Grasp Generation with Depth Estimation from Color Images

Van-Thiep Nguyen[1], Van-Duc Vu[1], Duy-Quang Vu [1],
Phuc-Quan Ngo[1], Ngoc-Anh Hoang[1], Khanh-Toan Phan[1],
Duc-Thanh Tran[1], Thu-Uyen Nguyen[1], Cong-Trinh Tran[1], Ngoc-Trung Ho[1], Dinh-Cuong Hoang[1],
[1]FPT University, Hanoi, 10000, Vietnam,
Corresponding author: Dinh-Cuong Hoang (e-mail: cuonghd7@fpt.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. Furthermore, we achieved a promising 84% success rate in real robot grasping experiments, underscoring the practical viability of our method. Our code and other materials are available at `https://github.com/hoangcuongbk80/DepthEstGrasp`.**

I. 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 [1]–[3]. While model-based grasp generation has been prevalent, its limitations become apparent, particularly when confronting unknown objects [4]–[7]. The reliance on predefined 3D models and grasp databases presents significant constraints in real-world scenarios where robots encounter diverse, unmodeled objects. Conventionally, model-based grasp generation methods rely on a 6D object pose estimation algorithm to align a Computer-Aided Design (CAD) model with measured data, followed by selecting grasps from a pre-computed database. However, synthesizing grasps for unknown objects becomes implausible, as these approaches presume the availability of a 3D model and a pre-defined grasp database [4].

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 [2], [3], [8]. 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 [4], [8], [9]. 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 [10] or 6D object pose estimation [11] from RGB images have exhibited remarkable performance. However, the domain of grasp detection from RGB images remains largely unexplored.

This study introduces a novel deep learning network for model-free 6-DoF grasping, exclusively leveraging an RGB image for accurate grasp estimation, building upon our prior work [1]. To derive geometric information crucial for prediction, we harness recent advancements in monocular depth estimation to extract 3D points. To the best of our knowledge, this is the first deep learning pipeline using 3D point clouds from estimated depth maps for grasp generation. Given an RGB image and a predicted 3D point cloud, we propose an attention-based adaptive fusion module to extract discriminative features. These features are subsequently fed into a deep Hough voting module, inspired by our prior work demonstrating the efficacy of voting mechanisms in addressing occlusions and ensuring collision-free grasps. The voting module, built upon our prior research, enables the identification of optimal grasp centers. Subsequently, 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 on real robot grasping application. The results show that

[1]FPT University, Hanoi, Vietnam.

even using only RGB images with estimated depth maps we still outperform state-of-the-art methods using depth images from sensors. Our method's evaluation on a standard dataset and real robot grasping applications underscores its superiority, showcasing that even with solely RGB images and estimated depth maps, our approach outperforms state-of-the-art methods utilizing depth images from sensors.

The remainder of this article is organized as follows: Section II, Literature Review, provides an overview of Learning-based Grasp Generation (II.A) and Monocular Depth Estimation (II.B). In Section III, Materials and Methods, we introduce our methodology. This includes Depth Estimation (III.A) and details our innovations: the Attention-based Adaptive Fusion Network (III.B) incorporating Visual-Guided 3D Geometric Feature Learning and Geometric-Guided Visual Feature Learning, as well as the Voting-based Grasp Generation (III.C) approach. Additionally, this section covers information about the Dataset (III.D). Section IV, Evaluation, includes the Implementation Details subsection (IV.A), discussing the technical specifics of our methods. The Evaluation on GraspNet-1Billion (IV.B) subsection presents the results and analysis based on the GraspNet-1Billion dataset. Finally, the Robotic Grasping Experiment (IV.C) subsection provides insights derived from real-world experiments in robotic grasping. Lastly, Section V, Conclusions, offers a summary of the key contributions and findings presented in this article. It also outlines potential avenues for future research, paving the way for further advancements in this domain.

II. Literature Review

In this section, we review relevant works, specifically focus- ing on existing learning-based grasp generation methods and datasets for monocular depth estimation.

A. Learning-based Grasp Generation

The grasp pose detection problem involves predicting multiple poses within a scene, enabling robots to manipulate objects effectively. Earlier approaches [12], [13] 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 [8] and PointNetGPD [9], 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. [14], sample wrist angles independently, while others, like Kokic et al. [15], 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. [16] and Yang et al. [17], predict grasp poses or transformation matrices directly from visual data, processing information holistically. Yet, approaches like GraspNet [3] and PointNet++ [18], 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 [1], [2] leverage a voting mechanism and contextual information to directly generate grasp configurations from 3D point clouds, addressing challenges in occlusion common in manipulationThe proposed method aligns closely with our prior studies [1], [2]. However, rather than relying on 3D data from depth sensors, we explore the utilization of depth images estimated via a monocular depth estimation framework.

B. Monocular Depth Estimation

The inception of monocular depth estimation was pioneered by [19], [20], employing hand-engineered features and Markov Random Fields (MRF). Subsequently, the advent of deep learning, spearheaded by Eigen et al. [21], 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. [22] introduced multi-scale networks to predict depth at multiple resolutions. Laina et al. [23] enhanced a ResNet architecture with improved up-sampling blocks to mitigate information loss. Xu et al. [24] 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 [25], depth edges, and normals [26]–[28] to refine depth predictions. Kendall et al. [29] explored uncertainty estimation's impact on scene understanding, while Yin et al. [30] 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. [31] integrated attention blocks into the decoder, while Transformer-based architectures gained traction [32], [33].

III. Materials and Methods

An overview of the proposed method is presented in Fig. 1. Our approach consists of several key components, including depth estimation, attention-based adaptive fusion incorporating visual-guided 3D geometric feature and geometric-guided visual feature Learning, 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 systems success.

A. Depth Estimation

Existing monocular depth estimation methods are primarily tailored for large outdoor scenes, posing challenges when applied to relatively smaller objects intended for manipulation. To address this limitation, our focus is on enhancing the depth map quality specifically for such objects. We leverage two distinct depth estimation networks, DPT [32] and iDisc [34], to derive individual depth images denoted as $\mathbf{I}_{d1}$ and $\mathbf{I}_{d2}$, respectively. By computing the disparity between these images, regions with significant differences beyond a predefined threshold are identified as uncertain areas. Our approach involves excluding these uncertain regions from the depth images and replacing the depth values within other areas with their mean values. This process aims to refine depth information specifically for small object manipulation, culminating in an enhanced and more accurate depth image, denoted as $\mathbf{I}_d$.

B. Attention-based Adaptive Fusion Network

Given a RGB image $\mathbf{I_v}$ and an estimated 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 [35] and PointNet++ [36] to extract visual features $\mathcal{F}_{vis}$ from RGB image and geometric feature $\mathcal{F}_{geo}$ from the point cloud $\mathbf{P}$ respectively. These networks facilitate bidirectional information flow through Visual-Guided Geometric Feature Learning (VGG) and Geometric-Guided Visual Feature Learning (GGV) modules, enabling each branch to utilize mutual local and global information for enhanced representation learning.

Visual-Guided 3D Geometric Feature Learning

To integrate visual information from $\mathcal{F}^i_{vis}$ into geometric features $\mathcal{F}^i_{geo}$ in the i-th stage, we introduce a novel Visual-Guided Geometric Feature Learning (VGG) module. Rather than globally compressing the RGB feature map and potentially losing intricate details, we utilize the aligned RGBD image. Each pixel's depth contributes to deriving its corresponding 3D point, establishing an XYZ map aligned with the RGB map. For every geometric feature paired with its 3D point coordinate, we retrieve visual features from $\mathcal{F}_{vis}$ by projecting its neighborhood, with a radius r_1, onto the image. Subsequently, we sample the k_1 nearest neighbor pixels within this region, gathering their visual features. In cases where fewer than k_1 pixels exist in the corresponding region, null features are padded. These collected visual features are integrated using max pooling and processed through Multi-Layer Perceptrons (MLPs) to match their channel size with the point cloud feature. This stage produces modified visual features $\mathcal{F}^{'}_{vis}$. Subsequently, we concatenate the integrated visual features $\mathcal{F}^{'}_{vis}$ with the geometric features $\mathcal{F}^i_{geo}$ and apply a shared MLP to obtain the fused geometric feature $\mathcal{F}^{fus}_{geo}$. Consequently, the network enriches N 3D points with high-dimensional features, denoted as $\mathcal{P} = \{p_i\}^N_{i=1}$ and $\mathcal{F}^{'}_{geo} = \{f_i\}^N_{i=1}$, where $p_i = [x_i; f_i]$. Here, $x_i \in \mathbb{R}^3$ signifies the point's location in 3D space, and f_i represents the associated feature vector. The enriched points $\{p_i\}^N_{i=1}$, now imbued with the fused features, are then inputted into our self-attention module to enhance the features $\mathcal{F}^{ei}_{geo}$. In accordance with [37], [38], the self-attention module is defined as follows:

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

$\mathcal{P}(i) \subseteq \mathcal{P}$ refers to a set of points in the local neighborhood of p_i. α, γ, δ, and β signify a mapping function, a relation function, a position encoding function, and pointwise feature transformation, respectively. The relation function γ uses 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 represent trainable transformations using multilayer perceptrons (MLPs). The mapping function α is an MLP with two linear layers and one ReLU nonlinearity, allowing the module to compute attention weights spatially and across channels while maintaining computational efficiency. To adapt to local data structures, we introduce spatial context using a trainable and parameterized position encoding function δ:

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

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.

Geometric-Guided Visual Feature Learning. The Geometric-Guided Visual Feature Learning (GGV) module provides an alternative approach to integrating geometric information from $\mathcal{F}^i_{geo}$ into visual features $\mathcal{F}^i_{vis}$ during the i-th stage. Rather than naively concatenating global point features, this module densely fuses features by identifying k_2 nearest points for each pixel from the point cloud, collecting corresponding point features, and integrating them via max pooling to produce $\mathcal{F}^{'}_{geo}$. These features are then passed through a spatial attention block M_{sa1} [39]. This mechanism is designed to discern informative regions, eliminating redundant geometric-guided features that may arise from noise or irrelevant areas, thereby facilitating a more effective integration with the visual features $\mathcal{F}^i_{vis}$. The block utilizes average-pooling to highlight informative regions, resulting in $\mathcal{F}^{avg}_{geo} \in \mathbb{R}^{W \times H}$. Subsequently, F^{avg}_{vis} undergoes a 7×7 filter convolution and normalization via the sigmoid function. The output, denoted as $M_{sa1}(\mathcal{F}^i_{geo})$, is then element-wise multiplied with the original geometric features, $\mathcal{F}^i_{geo}$, to acquire the initial enhanced geometric-guided features, $\mathcal{F}^{sa}_{geo}$. The summarized attention process is illustrated as:

$$M_{sa1}(\mathcal{F}^i_{geo}) = \sigma(f^{7\times7}(AvgPool(\mathcal{F}^i_{geo})) \tag{4}$$

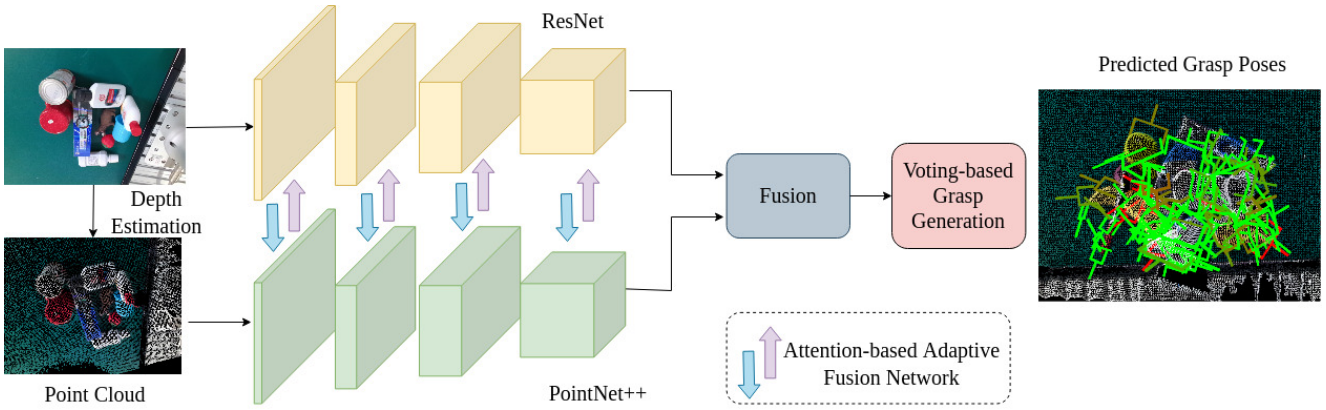


Fig. 1: Overview of our network architecture.

$$\mathcal{F}_{geo}^{sa} = M_{sa1}(\mathcal{F}_{geo}^{i}) \otimes \mathcal{F}_{geo}^{i} \tag{5}$$

Here, $\otimes$ denotes element-wise multiplication, σ represents the sigmoid function, and $f^{7\times7}$ denotes a convolution operation utilizing a 7×7 filter. Subsequently, $\mathcal{F}_{geo}^{sa}$ is integrated with the visual features $\mathcal{F}_{vis}^{i}$ through element-wise summation to produce the fused features $\mathcal{F}_{vis}^{fus}$:

$$\mathcal{F}_{vis}^{fus} = \mathcal{F}_{vis}^{i} \oplus \mathcal{F}_{geo}^{sa} \tag{6}$$

Where $\oplus$ signifies element-wise summation. To further refine the fused features $\mathcal{F}_{vis}^{fus}$, a channel attention block M_{ca} [40] is introduced. This block utilizes global average pooling to reduce each feature map within $\mathcal{F}_{vis}^{fus}$ to a single pixel, generating a 1D vector of length C. The vector undergoes an MLP network with a hidden layer and sigmoid activation, followed by element-wise multiplication with $\mathcal{F}_{vis}^{fus}$. This process recalibrates the feature responses, accentuating important channels while suppressing less relevant ones. The output of M_{ca}, denoted as F_{vis}^{c}, can be summarized as:

$$M_{ca}(\mathcal{F}_{vis}^{fus}) = \sigma(MLP(AvgPool(\mathcal{F}_{vis}^{fus})) \tag{7}$$

$$\mathcal{F}_{vis}^{c} = M_{ca}(\mathcal{F}_{vis}^{fus}) \otimes \mathcal{F}_{vis}^{fus} \tag{8}$$

Moreover, $\mathcal{F}_{vis}^{c}$ undergoes re-weighting by another spatial attention block, M_{sa2}, with components akin to M_{sa1}, producing $\mathcal{F}_{vis}^{cs}$. Finally, $\mathcal{F}_{vis}^{cs}$ is integrated with the visual features $\mathcal{F}_{vis}$ through element-wise summation, yielding the enhanced feature representation $\mathcal{F}_{vis}^{ei}$.

Fusion. Following bidirectional fusion in both VGG and GGV modules, distinct features are extracted by the visual and geometric branches. To generate reliable correspondences and obtain more distinctive features, a simple undirected fusion is performed in the final stage. By projecting each point to the image plane with the camera intrinsic matrix, correspondences between visual and geometry features are established. These pairs are concatenated to form the extracted dense fused feature $\mathcal{F}$, subsequently utilized in the voting-based grasp generation module in the subsequent step.

C. Voting-based 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 [1]. Each grasp comprises a center point $p \in \mathbb{R}^3$, a gripper orientation $R \in SO(3)$, a gripper width $w \in \mathbb{R}$, and a grasp score $q \in [0, 1]$. We generate M seeds $\{s_i\}_{i=1}^{M}$, where each seed $s_i = [x_i, f_i^s]$ holds the 3D spatial location $x_i \in \mathbb{R}^3$ and the corresponding feature vector $f_i^s \in \mathbb{R}^F$. Processing these seeds through an MLP computes J votes $\{\{v_{ij} = [y_{ij}; f_{ij}^v] \in \mathbb{R}^{3+F}\}_{i=1}^{M}\}_{j=1}^{J}$, leveraging fully connected layers, ReLU activation, and batch normalization. Each vote v_{ij} comprises a 3D point y_{ij} close to a grasp center in Euclidean space and a F-dimensional feature vector f_{ij}^v. Clustering the votes via uniform sampling and Euclidean distance identifies K votes $\{v_k\}_{k=1}^{K}$. Using iterative farthest point sampling (FPS) based on $\{y_i\}$, K clusters form from the sampled votes, employing a ball query to gather votes within a set radius of the query vote v_k.

To achieve collision-free grasps in complex environments, comprehending object relationships and contextual cues within features is essential. Our VoteNet integrates a contextual module inspired by self-attention models. It utilizes an MLP and max-pooling to process cluster votes, aggregating into $f_k^c \in \mathbb{R}^{F'}$. These vectors compile into a map $f^c = [f_1^c; f_2^c; ...; f_K^c] \in \mathbb{R}^{K\times F'}$, fostering inter-cluster feature communication, significantly enhancing grasp detection performance.

Following the computation of the contextual feature map, our model employs an MLP network to detect a ranked list of grasps $G = (p, R, w, q)$. The prediction layer includes $5 + V + 2A$ channels: 3 for grasp center regression values, 1 for gripper width regression value, 1 for grasp confidence regression value, V for viewpoint scores, and A each for angle scores and angle residual regression values for in-plane rotation. Here, V and A represent the numbers of sampled viewpoints and in-plane rotations, respectively.

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

$$L_{votegrasp} = \lambda_1 L_{vote} + \lambda_2 L_{grasp} \tag{9}$$

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

$$L_{vote} = \frac{1}{M_s} \sum_i \|y_i - c_i^g\|_H \cdot \mathbb{1}(x_i) \tag{10}$$

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:

$$L_{grasp} = L_{center} + \alpha L_{rot} + \beta L_{width} + \gamma L_{score} \tag{11}$$

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 [1].

D. Dataset

We conduct evaluations and comparisons on the publicly available GraspNet-1Billion dataset [3]. 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 [3], 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.

IV. Result and Discussion

A. Implementation Details

In our implementation[1], 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 depth images and utilize a PointNet++ [36]-based feature learning network, which also yields a 256-channel output. The detailed layer parameters of PointNet++ [36] are displayed in Table I. 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.

TABLE I: Layer parameters of PointNet++ [36] based feature learning network.

layer name	input layer	layer params
SA1	point cloud	(2048,0.025,[64,64,128])
SA2	SA1	(1024,0.05,[128,128,256])
SA3	SA2	(512,0.1,[128,128,256])
SA4	SA3	(256,0.2,[128,128,256])
FP1	SA3, SA4	[256,256]
FP2	SA2, SA3	[256,256]

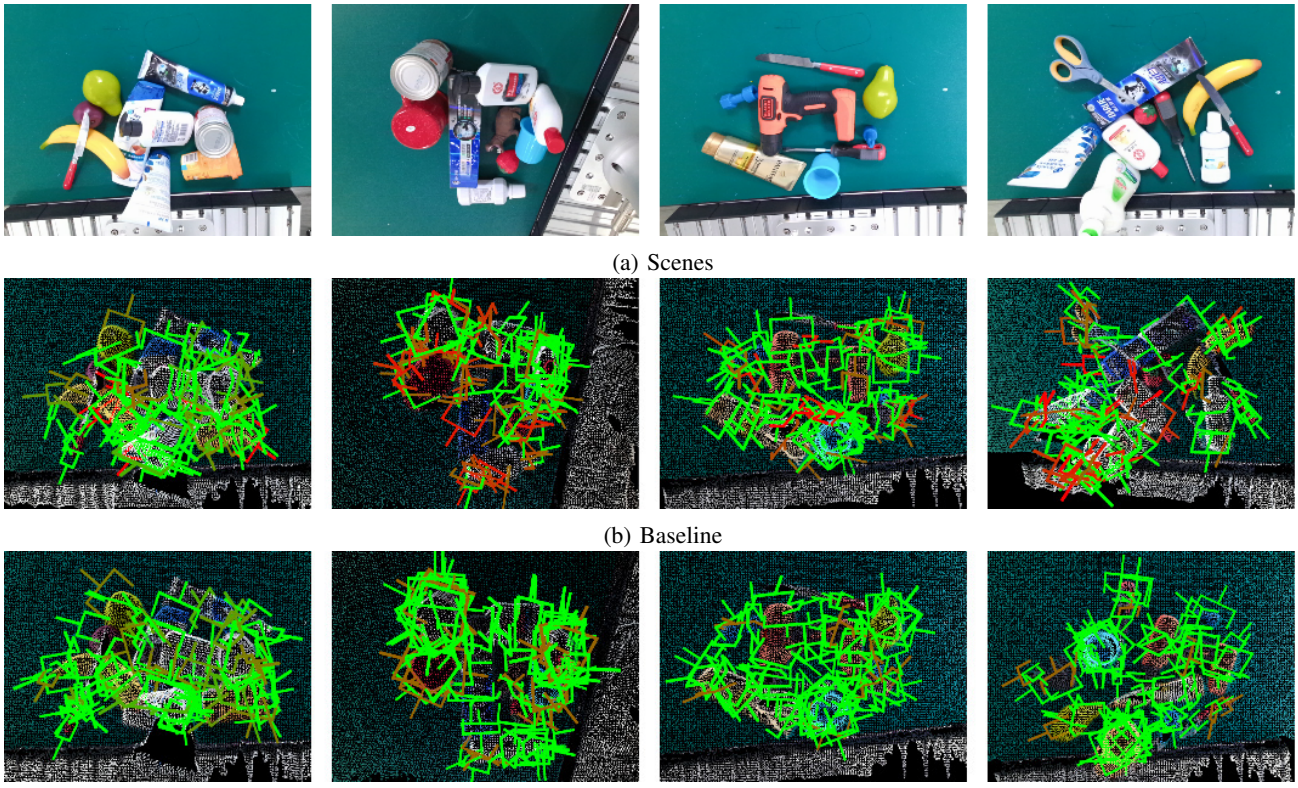
B. Evaluation on GraspNet-1Billion

We follow previous research [3] 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 [3], [8]. 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 II and Fig. 2 demonstrate the performance comparison between our approach and state-of-the-art methods. The evaluation utilized the evaluation metric adopted in [3], enabling a direct comparison with related works reported in [3], [43]. 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.

The distinctions between "Ours (-VGG)" and "Ours (-GGV)" when compared to our complete approach ("Ours") showcase the significant impact of these modules. When excluding the VGG module, the method lacks the ability to effectively integrate RGB features, leading to a considerable reduction in performance across all evaluation categories:

[1]Our code and other materials are available at `https://github.com/hoangcuongbk80/NovelVoteGrasp`



(a) Scenes

(b) Baseline

(c) Ours

Fig. 2: Examples of input scenes and predicted grasps from VoteGrasp [2] and 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.

TABLE II: 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 [41]	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. [42]	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 [8]	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
PointNetGPD [9]	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. [3]	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. [43]	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-GraspNet [44]	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 [2]	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 (-VGG)	32.0/35.1	36.4/43.7	22.1/25.4	31.0/33.8	38.3/41.2	18.3/22.4	14.8/16.1	15.0/16.3	9.1/9.3
Ours (-GGV)	31.4/34.6	35.4/42.1	21.4/24.2	30.0/32.1	37.4/40.0	17.0/21.2	13.5/15.8	14.8/15.5	8.3/8.8
Ours	**38.5/39.2**	**43.1/46.7**	**29.3/30.8**	**37.2/38.0**	**44.1/45.2**	**25.1/28.1**	**21.5/21.2**	**22.5/22.9**	**12.6/13.2**

Seen, Unseen (but similar), and Novel objects. Similarly, without the GGV module, the model fails to appropriately fuse depth-based features, resulting in notable performance degradation in grasp detection across the board. The performance drop in both cases reaffirms the critical role played by these modules in amalgamating complementary information from RGB and depth data. It highlights their significance in capturing nuanced visual cues from different sources, which are vital for accurate and robust grasp detection. This clear decline in performance underlines the necessity of the VGG and GGV modules in our model's architecture, demonstrating their collective contribution to a more comprehensive understanding of the scene by integrating information from diverse modalities. The substantial discrepancy in results showcases that these modules are not just supplementary but rather pivotal components in leveraging the combined strengths of RGB and depth information. Their absence leads to a significant loss in the model's ability to discern crucial features necessary for precise grasp detection, emphasizing the vital role of these fusion modules in enhancing the model's performance across various object scenarios.

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

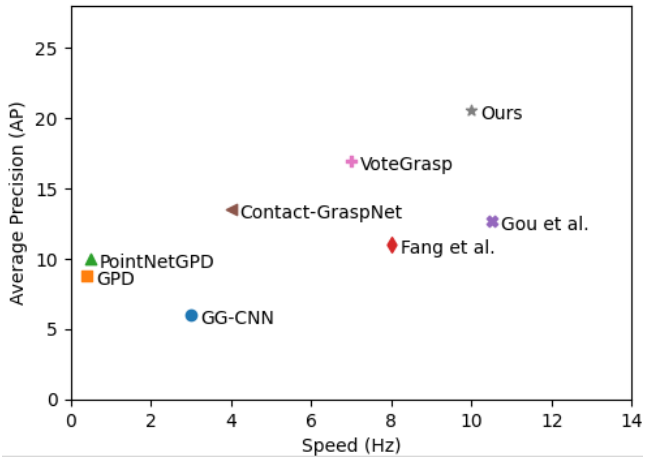


Fig. 3: Comparison of running speed (Hz) and AP on GraspNet-1Billion dataset.

of memory. The runtime analysis of all evaluated methods is graphically represented in Fig. 3. 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.

C. Robotic Grasping Experiment

The experiments were conducted with a Franka Emika Panda robot arm with 7-DOF, equipped with a parallel-jaw gripper as shown in Fig. 4. 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.

Fig. 4: Real-world grasping experiment.

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. GPD [8], PointNetGPD [9], and Contact-GraspNet [44] were trained using the hyperparameters specified in their respective papers. 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 III 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.

TABLE III: 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 [8]	300	195	65%
PointNetGPD [9]	300	201	67%
Fang et al. [3]	300	214	71%
Gou et al. [43]	300	218	73%
Contact-GraspNet [44]	300	222	74%
VoteGrasp [2]	300	234	78%
Ours	300	**251**	**84%**

V. 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 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] D.-C. Hoang, A.-N. Nguyen, V.-D. Vu, D.-Q. Vu, V.-T. Nguyen, T.-U. Nguyen, C.-T. Tran, K.-T. Phan, and N.-T. Ho, "Grasp configuration synthesis from 3d point clouds with attention mechanism," *Journal of Intelligent & Robotic Systems*, vol. 109, no. 3, p. 71, 2023.

[2] D.-C. Hoang, J. A. Stork, and T. Stoyanov, "Context-aware grasp generation in cluttered scenes," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 1492–1498.

[3] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 444–11 453.

[4] G. Du, K. Wang, S. Lian, and K. Zhao, "Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review," *Artificial Intelligence Review*, vol. 54, no. 3, pp. 1677–1734, 2021.

[5] D.-C. Hoang, A. J. Lilienthal, and T. Stoyanov, "Panoptic 3d mapping and object pose estimation using adaptively weighted semantic information," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1962–1969, 2020.

[6] ——, "Object-rpe: Dense 3d reconstruction and pose estimation with convolutional neural networks," *Robotics and Autonomous Systems*, vol. 133, p. 103632, 2020.

[7] D.-C. Hoang, L.-C. Chen, and T.-H. Nguyen, "Sub-obb based object recognition and localization algorithm using range images," *Measurement Science and Technology*, vol. 28, no. 2, p. 025401, 2016.

[8] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[9] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, "Pointnetgpd: Detecting grasp configurations from point sets," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3629–3635.

[10] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, 2023.

[11] G. Marullo, L. Tanzi, P. Piazzolla, and E. Vezzetti, "6d object position estimation from 2d images: a literature review," *Multimedia Tools and Applications*, vol. 82, no. 16, pp. 24 605–24 643, 2023.

[12] A. Bicchi and V. Kumar, "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, 2000, pp. 348–353.

[13] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 2. IEEE, 2003, pp. 1824–1829.

[14] X. Lou, Y. Yang, and C. Choi, "Learning to generate 6-dof grasp poses with reachability awareness," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1532–1538.

[15] M. Kokic, D. Kragic, and J. Bohg, "Learning task-oriented grasping from human activity datasets," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3352–3359, 2020.

[16] P. Schmidt, N. Vahrenkamp, M. Wächter, and T. Asfour, "Grasping of unknown objects using deep convolutional neural networks based on depth images," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 6831–6838.

[17] D. Yang, T. Tosun, B. Eisner, V. Isler, and D. Lee, "Robotic grasping through combined image-based grasp proposal and 3d reconstruction," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6350–6356.

[18] P. Ni, W. Zhang, X. Zhu, and Q. Cao, "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)*, 2020, pp. 3619–3625.

[19] A. Saxena, S. Chung, and A. Ng, "Learning depth from single monocular images," *Advances in neural information processing systems*, vol. 18, 2005.

[20] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 824–840, 2008.

[21] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," *Advances in neural information processing systems*, vol. 27, 2014.

[22] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2650–2658.

[23] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *2016 Fourth international conference on 3D vision (3DV)*. IEEE, 2016, pp. 239–248.

[24] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe, "Multi-scale continuous crfs as sequential deep networks for monocular depth estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5354–5362.

[25] J. Jiao, Y. Cao, Y. Song, and R. Lau, "Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 53–69.

[26] M. Ramamonjisoa and V. Lepetit, "Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[27] Z. Zhang, Z. Cui, C. Xu, Y. Yan, N. Sebe, and J. Yang, "Pattern-affinitive propagation across depth, surface normal and semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4106–4115.

[28] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," *arXiv preprint arXiv:1907.10326*, 2019.

[29] A. Kendall, Y. Gal, and R. Cipolla, "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*, 2018, pp. 7482–7491.

[30] W. Yin, Y. Liu, C. Shen, and Y. Yan, "Enforcing geometric constraints of virtual normal for depth prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5684–5693.

[31] T. Chen, S. An, Y. Zhang, C. Ma, H. Wang, X. Guo, and W. Zheng, "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, 2020, pp. 90–108.

[32] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 179–12 188.

[33] G. Yang, H. Tang, M. Ding, N. Sebe, and E. Ricci, "Transformer-based attention networks for continuous pixel-wise prediction," in *Proceedings of the IEEE/CVF International Conference on Computer vision*, 2021, pp. 16 269–16 279.

[34] L. Piccinelli, C. Sakaridis, and F. Yu, "idisc: Internal discretization for monocular depth estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 477–21 487.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[36] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[37] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 16 259–16 268.

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[39] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[40] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[41] D. Morrison, P. Corke, and J. Leitner, "Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach," in *Proc. of Robotics: Science and Systems (RSS)*, 2018.

[42] F.-J. Chu, R. Xu, and P. A. Vela, "Real-world multiobject, multigrasp detection," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3355–3362, 2018.

[43] M. Gou, H.-S. Fang, Z. Zhu, S. Xu, C. Wang, and C. Lu, "Rgb matters: Learning 7-dof grasp poses on monocular rgbd images," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 459–13 466.

[44] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," 2021.