IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 20 July 2024, accepted 13 August 2024, date of publication 15 August 2024, date of current version 26 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3444322

RESEARCH ARTICLE

Efficient Multimodal Fusion for Hand Pose Estimation With Hourglass Network

DINH-CUONG HOANG[1], PHAN XUAN TAN[2], (Member, IEEE), DUC-LONG PHAM[3], HAI-NAM PHAM[1], SON-ANH BUI[1], CHI-MINH NGUYEN[1], AN-BINH PHI[1], KHANH-DUONG TRAN[1], VIET-ANH TRINH[1], VAN-DUC TRAN[1], DUC-THANH TRAN[3], VAN-HIEP DUONG[3], KHANH-TOAN PHAN[3], VAN-THIEP NGUYEN[3], VAN-DUC VU[3], AND THU-UYEN NGUYEN[3]

[1]Greenwich Vietnam, FPT University, Hanoi 10000, Vietnam
[2]College of Engineering, Shibaura Institute of Technology, Tokyo 135-8548, Japan
[3]Department of Information Technology, FPT University, Hanoi 10000, Vietnam

Corresponding author: Dinh-Cuong Hoang (cuonghd12@fe.edu.vn)

ABSTRACT Hand pose estimation is vital for various applications, including virtual reality (VR), augmented reality (AR), gesture recognition, human-computer interaction (HCI), and robotics. Achieving accurate and real-time hand pose estimation is challenging due to factors such as the high degree of articulation in the human hand and the variability in hand shapes and sizes. While multimodal data offers advantages, developing a fast and resource-efficient hand pose estimation system remains challenging. Current state-of-the-art methods often require powerful graphics processing units (GPUs) for high performance, limiting deployment on edge platforms with limited computational resources. There is a critical need for higher efficiency without compromising accuracy, especially in real-world applications like mobile devices and embedded systems. Additionally, real-time performance is essential for practical applications, where systems must respond immediately to user interactions. Unfortunately, most current methods struggle to achieve real-time speeds, even on powerful GPUs, let alone on resource-constrained devices. To address these challenges, we propose an efficient hand pose estimation system that leverages both red-green-blue (RGB) and depth (RGBD) data through a unified fusion strategy. Our method combines appearance and geometric data early in the processing pipeline, significantly reducing computational complexity while maintaining real-time performance on resource-constrained devices. Experimental results show that the proposed model runs at over 110 fps on GPU, and 30 fps on the edge platform of NVidia Jetson NX Xavier, which is 4 to 5 times faster than existing methods, while achieving competitive accuracy.

INDEX TERMS Pose estimation, robot vision systems, intelligent systems, deep learning, supervised learning, machine vision.

I. INTRODUCTION

Hand pose estimation has emerged as a crucial technology for a variety of applications, including virtual reality, augmented reality, gesture recognition, human-computer interaction, and robotics [1], [2], [3]. In robotics, precise hand pose estimation is essential for tasks such as teleoperation, where an operator remotely controls a robot's hand movements, and for collaborative robots (cobots) that work alongside humans in industrial and domestic settings [4], [5], [6]. By accurately tracking human hand poses, robots can learn from human demonstrations, perform intricate manipulations, and safely interact with their environment and human partners [7], [8], [9]. Accurate and real-time hand pose estimation can significantly enhance the user experience in these domains by enabling more natural and intuitive interactions. Despite its potential, achieving robust and efficient hand pose estimation remains a challenging task due to the high degree of articulation in the human hand, occlusions, and variability in hand shapes and sizes.

The associate editor coordinating the review of this manuscript and approving it for publication was Zhenhua Guo.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

Traditional approaches to hand pose estimation primarily rely on either RGB or depth data [10], [11]. RGB images provide rich color and texture information, which is useful for identifying hand features, but they often struggle with occlusions and varying lighting conditions [10], [12], [13]. Depth images, on the other hand, offer geometric information about the hand's spatial structure, which is invaluable for understanding the 3D pose. However, depth data alone can be noisy and less informative about surface details [14], [15]. Therefore, a fusion of RGB and depth (RGBD) data can leverage the strengths of both modalities, potentially leading to more accurate and reliable hand pose estimation. Despite the advantages of multimodal data, developing a hand pose estimation system that is both fast and resource-efficient remains a significant challenge. Current state-of-the-art methods typically require powerful GPUs to achieve high performance, making them impractical for deployment on edge platforms with limited computational resources [16], [17], [18]. In many real-world applications, such as mobile devices and embedded systems, there is a critical need for higher efficiency without compromising accuracy [19], [20]. Moreover, real-time performance is essential for practical applications, where systems must provide immediate responses to user interactions. Unfortunately, most current state-of-the-art methods struggle to achieve real-time speeds even on powerful GPUs, let alone on resource-constrained devices.

This paper aims to address these challenges by developing a hand pose estimation system that is both efficient and capable of real-time performance on platforms with limited computational resources. By leveraging an effective fusion strategy for RGBD data and optimizing the network architecture for speed and resource efficiency, our approach seeks to deliver accurate hand pose estimation suitable for real-world applications without the need for powerful GPUs. To this end, we propose an efficient and real-time hand pose estimation method that effectively integrates RGB and depth data through a unified fusion strategy. Our approach begins by combining the RGB and depth images early in the process, forming a unified input tensor that is processed through shared convolutional layers. This early fusion reduces computational complexity while preserving the complementary information from both modalities. A key innovation in our method is the introduction of a channel attention mechanism, which dynamically balances the contributions of RGB and depth features, enhancing the most informative channels. To ensure computational efficiency suitable for real-time applications, we employ the MobileNetV2 architecture as the backbone for feature extraction. MobileNetV2 is renowned for its lightweight design, utilizing depthwise separable convolutions and inverted residuals with linear bottlenecks to maintain high performance with low computational cost. The refined feature maps from MobileNetV2 [21] are further processed by an Hourglass network [22], which performs multi-scale analysis to refine pose predictions. To integrate multi-scale features more effectively, we also incorporate intermediate feature maps from different layers of MobileNetV2 into the Hourglass network. This multi-scale fusion approach ensures that the model leverages a rich set of features at various levels of abstraction, improving the accuracy of hand pose estimation. The effectiveness of our approach is validated through extensive experiments, demonstrating its ability to achieve high accuracy in hand pose estimation while maintaining real-time performance.

- **Efficient Multimodal Hand Pose Estimation Network**. We present a deep learning approach for real-time multimodal hand pose estimation, which fuses depth cues with RGB images and predicts poses using limited onboard computational resources.
- **Efficient RGBD Fusion**. We introduce a novel, efficient modality-aware fusion module that effectively integrates complementary information from RGB and depth data while maintaining a lightweight and real-time architecture. This approach combines RGB and depth data early in the process and utilizes shared convolutional layers along with a channel attention mechanism to extract initial fused features.
- **Multi-Scale Feature Extraction**. We carefully design and integrate a lightweight MobileNetV2 backbone with a single Hourglass network to extract hierarchical features efficiently, capturing intricate hand pose patterns.

The structure of this article is as follows. In Section II, we present related work, specifically addressing RGB-based hand pose estimation in Section II-A, depth-based hand pose estimation in Section II-B, and RGBD fusion in Section II-C. Section III outlines our proposed methodology, breaking down the process into distinct components such as the early fusion of RGB and depth data (Section III-A), optimized multi-scale feature extraction using MobileNetV2 and (Section III-B), the modified Hourglass network (Section III-C), and loss function (Section III-D). Moving on to Section IV, we cover the evaluation process, including datasets (Section IV-A), training details (Section IV-B), evaluation metrics (Section IV-C), the results (Section IV-D), and ablation study (Section IV-E). Finally, in Section V, we draw conclusions. The detailed abbreviations and definitions used in the paper are listed in Table 1.

II. RELATED WORK

A. RGB-BASED HAND POSE ESTIMATION

Most of the current approaches have utilized either RGB or depth data [10], [11]. RGB-based hand pose estimation methods rely on the color and texture information available in RGB images [23]. These methods have been extensively researched and developed due to the widespread availability of RGB cameras and the rich visual information they provide. RGB images capture detailed surface textures and color variations, which can be beneficial for identifying and tracking hand features. Early approaches in RGB-based hand pose estimation utilized traditional computer vision techniques, such as edge detection, template matching, and skin color segmentation [10], [12], [24], [25]. These methods were

TABLE 1. List of abbreviation and acronyms.

Abbreviation	Definition
RGB	Red-Green-Blue
RGBD	Red-Green-Blue and Depth
VR	Virtual reality
GPU	Graphics processing unit
CNN	Convolutional neural network
DNN	Deep neural network
CMR	Camera-space mesh recovery
FIT	Feature injecting transformer
SET	Self-enhancing transformer
2D	Two-dimensional
3D	Three-dimensional
LiDAR	Light Detection and Ranging
BEV	Bird's eye view
ReLU	Rectified linear unit
BN	Batch normalization

often limited by their reliance on handcrafted features and their inability to generalize well to different hand shapes and poses. With the advent of deep learning, significant advancements have been made in RGB-based hand pose estimation. Convolutional neural networks (CNNs) have been employed to automatically learn features from large datasets of hand images. These networks can capture complex patterns and structures, enabling more accurate and robust hand pose estimation. For instance, Zimmermann and Brox [26] introduced a deep learning framework that predicts 3D hand joint positions from RGB images using a two-stage approach: first, a CNN estimates 2D joint locations, and then a deep neural network (DNN) lifts these 2D locations to 3D coordinates. Another notable approach is the work by Iqbal et al. [27], which proposed an approach for estimating 3D hand poses from monocular images, leveraging a unique 2.5D pose representation. This new representation captures pose details while accommodating for scaling variations, which can be refined with prior knowledge of hand size. The method involves the implicit learning of depth maps and heatmap distributions using an CNN architecture. Reference [13] decomposes the task into three stages: the joint stage predicts the 3D coordinates of hand joints and the hand segmentation mask; the mesh stage estimates a rough 3D hand mesh; and the refine stage aggregates local and global features from earlier layers, learning to regress per-vertex offset vectors for aligning the rough mesh to the hand image with finger-level precision. On a similar note, [28] present a camera-space mesh recovery (CMR) framework that unifies the tasks of 3D hand mesh and root recovery into a coherent system, involving 2D cue extraction, 3D mesh recovery, and global mesh registration phases. Inspired by the advancements in neural language processing, [29] utilize a transformer encoder to jointly model vertex-vertex and vertex-joint interactions, producing 3D joint coordinates and mesh vertices simultaneously. Addressing occlusion challenges posed by objects, [30] introduce HandOccNet, a 3D hand mesh estimation network that harnesses information from occluded regions to enhance image features. HandOccNet integrates two successive Transformer-based modules, the feature injecting transformer (FIT), and self-enhancing transformer (SET). FIT injects hand information into occluded regions by considering their correlation, while SET refines the FIT output using a self-attention mechanism. Through the infusion of hand information into occluded regions, HandOccNet exhibits promising performance on 3D hand mesh benchmarks, particularly those featuring challenging hand-object occlusions.

Despite these advancements, RGB-based methods still face challenges related to occlusions, complex backgrounds, and varying lighting conditions. Depth images, on the other hand, provide geometric information about the hand's spatial structure, which is invaluable for understanding the 3D pose. Depth data directly offers distance measurements from the sensor to the hand surface, which helps in mitigating issues related to occlusions and varying illumination.

B. DEPTH-BASED HAND POSE ESTIMATION

Depth-based Hand Pose Estimation focuses on reconstructing 3D hand shapes from single depth maps, overcoming the limitations of 2D images by leveraging depth information. Reference [31] introduced a hierarchical PointNet architecture for point-to-point regression, achieving precise pose estimation by capturing detailed geometric features from depth maps. Meanwhile, [14] proposed V2V-PoseNet, converting 2D depth images into 3D volumetric forms for accurate 3D hand and human pose estimation. Spatial attention-based method [32] enhances prediction accuracy by focusing on relevant parts of depth maps. Reference [33] utilizes end-to-end learning with multiple anchor points for improved prediction accuracy by capturing global and local spatial contexts. Additionally, [34] unifies dense representation and hand joint regression with Adaptive Weighting Regression (AWR), enhancing robustness and accuracy. Furthermore, methods like HandVoxNet++ [35] and HandFoldingNet [36] incorporate advanced techniques such as TSDF-based voxel-to-voxel networks and folding-based decoders, respectively, to capture fine-grained details of hand shapes.

Nevertheless, depth data alone can be noisy and less informative about surface details. Noise in depth data, especially around the edges and at distant points, can degrade the accuracy of pose estimation. The fusion of RGB and depth data, known as RGBD Fusion, has emerged as a prominent research area, particularly with the proliferation of accessible RGBD sensors [37]. The fusion of RGB and depth modalities aims to exploit the complementary strengths of both, potentially leading to more robust and accurate computer vision applications.

C. RGBD FUSION

Traditional RGBD fusion methods typically involve extracting and combining handcrafted features from RGB images and depth maps [49], [50], [51], [52]. However, these methods often fall short due to the inherent limitations of handcrafted features, which lack the complexity and adaptability

TABLE 2. Deep learning-based RGBD fusion methods.

	Undirected	Unidirectional	Bidirectional
Wang et al. [38]	✓		
Gupta et al. [39]	✓		
He et al. [40]	✓		
Wang et al. [41]	✓		
Liang et al. [42]		✓	
Huang et al. [43]		✓	
Wang et al. [16]		✓	
Qi et al. [44]		✓	
Hoang et al. [45]		✓	
Li et al. [46]			✓
Hu et al. [47]			✓
He et al. [18]			✓
Chen et al. [48]			✓

needed for robust performance. With the advent of deep convolutional neural networks (CNNs), the field has seen a paradigm shift. CNN-based approaches leverage the powerful feature extraction and representation capabilities of deep learning, leading to substantial improvements in RGBD fusion performance and setting new standards [53], [54], [55]. Fusion schemes can be categorized into three types based on how information flows (as shown in Table 2): undirected fusion [38], [39], [40], [41], unidirectional fusion [16], [42], [43], [44], [45], and bidirectional fusion [18], [46], [47], [48].

Undirected fusion is most commonly implemented by directly concatenating or adding the separately extracted features. For instance, DenseFusion [41] and PVN3D [40] process RGB and depth data individually, using a dense fusion network to extract pixel-wise dense feature embeddings. Geometric features are derived by converting depth pixels into a 3D point cloud with camera intrinsics and utilizing a PointNet-like architecture. Simultaneously, color features are obtained through a CNN-based encoder-decoder that transforms images into a dense feature space, where each pixel is represented by a multi-dimensional vector. These features are then fused locally on a per-pixel basis: each point's geometric feature is paired with its corresponding image pixel feature, concatenated, and processed by another network to produce a global feature vector.

Unidirectional fusion [16], [42], [43], [44] leverages information flow from one modality (either RGB or depth) to guide or enhance the processing of the other modality. In these approaches, features from one modality refine or augment the features of the other, ensuring that the strengths of both modalities are utilized effectively. Huang et al. [43] introduced a LiDAR-guided Image Fusion (LI-Fusion) module, where semantic features from images enhance point features derived from LiDAR data. The module establishes point-wise correspondence between raw point cloud data and camera images, adaptively weighing the importance of image semantic features. This method enhances useful image features while suppressing interfering ones, thus improving the quality of point features. Liang et al. [42] proposed a method that projects image features extracted by a convolutional network into bird's eye view (BEV) and integrates them with the convolution layers of a LiDAR-based network. This fusion process involves interpolating discrete image features to create a dense BEV feature map, using continuous convolutions to extract relevant information from the nearest corresponding image features for each point in BEV space. Wang et al. [16] developed a Geometry-Aware Visual Feature Extractor (GAVE) to produce distinctive and comprehensive geometric-visual features from RGBD images. This extractor facilitates better point cloud registration by reliably estimating correspondences. Within the GAVE module, a Local Linear Transformation (LLT) technique uses geometric features (from a geometric feature extractor) as guiding signals, converting them into point-wise linear coefficients. These coefficients are then applied to enhance the visual features (from a visual feature extractor) through point-wise linear transformation, improving the overall feature representation.

Bidirectional fusion [18], [47], [56] facilitates a two-way exchange of information between RGB and depth modalities, allowing each to iteratively refine and enhance the other throughout the network. This approach leverages the strengths of both data types to create more comprehensive feature representations. Hu et al. [47] introduced the Bidirectional Projection Network (BPNet) for joint 2D and 3D reasoning in an end-to-end manner. BPNet features symmetric 2D and 3D sub-networks connected by a Bidirectional Projection Module (BPM). The BPM facilitates interaction between complementary 2D and 3D information across multiple layers, significantly enhancing scene recognition capabilities by allowing detailed and nuanced feature sharing. He et al. [18] developed the FFB6D network, which integrates appearance and geometry information to improve both representation learning and output selection. This network employs bidirectional fusion modules that merge information at every encoding and decoding layer. It extracts features from RGB images with a CNN and from point clouds with a point cloud network, using the fusion modules to enable the exchange of complementary information. This process enhances the distinctiveness of the features and addresses challenges like incomplete depth data in point clouds and similar object appearances in RGB images. Bidirectional fusion effectively utilizes the strengths of both RGB and depth modalities, enabling robust and accurate feature representations. By allowing continuous and mutual refinement of features, this approach mitigates issues such as noisy depth data and ambiguous visual cues, leading to improved performance in complex tasks.

While the above fusion methods have achieved promising results in terms of accuracy, they often result in more complex network architectures. These architectures require separate subnetworks and additional fusion layers, leading to increased training requirements, more parameters to optimize, and greater demands on data and computational resources. Beyond performance, efficiency is crucial for practical deployment. Instead of using two separate streams to handle RGB and depth information as existing methods do, we propose an approach that combines RGB and depth data at an early stage, processing them through a single network. To effectively capture the complex relationships between RGB and depth data within this streamlined architecture, we employ a channel attention mechanism. This mechanism adaptively balances the contributions of RGB and depth features, enhancing the model's ability to integrate complementary information from both modalities. We further enhance the network by incorporating intermediate feature maps from different layers into an Hourglass network for tasks such as hand pose estimation. This multi-scale fusion approach ensures that the model leverages a rich set of features at various levels of abstraction, leading to improved performance while maintaining a simpler and more efficient architecture.

III. METHODOLOGY

In this section, we present our Deep Neural Network (DNN) architecture designed to recover hand configuration from a single RGBD image in one forward pass. As illustrated in Figure 1, the architecture comprises three main components: efficient RGBD fusion, a backbone network, and the Hourglass network for hand pose estimation. We employ the MANO parametric model [57] to represent the hand. The network outputs the MANO pose and shape parameters, providing a comprehensive hand configuration.

A. EFFICIENT RGBD FUSION

To effectively integrate the complementary information from RGB and depth data while maintaining a lightweight and real-time architecture, we propose a unified fusion strategy (as shown in Figure 2). This approach combines RGB and depth data early in the process and utilizes shared convolutional layers to extract features, thereby reducing computational complexity. Initially, let $\mathcal{I}_{\text{RGB}} \in \mathbb{R}^{H\times W\times 3}$ and $\mathcal{I}_{\text{D}} \in \mathbb{R}^{H\times W\times 1}$ denote the RGB and depth images, respectively. These inputs are concatenated along the channel dimension to form a unified input tensor $\mathcal{I}_{\text{RGBD}} \in \mathbb{R}^{H\times W\times 4}$:

$$\mathcal{I}_{\text{RGBD}} = [\mathcal{I}_{\text{RGB}}, \mathcal{I}_{\text{D}}] \tag{1}$$

The concatenated RGBD input is then processed through a series of shared convolutional layers designed to extract joint features from both modalities efficiently. Each convolutional layer is followed by batch normalization (BN) and ReLU activation. The first convolutional layer uses a 3 × 3 filter size with 32 filters, a stride of 2, and padding set to 'same', resulting in output dimensions of 224 × 224 × 32. Let $\mathcal{F}_{\text{RGBD}} \in \mathbb{R}^{224\times 224\times 32}$ represent the feature map obtained after this convolutional processing. To dynamically balance the contributions of RGB and depth features, we introduce a channel attention mechanism. This mechanism computes a channel attention vector $\mathbf{a} \in \mathbb{R}^{32}$ that adjusts the importance of each channel in the feature map. Channel attention can effectively integrate information from both RGB and depth data by dynamically weighing the importance of different channels. This is particularly useful when dealing with multimodal inputs, as it can adaptively prioritize channels that carry more significant information from either modality. The channel attention vector is computed as follows:

$$\mathbf{a} = \sigma(\text{FC}(\text{GAP}(\mathcal{F}_{\text{RGBD}}))) \tag{2}$$

where GAP denotes global average pooling, FC represents a fully connected layer that reduces the pooled features to a vector of length 32, and σ is the sigmoid function. The channel attention vector $\mathbf{a}$ is then used to weight the channels of the feature map:

$$\mathcal{F}_{\text{RGBD}}^{\text{att}} = \mathbf{a} \odot \mathcal{F}_{\text{RGBD}} \tag{3}$$

where $\odot$ denotes element-wise multiplication. This operation enhances the most informative features across the channels. The final fused feature map $\mathcal{F}_{\text{fused}} \in \mathbb{R}^{224\times 224\times 32}$ is then passed through an additional convolutional layer to ensure compatibility with the input requirements of the subsequent backbone network MobileNetV2. This layer uses a 1 × 1 convolution with 3 filters, batch normalization (BN), and ReLU activation to reduce the number of channels to 3, making it suitable for the backbone:

$$\mathcal{F}_{\text{fused}} = \text{Conv}_{1\times 1}(\mathcal{F}_{\text{RGBD}}^{\text{att}}, 3) \rightarrow \text{BatchNorm} \rightarrow \text{ReLU} \tag{4}$$

This results in a feature map with dimensions $\mathbb{R}^{224\times 224\times 3}$, which is compatible with the input requirements of the backbone MobileNetV2 [21]. This efficient RGBD fusion module combines RGB and depth data in a unified manner and utilizes attention mechanisms to dynamically enhance the feature representation. By leveraging shared convolutional layers and channel attention, the proposed method maintains computational efficiency and is well-suited for real-time hand pose estimation.

B. BACKBONE

The initial fusion stage provides a combined representation of RGB and depth data, but these features are relatively low-level and local. The backbone network, used in this step, plays a crucial role in the hierarchical extraction of high-level features from the initial fused RGBD input. While the Efficient RGBD Fusion primarily focuses on combining and enhancing the complementary information from RGB and depth data, the backbone network is responsible for further processing this fused input to extract more abstract and discriminative features.

MobileNetV2 [21] backbone is used for feature extraction from the initial RGBD features $\mathcal{F}_{\text{fused}}$. MobileNetV2 is known for its efficient and lightweight architecture, making it ideal for real-time applications on mobile and embedded devices. MobileNetV2 employs depthwise separable convolutions, which reduce the computational cost by separating the spatial and channel-wise convolutions. It also uses inverted residuals, which employ an inverted residual structure with linear bottlenecks, expanding the intermediate layers and projecting back to a lower-dimensional space. The ReLU6 activation function is used, clipping the ReLU activation at 6 to ensure robustness in low-precision computations.

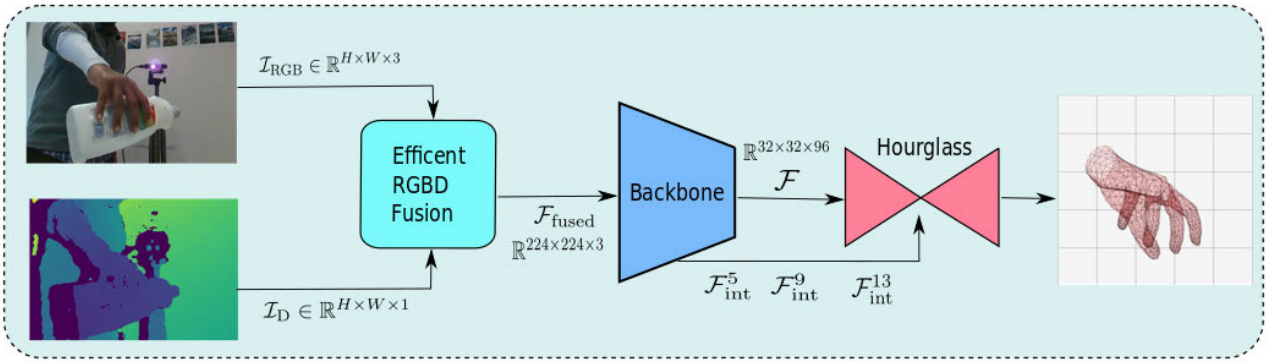

$\mathcal{I}_{\text{RGB}} \in \mathbb{R}^{H\times W\times 3}$
$\mathcal{I}_{\text{D}} \in \mathbb{R}^{H\times W\times 1}$
Efficent RGBD Fusion
$\mathcal{F}_{\text{fused}}$
$\mathbb{R}^{224\times224\times3}$
Backbone
$\mathbb{R}^{32\times32\times96}$
$\mathcal{F}$
Hourglass
$\mathcal{F}^5_{\text{int}}$ $\mathcal{F}^9_{\text{int}}$ $\mathcal{F}^{13}_{\text{int}}$

FIGURE 1. Overview of the proposed network for efficient and accurate hand pose estimation. The network begins with an early fusion module for initial feature extraction and fusion from the RGB and depth inputs, forming a unified input tensor that is processed through shared convolutional layers and channel attention. This early fusion reduces computational complexity while preserving complementary information from both modalities. To ensure computational efficiency suitable for real-time applications, we employ the MobileNetV2 [21] architecture as the backbone for feature extraction. The refined feature maps from MobileNetV2 are further processed by a single Hourglass network, performing multi-scale analysis to refine pose predictions. Intermediate feature maps from different layers of MobileNetV2 are integrated into a single Hourglass network [22] to leverage a rich set of features at various levels of abstraction, improving the accuracy of hand pose estimation.

The 13th bottleneck layer outputs a feature map with a spatial resolution of 32 × 32 and 96 channels. This layer strikes a balance between maintaining sufficient spatial resolution for detailed feature extraction and having a rich set of features for subsequent processing by the Hourglass network. Truncating the MobileNetV2 after the 13th bottleneck layer ensures that the model remains computationally efficient, which is crucial for real-time applications.

The feature map output from MobileNetV2 backbone is denoted as $\mathcal{F} \in \mathbb{R}^{32\times32\times96}$. A transition layer is introduced to adapt the output feature map from MobileNetV2 to the input requirements of the Hourglass network. This transition layer includes a 1 × 1 convolution that adjusts the number of channels from 96 to 256, ensuring compatibility with the Hourglass network while preserving spatial dimensions. The transformed feature map is denoted as $\mathcal{F}_T \in \mathbb{R}^{32\times32\times256}$.

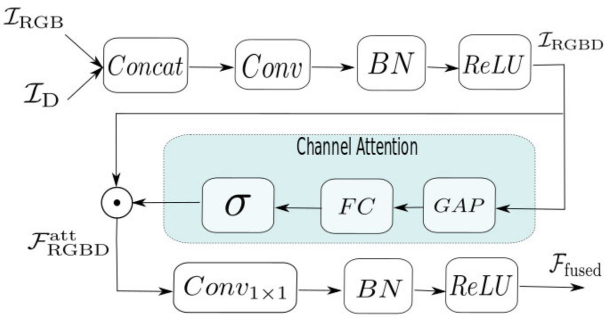

$\mathcal{I}_{\text{RGB}}$
$\mathcal{I}_{\text{D}}$
$Concat$
$Conv$
BN
$ReLU$
$\mathcal{I}_{\text{RGBD}}$
Channel Attention
σ
FC
GAP
$\mathcal{F}^{\text{att}}_{\text{RGBD}}$
$Conv_{1\times1}$
BN
$ReLU$
$\mathcal{F}_{\text{fused}}$

FIGURE 2. Efficient RGBD fusion module.

C. HAND POSE ESTIMATION WITH HOURGLASS NETWORK

The Hourglass network [22] processes the features extracted by MobileNetV2, allowing the model to refine pose predictions through multi-scale analysis. We use a single hourglass module for this purpose. The hourglass module consists of downsampling followed by upsampling layers, with skip connections between corresponding levels to combine high-resolution and low-resolution features. In the downsampling path, 3 × 3 convolutions reduce spatial resolution while increasing the number of channels, batch normalization normalizes the feature maps to accelerate training and improve stability, ReLU activation introduces non-linearity, and max pooling reduces spatial dimensions by half. The downsampling path reduces the resolution to 4 × 4 while increasing the channels to 512. The bottleneck layer processes the condensed features at the lowest resolution to capture global context and interactions. In the upsampling path, nearest neighbor upsampling increases spatial resolution by a factor of 2, 3 × 3 convolutions refine the upsampled features, and batch normalization and ReLU activation are applied after each convolution. The upsampling path restores the resolution to 32 × 32 while reducing the channels back to 256. Skip connections link corresponding downsampling and upsampling layers to merge fine-grained details from higher resolutions with contextual information from lower resolutions. The feature map output from the Hourglass network is denoted as $\mathcal{F}_H \in \mathbb{R}^{32\times32\times256}$.

To further enhance the model's performance, features from different layers of MobileNetV2 are combined with features from the Hourglass network, enabling the model to leverage multi-scale features more effectively. Specifically, we extract intermediate feature maps from the 5th, 9th, and 13th bottleneck layers of MobileNetV2, denoted as $\mathcal{F}^5_{\text{int}} \in \mathbb{R}^{64\times64\times24}$, $\mathcal{F}^9_{\text{int}} \in \mathbb{R}^{32\times32\times32}$, and $\mathcal{F}^{13}_{\text{int}} \in \mathbb{R}^{32\times32\times96}$, respectively. These intermediate feature maps provide varying levels of spatial resolution and feature richness. At each corresponding downsampling and upsampling stage of the Hourglass network, these intermediate features from MobileNetV2 are concatenated with the feature maps of the Hourglass network. For instance, at the first downsampling stage of the Hourglass network, $\mathcal{F}^5_{\text{int}}$ is resized to match the spatial dimensions of the Hourglass network's feature map and then concatenated, resulting in $\mathcal{F}^1_{\text{cat}} = [\mathcal{F}^1_H, \mathcal{F}^5_{\text{int}}]$. Similarly, at subsequent stages, $\mathcal{F}^9_{\text{int}}$ and $\mathcal{F}^{13}_{\text{int}}$ are concatenated with the Hourglass feature maps at appropriate resolutions. After concatenation, 1×1 convolutions are used to fuse the combined features, ensuring the number of channels is appropriate for the next layer. This operation can be mathematically expressed as:

$$\mathcal{F}^i_{\text{fused}} = \text{Conv}_{1\times 1}([\mathcal{F}^i_H, \mathcal{F}^i_{\text{int}}]) \tag{5}$$

where $\mathcal{F}^i_{\text{fused}}$ represents the fused feature map at stage i, $\text{Conv}_{1\times 1}$ denotes a 1×1 convolution, and $[\cdot]$ denotes the concatenation operation. The output layer generates a set of heatmaps, each corresponding to a keypoint. The heatmaps have the same spatial resolution as the final feature map (32×32). A 1×1 convolution is applied to the final feature map to produce the heatmaps. Each heatmap represents the probability distribution of a keypoint's location, with a peak indicating the predicted keypoint position. The heatmaps are upsampled to match the original input image resolution (256×256). The set of output heatmaps is denoted as $\mathcal{H} \in \mathbb{R}^{32\times 32\times K}$, where K is the number of keypoints.

Subsequently, the feature maps generated by the Hourglass network and the heatmaps are fused via 1×1 convolutions and element-wise addition. This fusion process ensures that the refined features from the Hourglass network and the probabilistic keypoint information from the heatmaps are combined effectively, enhancing the accuracy and robustness of the hand pose estimation. The fused feature maps are then fed into four successive residual blocks, which are designed to further refine the features while preserving spatial information through skip connections. Each residual block consists of two 3×3 convolutional layers with batch normalization and ReLU activation, followed by a skip connection that adds the input of the block to its output. These residual blocks help in learning more complex representations and improving the gradient flow during training.

The output of the final residual block is a high-dimensional feature map that encapsulates detailed information about the hand pose and shape. This feature map is then flattened into a 1024-dimensional vector. Flattening converts the spatial dimensions into a single vector, making it suitable for subsequent fully connected layers. This 1024-dimensional vector is then fed into two fully connected layers to predict the hand pose and shape parameters according to the MANO model [57]. The first fully connected layer reduces the dimensionality of the vector while the second fully connected layer outputs the MANO parameters. These parameters include the hand pose parameters $\boldsymbol{\theta} \in \mathbb{R}^{48}$ and the shape parameters $\boldsymbol{\beta} \in \mathbb{R}^{10}$. The pose parameters $\boldsymbol{\theta}$ represent the joint angles of the hand, while the shape parameters $\boldsymbol{\beta}$ capture variations in hand shape. With the obtained MANO parameters, we use the MANO model to generate the estimated 3D hand mesh $\mathbf{V} \in \mathbb{R}^{778\times 3}$, which provides a detailed surface representation of the hand. Additionally, the 3D coordinates of the hand joints $\mathbf{J} \in \mathbb{R}^{21\times 3}$ are computed, representing the keypoints necessary for various hand pose estimation tasks. These outputs can be used for applications such as gesture recognition, virtual reality interactions, and hand tracking in augmented reality.

D. LOSS FUNCTION

To train our network, we minimize a loss function defined as a combination of L2 distances between the predicted and ground truth values of $\mathcal{H}$, $\boldsymbol{\theta}$, $\boldsymbol{\beta}$, $\mathbf{V}$, and $\mathbf{J}$. The overall loss function for the hand pose estimation task, L_{hand}, is expressed as follows:

$$L_{hand} = L_H + L_{3d} + L_{mano} \tag{6}$$

L_H denotes the L2 loss for 2D joint point detection, imposed on the heatmaps $\mathcal{H}$. This loss ensures that the predicted heatmaps accurately represent the 2D locations of the hand keypoints:

$$L_H = \sum_{i=1}^{K} \left\| \mathcal{H}_i - \mathcal{H}^{gt}_i \right\|_2^2 \tag{7}$$

where $\mathcal{H}_i$ and $\mathcal{H}^{gt}_i$ are the predicted and ground truth heatmaps for the i-th keypoint, respectively. L_{3d} stands for the L2 loss imposed on the 3D vertices $\mathbf{V}$ and 3D joint coordinates $\mathbf{J}$. This loss ensures that the predicted 3D mesh and joint positions closely match the ground truth:

$$L_{3d} = \left\| \mathbf{V} - \mathbf{V}^{gt} \right\|_2^2 + \left\| \mathbf{J} - \mathbf{J}^{gt} \right\|_2^2 \tag{8}$$

where $\mathbf{V}$ and $\mathbf{V}^{gt}$ are the predicted and ground truth 3D vertices, and $\mathbf{J}$ and $\mathbf{J}^{gt}$ are the predicted and ground truth 3D joint coordinates. L_{mano} is the L2 loss on the MANO parameters $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$, ensuring that the predicted hand pose and shape parameters are accurate:

$$L_{mano} = \left\| \boldsymbol{\beta} - \boldsymbol{\beta}^{gt} \right\|_2^2 + \left\| \boldsymbol{\theta} - \boldsymbol{\theta}^{gt} \right\|_2^2 \tag{9}$$

where $\boldsymbol{\beta}$ and $\boldsymbol{\beta}^{gt}$ are the predicted and ground truth shape parameters, and $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^{gt}$ are the predicted and ground truth pose parameters.

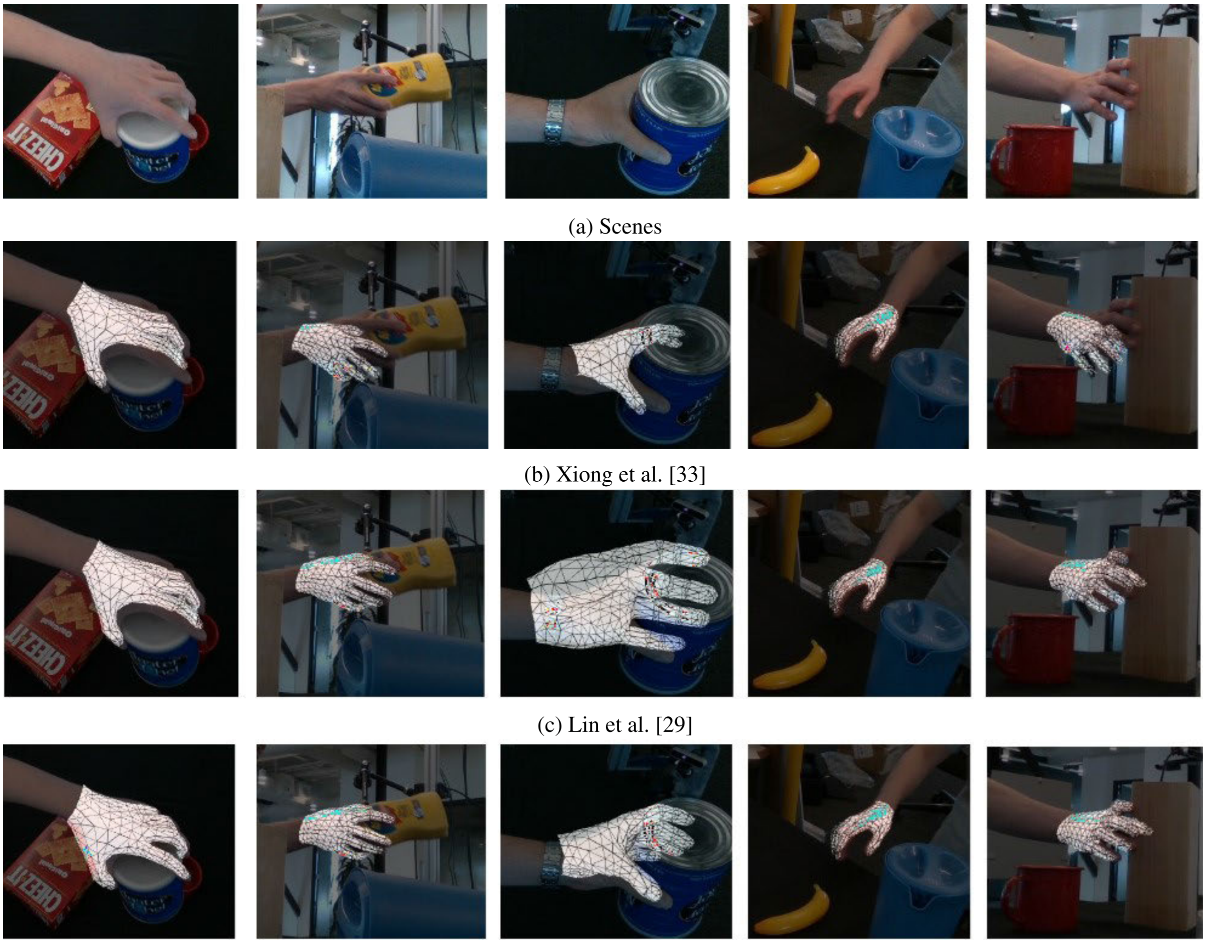
IV. EVALUATION

In this section, we extensively evaluate our proposed method using three publicly available RGBD datasets: HO-3D [58], FPHAB [59], and DexYCB [60]. These datasets are specifically curated to capture hand poses in real-world scenarios, providing a robust testing ground for evaluating the performance of hand pose estimation methods under realistic conditions. Our evaluation includes comparisons with state-of-the-art RGB-based and depth-based approaches, enabling us to assess the effectiveness of our proposed system.

A. DATASETS

1) HO-3D DATASET [58]

This dataset is specifically designed for studying hand-object interactions. It supports the development and evaluation of algorithms for accurate hand pose estimation, crucial for advancing manipulation tasks. The dataset comprises RGBD video sequences with detailed annotations of hand and object poses, including the 3D positions and orientations of hand joints and 6D poses of objects. Capturing realistic interactions such as grasping and manipulating various objects,



(a) Scenes

(b) Xiong et al. [33]

(c) Lin et al. [29]

(d) Ours

FIGURE 3. Qualitative comparison of the proposed method and state-of-the-art hand pose estimation methods.

the HO-3D dataset ensures a diverse range of shapes, sizes, and scenarios. It includes 77,558 frames distributed across 68 sequences, capturing interactions involving 10 individuals and 10 different objects. Such diversity facilitates the exploration of complex hand-object interactions under varying conditions, ensuring a rich array of scenarios for algorithm development and evaluation. By providing detailed data on hand-object interactions, including challenging occlusions, the HO-3D dataset significantly advances research and development in hand pose estimation.

2) FPHAB DATASET [59]

The First-Person Hand Action Benchmark (FPHAB) dataset comprises RGBD video sequences, encompassing over 100,000 frames capturing 45 distinct daily hand action categories. These actions involve 26 different objects and cover various hand configurations. Hand pose annotations were acquired through a proprietary motion capture (mocap) system, which utilizes six magnetic sensors and inverse kinematics to automatically infer the 3D location of each of the 21 joints in a hand model. The 45 diverse daily hand action categories were meticulously designed to encompass a wide array of hand configurations, ensuring diversity in both hand pose and action space. Each object is linked to a minimum of one and a maximum of four associated actions, providing a rich dataset for developing robust algorithms. The recorded hand actions are organized into three distinct scenarios for comprehensive coverage. A unique aspect of the FPHAB dataset is its first-person view, captured using a head-mounted camera, providing a realistic and immersive perspective of hand actions. The multi-modal data, including both RGB and depth information, allows for the development of robust algorithms. Applications of the FPHAB dataset include hand action recognition, hand pose estimation, object interaction analysis, and improvements in augmented reality (AR) and virtual reality (VR) environments. The dataset is particularly valuable for training and testing hand pose estimation algorithms.

3) DexYCB DATASET [60]

This is a large dataset used in robotics and computer vision, particularly for tasks involving hand-object interaction, grasping, and manipulation. It is designed to support the development and benchmarking of algorithms by providing

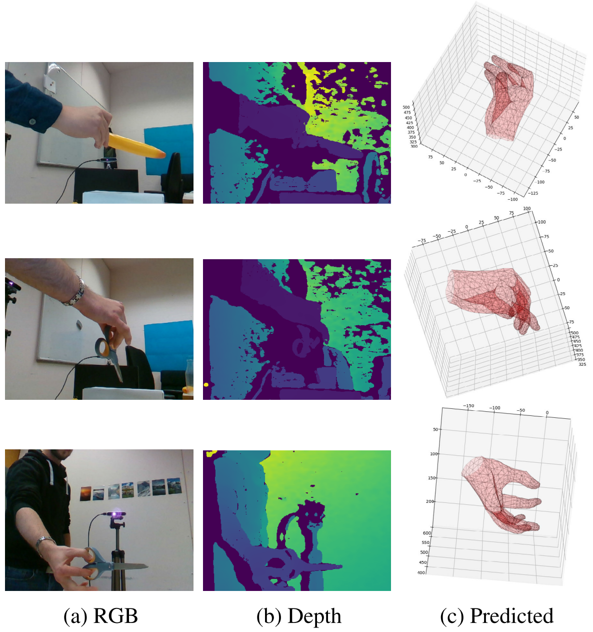

(a) RGB (b) Depth (c) Predicted

FIGURE 4. Qualitative results of the proposed method on HO-3D dataset.

synchronized RGBD video sequences, detailed hand poses, and object poses. The dataset features a variety of everyday objects from the YCB (Yale-CMU-Berkeley) object set, including tools, kitchenware, and other household items. It encompasses a substantial dataset of 582,000 RGBD frames distributed across 1,000 sequences, involving 10 subjects interacting with 20 distinct objects from eight different viewpoints. A standout feature of the DexYCB dataset is its detailed hand pose annotations. These annotations include the 3D positions and orientations of the hand joints, capturing the intricate movements and interactions of the human hand with various objects. The dataset provides accurate, frame-by-frame hand pose data, essential for training and evaluating models in hand pose estimation. This allows for the development of algorithms that can predict the 3D pose of the human hand from visual data with high precision. Noteworthy in its setup, DexYCB employs an instrumentation of eight RGBD cameras configured to capture an expansive workspace, allowing human subjects to interact with objects freely. This multi-camera setup provides comprehensive coverage from eight different viewpoints, ensuring diverse and challenging interaction scenarios where a human hand manipulates objects in realistic conditions. The multi-modal nature of the dataset, combining RGBD data, makes it valuable for algorithms that leverage both visual and depth information. These features make the DexYCB dataset ideal for applications in hand pose estimation, object grasping and manipulation, human-robot interaction, and augmented reality (AR) and virtual reality (VR).

B. TRAINING NETWORK

We train our network and comparative models on the three datasets. The HO-3D dataset was split into 66,034 images for training and 11,524 images for testing. Similarly, the FPHAB dataset was divided into 82,545 training images and 16,986 testing images, while the DexYCB dataset included 118,575 images for training and 23,187 images for testing. During data preparation, we applied augmentation techniques such as random cropping, rotation, and scaling, along with normalization, to enhance the model's robustness. We used the Adam optimizer with an initial learning rate of 0.001, adjusted using a cosine annealing schedule, and trained the model with a batch size of 16 for 180 epochs, incorporating early stopping to prevent overfitting. Regularization techniques like dropout and batch normalization were employed to improve generalization.

For the training phase, we utilized a 32GB Tesla V100 GPU, which provided the necessary computational power and memory capacity to handle the large datasets and complex model architecture efficiently. For inference, all comparison experiments are executed on the same GPU. In addition, we tested our trained model on NVIDIA Jetson NX Xavier, a powerful yet compact platform designed for edge computing. This deployment strategy was chosen to ensure that our hand pose estimation model could operate efficiently in real-world applications where computational resources are limited. The Jetson NX Xavier provides a balance of performance and power efficiency, making it ideal for testing the model's real-time capabilities in practical scenarios. Experimental results show that our trained model can run at over 110 fps on GPU, and 30 fps on the edge platform of NVidia Jetson NX Xavier.

C. EVALUATION METRICS

The evaluation of 3D hand pose estimation methods typically relies on two key metrics: mean End-Point-Error (EPE) [61] and Area Under the Curve (AUC) on the Percentage of Correct Keypoints (PCK) [62]. Mean End-Point-Error (EPE) quantifies accuracy by calculating the average distance between the predicted 3D keypoint positions and their corresponding ground truth locations. On the other hand, the PCK metric assesses accuracy by determining the percentage of keypoints that are correctly predicted within a specified distance threshold d. Specifically, a keypoint is considered accurate if its distance from the ground truth is less than or equal to d. The PCK score is computed across various distance thresholds, and the AUC on PCK represents the integral of the PCK curve over these thresholds. A higher AUC indicates a more precise estimation across a broader range of distance thresholds. In our study, we define a keypoint prediction as correct if it lies within 50 mm of the ground-truth position for computing PCK. We calculate the AUC on PCK with 100 intervals and also report the absolute 3D positional errors of the predicted joints. Additionally, as recommended by [60], we include two post-processing error indicators by aligning the predicted joint positions with the ground truth. We use two alignment techniques: root-relative and procrustes. The root-relative method mitigates translation discrepancies by aligning the predicted root (wrist) position with the ground truth. The procrustes method, however, focuses on adjusting for translation, rotation, and scale, concentrating on the relative articulation of the hand.

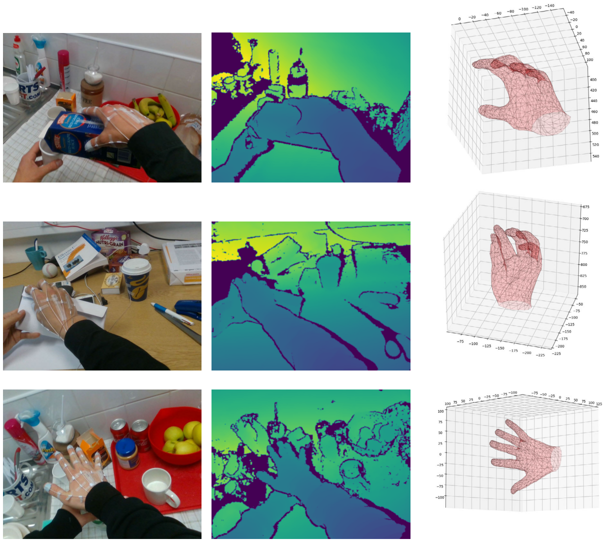

(a) RGB (b) Depth (c) Predicted

FIGURE 5. Qualitative results of the proposed method on FPHAB dataset.

D. RESULTS

Tables 3, 4, and 5 present the qualitative results, while Figure 3, 4, 5, and 6 illustrates the quantitative results. The quantitative evaluation of hand pose estimation on the HO-3D dataset demonstrates significant performance improvements by our method compared to existing state-of-the-art approaches. The evaluation metrics used include End-Point-Error (EPE) and Area Under the Curve (AUC) on the Percentage of Correct Keypoints (PCK), with detailed results presented in Table 3. We assess three types of EPE: absolute (abs), root-relative (root-rel), and procrustes (pro), and report the speed in frames per second (FPS). Our method, which utilizes both depth and RGB input modalities, achieves the lowest EPE across all three metrics: absolute (12.15 mm), root-relative (11.23 mm), and procrustes (6.78 mm). This indicates a substantial improvement in accuracy over other methods. For example, Moon et al. [14], using only depth input, report EPE values of 23.31 mm (abs), 17.43 mm (root-rel), and 11.90 mm (pro). Similarly, Xiong et al. [33] and Huang et al. [34] also demonstrate higher EPE values compared to our approach. In terms of AUC on PCK, our method again outperforms others with values of 0.813 (abs), 0.837 (root-rel), and 0.859 (pro). This indicates a higher overall accuracy across varying distance thresholds. For instance, Lin et al. [29], which use only RGB input, achieve AUC values of 0.692 (abs), 0.722 (root-rel), and 0.763 (pro), while Park et al. [30] show AUC values of 0.798 (abs), 0.823 (root-rel), and 0.845 (pro). Our method also excels in terms of processing speed, achieving a remarkable 112 FPS. This is significantly faster than the next best methods, such as Spurr et al. [63] at 36 FPS and Park et al. [30] at 32 FPS. This high speed is crucial for real-time applications, making our approach not only more accurate but also more practical for deployment in real-world scenarios. We also compare our method to Modified DenseFusion [41], Modified FFB6D [18], and Modified PointMBF [65]. These networks were originally designed for object pose estimation or registration using RGBD images but were adapted for hand pose estimation in our study. We retained the same RGBD fusion and feature extraction techniques as described in their original papers but modified the networks for hand pose estimation. The Modified DenseFusion achieves EPE values of 17.42 mm (abs), 16.37 mm (root-rel), and 14.79 mm (pro), with AUC values of 0.670 (abs), 0.705 (root-rel), and 0.719 (pro), and operates at 18 FPS. Modified FFB6D performs better with EPE values of 14.32 mm (abs), 13.21 mm (root-rel), and 11.30 mm (pro), AUC values of 0.732 (abs), 0.761 (root-rel), and 0.782 (pro), and runs at 23 FPS. The Modified PointMBF also shows strong results with EPE values of 13.86 mm (abs), 12.82 mm (root-rel), and 10.34 mm (pro), AUC values of 0.745 (abs), 0.769 (root-rel), and 0.788 (pro), and a speed of 20 FPS. Despite these strong performances, our proposed method surpasses all of these modified approaches both in terms of accuracy and speed. The superior results of our method underscore its effectiveness in leveraging both RGB and depth data for precise and efficient 3D hand pose estimation.

Similarly, the results, summarized in Table 4, demonstrate the superior performance of our method compared to existing state-of-the-art approaches on FPHAB dataset [59]. The proposed network outperforms existing methods, achieving the lowest End-Point-Error (EPE) values of 12.23 mm (abs), 11.38 mm (root-rel), and 6.92 mm (pro). This is significantly better compared to Moon et al. [14], who reported 20.58 mm, 16.90 mm, and 10.87 mm, respectively. Our method also excels in Area Under the Curve (AUC) on Percentage of Correct Keypoints (PCK), with scores of 0.802 (abs), 0.831 (root-rel), and 0.843 (pro), outperforming methods like Huang et al. [34] and Malik et al. [35].

Extending the evaluation to the DexYCB dataset, our method again demonstrates superior performance compared to other state-of-the-art methods, as shown in Table 5. Our method achieves the lowest End-Point-Error (EPE) values of 23.64 mm (abs), 13.88 mm (root-rel), and 5.74 mm (pro), outperforming methods such as Moon et al. [14] and Xiong et al. [33], which reported significantly higher EPE values. In terms of Area Under the Curve (AUC) on Percentage of Correct Keypoints (PCK), our method also excels, with scores of 0.761 (abs), 0.812 (root-rel), and 0.874 (pro). These results are superior to those of Malik et al. [35] and Lin et al. [29]. Furthermore, our method operates at 110 FPS, demonstrating a significant speed advantage over methods like Spurr et al. [63] at 35 FPS and Tang et al. [13] at 30 FPS. We also adapted and evaluated methods originally designed for object pose estimation, including Modified DenseFusion [41], Modified FFB6D [18], and Modified PointMBF [65]. Despite their strong performance, with EPE values of 24.47 mm and AUC

TABLE 3. Quantitative evaluation of hand pose on the HO-3D dataset. Absolute (abs), root-relative (root-rel), procrustes (pro). The join error EPE is in mm. Speed in FPS (frames per second).

	Input Modality		End-Point-Error (EPE)			Area Under the Curve (AUC)			Speed
	Depth	RGB	abs	root-rel	pro	abs	root-rel	pro	FPS
Moon et al. [14]	√		23.31	17.43	11.90	0.652	0.675	0.725	17
Xiong et al. [33]	√		21.22	17.70	11.53	0.631	0.783	0.726	12
Huang et al. [34]	√		19.65	17.42	11.04	0.648	0.682	0.729	28
Cheng et al. [36]	√		24.74	19.21	13.87	0.601	0.632	0.689	27
Malik [35]	√		20.32	18.04	10.87	0.643	0.678	0.702	24
Spurr et al. [63]		√	25.33	12.98	7.67	0.489	0.645	0.733	36
Tang et al. [13]		√	24.16	13.79	9.43	0.569	0.654	0.715	31
Chen et al. [28]		√	20.54	19.31	17.45	0.601	0.631	0.6579	25
Lin et al. [29]		√	14.49	13.06	12.11	0.692	0.722	0.763	28
Park et al. [30]		√	13.43	12.08	9.53	0.798	0.823	0.845	32
Kazakos et al. [64]	√	√	21.43	19.43	18.27	0.589	0.623	0.684	16
Modified DenseFusion [41]	√	√	17.42	16.37	14.79	0.670	0.705	0.719	18
Modified FFB6D [18]	√	√	14.32	13.21	11.30	0.732	0.761	0.782	23
Modified PointMBF [65]	√	√	13.86	12.82	10.34	0.745	0.769	0.788	20
Ours	√	√	**12.15**	**11.23**	**6.78**	**0.813**	**0.837**	**0.859**	**112**

TABLE 4. Quantitative evaluation of hand pose on the FPHAB dataset. Absolute (abs), root-relative (root-rel), procrustes (pro). The join error EPE is in mm. Speed in FPS (frames per second).

	Input Modality		End-Point-Error (EPE)			Area Under the Curve (AUC)			Speed
	Depth	RGB	abs	root-rel	pro	abs	root-rel	pro	FPS
Moon et al. [14]	√		20.58	16.90	10.87	0.681	0.701	0.765	18
Xiong et al. [33]	√		19.32	16.46	10.24	0.695	0.722	0.778	13
Huang et al. [34]	√		18.21	16.05	9.77	0.708	0.729	0.780	27
Cheng et al. [36]	√		20.87	17.80	11.61	0.645	0.687	0.749	27
Malik et al. [35]	√		18.14	16.03	9.70	0.714	0.738	0.789	24
Spurr et al. [63]		√	28.11	12.36	7.14	0.513	0.745	0.818	35
Tang et al. [13]		√	30.25	13.57	7.33	0.490	0.733	0.805	31
Chen et al. [28]		√	19.62	18.81	16.82	0.628	0.658	0.688	24
Lin et al. [29]		√	20.19	19.03	17.66	0.610	0.642	0.664	29
Park et al. [30]		√	18.53	17.68	15.55	0.675	0.710	0.715	31
Kazakos et al. [64]	√	√	20.61	18.24	17.75	0.602	0.646	0.693	17
Modified DenseFusion [41]	√	√	16.36	15.01	14.23	0.696	0.734	0.748	18
Modified FFB6D [18]	√	√	14.45	13.37	11.42	0.722	0.754	0.767	24
Modified PointMBF [65]	√	√	14.05	13.24	10.65	0.731	0.755	0.776	21
Ours	√	√	**12.23**	**11.38**	**6.92**	**0.802**	**0.831**	**0.843**	**110**

scores up to 0.869, our method consistently achieved better results. This reinforces the effectiveness of our approach in 3D hand pose estimation tasks across diverse datasets.

E. ABLATION STUDY

The ablation study evaluates the contribution of various components in our proposed architecture. The experiments assess different configurations by altering or removing specific modules and comparing their performance on the HO-3D, FPHAB, and DexYCB datasets. Table 6 presents detailed results of these ablation studies.

1) OURS (CONCAT FUSION)

Replacing our efficient RGBD fusion module with a simple concatenation of RGB and depth images while keeping other components unchanged results in a slight drop in performance. The AUC scores are 0.772 (abs), 0.795 (root-rel), and 0.826 (pro) on the HO-3D dataset, which are lower compared to the full model. This shows the importance of our efficient RGBD fusion module in enhancing the performance of the network. However, this configuration achieves the highest speed at 116 FPS, indicating that while simpler fusion methods might be faster, they compromise on accuracy.

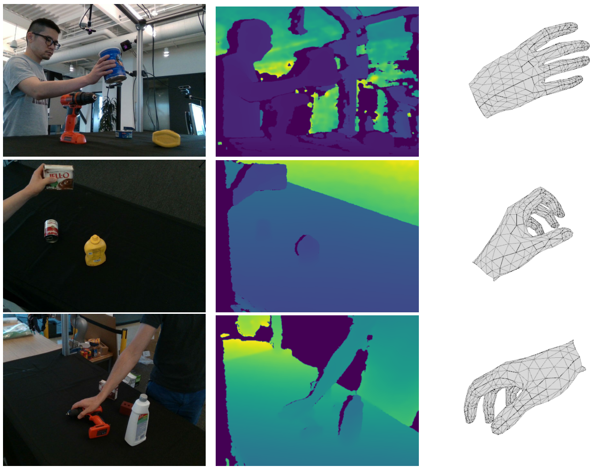

(a) RGB (b) Depth (c) Predicted

FIGURE 6. Qualitative results of the proposed method on DexYCB Dataset.

2) OURS (DENSE FUSION) [41]

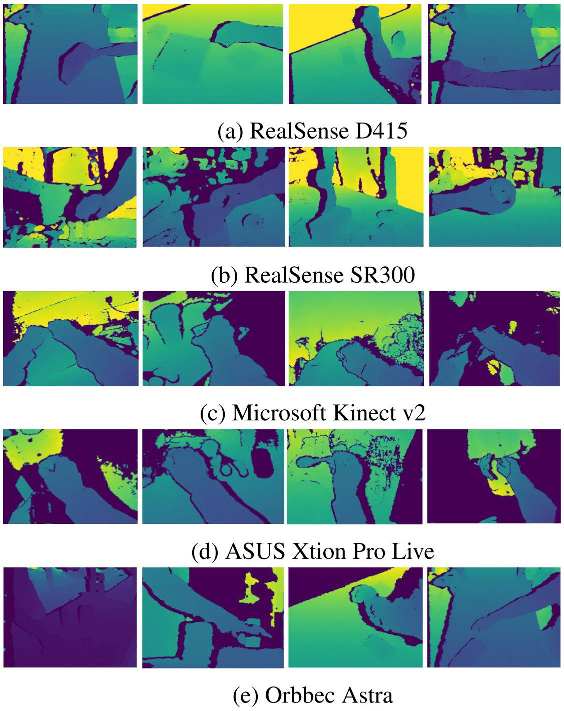
Using Dense Fusion instead of our efficient RGBD fusion module and MobileNetv2 increases AUC scores to 0.811 (abs), 0.835 (root-rel), and 0.853 (pro) on the HO-3D dataset.

TABLE 5. Quantitative evaluation of hand pose on the DexYCB dataset. Absolute (abs), root-relative (root-rel), procrustes (pro). The join error EPE is in mm. Speed in FPS (frames per second).

	Input Modality		End-Point-Error (EPE)			Area Under the Curve (AUC)			Speed
	Depth	RGB	abs	root-rel	pro	abs	root-rel	pro	FPS
Moon et al. [14]	√		28.78	24.74	13.79	0.574	0.605	0.706	18
Xiong et al. [33]	√		27.53	23.93	12.07	0.612	0.588	0.760	13
Huang et al. [34]	√		27.15	24.33	12.25	0.623	0.653	0.741	27
Cheng et al. [36]	√		28.25	24.31	13.00	0.597	0.621	0.720	27
Malik et al. [35]	√		26.31	22.62	11.76	0.668	0.626	0.784	24
Spurr et al. [63]		√	52.26	17.34	6.83	0.328	0.698	0.864	35
Tang et al. [13]		√	52.41	17.63	7.22	0.307	0.672	0.823	30
Chen et al. [28]		√	27.68	14.21	12.30	0.672	0.680	0.785	26
Lin et al. [29]		√	27.37	19.91	11.36	0.692	0.705	0.803	28
Park et al. [30]		√	26.05	18.55	8.24	0.714	0.742	0.851	31
Kazakos et al. [64]	√	√	59.32	27.85	17.43	0.284	0.399	0.512	16
Modified DenseFusion	√	√	25.01	20.23	9.14	0.624	0.642	0.783	18
Modified FFB6D [18]	√	√	24.47	14.56	6.16	0.742	0.781	0.850	24
Modified PointMBF [65]	√	√	24.62	14.66	6.22	0.756	0.810	0.869	21
Ours	√	√	**23.64**	**13.88**	**5.74**	**0.761**	**0.812**	**0.874**	**110**

TABLE 6. Ablation study. We run trained models on a 32GB Tesla V100 GPU (V100) and an edge platform of NVidia Jetson NX Xavier (Xavier).

	HO-3D Dataset			FPHAB Dataset			DexYCB Dataset			Speed (fps)	
	abs	root-rel	pro	abs	root-rel	pro	abs	root-rel	pro	V100	Xavier
Ours (Concat Fusion)	0.772	0.795	0.826	0.764	0.794	0.817	0.722	0.777	0.832	116	32
Ours (Dense Fusion) [41]	0.811	0.835	0.853	0.801	0.830	0.842	0.760	0.810	0.871	23	6
Ours (FFB6D Fusion [18])	0.815	0.838	0.859	0.804	0.835	0.845	0.763	0.813	0.877	20	5
Ours (PointMBF Fusion [65])	0.816	0.839	0.861	0.803	0.833	0.844	0.765	0.816	0.877	20	5
Ours (ResNet Backbone [66])	0.817	0.839	0.865	0.806	0.837	0.848	0.764	0.815	0.877	18	4
Ours (Stacked Hourglass Networks [22])	0.793	0.811	0.832	0.778	0.803	0.812	0.732	0.798	0.853	35	7
Ours (-Intermediate Feature Fusion)	0.803	0.822	0.841	0.795	0.818	0.832	0.752	0.801	0.864	113	31
Ours (full)	0.813	0.837	0.859	0.802	0.831	0.843	0.761	0.812	0.874	111	30


(a) RealSense D415

(b) RealSense SR300

(c) Microsoft Kinect v2

(d) ASUS Xtion Pro Live

(e) Orbbec Astra

FIGURE 7. RGBD images captured by different cameras.

Despite this, the speed drops significantly to 23 FPS. This indicates that while Dense Fusion can improve accuracy, it is much less efficient, highlighting the advantage of our approach in balancing performance and speed.

3) OURS (FFB6D FUSION) [18]

Replacing our modules with FFB6D Fusion results in AUC scores of 0.815 (abs), 0.838 (root-rel), and 0.859 (pro) on the HO-3D dataset. This configuration is similar in accuracy to Dense Fusion but slightly better. The speed, however, remains low at 20 FPS, demonstrating a trade-off between accuracy and computational efficiency.

4) OURS (POINTMBF FUSION) [65]

Similarly, using PointMBF Fusion shows AUC scores of 0.816 (abs), 0.839 (root-rel), and 0.861 (pro) on the HO-3D dataset. The performance is comparable to FFB6D Fusion, with slightly better AUC scores but at the same speed of 20 FPS. This reinforces the need for an efficient fusion module to maintain a balance between accuracy and speed.

5) OURS (RESNET BACKBONE [66])

Replacing MobileNetv2 with ResNet-50 in our architecture yields slightly higher AUC scores of 0.817 (abs), 0.839 (root-rel), and 0.865 (pro) on the HO-3D dataset. However, the speed drops to 18 FPS, the lowest among all configurations. This highlights that while ResNet-50 can slightly improve accuracy, it significantly affects the real-time performance of the network.

6) OURS (STACKED HOURGLASS NETWORKS) [22]

Using the Stacked Hourglass Network instead of our efficient fusion module results in AUC scores of 0.793 (abs), 0.811 (root-rel), and 0.832 (pro) on the HO-3D dataset. This configuration shows a decrease in accuracy and operates at 35 FPS, which is faster than Dense Fusion and FFB6D but still not optimal.

7) OURS (-INTERMEDIATE FEATURE FUSION)

Removing the intermediate feature fusion between MobileNetv2 and the Hourglass network slightly decreases the AUC scores to 0.803 (abs), 0.822 (root-rel), and 0.841 (pro) on the HO-3D dataset. The speed, however, remains high at 113 FPS, indicating that intermediate feature fusion contributes positively to accuracy without significantly affecting speed.

8) OURS (FULL)

The full configuration of our method achieves AUC scores of 0.813 (abs), 0.837 (root-rel), and 0.859 (pro) on the HO-3D dataset, with a speed of 111 FPS. This demonstrates that our efficient RGBD fusion module, in combination with MobileNetv2 and intermediate feature fusion, provides the best balance of accuracy and speed across the datasets.

9) PERFORMANCE ON VARIOUS RGBD CAMERAS

Figure 7 shows depth images captured by different RGBD cameras. Table 7 presents an ablation study evaluating the performance of our method using depth images from various sensors: Orbbec Astra, ASUS Xtion Pro Live, Microsoft Kinect v2, Intel RealSense SR300, and Intel RealSense D435. The results demonstrate that the Intel RealSense D435 sensor achieves the best performance with values of 0.821 (abs), 0.844 (root-rel), and 0.862 (pro). In comparison, the Intel RealSense SR300 also performs well, with values of 0.810 (abs), 0.835 (root-rel), and 0.850 (pro). The Microsoft Kinect v2 and ASUS Xtion Pro Live show similar, moderate performance, while the Orbbec Astra sensor records the lowest performance metrics among the tested sensors. These results indicate that the choice of depth sensor significantly impacts the accuracy of our proposed method, with the Intel RealSense D435 sensor providing the highest accuracy. The Intel RealSense D435 offers the highest resolution at 1280 × 720 and the widest range from 0.2 to 10 meters, making it suitable for capturing detailed hand movements at various distances, with a frame rate that can reach up to 90 fps at lower resolutions. The Microsoft Kinect v2, known for its reliable body tracking and depth imaging, provides high accuracy within a range of 0.5 to 4.5 meters and produces low-noise depth images. The Orbbec Astra and ASUS Xtion Pro Live, both offering a resolution of 640 × 480, are suitable for general applications with moderate noise levels, though the Xtion Pro Live has a shorter range of 0.8 to 3.5 meters. The Intel RealSense SR300 is optimized for close-range applications, with high accuracy within a range of 0.2 to 1.5 meters and low noise levels, making it particularly useful for detailed hand pose estimation at close distances. Therefore, the choice of camera for hand pose estimation depends on specific requirements, with the D435 being the most versatile, and the SR300 being ideal for close-range precision tasks.

10) PERFORMANCE ON LIMITED COMPUTATIONAL RESOURCES

To evaluate the performance of our model on limited computational resources, we conducted tests on an edge platform, the NVIDIA Jetson NX Xavier. Our full model achieves an impressive speed of 30 FPS on the Xavier, demonstrating its suitability for real-time applications in resource-constrained environments. Even with reduced computational power, the model maintains a balance of accuracy and speed, reinforcing the effectiveness of our efficient fusion and feature extraction techniques. This makes our approach practical for deployment in scenarios requiring lightweight and efficient hand pose estimation.

TABLE 7. Performance comparison of our method using depth images from different sensors.

	abs	root-rel	pro
Orbbec Astra	0.788	0.801	0.822
ASUS Xtion Pro Live	0.795	0.812	0.843
Microsoft Kinect v2	0.793	0.815	0.840
Intel Realsense SR300	0.810	0.835	0.850
Intel Realsense D435	**0.821**	**0.844**	**0.862**

V. CONCLUSION

In this paper, we have presented a novel approach for hand pose estimation using an efficient multimodal fusion method that leverages both RGB and depth data. Central to our approach is the innovative RGBD fusion module, which combines appearance and geometric data early in the processing pipeline. This module significantly reduces computational complexity while maintaining real-time performance on resource-constrained devices. Our method integrates MobileNetv2 with an hourglass network and utilizes intermediate feature fusion to enhance accuracy. Extensive evaluations on three publicly available datasets demonstrate that our approach achieves state-of-the-art performance in terms of End-Point-Error (EPE) and Area Under the Curve (AUC) on the Percentage of Correct Keypoints (PCK). Ablation studies further confirmed that each component of our architecture contributes to the overall performance. Compared to other configurations, our full method consistently outperformed alternative fusion strategies and backbone networks, reinforcing the effectiveness of our design choices. Future work will focus on integrating the current hand pose estimation system into robotic systems to enhance their interaction capabilities. One promising direction is to incorporate our hand pose estimation method into robotic manipulators and humanoid robots, enabling more precise and intuitive control for tasks such as object manipulation, assembly, and human-robot collaboration.

REFERENCES

[1] M.-Y. Wu, P.-W. Ting, Y.-H. Tang, E.-T. Chou, and L.-C. Fu, "Hand pose estimation in object-interaction based on deep learning for virtual reality applications," *J. Vis. Commun. Image Represent.*, vol. 70, Jul. 2020, Art. no. 102802.

[2] Z. Ren, J. Meng, and J. Yuan, "Depth camera based hand gesture recognition and its applications in human-computer-interaction," in *Proc. 8th Int. Conf. Inf., Commun. Signal Process.*, Dec. 2011, pp. 1–5.

[3] 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," *J. Intell. Robotic Syst.*, vol. 109, no. 3, p. 71, Nov. 2023.

[4] J. Qi, L. Ma, Z. Cui, and Y. Yu, "Computer vision-based hand gesture recognition for human–robot interaction: A review," *Complex Intell. Syst.*, vol. 10, no. 1, pp. 1581–1606, Feb. 2024.

[5] X. Yin and M. Xie, "Finger identification and hand posture recognition for human–robot interaction," *Image Vis. Comput.*, vol. 25, no. 8, pp. 1291–1300, Aug. 2007.

[6] D.-C. Hoang, A.-N. Nguyen, V.-D. Vu, T.-U. Nguyen, D.-Q. Vu, P.-Q. Ngo, N.-A. Hoang, K.-T. Phan, D.-T. Tran, V.-T. Nguyen, Q.-T. Duong, N.-T. Ho, C.-T. Tran, V.-H. Duong, and A.-T. Mai, "Graspability-aware object pose estimation in cluttered scenes," *IEEE Robot. Autom. Lett.*, vol. 9, no. 4, pp. 3124–3130, Apr. 2024.

[7] T. Grzejszczak, A. Legowski, and M. Niezabitowski, "Application of hand detection algorithm in robot control," in *Proc. 17th Int. Carpathian Control Conf. (ICCC)*, May 2016, pp. 222–225.

[8] A. Albini, S. Denei, and G. Cannata, "Human hand recognition from robotic skin measurements in human–robot physical interactions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 4348–4353.

[9] D.-C. Hoang, J. A. Stork, and T. Stoyanov, "Context-aware grasp generation in cluttered scenes," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 1492–1498.

[10] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: A review," *Comput. Vis. Image Understand.*, vol. 108, nos. 1–2, pp. 52–73, Oct. 2007.

[11] R. Li, Z. Liu, and J. Tan, "A survey on 3D hand pose estimation: Cameras, methods, and datasets," *Pattern Recognit.*, vol. 93, pp. 251–272, Sep. 2019.

[12] L. Bretzner, I. Laptev, and T. Lindeberg, "Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering," in *Proc. 5th IEEE Int. Conf. Autom. Face Gesture Recognit.*, May 2002, pp. 423–428.

[13] X. Tang, T. Wang, and C.-W. Fu, "Towards accurate alignment in real-time 3D hand-mesh reconstruction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 11678–11687.

[14] J. Y. Chang, G. Moon, and K. M. Lee, "V2V-PoseNet: Voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5079–5088.

[15] D.-C. Hoang, J. A. Stork, and T. Stoyanov, "Voting and attention-based pose relation learning for object pose estimation from 3D point clouds," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 8980–8987, Oct. 2022.

[16] Z. Wang, X. Huo, Z. Chen, J. Zhang, L. Sheng, and D. Xu, "Improving RGB-D point cloud registration by learning multi-scale local linear transformation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 175–191.

[17] D.-C. Hoang, A.-N. Nguyen, C.-M. Nguyen, A.-B. Phi, Q.-T. Duong, K.-D. Tran, V.-A. Trinh, V.-D. Tran, H.-N. Pham, P.-Q. Ngo, D.-Q. Vu, T.-U. Nguyen, V.-D. Vu, D.-T. Tran, and V.-T. Nguyen, "Collision-free grasp detection from color and depth images," *IEEE Trans. Artif. Intell.*, early access, Jul. 1, 2024, doi: 10.1109/TAI.2024.3420848.

[18] Y. He, H. Huang, H. Fan, Q. Chen, and J. Sun, "FFB6D: A full flow bidirectional fusion network for 6D pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3002–3012.

[19] M. Hussien, K. K. Nguyen, and M. Cheriet, "A learning framework for bandwidth-efficient distributed inference in wireless IoT," *IEEE Sensors J.*, vol. 23, no. 15, pp. 17656–17666, Aug. 2023.

[20] M. Kayaalp, Y. Inan, E. Telatar, and A. H. Sayed, "On the arithmetic and geometric fusion of beliefs for distributed inference," *IEEE Trans. Autom. Control*, vol. 69, no. 4, pp. 2265–2280, Apr. 2024.

[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[22] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands. Cham, Switzerland: Springer, Oct. 2016, pp. 483–499.

[23] Y. Liu, J. Jiang, and J. Sun, "Hand pose estimation from RGB images based on deep learning: A survey," in *Proc. IEEE 7th Int. Conf. Virtual Reality (ICVR)*, May 2021, pp. 82–89.

[24] Y. Wu and T. S. Huang, "View-independent recognition of hand postures," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2000, pp. 88–94.

[25] V. Athitsos and S. Sclaroff, "Estimating 3D hand pose from a cluttered image," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2003, p. 432.

[26] C. Zimmermann and T. Brox, "Learning to estimate 3D hand pose from single RGB images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4913–4921.

[27] U. Iqbal, P. Molchanov, T. B. J. Gall, and J. Kautz, "Hand pose estimation via latent 2.5D heatmap regression," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 118–134.

[28] X. Chen, Y. Liu, C. Ma, J. Chang, H. Wang, T. Chen, X. Guo, P. Wan, and W. Zheng, "Camera-space hand mesh recovery via semantic aggregation and adaptive 2D-1D registration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13269–13278.

[29] K. Lin, L. Wang, and Z. Liu, "End-to-end human pose and mesh reconstruction with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1954–1963.

[30] J. Park, Y. Oh, G. Moon, H. Choi, and K. M. Lee, "HandOccNet: Occlusion-robust 3D hand mesh estimation network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1486–1495.

[31] L. Ge, Z. Ren, and J. Yuan, "Point-to-point regression pointnet for 3D hand pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 475–491.

[32] Q. Ye, S. Yuan, and T.-K. Kim, "Spatial attention deep net with partial PSO for hierarchical hybrid hand pose estimation," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands. Cham, Switzerland: Springer, Oct. 2016, pp. 346–361.

[33] F. Xiong, B. Zhang, Y. Xiao, Z. Cao, T. Yu, J. T. Zhou, and J. Yuan, "A2J: Anchor-to-joint regression network for 3D articulated pose estimation from a single depth image," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 793–802.

[34] W. Huang, P. Ren, J. Wang, Q. Qi, and H. Sun, "AWR: Adaptive weighting regression for 3D hand pose estimation," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2020, pp. 11061–11068.

[35] J. Malik, S. Shimada, A. Elhayek, S. A. Ali, C. Theobalt, V. Golyanik, and D. Stricker, "HandVoxNet++: 3D hand shape and pose estimation using voxel-based neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 8962–8974, Dec. 2022.

[36] W. Cheng, J. H. Park, and J. H. Ko, "HandFoldingNet: A 3D hand pose estimation network using multiscale-feature guided folding of a 2D hand skeleton," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 11240–11249.

[37] A. Lopes, R. Souza, and H. Pedrini, "A survey on RGB-D datasets," *Comput. Vis. Image Understand.*, vol. 222, Sep. 2022, Art. no. 103489.

[38] C. Wang, C. Ma, M. Zhu, and X. Yang, "PointAugmenting: Cross-modal augmentation for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11789–11798.

[39] J. Gupta, "Learning rich features from RGB-D images for object detection and segmentation," in *Proc. 13th Eur. Conf. Comput. Vis. (ECCV)*, Zurich, Switzerland. Cham, Switzerland: Springer, Sep. 2014, pp. 345–360.

[40] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "PVN3D: A deep point-wise 3D keypoints voting network for 6DoF pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11629–11638.

[41] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "DenseFusion: 6D object pose estimation by iterative dense fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3338–3347.

[42] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3D object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 641–656.

[43] T. Huang, Z. Liu, X. Chen, and X. Bai, "EPNet: Enhancing point features with image semantics for 3D object detection," in *Proc. 16th Eur. Conf. Comput. Vis. (EECV)*, Glasgow, U.K. Cham, Switzerland: Springer, Aug. 2020, pp. 35–52.

[44] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun, "3D graph neural networks for RGBD semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5209–5218.

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

[46] H. Li, Y. Chen, Q. Zhang, and D. Zhao, "BiFNet: Bidirectional fusion network for road segmentation," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 8617–8628, Sep. 2022.

[47] W. Hu, H. Zhao, L. Jiang, J. Jia, and T.-T. Wong, "Bidirectional projection network for cross dimension scene understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14368–14377.

[48] X. Chen, K.-Y. Lin, J. Wang, W. Wu, C. Qian, H. Li, and G. Zeng, "Bi-directional cross-modality feature propagation with separation-and-aggregation gate for RGB-D semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 561–577.

[49] K. Desingh, K. M. Krishna, D. Rajan, and C. Jawahar, "Depth really matters: Improving visual salient region detection with depth," in *Proc. Procedings Brit. Mach. Vis. Conf.*, 2013, pp. 98.1–98.11.

[50] L.-C. Chen, D.-C. Hoang, H.-I. Lin, and T.-H. Nguyen, "Innovative methodology for multi-view point cloud registration in robotic 3D object scanning and reconstruction," *Appl. Sci.*, vol. 6, no. 5, p. 132, May 2016.

[51] D. Feng, N. Barnes, S. You, and C. McCarthy, "Local background enclosure for RGB-D salient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2343–2350.

[52] D.-C. Hoang, L.-C. Chen, and T.-H. Nguyen, "Sub-OBB based object recognition and localization algorithm using range images," *Meas. Sci. Technol.*, vol. 28, no. 2, Feb. 2017, Art. no. 025401.

[53] C. Wang, C. Wang, W. Li, and H. Wang, "A brief survey on RGB-D semantic segmentation using deep learning," *Displays*, vol. 70, Dec. 2021, Art. no. 102080.

[54] H. Zhang, V. S. Sheng, X. Xi, Z. Cui, and H. Rong, "Overview of RGBD semantic segmentation based on deep learning," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 10, pp. 13627–13645, Oct. 2023.

[55] D.-C. Hoang, P. Xuan Tan, A.-N. Nguyen, D.-Q. Vu, V.-D. Vu, T.-U. Nguyen, Q.-T. Duong, V.-T. Nguyen, N.-A. Hoang, K.-T. Phan, D.-T. Tran, N.-T. Ho, C.-T. Tran, V.-H. Duong, and P.-Q. Ngo, "Object pose estimation using color images and predicted depth maps," *IEEE Access*, vol. 12, pp. 65444–65461, 2024.

[56] P. Xuan Tan, D.-C. Hoang, A.-N. Nguyen, V.-T. Nguyen, V.-D. Vu, T.-U. Nguyen, N.-A. Hoang, K.-T. Phan, D.-T. Tran, D.-Q. Vu, P.-Q. Ngo, Q.-T. Duong, N.-T. Ho, C.-T. Tran, V.-H. Duong, and A.-T. Mai, "Attention-based grasp detection with monocular depth estimation," *IEEE Access*, vol. 12, pp. 65041–65057, 2024.

[57] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *ACM Trans. Graph.*, vol. 36, no. 6, pp. 1–17, 2017.

[58] S. Hampali, M. Rad, M. Oberweger, and V. Lepetit, "HOnnotate: A method for 3D annotation of hand and object poses," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3193–3203.

[59] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, "First-person hand action benchmark with RGB-D videos and 3D hand pose annotations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 409–419.

[60] Y.-W. Chao, W. Yang, Y. Xiang, P. Molchanov, A. Handa, J. Tremblay, Y. S. Narang, K. Van Wyk, U. Iqbal, S. Birchfield, J. Kautz, and D. Fox, "DexYCB: A benchmark for capturing hand grasping of objects," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 9044–9053.

[61] T. Chatzis, A. Stergioulas, D. Konstantinidis, K. Dimitropoulos, and P. Daras, "A comprehensive study on deep learning-based 3D hand pose estimation methods," *Appl. Sci.*, vol. 10, no. 19, p. 6850, Sep. 2020.

[62] K. Ahuja, P. Streli, and C. Holz, "TouchPose: Hand pose prediction, depth estimation, and touch classification from capacitive images," in *Proc. 34th Annu. ACM Symp. User Interface Softw. Technol.*, Oct. 2021, pp. 997–1009.

[63] A. Spurr, U. Iqbal, P. Molchanov, O. Hilliges, and J. Kautz, "Weakly supervised 3D hand pose estimation via biomechanical constraints," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 211–228.

[64] E. Kazakos, C. Nikou, and I. A. Kakadiaris, "On the fusion of RGB and depth information for hand pose estimation," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 868–872.

[65] M. Yuan, K. Fu, Z. Li, Y. Meng, and M. Wang, "PointMBF: A multi-scale bidirectional fusion network for unsupervised RGB-D point cloud registration," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 17694–17705.

[66] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

DINH-CUONG HOANG received the Ph.D. degree in computer science from Örebro University, Sweden, in 2021. He is currently a Lecturer with Greenwich Vietnam, FPT University. His research interests include computer vision, robotics, and machine learning. He is particularly interested in topics involving autonomy for robots, with a focus on perception algorithms.

PHAN XUAN TAN (Member, IEEE) received the B.E. degree in electrical-electronic engineering from the Military Technical Academy, Vietnam, the M.E. degree in computer and communication engineering from Hanoi University of Science and Technology, Vietnam, and the Ph.D. degree in functional control systems from Shibaura Institute of Technology, Japan. He is currently an Associate Professor with Shibaura Institute of Technology. His current research interests include deep learning for visual computing, image and video processing, computational light field, 3D view synthesis, multimedia quality of experience, and multimedia networking.

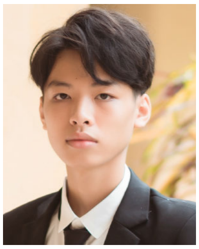
DUC-LONG PHAM is currently pursuing the B.S. degree in artificial intelligence with FPT University, Hanoi, Vietnam, with a primary focus on research in the field of computer vision.

HAI-NAM PHAM is currently pursuing the B.S. degree in computing with Greenwich Vietnam, FPT University, Hanoi, Vietnam, with a primary focus on research in the field of computer vision and robotics.

SON-ANH BUI is currently pursuing the B.S. degree in computing with Greenwich Vietnam, FPT University, Hanoi, Vietnam, with a primary focus on research in the field of computer vision and robotics.

CHI-MINH NGUYEN is currently pursuing the B.S. degree in computing with Greenwich Vietnam, FPT University, Hanoi, Vietnam, with a primary focus on research in the field of computer vision and robotics.

AN-BINH PHI is currently pursuing the B.S. degree in computing with Greenwich Vietnam, FPT University, Hanoi, Vietnam, with a primary focus on research in the field of computer vision and robotics.

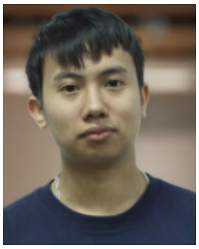


KHANH-DUONG TRAN is currently pursuing the B.S. degree in computing with Greenwich Vietnam, FPT University, Hanoi, Vietnam, with a primary focus on research in the field of computer vision and robotics.

VIET-ANH TRINH is currently pursuing the B.S. degree in computing with Greenwich Vietnam, FPT University, Hanoi, Vietnam, with a primary focus on research in the field of computer vision and the Internet of Things (IoT).

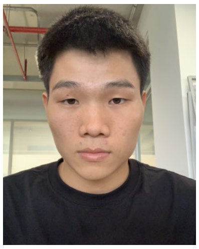
VAN-DUC TRAN is currently pursuing the B.S. degree in computing with Greenwich Vietnam, FPT University, Hanoi, Vietnam, with a primary focus on research in the field of computer vision and robotics.

DUC-THANH TRAN is currently pursuing the B.S. degree in artificial intelligence with FPT University, Hanoi, Vietnam, with a primary focus on research in the field of computer vision.



VAN-HIEP DUONG is currently pursuing the B.S. degree in artificial intelligence with FPT University, Hanoi, Vietnam, with a primary focus on research in the field of computer vision and robotics.

KHANH-TOAN PHAN is currently pursuing the B.S. degree in artificial intelligence with FPT University, Hanoi, Vietnam, with a primary focus on research in the field of computer vision and robotics.



VAN-THIEP NGUYEN is currently pursuing the B.S. degree in artificial intelligence with FPT University, Hanoi, Vietnam, with a primary focus on research in the field of computer vision.

VAN-DUC VU is currently pursuing the B.S. degree in artificial intelligence with FPT University, Hanoi, Vietnam, with a primary focus on research in the field of computer vision.

THU-UYEN NGUYEN is currently pursuing the B.S. degree in artificial intelligence with FPT University, Hanoi, Vietnam, with a primary focus on research in the field of computer vision.