Grasp Generation with Depth Estimation from Color Images

Van-Thiep Nguyen
FPT University
Hanoi, Vietnam
thiepnvhe173027@fpt.edu.vn

Van-Duc Vu
FPT University
Hanoi, Vietnam
ducvvhe176438@fpt.edu.vn

Ngoc-Anh Hoang
FPT University
Hanoi, Vietnam
anhhnhe186401@fpt.edu.vn

Thu-Uyen Nguyen
FPT University
Hanoi, Vietnam
uyennthe176614@fpt.edu.vn

Duy-Quang Vu
FPT University
Hanoi, Vietnam
quangvdhe163133@fpt.edu.vn

Duc-Thanh Tran
FPT University
Hanoi, Vietnam
thanhtdhe176812@fpt.edu.vn

Khanh-Toan Phan
FPT University
Hanoi, Vietnam
toanpkhe170983@fpt.edu.vn

Anh-Truong Mai
FPT University
Hanoi, Vietnam
TruongMAHE182474@fpt.edu.vn

Van-Hiep Duong
FPT University
Hanoi, Vietnam
hiepdvhe181185@fpt.edu.vn

Cong-Trinh Tran
FPT University
Hanoi, Vietnam
trinhtche160916@fpt.edu.vn

Ngoc-Trung Ho
FPT University
Hanoi, Vietnam
trunghnhe172033@fpt.edu.vn

Quang-Tri Duong
FPT University
Hanoi, Vietnam
TriDQGCH210221@fpt.edu.vn

Phuc-Quan Ngo
FPT University
Hanoi, Vietnam
QuanNPGCH211110@fpt.edu.vn

Dinh-Cuong Hoang
FPT University
Hanoi, Vietnam
cuonghd12@fe.edu.vn

Abstract

Grasp generation plays a fundamental role in robot manipulation, often relying on three-dimensional (3D) point cloud data acquired through specialized depth cameras. However, the limited availability of such sensors in practical scenarios emphasizes the necessity for alternative approaches. This paper introduces an innovative method for grasp generation directly from color (RGB) images, negating the reliance on dedicated depth sensors. The proposed method employs tailored deep learning techniques for depth estimation from color images. Instead of traditional depth sensors, our approach computes predicted point clouds from estimated depth images directly generated from RGB inputs. A significant contribution lies in the design of a fusion module adept at seamlessly integrating features extracted from RGB images with those inferred from the predicted point clouds. This fusion process significantly strengthens the grasp generation pipeline by strengthening the advantages of both modalities, yielding notably improved grasp configurations. Experimental evaluations on standard datasets validate the efficacy of our approach, demonstrating its superior performance in generating grasp configurations compared to existing methods.

CCS Concepts: • **Computing methodologies → Computer vision**.

Keywords: Pose estimation, robot vision systems , intelligent systems, deep learning, supervised learning, machine vision.

1 Introduction

Grasp configuration generation stands as a critical element in robotic manipulation, and vision-based methodologies have played a pivotal role in addressing this challenge [9, 15, 16]. While model-based grasp generation has been prevalent, its limitations become apparent, particularly when confronting unknown objects [2, 6, 12–14, 17, 18, 29, 39]. An alternative avenue involves generating grasp configurations directly from sensor data without presuming knowledge of the object's 3D model or pre-computed grasps, referred to as grasp generation or grasp detection [9, 16]. Current methods fall into two categories: planar grasping and six Degrees of Freedom (6-DoF) grasping. Planar grasping utilizes a simple yet effective representation defining grasps as oriented bounding boxes. While this low degree of freedom (DoF) representation simplifies the task to a detection problem, it restricts performance in 3D manipulation tasks. On the other hand, 6-DoF grasping offers greater dexterity, suitable for complex scenarios. However, accurate generation of 6-DoF grasps necessitates geometric information, leading many existing methods to rely on 3D point cloud data. Despite significant

progress achieved by grasp generation methods using point clouds, challenges persist due to measurement noise, occlusions, and environmental interference, making generating feasible and reliable grasps in cluttered scenes difficult. Additionally, many methods require time-consuming multi-stage processing for sampling grasp candidates and evaluating grasp quality, while the unavailability of 3D point cloud data in numerous applications exacerbates this issue [6, 24, 37]. In contrast to 3D point clouds, acquiring color (RGB) images is more cost-effective and straightforward. With the advancements in deep learning, tasks such as object detection or 6D object pose estimation from RGB images have exhibited remarkable performance. However, the domain of grasp detection from RGB images remains largely unexplored.

In this work, we present a deep learning approach for grasp generation, focusing exclusively on leveraging RGB images to achieve accurate grasp estimation, building upon our previous research efforts [15, 16]. To obtain crucial geometric information for prediction, we employ recent advancements in monocular depth estimation to extract 3D points. Using an RGB image and the predicted 3D point cloud, we introduce an adaptive fusion module to extract discriminative features. These features are then input into a deep Hough voting module, inspired by our prior works [15, 16], which has demonstrated the effectiveness of voting mechanisms in addressing occlusions and ensuring collision-free grasps. Following the voting module, the collected votes undergo clustering and regression processes to precisely determine the essential grasp parameters. We evaluate the proposed method on a standard dataset and in a real-world robot grasping application. The results demonstrate promising outcomes, indicating that even with the utilization of only RGB images and estimated depth maps, we achieve noteworthy results.

2 Literature Review

2.1 Learning-based Grasp Generation

The grasp pose detection problem involves predicting multiple poses within a scene, enabling robots to manipulate objects effectively. Earlier approaches [1, 26] assumed complete 2D or 3D object knowledge or simplified objects as primitive shapes, facing limitations in obtaining accurate 3D models. Learning-based methods emerged, utilizing large-scale data and automated feature extraction. Some focused on 4-DoF grasp poses on the camera plane, known as "top-down grasping," restricting degrees of freedom and potentially missing crucial grasp poses, like those along object edges. In contrast, 6-DoF grasp poses offer increased flexibility and complexity, allowing grasping from various directions, necessitating six parameters to define location and rotation, with potential inclusion of additional degrees of freedom, like gripper width or height. Learning-based grasp generation can be categorized into two primary algorithmic methodologies for grasp synthesis: grasp pose sampling and and regressing grasp pose directly. Sampling-based approaches, like GPD [37] and PointNetGPD [24], evaluate individual grasp samples. Despite dense sampling, they struggle in regions like the rims of objects where surface normals estimation is unreliable. Some methods, such as Lou et al. [25], sample wrist angles independently, while others, like Kokic et al. [21], sample grasp, roll angles, and offset distances. However, these approaches often trade computation time for generated grasp poses, resulting in limited poses per scene and a focus on local object features. Direct regression methods, exemplified by Schmidt et al. [35] and Yang et al. [41], predict grasp poses or transformation matrices directly from visual data, processing information holistically. Yet, approaches like GraspNet [9] and PointNet++ [28], utilizing entire scene point clouds, lack consideration for inter-object relationships, limiting performance in cluttered scenes and under occlusion. To overcome these limitations, our previous work [15, 16] leverage a voting mechanism and contextual information to directly generate grasp configurations from 3D point clouds, addressing challenges in occlusion common in manipulation- The proposed method aligns closely with our prior studies [15, 16]. However, rather than relying on 3D data from depth sensors, we explore the utilization of depth images estimated via a monocular depth estimation framework.

2.2 Monocular Depth Estimation

The inception of monocular depth estimation was pioneered by [33, 34], employing hand-engineered features and Markov Random Fields (MRF). Subsequently, the advent of deep learning, spearheaded by Eigen et al. [8], revolutionized depth estimation. However, learned depth regression encounters challenges in the decoder phase due to the loss of fine details from successive convolution layers in neural networks. Numerous approaches have addressed this issue diversely. [7] introduced multi-scale networks to predict depth at multiple resolutions. Laina et al. [22] enhanced a ResNet architecture with improved up-sampling blocks to mitigate information loss. Xu et al. [40] combined deep learning with conditional random fields (CRF) for feature fusion at different scales. Another line of research pursued multitask learning, simultaneously predicting semantic labels [19], depth edges, and normals [23, 31, 44] to refine depth predictions. Kendall et al. [20] explored uncertainty estimation's impact on scene understanding, while Yin et al. [43] used surface geometry to estimate 3D point clouds from predicted depth maps. Recent works by Bhat et al. propose a classification-based formulation for distance prediction [29]. Tian et al. [3] integrated attention blocks into the decoder, while Transformer-based architectures gained traction [32, 42].

3 Materials and Methods

An overview of the proposed method is presented in Fig. 1. Our approach consists of several key components, including

depth estimation, adaptive fusion and voting-based grasp generation. These components work synergistically to enhance the dis- criminability and robustness of features, ultimately leading to more accurate and efficient grasp pose generation. We provide a detailed explanation of each of these components and their role in our systemâĂŹs success.

3.1 Depth Estimation

Given an RGB image $\mathbf{I_v}$, we employ an off-the-shelf monocular depth estimation method to generate a depth map $\mathbf{I}_d$. Specifically, we choose DPT [32], a dense prediction architecture based on an encoder-decoder design that utilizes a transformer as the fundamental computational building block of the encoder. In particular, DPT adopts the vision transformer (ViT) [5] as its backbone architecture. DPT reconstructs the bag-of-words representation provided by ViT into image-like feature representations at various resolutions. It progressively combines these feature representations into the final dense prediction using a convolutional decoder. Unlike fully-convolutional networks, the vision transformer backbone avoids explicit downsampling operations after computing an initial image embedding and maintains a representation with constant dimensionality throughout all processing stages. Additionally, it ensures a global receptive field at every stage. We demonstrate that these characteristics are particularly advantageous for dense prediction tasks, naturally leading to fine-grained and globally coherent predictions.

3.2 Feature Extraction and Fusion

Given a input RGB image $\mathbf{I_v}$ and an predicted depth map $\mathbf{I_d}$, our initial step involves elevating the depth image $\mathbf{I_d}$ to a point cloud $\mathbf{P}$ using the camera intrinsic matrix. Subsequently, we employ ResNet34 [11] and PointNet++ [30] to extract visual features $\mathcal{F}_{vis}$ from RGB image and geometric feature $\mathcal{F}_{geo}$ from the point cloud $\mathbf{P}$ respectively.

To achieve the fusion of RGB features $\mathcal{F}_{vis}$ with geometric features $\mathcal{F}_{geo}$, we employ an innovative adaptive fusion module. Departing from the conventional approach of globally compressing the RGB feature map, which may result in the loss of intricate details, we capitalize on the information provided by the aligned RGBD image. The depth information associated with each pixel is instrumental in establishing an XYZ map that is precisely aligned with the RGB map, forming a coherent representation of the scene in three-dimensional space. For each geometric feature, paired with its corresponding 3D point coordinate derived from the aligned RGBD image, we employ a novel strategy to extract pertinent visual information. This involves projecting a defined neighborhood around each 3D point onto the RGB image, utilizing a specified radius r. By doing so, we selectively capture the visual context surrounding each point, taking into account its relationship with the RGB features. Subsequently, a crucial step in this fusion process is the extraction of visual features from $\mathcal{F}_{vis}$ for each geometric feature. This is achieved by sampling the k nearest neighbor pixels within the projected neighborhood. The visual features associated with these sampled pixels are then collected and integrated using max pooling. In situations where fewer than k pixels exist in the specified region, null features are padded to maintain the consistency of the sampling process. The collected visual features are then processed through Multi-Layer Perceptrons (MLPs) to ensure their channel size matches that of the original point cloud feature. This modification results in a set of adapted visual features denoted as $\mathcal{F}^{'}_{vis}$, reflecting a refined representation of the RGB information in the context of the geometric features.

Following this adaptation, the integrated visual features $\mathcal{F}^{'}_{vis}$ are concatenated with the original geometric features $\mathcal{F}_{geo}$. This concatenated feature set undergoes further refinement through the application of a shared Multi-Layer Perceptron (MLP), culminating in the generation of the fused geometric feature $\mathcal{F}_{fus}$. This fused representation embodies a synergistic combination of RGB and geometric information, capturing the nuanced interplay between color and spatial features.

Consequently, the network enriches a set of N 3D points with high-dimensional features, denoted as $\mathcal{P} = \{p_i\}_{i=1}^{N}$ and $\mathcal{F}_{fus} = \{f_i\} i = 1^N$, where each p_i is a concatenation of the 3D point's location $x_i \in \mathbb{R}^3$ and its associated feature vector f_i. The enriched points $\{p_i\}_{i=1}^{N}$, now imbued with the fused features, serve as input to our self-attention module, producing enhanced features denoted as $\mathcal{F}$. This self-attention mechanism, inspired by [38], dynamically weighs the importance of different enriched features, facilitating the extraction of meaningful relationships and patterns within the spatial context of the 3D points. The resulting enhanced features $\mathcal{F}$ capture the fused information from both RGB and geometric domains, contributing to improved performance in downstream tasks. The self-attention module, as defined in [38], is expressed as:

$$y_i = \sum_{p_j \in \mathcal{P}(i)} (\alpha(\gamma(p_i, p_j) + \delta) \odot \beta(p_j)) \tag{1}$$

Here, $\mathcal{P}(i) \subseteq \mathcal{P}$ refers to a set of points in the local neighborhood of p_i. The terms α, γ, δ, and β represent a mapping function, a relation function, a position encoding function, and pointwise feature transformation, respectively. The relation function γ utilizes subtraction to output a vector representing the features of p_i and p_j:

$$\gamma(p_i, p_j) = \varphi(p_i) - \psi(p_j) \tag{2}$$

Here, φ and ψ are trainable transformations implemented using multilayer perceptrons (MLPs). The mapping function α is an MLP with two linear layers and one ReLU nonlinearity, enabling the module to compute attention weights spatially and across channels while maintaining computational efficiency. To adapt to local data structures, we introduce spatial

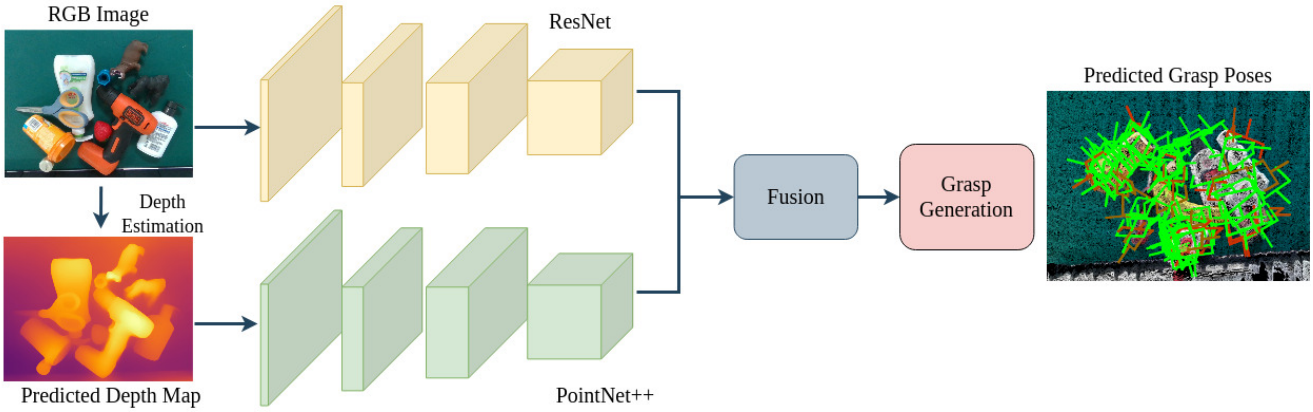


Figure 1. Overview of our network architecture.

context using a trainable and parameterized position encoding function δ:

$$\delta = \phi(x_i - x_j) \tag{3}$$

Here, x_i and x_j denote the 3D point coordinates for points i and j, respectively. The encoding function ϕ is an MLP with two linear layers and one ReLU nonlinearity. This comprehensive self-attention mechanism allows the network to capture intricate dependencies among enriched features, enhancing the model's capacity for contextualized information processing.

3.3 Grasp Generation

Given the extracted dense fused feature $\mathcal{F} = \{f_i\}$, we predict grasp poses using the voting-based grasp generation module in our previous work [15].

Loss Function: The learning of modules is supervised jointly using a multi-task loss:

$$L_{votegrasp} = \lambda_1 L_{vote} + \lambda_2 L_{grasp} \tag{4}$$

The voting loss L_{vote} is a regression loss formulated as:

$$L_{vote} = \frac{1}{M_s} \sum_i \|y_i - c_i^g\|_H \cdot \mathbb{1}(x_i) \tag{5}$$

Here, M_s represents the total number of seed points on the object surface, c_i^g is the closest ground truth grasp center, $\|\cdot\|_H$ denotes the Huber norm, and $\mathbb{1}(\cdot)$ is a binary function determining whether a seed point s_i belongs to an object.

The grasp loss function L_{grasp} is defined as:

$$L_{grasp} = L_{center} + \alpha L_{rot} + \beta L_{width} + \gamma L_{score} \tag{6}$$

The L_{grasp} comprises losses for grasp center regression (L_{center}), rotation (L_{rot}), gripper width regression (L_{width}), and grasp confidence score regression (L_{score}). The grasp center loss includes viewpoint classification loss ($L_{viewpoint}$) and in-plane rotation loss ($L_{in-plane}$), which consists of classification ($L_{angle-cls}$) and regression ($L_{angle-reg}$) losses. Regression losses employ $L1$-smooth loss, while classification losses use standard cross-entropy loss. More details can be found in [15].

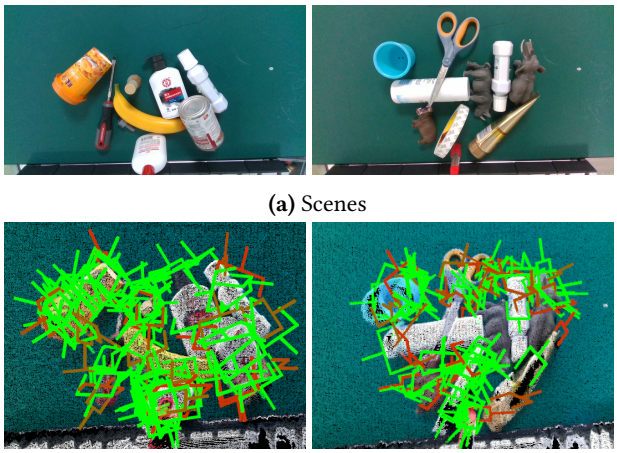
3.4 Dataset

We conduct evaluations and comparisons on the publicly available GraspNet-1Billion dataset [9]. This dataset comprises 97,280 RGB-D images from 190 cluttered scenes, providing over one billion grasp poses for 88 distinct objects within these scenes. These objects exhibit diversity in shape, texture, size, material, and occlusion conditions, making it an ideal benchmark for assessing our model's generalization capacity and robustness to occlusions. Each object in the dataset is associated with an accurate 3D mesh model, along with camera poses, 6D object poses, object masks, and bounding boxes for all frames. This extensive annotation facilitates straightforward generation of ground truth votes and grasp configurations. Following the methodology of [9], we partitioned the dataset into training and testing sets. Specifically, 100 scenes were allocated for training purposes, while 90 scenes were reserved for testing. To evaluate the model's generalizability, the test dataset is further divided into subsets: scenes with novel objects, scenes featuring unseen yet similar objects, and scenes containing previously encountered objects. This deliberate partitioning allows for a comprehensive assessment of our model's performance across diverse scenarios.

4 Result and Discussion

4.1 Implementation Details

In our implementation, we employ a pre-trained ResNet34 model trained on the ImageNet dataset as the encoder for RGB images. The output appearance feature from this encoder-decoder architecture comprises 256 channels. For point cloud feature extraction, we randomly sample 12,288 points from



(a) Scenes

(b) Predicted Grasps

Figure 2. Examples of input scenes and predicted grasps from the proposed method. The different intensity of grasp color denotes the confidence score of grasps. Green refers to the highest quality grasps and red refers to the lowest ones.

depth images and utilize a PointNet++ [30]-based feature learning network, which also yields a 256-channel output. In the voting and context learning modules, we form $K = 128$ clusters and produce a new feature map $\mathcal{F}_{context} \in 128 \times 512$. Subsequently, 128 grasps are generated from this new feature map. The prediction layer comprises $5+V+2A$ channels, with $V = 120$ and $A = 6$. We set $\lambda_1 = \lambda_2 = 1.0$ and α=β=γ=1.0. Our network is trained entirely using a batch size of 8 and optimized with Adam, employing a learning rate of 0.001 for 200 epochs. Training on a single Nvidia GeForce RTX 2080 Ti 11GB GPU takes approximately 20 hours. Regarding inference, our method requires 90ms for a single scene during the forward pass.

4.2 Evaluation on GraspNet-1Billion

We follow previous research [9] and evaluate our results on the dataset using *Precision@k*. This metric quantifies the precision of the top-k ranked grasps. To identify a predicted grasp (G_p) as a true positive, it must satisfy three conditions: (i) containing an object inside the gripper; (ii) being collision-free; (iii) exhibiting an antipodal grasp under a given friction coefficient μ. The third condition is calculated based on prior works [9, 37]. We denote AP_μ as the average *Precision@k* for k values ranging from 1 to 50, given a friction coefficient μ. Additionally, we present the average of AP_μ across $\mu = \{0.2, 0.4, 0.6, 0.8, 1.0\}$, denoted as AP.

Table 1 and Fig. 2 demonstrate the performance comparison between our approach and state-of-the-art methods. The evaluation utilized the evaluation metric adopted in [9], enabling a direct comparison with related works reported in [9, 10]. The table showcases the evaluation outcomes categorized into "Seen," "Unseen (but similar)," and "Novel" objects, aiding in assessing the model's generalization capability. The results indicate superior performance on scenes featuring seen objects across all methods, while notably, our proposed approach consistently outperforms others, even in the challenging "Novel" category, underscoring its robust generalization capabilities.

Computational Time. Our experiments were conducted on an Intel Xeon E-2716G CPU clocked at 3.7 GHz, paired with an Nvidia GeForce RTX 2080 Ti GPU featuring 11GB of memory. Our approach achieves a runtime of 100 ms per RGBD image. This fine balance between accuracy and speed empowers our method to proficiently generate grasp configurations in cluttered scenes, rendering it well-suited for diverse real-world scenarios.

4.3 Robotic Grasping Experiment

The experiments were conducted with a Franka Emika Panda robot arm with 7-DOF, equipped with a parallel-jaw gripper. To capture RGBD data, we used either ASUS Xtion PRO LIVE sensor or Microsoft Kinect sensor v2. The whole system is implemented using the ROS and MoveIt! frameworks.

We conducted a real-world evaluation of state-of-the-art grasp detection methods, each trained on the GraspNet-1Billion dataset for a fair comparison. We selected novel objects tailored to fit the gripper's shapes and sizes. In each scenario, a random subset of 10-15 objects was arranged in a haphazard manner on a table, mirroring the unpredictability of real-world environments. Each method underwent 300 grasp attempts, with the robot randomly selecting objects. A grasp was deemed successful if the robot could grasp and lift the object within a single attempt. The results in Table 2 demonstrate our method's superiority, achieving an 84% success rate outperforming all other methods. This highlights the proposed framework's efficacy in real-world grasping scenarios, attributing the increased success rate to the integration of estimated depth data, underscoring the significance of richer input data for precise and effective.

5 Conclusions

In this study, we addressed the fundamental challenge of grasp generation in robotic manipulation by introducing an innovative approach that bypasses the need for specialized depth sensors. Our method revolutionizes grasp generation by leveraging tailored deep learning techniques to estimate depth from color (RGB) images directly. This paradigm shift allows the computation of predicted point clouds solely from RGB inputs, eliminating the dependency on traditional depth sensors. A pivotal contribution lies in the development of a fusion module adept at seamlessly integrating features derived from RGB images with those inferred from predicted point clouds. This fusion process harnesses the strengths

Table 1. The table shows the results on GraspNet-1Billion test set captured by RealSense/Kinect sensors respectively.

	Seen			Unseen (but similar)			Novel		
	AP	$AP_{0.8}$	$AP_{0.4}$	*AP*	$AP_{0.8}$	$AP_{0.4}$	*AP*	$AP_{0.8}$	$AP_{0.4}$
GG-CNN [27]	15.5/16.9	21.8/22.5	10.3/11.2	13.3/15.1	18.4/19.8	4.6/6.2	5.5/7.4	5.9/8.8	1.9/1.3
Chu et al. [4]	16.0/17.6	23.7/24.7	10.8/12.7	15.4/17.4	20.2/21.6	7.1/8.9	7.6/8.0	8.7/9.3	2.5/1.8
GPD [37]	22.9/24.4	28.5/30.2	12.8/13.5	21.3/23.2	27.8/28.6	9.6/11.3	8.2/9.6	8.9/10.1	2.7/3.2
PN-GDP [24]	26.0/27.6	33.0/34.2	15.4/17.8	22.7/24.4	29.2/30.8	10.8/12.8	9.2/10.7	9.9/11.2	2.7/3.2
Fang et al. [9]	27.6/29.9	33.4/36.2	17.0/19.3	26.1/27.8	34.2/33.2	14.2/16.6	10.6/11.5	11.3/12.9	4.0/3.6
Gou et al. [10]	28.0/32.1	33.5/39.5	17.8/20.9	27.2/30.4	36.3/37.9	15.6/18.7	12.3/13.1	12.5/13.8	5.6/6.0
Contact [36]	29.9/31.4	35.2/39.0	19.5/21.6	28.2/29.0	37.0/35.2	16.3/18.9	13.2/13.9	13.5/14.7	6.8/7.7
VoteGrasp [16]	34.1/37.5	38.9/45.6	24.0/27.7	33.0/35.9	40.8/43.3	20.5/24.7	16.9/18.5	17.0/18.5	10.0/10.6
Ours	**34.9/37.9**	**39.5/46.1**	**24.3/28.8**	**33.2/36.0**	**41.1/44.2**	**26.1/25.1**	**17.5/25.2**	**17.5/18.9**	**10.6/11.2**

Table 2. Results of real robot experiments. The networks were trained on the GraspNet-1Billion dataset. The table shows the number of attempts, the number of successful attempts, and the grasp success rate.

Method	Attempt	Success	Success Rate
GPD [37]	300	195	65%
PointNetGPD [24]	300	201	67%
Fang et al. [9]	300	214	71%
Gou et al. [10]	300	218	73%
Contact-GraspNet [36]	300	222	74%
VoteGrasp [16]	300	234	78%
Ours	300	**240**	**80**%

of both modalities, significantly enhancing grasp configurations. Experimental evaluations unequivocally validate the effectiveness of our approach, demonstrating its superiority in generating grasp configurations compared to existing methods. Future endeavors in these outlined directions hold the promise of further enhancing the versatility, adaptability, and real-world applicability of grasp generation in robotics.

References

[1] Antonio Bicchi and Vijay Kumar. 2000. Robotic grasping and contact: A review. In *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065)*, Vol. 1. IEEE, 348–353.

[2] Liang-Chia Chen, Dinh-Cuong Hoang, Hsien-I Lin, and Thanh-Hung Nguyen. 2016. Innovative methodology for multi-view point cloud registration in robotic 3D object scanning and reconstruction. *Applied Sciences* 6, 5 (2016), 132.

[3] Tian Chen, Shijie An, Yuan Zhang, Chongyang Ma, Huayan Wang, Xiaoyan Guo, and Wen Zheng. 2020. Improving monocular depth estimation by leveraging structural awareness and complementary datasets. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer, 90–108.

[4] Fu-Jen Chu, Ruinian Xu, and Patricio A Vela. 2018. Real-world multiobject, multigrasp detection. *IEEE Robotics and Automation Letters* 3, 4 (2018), 3355–3362.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[6] Guoguang Du, Kai Wang, Shiguo Lian, and Kaiyong Zhao. 2021. Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. *Artificial Intelligence Review* 54, 3 (2021), 1677–1734.

[7] David Eigen and Rob Fergus. 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*. 2650–2658.

[8] David Eigen, Christian Puhrsch, and Rob Fergus. 2014. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems* 27 (2014).

[9] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. 2020. GraspNet-1Billion: A Large-Scale Benchmark for General Object Grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11444–11453.

[10] Minghao Gou, Hao-Shu Fang, Zhanda Zhu, Sheng Xu, Chenxi Wang, and Cewu Lu. 2021. Rgb matters: Learning 7-dof grasp poses on monocular rgbd images. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 13459–13466.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[12] Dinh-Cuong Hoang, Liang-Chia Chen, and Thanh-Hung Nguyen. 2016. Sub-OBB based object recognition and localization algorithm using range images. *Measurement Science and Technology* 28, 2 (2016), 025401.

[13] Dinh-Cuong Hoang, Achim J Lilienthal, and Todor Stoyanov. 2020. Object-RPE: Dense 3D reconstruction and pose estimation with convolutional neural networks. *Robotics and Autonomous Systems* 133 (2020), 103632.

[14] Dinh-Cuong Hoang, Achim J Lilienthal, and Todor Stoyanov. 2020. Panoptic 3D mapping and object pose estimation using adaptively weighted semantic information. *IEEE Robotics and Automation Letters* 5, 2 (2020), 1962–1969.

[15] Dinh-Cuong Hoang, Anh-Nhat Nguyen, Van-Duc Vu, Duy-Quang Vu, Van-Thiep Nguyen, Thu-Uyen Nguyen, Cong-Trinh Tran, Khanh-Toan Phan, and Ngoc-Trung Ho. 2023. Grasp Configuration Synthesis from 3D Point Clouds with Attention Mechanism. *Journal of Intelligent and Robotic Systems* 109, 3 (2023), 71.

[16] Dinh-Cuong Hoang, Johannes A Stork, and Todor Stoyanov. 2022. Context-aware grasp generation in cluttered scenes. In *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 1492–1498.

[17] Dinh-Cuong Hoang, Johannes A Stork, and Todor Stoyanov. 2022. Voting and attention-based pose relation learning for object pose estimation from 3d point clouds. *IEEE Robotics and Automation Letters* 7, 4 (2022), 8980–8987.

[18] Dinh-Cuong Hoang, Todor Stoyanov, and Achim J Lilienthal. 2019. Object-rpe: Dense 3d reconstruction and pose estimation with convolutional neural networks for warehouse robots. In *2019 European Conference on Mobile Robots (ECMR)*. IEEE, 1–6.

[19] Jianbo Jiao, Ying Cao, Yibing Song, and Rynson Lau. 2018. Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In *Proceedings of the European conference on computer vision (ECCV)*. 53–69.

[20] Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7482–7491.

[21] Mia Kokic, Danica Kragic, and Jeannette Bohg. 2020. Learning task-oriented grasping from human activity datasets. *IEEE Robotics and Automation Letters* 5, 2 (2020), 3352–3359.

[22] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. 2016. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*. IEEE, 239–248.

[23] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. 2019. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326* (2019).

[24] Hongzhuo Liang, Xiaojian Ma, Shuang Li, Michael Gorner, Song Tang, Bin Fang, Fuchun Sun, and Jianwei Zhang. 2019. Pointnetgpd: Detecting grasp configurations from point sets. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 3629–3635.

[25] Xibai Lou, Yang Yang, and Changhyun Choi. 2020. Learning to generate 6-dof grasp poses with reachability awareness. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 1532–1538.

[26] Andrew T Miller, Steffen Knoop, Henrik I Christensen, and Peter K Allen. 2003. Automatic grasp planning using shape primitives. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, Vol. 2. IEEE, 1824–1829.

[27] Douglas Morrison, Peter Corke, and Jurgen Leitner. 2018. Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach. In *Robotics: Science and Systems (RSS)*.

[28] P. Ni, W. Zhang, X. Zhu, and Q. Cao. 2020. PointNet++ Grasping: Learning An End-to-end Spatial Grasp Generation Algorithm from Sparse Point Clouds. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 3619–3625. https://doi.org/10.1109/ICRA40945.2020.9196740

[29] Alessandro Palleschi, Marco Gugliotta, Chiara Gabellieri, Dinh-Cuong Hoang, Todor Stoyanov, Manolo Garabini, and Lucia Pallottino. 2020. Fully autonomous picking with a dual-arm platform for intralogistics. In *Proc. I-RIM Conf. I-RIM*. 109–111.

[30] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems* 30 (2017).

[31] Michael Ramamonjisoa and Vincent Lepetit. 2019. Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 0–0.

[32] Rene Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. 2021. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*. 12179–12188.

[33] Ashutosh Saxena, Sung Chung, and Andrew Ng. 2005. Learning depth from single monocular images. *Advances in neural information processing systems* 18 (2005).

[34] Ashutosh Saxena, Min Sun, and Andrew Y Ng. 2008. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence* 31, 5 (2008), 824–840.

[35] Philipp Schmidt, Nikolaus Vahrenkamp, Mirko Wachter, and Tamim Asfour. 2018. Grasping of unknown objects using deep convolutional neural networks based on depth images. In *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 6831–6838.

[36] Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. 2021. Contact-GraspNet: Efficient 6-DoF Grasp Generation in Cluttered Scenes. (2021).

[37] Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. 2017. Grasp pose detection in point clouds. *The International Journal of Robotics Research* 36, 13-14 (2017), 1455–1473.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[39] Van-Duc Vu, Dinh-Dai Hoang, Phan Xuan Tan, Van-Thiep Nguyen, Thu-Uyen Nguyen, Ngoc-Anh Hoang, Khanh-Toan Phan, Duc-Thanh Tran, Duy-Quang Vu, Phuc-Quan Ngo, et al. 2024. Occlusion-Robust Pallet Pose Estimation for Warehouse Automation. *IEEE Access* (2024).

[40] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. 2017. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5354–5362.

[41] Daniel Yang, Tarik Tosun, Benjamin Eisner, Volkan Isler, and Daniel Lee. 2021. Robotic grasping through combined image-based grasp proposal and 3d reconstruction. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 6350–6356.

[42] Guanglei Yang, Hao Tang, Mingli Ding, Nicu Sebe, and Elisa Ricci. 2021. Transformer-based attention networks for continuous pixel-wise prediction. In *Proceedings of the IEEE/CVF International Conference on Computer vision*. 16269–16279.

[43] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. 2019. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5684–5693.

[44] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. 2019. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4106–4115.