JENGA: Object selection and pose estimation for robotic grasping from a stack

Sai Srinivas Jeevanandam[1], Sandeep Inuganti[1,2], Shreedhar Govil[1], Didier Stricker[1,2], Jason Rambach[1]
[1]*German Research Center for Artificial Intelligence (DFKI)*, [2]*RPTU Kaiserslautern*

***Abstract*— Vision-based robotic object grasping is typically investigated in the context of isolated objects or unstructured object sets in bin picking scenarios. However, there are several settings, such as construction or warehouse automation, where a robot needs to interact with a structured object formation such as a stack. In this context, we define the problem of selecting suitable objects for grasping along with estimating an accurate 6DoF pose of these objects. To address this problem, we propose a camera-IMU based approach that prioritizes unobstructed objects on the higher layers of stacks and introduce a dataset for benchmarking and evaluation, along with a suitable evaluation metric that combines object selection with pose accuracy. Experimental results show that although our method can perform quite well, this is a challenging problem if a completely error-free solution is needed. Finally, we show results from the deployment of our method for a brick-picking application in a construction scenario.**

***Index Terms*— Robotic Grasping, Object Pose, Object Picking, Pose Estimation, Picking from a Stack**

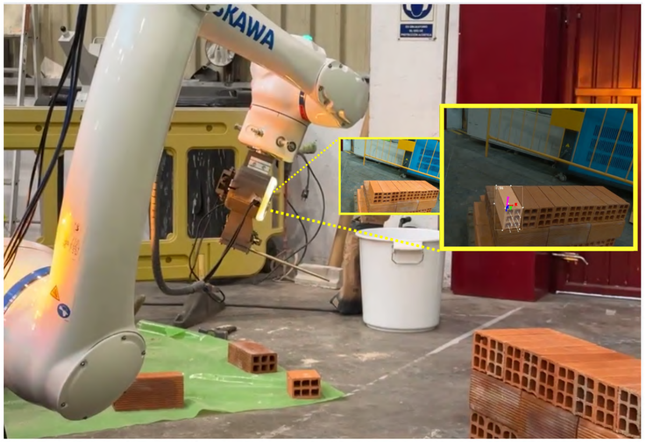

Fig. 1: When performing robotic grasping from an object stack, detection and 6DoF object pose are not sufficient. We present an approach that is able to select suitable objects for grasping.

I. Introduction

In modern robotic applications, such as warehouse automation and domestic assistance, robots frequently encounter scenes where objects are *stacked* rather than merely scattered or isolated. Stacked arrangements introduce substantial *occlusion* and *partial observability*, often leading to ambiguous or erroneous pose estimates. Although a state-of-the-art 6DoF (6 Degrees of Freedom) pose estimation algorithm [1] can generate poses for multiple objects, many of these might be *unreliable* or *physically obstructed*, making them unsuitable for grasping. Specifically, objects may be nestled beneath or alongside other objects in the stack, creating challenges that most pose estimation methods do not adequately address.

On the other hand, research on robotic grasping has put significant effort on the bin-picking scenario where objects are placed in an unordered manner inside a container and are grasped from above [2], [3]. In this case object grasping is less challenging and has less potential impact on a failed grasp than picking from a stack. Since object stacks are common structures encountered in daily life, we advocate the need for the development of robotic grasping policies for such structures as well.

To achieve this, we initially created a synthetic dataset of stacked objects (bricks in our case) representing a common use-case of objects stacked in pallets in construction. Further, we formally define the problem of object picking from stacks and design appropriate metrics for its evaluation that combine object detection and pose estimation with object suitability for grasping. We investigate and evaluate baseline approaches for the object selection, such as relying on the confidence of the object detector to select objects of lower occlusion or on inertial sensor measurements to locate the objects on the top of the stack. Finally, we propose a novel method to select suitable grasping candidates that combines pose information, object mask visibility, and inertial sensor cues. In summary, this work introduces the following contributions:

- We formally define the problem of object grasping from structured stacks as opposed to the common bin-picking scenario.
- We introduce a novel, procedurally generated dataset for the stacked object pose estimation and grasping task along with a suitable evaluation protocol to quantitatively benchmark solutions.
- We propose a novel visual-inertial method for stacked object selection and pose estimation and an evaluation against appropriate baselines.

II. Related Work

A. Object Pose Estimation

Object pose estimation plays a crucial role in robotic grasping, enabling robots to localize and manipulate objects in various environments accurately. Since robotic grasping

Contact: {sai_srinivas.jeevanandam, jason.rambach}@dfki.de
Project Website: https://sjeevanandam.github.io/jenga_iros_2025

is a task that requires highly accurate poses, we focus on instance-level pose estimation methods where a 3D reference model of the exact object whose pose we wish to estimate, is available. Traditional object pose estimation methods such as Iterative Closest Point (ICP) [4] have been widely adopted for aligning known 3D models to observed point clouds or image features. However, these methods often struggle in cluttered or occluded environments. Recently, deep learning approaches have significantly improved object pose estimation by leveraging large datasets. While some early methods directly regressed object poses or 3D bounding boxes [5], [6], methods such as PoseCNN [7], and HiPose [1] utilize deep feature representations to improve robustness to occlusions and lighting variations. ZebraPose [8] uses a binary surface encoding representation to establish image-to-model (2D-3D) correspondences leading to a high accuracy method while being computationally efficient and suitable for real-time applications. A complete overview of the state-of-the-art in the topic can be found in the report of the BOP challenge [9].

B. Robotic Grasping

Robotic grasping has been extensively studied in diverse settings—such as picking objects from tables, bins, and conveyor belts. In these scenarios, robots employ a combination of grasping and localization techniques to manipulate or retrieve objects.

Robot grasping methods can be broadly classified into two categories: geometric-based and data-driven [10], [11]. Geometric (or sometimes called analytical) methods typically first analyze the shape of the target objects to plan an appropriate grasping pose. Meanwhile, data-driven (or sometimes called empirical) methods are based on machine learning and have become increasingly popular. They have made substantial improvement in recent years owing to the increased data availability, better computational resources, and algorithmic improvements. We refer the reader to standard surveys on the subject for a more complete treatment [10], [12], [13].

Despite substantial progress in various grasping methods, the challenge of handling objects in stacks remains underexplored. Stacked objects introduce unique difficulties, such as ensuring stability and executing precise grasps without disturbing neighboring items. In this paper, we tackle these challenges by proposing a novel camera-IMU based approach and presenting an evaluation benchmark dataset.

III. Problem Formulation

A. Notation

We define a *6DoF pose* as a transformation $\mathbf{T}_{co} = [\mathbf{R}_{co} \in \mathbb{R}^{3\times 3}, \mathbf{o}_c \in \mathbb{R}^3]$ between the camera coordinate system C and an object coordinate system O.

An inertial sensor (IMU) measures linear acceleration $\alpha \in \mathbb{R}^3$ and angular velocity $\omega \in \mathbb{R}^3$ with respect to the sensor coordinate system S. When no linear acceleration due to sensor motion is present, the measured acceleration corresponds to the gravity vector $\mathbf{g}_{global} = [0, 0, -9.81]$ projected on the sensor coordinate system as $\alpha_\mathbf{s} = \mathbf{R}_{sg}\, \mathbf{g}_{global}$, where $\mathbf{R}_{sg} \in \mathbb{R}^{3\times 3}$ is the rotation matrix from the global coordinate frame G to the sensor coordinate frame S.

The rigid transformation between camera and inertial sensor coordinate systems, is given by $\mathbf{T}_{sc} = [\mathbf{R}_{sc}, \mathbf{c}_s]$.

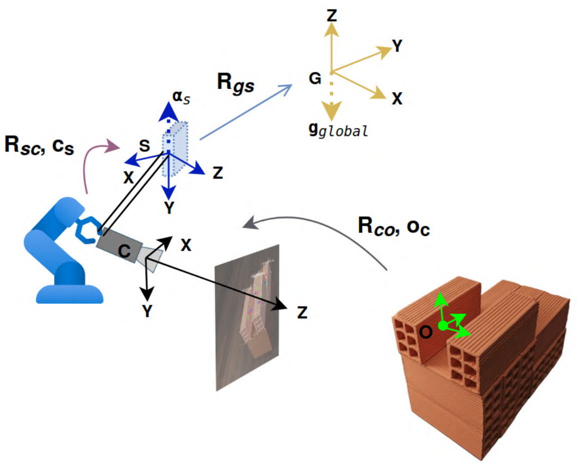
Z
Y
X
G
$\mathbf{R}_{gs}$
$\mathbf{g}_{global}$
α_s
$\mathbf{R}_{sc}, \mathbf{c}_s$
S
X
Z
Y
C
X
Y
Z
$\mathbf{R}_{co}, \mathbf{o}_c$
O

Fig. 2: Illustration of the coordinate systems in our pipeline and their transformations

B. Problem Definition

Given an RGB image $\mathcal{I}$ containing a set of N stacked objects $\mathcal{O} = [O_1, \cdots, O_N]$ and optionally a corresponding inertial acceleration measurement $\alpha_s^{\mathcal{I}}$, the goal is to identify and estimate the 6DoF poses of the objects that are graspable, both in terms of occlusion and stability of the stack.

Formally, given a set of pose hypotheses

$$\mathbf{\Pi} = \{\mathbf{T}_{co}^1, \mathbf{T}_{co}^2, \ldots, \mathbf{T}_{co}^n\}, \tag{1}$$

we aim to *filter* the raw set of pose estimations $\mathbf{\Pi}$ to yield a smaller, *physically consistent* set of **graspable** poses:

$$\mathcal{G} = \mathcal{F}(\mathbf{\Pi}, \mathcal{I}, \mathcal{M}, \alpha_\mathbf{s}), \tag{2}$$

where $\mathcal{M}$ represents 3D object models.
Figure 2 illustrates the different coordinate frames and their transformations including the pose between the camera and the object.

IV. Methodology

A. Overview

Our approach consists of four major components: **(i)** detection of 2D regions of interest (RoIs), **(ii)** 6DoF object pose estimation, **(iii)** Vision-based object filtering, and **(iv)** IMU-based object height filtering. This pipeline refines the raw pose hypotheses from **(i)-(ii)** into a final set of *unobstructed* and *visually consistent* grasping candidates. The modules are detailed below:

B. 2D Detection

An RoI defines the subregion of an image that is hypothesized to contain an object. To achieve this, we use YOLOv8 [14] as the 2D detector. For an input image $\mathcal{I}$, the detector yields a set of bounding boxes $\mathcal{D}$ for each recognized object. These bounding boxes provide the RoIs needed as input to the pose estimation algorithm.

C. 6DoF Object Pose Estimation

From each bounding box $d \in \mathcal{D}$, we crop or resize the RoI and feed it into a state-of-the-art pose estimation algorithm. In this work we use the RGB-only variant of **ZebraPose** by Su et al [8] due to its accuracy (despite using just an RGB input), computational efficiency(see BOP challenge results [9]) and its design to handle *challenging viewpoints* and *occlusions*. ZebraPose encodes every point on a 3D model with a binary code, then trains a network to predict those codes for each object pixel in an RGB image, before solving a PnP problem using 2D-3D correspondences to recover the pose. The output of each bounding box is produced as a **6DoF pose** $\mathbf{T}^i_{co}$, forming the set of pose estimations $\mathbf{\Pi}$ as described in Eq. (1).

D. Vision-based Object Filtering

In the vision-based filtering we adopt a *mask-based* strategy to assess how much of the object is actually visible. As there is a strong correlation between object visibility and graspability, eliminating predicted pose estimates with low visibility is expected to lead to an improvement of our set of grasping candidates. For each pose $\mathbf{T}^i_{co} \in \mathbf{\Pi}$ we use the:

1) **Amodal Mask ($\mathbf{M}^i_a$):** We render the full silhouette of the object (including occluded parts) from the known 3D model $\mathcal{M}$ using pose $\mathbf{T}^i_{co}$.
2) **Modal Mask ($\mathbf{M}^i_m$):** We obtain the *visible* mask from the pose estimator's segmentation output.

Using the amodal and modal masks we can define a **Visibility ratio** as:

$$r^i = \frac{|\mathbf{M}^i_m|}{|\mathbf{M}^i_a|}, \tag{3}$$

where $|\cdot|$ denotes the pixel area. A higher r^i indicates the object is more exposed and thus more *graspable*. If r^i falls below a threshold ϵ_{vis}, we discard $\mathbf{T}^i$ as insufficiently visible. The remaining set $\mathcal{G}_v \subseteq \mathbf{\Pi}$ which contains the poses that pass the visibility threshold is given as

$$\begin{aligned} \mathcal{G}_v \;&=\; \mathcal{F}_v(\mathbf{\Pi}, \mathcal{I}, \mathcal{M}) \\ &= \left\{ \mathbf{T}^i \in \mathbf{\Pi} \;\middle|\; r^i \;\geq\; \epsilon_{\text{vis}} \right\}. \end{aligned} \tag{4}$$

E. IMU-based height filtering

The object positions are described by the pose estimator in the camera coordinate systems, i.e. for object i we have $\mathbf{o}^i_c$, which describes the position of the object coordinate origin in the camera coordinate system. However, using the IMU information it is possible to recover $\mathbf{o}^{i,z}_g$, i.e. the position of the object in the global frame with respect to the direction of the gravity vector. To retrieve this, first the object position is transformed to the sensor coordinate system as:

$$\mathbf{o}^i_s = \mathbf{T}^i_{sc}\mathbf{o}^i_c. \tag{5}$$

The object position in the global frame $\mathbf{o}_g$ would be given as:

$$\mathbf{o}^i_g = \mathbf{T}^i_{gs}\mathbf{o}^i_s. \tag{6}$$

However, since we are only interested in the position along the gravity axis (height of object position), Eq.(6) is simplified to:

$$\mathbf{o}^{i,z}_g = \alpha_s \mathbf{o}_s. \tag{7}$$

We can thus rank all $\mathbf{T}_i \in \mathbf{\Pi}$ by their descending height $\mathbf{o}^{i,z}_g$ (descending) and keep only the *top-k* poses with the highest values, forming the subset $\mathcal{G}_h \subseteq \mathbf{\Pi}$. This is defined as

$$\mathcal{G}_h \;=\; \mathcal{F}_h(\mathbf{\Pi}, \mathcal{I}, \mathcal{M}, \alpha_s) \tag{8}$$

F. Final Set of Graspable Poses

We combine both the *IMU-based height filtering* with *vision based filtering* by applying them one after the other. Typically, we first apply the inertial filter to obtain $\mathcal{G}_h \subseteq \mathbf{\Pi}$, followed by the vision filter to the get final set of graspable poses $\mathcal{G} \subseteq \mathcal{G}_h$. The final filtering function $\mathcal{F} = \mathcal{F}_v \circ \mathcal{F}_h$, given as

$$\mathcal{G} \;=\; \mathcal{F}(\mathbf{\Pi}, \mathcal{I}, \mathcal{M}, \alpha_s) \tag{9}$$

This final set $\mathcal{G}$ contains filtered object poses prioritizing unobstructed objects on the top of a stack to ensure higher grasping success rates.

V. Dataset and Ground Truth Generation

Due to the absence of publicly available stacked object datasets, we construct a comprehensive synthetic dataset with 6DoF object pose and binary graspability labels using construction bricks as our example object. The data generation and annotation process is described in the following subsections. The dataset will be made available upon acceptance of the paper.

A. Synthetic Dataset Creation

We generated a synthetic and procedurally designed dataset using *BlenderProc* [15] designed for the BOP challenge [9].

To replicate real-world construction scenarios, we utilized a scanned 3D model of a brick preserving geometry and texture details. We created *3×3* brick units and procedurally stacked them to form various configurations, including regular and irregular stacks with different levels of occlusion. Utilizing BlenderProc, we simulated real-world lighting conditions along with randomized textures for walls and floors.

The bricks were stacked in the middle of a room, and camera poses were sampled within a predefined spherical radius around the scene center. The process yields RGB images, depth maps, segmentation masks, and accurate 6DoF object pose annotations. Synthetic data examples from our dataset are shown in Figure 3.

B. Ground Truth Generation

To produce ground truth annotations of graspable objects, we employ a KD-Tree based method combined with an IoU based validation. Each object i is represented by its 6DoF pose in world scene coordinates, represented as $\mathbf{T}^i_{wo}$.

Using this transformation, the eight corner points of a object are used to form a bounding box. A KD-Tree is

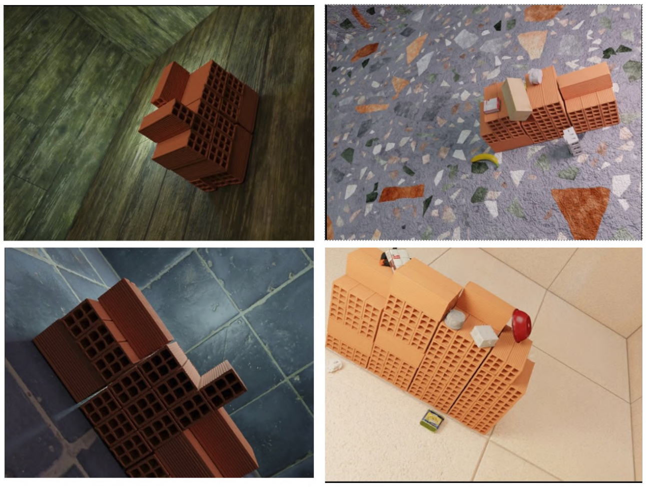

Fig. 3: Example images from the synthetic dataset of stacked objects. We simulate different stack topologies with varying lighting conditions, poses, and backgrounds. Clutter objects are added in some of the images.

constructed from the centers of these bounding boxes, enabling efficient nearest neighbor queries. For each object with center c^i and corresponding bounding box B^i, the algorithm “moves” the center along each of the six principal directions (i.e., $\pm X, \pm Y, \pm Z$) by an amount equal to the object’s size in that dimension. For each shifted bounding box, we compute the bounding boxes of its neighboring objects in the scene. We then verify if the shifted bounding box falls within the bounds of any other neighbors. This is done by checking if the shifted box has positive IoU with any of its neighbors. If no such neighbors exist, then we consider this a *missing neighbor* in that direction. An object is then classified as graspable if it is missing neighbors in at least four out of the six principal directions. In cases where exactly three directions are missing, additional structural constraints are applied: (i) there must be no neighbor in the upward ($Z+$) direction; (ii) at least one of the left/right ($X+$ or $X-$) directions must be missing a neighbor; and (iii) at least one of the front/back ($Y+$ or $Y-$) directions must be missing a neighbor. The detailed pseudo code of the ground truth generation process is provided in Algorithm 1.

Algorithm 1: Ground Truth Generation for graspable Bricks

```
Input: Brick poses $\mathcal{P} = \{\mathbf{T}^i_{wo}\}_{i=1}^n$, 3D model points $\mathcal{M}$
Output: Set of graspable brick centers $\mathcal{Q}$
Initialize: $\mathcal{B} \leftarrow []$, $\mathcal{C} \leftarrow []$, $\mathcal{S} \leftarrow []$;
for $i = 1$ to $n$ do
    $P^i \leftarrow \mathbf{T}^i_{wo}\mathcal{M}$;   $B^i \leftarrow \text{BBoxCorners}(P^i)$
      $c^i \leftarrow \text{mean}(B^i)$ ; // Compute center
    $s^i \leftarrow \max(B^i) - \min(B^i)$ ; // Compute Size
    Append $B^i$, $c^i$, $s^i$ to $\mathcal{B}$, $\mathcal{C}$, $\mathcal{S}$;
Build KD-Tree $T$ from centers $\mathcal{C}$;
$\mathcal{Q} \leftarrow \emptyset$;
for $i = 1$ to $n$ do
    $m \leftarrow 0$;
    for direction $d \in \{X, Y, Z, -X, -Y, -Z\}$ do
        $\tilde{c}^i \leftarrow c^i + s^i\tilde{d}$ ; // $\tilde{d}$ is unit vector in
          direction d
        $\tilde{B}^i \leftarrow B^i$ shifted by $s^i\tilde{d}$;
        $N^i \leftarrow T.\text{query}(\tilde{c}^i)$ ; // get neighbors
        if for all $j \in N^i$, $j \neq i$, $IoU(\tilde{B}^i, B^j) > 0$ then
            $m \leftarrow m + 1$;
    if $m \geq 4$ then
        Add $c^i$ to $\mathcal{Q}$;
return $\mathcal{Q}$
```

VI. Experiments

A. Evaluation Metrics

In this section we define the metrics we use to evaluate the correctness of the filtered candidate poses $\mathcal{G}$. Given an estimated candidate pose from $\mathbf{T}^i_{co} \in \mathcal{G}$ and a ground truth pose $\hat{\mathbf{T}}_{co}$, we compute a score between them using the error functions Average Distance of Model Points (ADD) [16] and Maximum Symmetry-Aware Surface Distance (MSSD) [9].

1) ADD-S: ADD quantifies how close the alignment is between a predicted pose and a ground-truth pose. Concretely, ADD measures the average pairwise distance between each model point $\mathbf{x} \in \mathcal{M}$ under these two poses. For objects with indistinguishable symmetries (e.g., rotational or reflective), the standard ADD metric may mistakenly penalize correct poses. To handle this we use the symmetric variant of ADD called *ADD for Symmetric Objects* (ADD-S) [16] where the distance of each projected 3D point $\mathbf{x_1}$ under the predicted pose is compared to its closest symmetric point $\mathbf{x_2}$ under the ground-truth pose. The error score e^i_{ADD-S} for a predicted pose $\mathbf{T}^i_{co}$ is defined as

$$e^i_{ADD\text{-}S} = \frac{1}{|\mathcal{M}|} \sum_{\mathbf{x_1} \in \mathcal{M}} \min_{\mathbf{x_2} \in \mathcal{M}} \left\| \mathbf{T}^i_{co}\,\mathbf{x_1} - \hat{\mathbf{T}}_{co}\,\mathbf{x_2} \right\|_2. \tag{10}$$

2) Maximum Symmetry-Aware Surface Distance (MSSD): MSSD is a pose error metric that focuses on the *largest* deviation between the ground-truth and predicted surfaces of an object, while accounting for global symmetries. This is especially critical in robotics, where local misalignments can lead to collisions or failed grasps. The error score e^i_{MSSD} between a estimated pose $\mathbf{T}^i_{co}$ and the ground truth pose $\hat{\mathbf{T}}_{co}$ is defined as

$$e^i_{MSSD} = \max_{\mathbf{S} \in \mathbf{S_M}} \min_{\mathbf{x} \in \mathcal{M}} \left\| \mathbf{T}^i_{co}\,\mathbf{x} - \hat{\mathbf{T}}_{co}\,\mathbf{S}\,\mathbf{x} \right\|_2. \tag{11}$$

where the set $\mathbf{S}$ consists of global symmetry transformation of the object model $\mathcal{M}$.

3) Accuracy Score: Unlike the typical evaluation of object pose estimation methods where the pose estimates are evaluated using Average Recall (AR) scores [9], in our task we need to incorporate the graspability aspect of the object in to the evaluation. Thus, we focus on the filtered pose

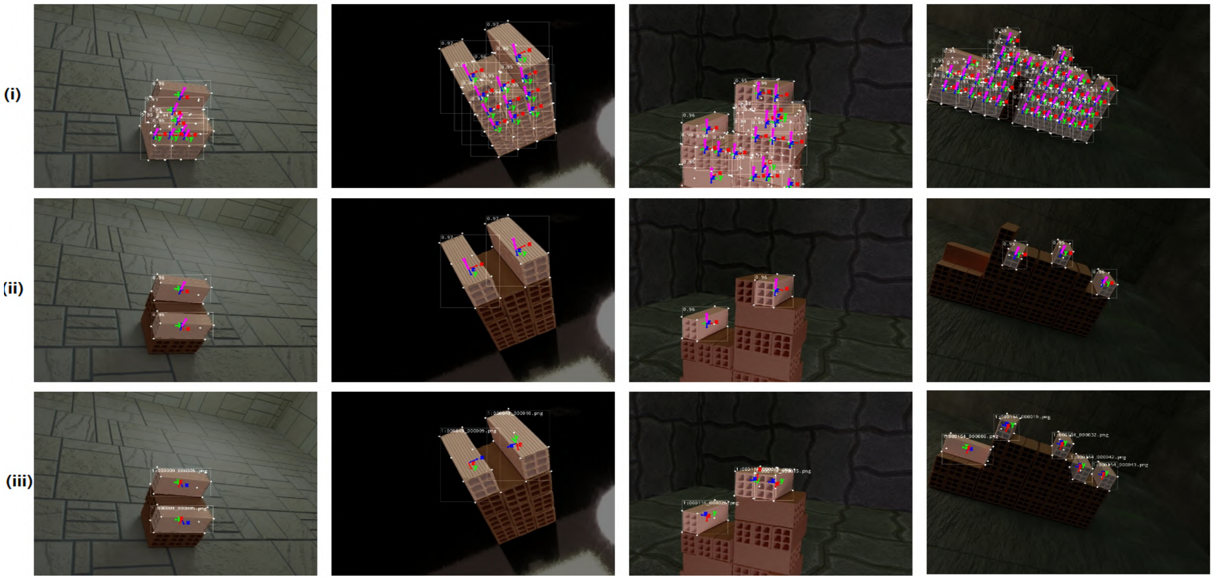

Fig. 4: Qualitative results of object selection and pose estimation with our algorithm. *(i)* shows all the pose estimates $\mathbf{\Pi}$ *(ii)* shows the graspable candidates $\mathcal{G}$ using our filtering algorithm $\mathcal{F}$ *(iii)* shows the poses from the Ground Truth.

estimates set $\mathcal{G}$ and check if they are actually graspable candidates by comparing with the ground truth graspability labels. Therefore, we focus on the Precision values as the proposed metric and compute the Average Precision (AP). A higher AP tells us how well the filtered pose candidates actually correspond to the ground truth candidates.

An estimated pose is deemed correct if its corresponding error e satisfies $e < (\theta \in \Theta_e)$, where $e \in \{e_{\text{ADD-S}}, e_{\text{MSSD}}\}$ and Θ_e is the set predefined thresholds for correctness. To compute Precision values, we perform optimal matching between predicted candidate poses and ground-truth candidates using the Hungarian (Kuhn-Munkres) algorithm [17] applied to a cost matrix of error scores. *True Positives (TP)* are matched poses whose error scores are below θ, *False Positives (FP)* are predicted poses that either fail to match any ground-truth or whose error scores are above θ . The Average Precision AP_e is then defined as the average of the Precision rates calculated for multiple settings of the threshold θ_e. The $Precision$ and the AP are defined as

$$Precision = \frac{TP}{TP + FP}, \quad AP_e = \frac{1}{|\Theta_e|} \sum_{\theta \in \Theta_e} Precision(\theta) \tag{12}$$

The settings for the Θ_e are similar to the BOP [9].

B. Implementation Details

For training and evaluating our methods, we generate a synthetic dataset with 12K images in total, consisting of 6DoF poses, modal & amodal masks, depth and RGB images saved in BOP [9] and COCO [18] formats. We split our dataset into 10K, 1K and 1K images for train, val and test splits respectively. In our experiments, for the filtering function $\mathcal{F}$ we set ϵ_{vis} to 0.80, as this allows only the bricks with the high visibility and thus least occlusions. Note that our approach is generic and independent of the selection of the exact detector and pose estimator.

C. Benchmark evaluation

The benchmark results for our approach are shown in Table I. We first run an evaluation directly on the pose estimates set $\mathbf{\Pi}$ predicted by the pose estimator. We compare the unfiltered poses with the ground truth using the metrics defined in Section VI-A. Then, we evaluate with $\mathbf{top\_k} = \mathbf{1}$ i.e. the top brick from the filtered pose estimates is selected for evaluation. Next, we also check for $\mathbf{top\_k} = \mathbf{2}, \mathbf{3}$ where 2 and 3 best bricks from the filtered pose estimates is selected for evaluation with ground truth. The results show that our method achieves good precision for selecting a single candidate for grasping and acceptable precision when we attempt to select more than one object for grasping from a single image. Qualitative examples are shown in Figure 4.

	AP$_{\text{ADD}}$↑	**AP$_{\text{MSSD}}$↑**	**mAP↑**
all poses	0.19	0.16	0.17
OURS (top_k=1)	**0.84**	**0.74**	**0.79**
OURS (top_k=2)	0.74	0.66	0.70
OURS (top_k=3)	0.68	0.62	0.65

TABLE I: Evaluation of precision over ADD and MSSD of filtered 6DoF pose estimates for grasping. *all poses* corresponds to no filtering, *top_k* corresponds to selecting k grasp candidates.

D. Ablation Study

Table II shows our ablation experiments where we compare to baselines and to partial versions of our algorithm. These experiments include: (i) Selecting random pose from the list of pose estimates (ii) Selecting the pose with highest detector confidence (iii) Only using the IMU information i.e., using only the height based filtering $\mathcal{G}_h$ as explained in IV-E and (iv) Only using the visual information i.e., using the vision based filter $\mathcal{G}_v$ as explained in Section IV-D.

The results clearly indicate the superiority of the combined visual-inertial method for object selection.

	$\mathbf{AP}_{\text{ADD}}\uparrow$	$\mathbf{AP}_{\text{MSSD}}\uparrow$	$\mathbf{mAP}\uparrow$
Random choice	0.18	0.17	0.17
Detector confidence	0.64	0.57	0.60
OURS - inertial only ($\mathcal{G}_h$)	0.68	0.55	0.61
OURS - vision only ($\mathcal{G}_v$)	0.81	0.66	0.73
OURS ($\mathcal{G}$) **(top_k=1)**	**0.84**	**0.74**	**0.79**

TABLE II: Ablation study. We compare to two baselines for object selection, random selection from all detections and top confidence detection selection. We also evaluate our visual and inertial approaches separately and combined.

E. Deployment experiments

In addition to the evaluation on the synthetic dataset shown above, we show in the supplementary video: 1) Qualitative examples from the use of our object selection and pose estimation approach in a real video of a brick stack. 2) Videos from the deployment of the algorithm on a robot arm for brick selection and grasping from a stack. To demonstrate in a real-world scenario, we used our method at a construction site to select the best graspable bricks from a brick pallette. We deployed it on the Yaskawa six-axis HC20DTP collaborative robot arm with a horizontal reach of 1700mm, a vertical reach of 3400mm and payload capacity of 20 kg with a custom two pronged gripper. We use an Intel Realsense d435i as our camera and IMU sensors which is connected to the flange of the robot.

VII. Conclusion

In this paper, we formally introduced the problem of selecting an object to grasp from a stack and estimating its pose. Due to the novelty of the problem, it was necessary to develop a new dataset with ground truth object pose and graspability labels to serve as a benchmark for evaluations. Our proposed approach combines visual and inertial information to favor objects with low occlusion that are also situated on the higher levels of the stacks to minimize grasping risk. The quantitative evaluation shows that the proposed method is highly effective compared to baseline approaches, while also confirming that the introduced problem is complex and worthy of additional future research attention due to its significance for real-world robotic application tasks. This is also demonstrated in the deployment of the method on a robotic arm in a construction-domain brick grasping scenario.

Acknowledgments

This work has been partially funded by the EU Horizon Europe Framework Program under grant agreement 101058236 (HumanTech) and by German Ministry for Economics and Climate Action (BMWK) under Grant Agreement 13IK010 (TWIN4TRUCKS). We thank our HumanTech partners for the collaboration: BAUBOT (Platform), SINTEF Manufacturing (Control) and ACCIONA (testing site and material).

References

[1] Y. Lin, Y. Su, P. Nathan, S. Inuganti, Y. Di, M. Sundermeyer, F. Manhardt, D. Stricker, J. Rambach, and Y. Zhang, “Hipose: Hierarchical binary surface encoding and correspondence pruning for rgb-d 6dof object pose estimation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 10 148–10 158.

[2] A. Cordeiro, L. F. Rocha, C. Costa, P. Costa, and M. F. Silva, “Bin picking approaches based on deep learning techniques: A state-of-the-art survey,” in *2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. IEEE, 2022, pp. 110–117.

[3] B. Drost, M. Ulrich, P. Bergmann, P. Hartinger, and C. Steger, “Introducing mvtec itodd-a dataset for 3d object recognition in industry,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 2200–2208.

[4] P. Besl and N. D. McKay, “A method for registration of 3-d shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[5] M. Rad and V. Lepetit, “Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3828–3836.

[6] Y. Su, J. Rambach, A. Pagani, and D. Stricker, “Synpo-net—accurate and fast cnn-based 6dof object pose estimation using synthetic training,” *Sensors*, vol. 21, no. 1, p. 300, 2021.

[7] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, “Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes,” in *Robotics: Science and Systems*, 2018.

[8] Y. Su, M. Saleh, T. Fetzer, J. Rambach, N. Navab, B. Busam, D. Stricker, and F. Tombari, “Zebrapose: Coarse to fine surface encoding for 6dof object pose estimation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 6738–6748.

[9] T. Hodan, M. Sundermeyer, Y. Labbe, V. N. Nguyen, G. Wang, E. Brachmann, B. Drost, V. Lepetit, C. Rother, and J. Matas, “Bop challenge 2023 on detection segmentation and pose estimation of seen and unseen rigid objects,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 5610–5619.

[10] J. Bohg, A. Morales, T. Asfour, and D. Kragic, “Data-driven grasp synthesis—a survey,” *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2014.

[11] A. Sahbani, S. El-Khoury, and P. Bidaud, “An overview of 3d object grasp synthesis algorithms,” *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 326–336, 2012, autonomous Grasping.

[12] H. Zhang, J. Tang, S. Sun, and X. Lan, “Robotic grasping from classical to modern: A survey,” 2022. [Online]. Available: https://arxiv.org/abs/2202.03631

[13] K. Kleeberger, R. Bormann, W. Kraus, and M. F. Huber, “A survey on learning-based robotic grasping,” *Current Robotics Reports*, vol. 1, no. 4, pp. 239–249, 2020.

[14] G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics yolov8,” 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[15] M. Denninger, D. Winkelbauer, M. Sundermeyer, W. Boerdijk, M. Knauer, K. H. Strobl, M. Humt, and R. Triebel, “Blenderproc2: A procedural pipeline for photorealistic rendering,” *Journal of Open Source Software*, vol. 8, no. 82, p. 4901, 2023.

[16] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, “Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes,” in *11th Asian Conference on Computer Vision (ACCV)*. Springer, 2013, pp. 548–562.

[17] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *13th European conference on Computer Vision (ECCV)*. Springer, 2014, pp. 740–755.