K., de Oliveira, D.M., Lemos, C.B., and Conceição, A.G.S. (2020). Redes neurais convolucionais para identificação e preensão robótica de objetos. *Anais da Sociedade Brasileira de Automática*, 2(1). doi:https://doi.org/10.48011/asba.v2i1.1163.