SPILL: Size, Pose, and Internal Liquid Level Estimation of Transparent Glassware for Robotic Bartending

Louis Adriaens[1], Thomas Lips[1], Mathieu De Coster[1], Andreas Verleysen[1] and Francis wyffels[1]

***Abstract*— Robotic perception of transparent objects presents unique challenges due to their refractive properties, lack of texture, and limitations of conventional RGB-D sensors in capturing reliable depth information. These challenges significantly hinder robotic manipulation capabilities in real-world settings such as household assistance, hospitality, and healthcare. To address these issues, we propose SPILL: A lightweight perception pipeline for Size, Pose, and Internal Liquid Level estimation of unknown transparent glassware using a single view. SPILL combines object detection with semantic keypoint detection, and operates without requiring object-specific 3D models or depth completion. We demonstrate its effectiveness in autonomous robotic pouring tasks. Additionally, to enhance the robustness and generalization of keypoint detection to diverse real-world scenarios, we introduce *Glasses-in-the-Wild*, a new dataset that captures a wide variety of glass types in realistic environments. Evaluated on a robot manipulator, SPILL achieves a 93.6% success rate across 500 autonomous pours with 20 unseen glasses in three diverse real-world scenes. We further demonstrate robustness through multiple live public events in real-world, human-centered environments. In one recorded session, the robot autonomously served 62 drinks with a 98.3% success rate. These results demonstrate that task-relevant keypoint detection enables scalable, real-world transparent object interaction, paving the way for practical applications in service and assistive robotics - without spilling a drop. Dataset and code are available at https://github.com/Louadria/SPILL.**

***Index Terms*— Perception for Grasping and Manipulation, Object Detection, Segmentation and Categorization, RGB-D perception, Data Sets for Robotic Vision, Service robotics**

I. Introduction

Robots are increasingly being deployed in human-centered environments, where they must interact with diverse and unstructured scenes. To operate safely and effectively in such settings, robots require reliable perception systems capable of identifying and localizing objects in their surroundings. While opaque, rigid objects have been extensively studied, transparent and reflective items, such as glasses or cups, remain a major challenge for vision-based robotic systems. Yet, such objects are pervasive in everyday life, appearing in a wide range of contexts: from glassware and plastic bottles on tabletops to laboratory equipment such as Erlenmeyer flasks and beakers. Robust methods to localize these objects are critical for robotic tasks in industries ranging from hospitality and healthcare to scientific research.

Transparent objects pose unique challenges due to their lack of texture and distortion of background features.

Supported by the Research Foundation Flanders (FWO, 1S56022N) and the EU Horizon Europe project euROBIn (101070596).

[1]AI and Robotics Lab (IDLab–AIRO), Ghent University–imec, Technologiepark 126, 9052 Zwijnaarde, Belgium.

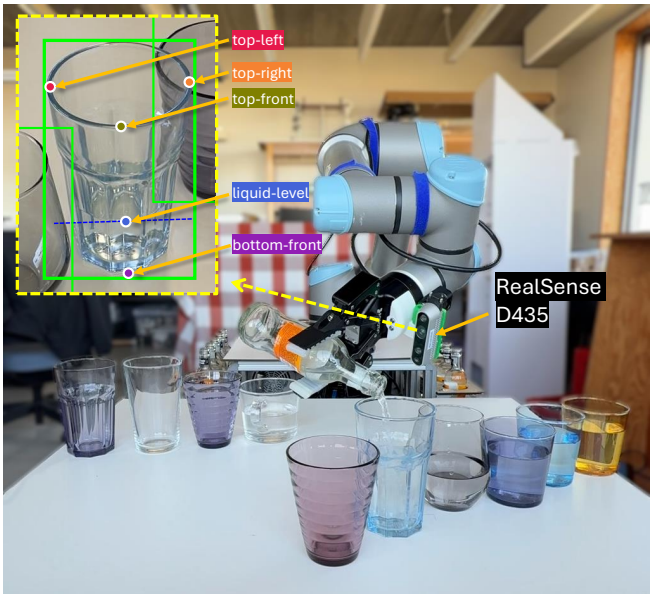

Fig. 1. The UR5e-based bartender robot setup in action. The inset highlights the perception output captured using the RealSense D435 camera, showing the detected glasses bounding boxes (green) along with predicted keypoints (top-left, top-right, top-front, fluid-level and bottom-front).

These reflective and refractive surfaces exhibit strong non-Lambertian properties, i.e., they do not reflect light equally in all directions, confounding RGB-based approaches, which results in inaccurate depth information when captured by RGB-D sensors [1]. These limitations hamper traditional perception pipelines, such as point cloud-based methods, which rely on accurate depth to reconstruct scene geometry.

These challenges are amplified when the container holds liquid. Reflections and refractions at the fluid–air interface introduce additional visual ambiguities, while colored or opaque liquids obscure the container's internal structure. In robotic tasks such as pouring a drink, estimating the liquid level and available volume is essential, adding another layer of complexity to an already difficult perception problem.

Addressing these challenges is an active area of research. A common strategy is depth completion, where missing or noisy depth data is inferred using methods such as surface normal estimation [2], [3], Neural Radiance Fields (NeRF) [4], [5], or end-to-end depth completion neural networks [6]–[9]. Others integrate additional sensing modalities, such as light-field cameras [10], to better capture the complex optical properties of transparent objects.

Moreover, many existing robotic systems assume a fixed set of known objects or require environment-specific cali-

bration [10]–[13], limiting generalization to new settings or previously unseen objects - capabilities that are critical for deployment in unstructured, real-world environments.

While these works focus on transparent object perception or depth completion in isolation, none - to the best of our knowledge - demonstrate complete robotic systems that integrate perception with complex physical interaction. End-to-end deployment, especially in unstructured, real-world environments, remains rare, highlighting the gap between lab-scale results and practical applicability.

To address this gap, we propose a lightweight, real-time system named SPILL, which combines object detection with semantic keypoint detection for Size, Pose, and Internal Liquid Level estimation of unknown transparent glassware using a single view, without object-specific models or predetermined scene configurations. Running at 25.3 ms per frame ($\approx$ 40 FPS) on an i7 CPU with an RTX 4060 GPU, SPILL operates efficiently on standard hardware. We demonstrate its effectiveness in a real-world robotic bartender setup, as shown in Fig. 1. To the best of our knowledge, this is the first system to achieve accurate pouring into unknown transparent glasses in an unstructured, human-facing environment.

Our method combines YOLOv8-based object detection with the detection of a small set of semantically meaningful and interpretable keypoints, such as the rim (*top-left, top-right, top-front*), base (*bottom-front*), and liquid level (*liquid-level* in Fig. 1), to construct a compact representation that generalizes across glassware types. This structure enables accurate determination of the size, 6D pose, and contained liquid volume of glasses, all without relying on explicit depth completion, object-specific models, or scene-specific calibration, while also improving transparency and explainability compared to black-box approaches.

To validate SPILL, we deploy the system on a real robotic platform: A UR5e-based autonomous bartender that detects and fills glasses (Fig. 1). Across 500 pouring attempts involving 20 types of previously unseen glassware, the system achieved a 93.6% success rate. We further evaluated its real-world performance through multiple public demonstrations, where it served over 200 drinks. In one recorded session, 61 of 62 pours were successful, with a single failure occurring in a complex, cluttered scene, confirming SPILL's robustness and reliability in practical use cases.

To summarize, we make the following contributions:

1) We introduce **SPILL: Size, Pose, and Internal Liquid Level estimation of unknown transparent objects from a single RGB-D image**. This lightweight perception pipeline uses semantic keypoints to estimate the 6D pose, size, and internal liquid level of transparent glassware, without requiring object-specific models, depth completion, or environment-specific calibration. SPILL operates on a single RGB-D image, and generalizes well to previously unseen glasses with diverse shapes, sizes, colors, and patterns. It demonstrates robustness to clutter, occlusion, reflections, and varying lighting, enabling practical manipulation tasks such as autonomous drink pouring into unknown containers.
2) By training SPILL's keypoint detector solely on RGB images, we enable scalable data collection, realized through the **Glasses-in-the-Wild dataset**, a crowd-sourced dataset of transparent glassware in diverse domestic and real-world environments, annotated with bounding box and keypoints. This dataset complements ClearPose [14] by covering edge cases such as opaque or colored glasses, exotic shapes, and background distractors, significantly boosting our model's robustness.

II. Related Work

Perceiving and manipulating transparent objects remains a challenging problem in computer vision and robotics. In this section, we review prior work across three areas most relevant to our approach: (i) transparent object perception, (ii) keypoint-based pose estimation, and (iii) robotic pouring and fluid level estimation.

A. Transparent Object Perception

Standard RGB-D sensors find it challenging to detect transparent objects due to their refractive and specular properties, which distort or obscure depth information. A range of techniques have been developed to address this issue.

Early methods addressed transparent object perception using RGB-D segmentation and pose estimation [13], [18]. Later work shifted toward depth completion, recovering missing or inaccurate depth by leveraging RGB cues, surface normals, or learned priors [2], [3], [19]–[21].For example, ClearGrasp [20] and TransNet [22] estimate surface normals as an intermediate step toward depth prediction, while RFTrans [2] models refraction explicitly via a refractive flow field. Though effective, these approaches rely on deep, compute-heavy networks, limiting real-time deployment.

Others use multi-view setups, including stereo cameras [23]–[25] and light field imagery [10] to overcome depth ambiguity by aggregating multiple perspectives. For example, KeyPose [23] and GhostPose [24] triangulate 3D keypoints from synchronized image pairs or sequences. While effective, these methods add hardware and computational complexity, requiring multiple camera setups or specialized sensors.

Other techniques rely on known object priors, such as 3D CAD models, to estimate pose [10]–[13], but these limit generalization. In contrast, our method targets unknown transparent objects, for which no CAD model or prior training data is available. We make no assumptions on the type of glass, allowing our method to operate in dynamic, unstructured environments where the robot must pour into a wide range of unseen glasses, mugs, or cups.

B. Keypoint Detection

Keypoints provide a compact and interpretable object representation for robotic manipulation, allowing task-specific reasoning about object geometry. For example, kPAM [26] introduced semantic 3D keypoints to generalize across object instances without relying on rigid templates, while ignoring task-irrelevant shape details. Similarly, KETO [27] demonstrated that task-specific keypoints can provide structured object understanding, which can guide tool-use manipulation.

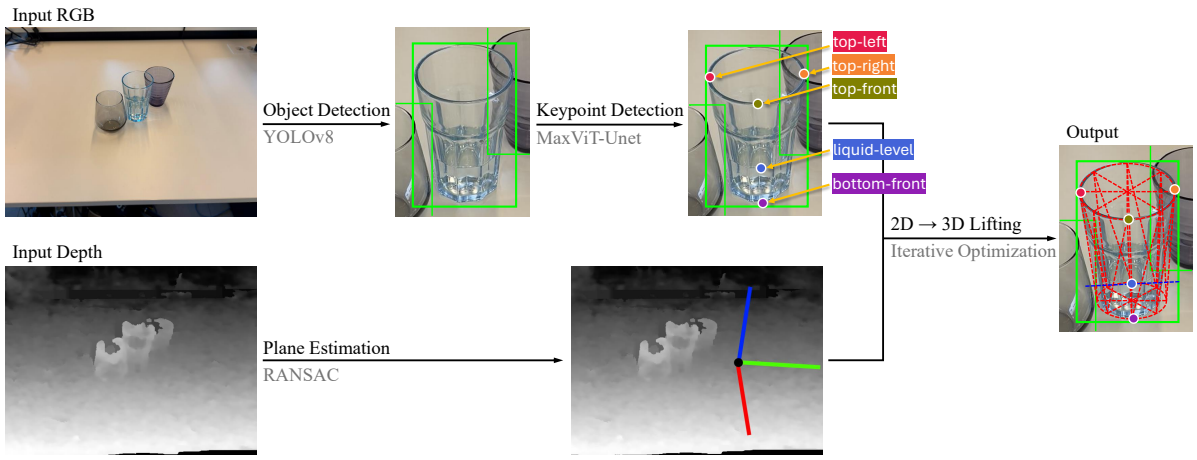

Fig. 2. Overview of the SPILL pipeline for Size, Pose, and Internal Liquid Level estimation of transparent glassware. The system takes a single RGB-D image as input. YOLOv8 [15] extracts bounding boxes around glassware from the RGB image, and the cropped regions are passed to a MaxViT-UNet [16], [17] keypoint detector to extract 2D semantic keypoints, while the depth map is used to extract the support plane via RANSAC. Together, these inputs are used to lift the 2D keypoints to 3D using an iterative optimization step and reconstruct a tapered-cylinder model.

In transparent object perception, methods like KeyPose [23] and GhostPose [24] also leverage keypoints, but require multi-view input (e.g., stereo image pairs or batches of images) to directly regress 3D coordinates or triangulate them from 2D views. These methods avoid the depth ambiguity of transparent materials but rely on synchronized cameras or sequence-based processing, increasing system complexity.

C. Robotic Pouring and Fluid Estimation

Most prior work on transparent object interaction focuses on *grasping*, typically using depth completion techniques or surface normal estimation to obtain usable geometric cues, which are then used to compute feasible grasp points. Some techniques even bypass explicit 6-DoF pose estimation altogether in favor of directly predicting a suitable grasp pose from partially reconstructed geometry [10].

Research on *pouring*, in contrast, generally assumes the target vessel is already localized and focuses on trajectory generation and control [28]. Fluid level or volume estimation has also been explored using vision-based methods, including depth-based regression [29], optical flow [30], and self-supervised learning [31]. However, these techniques tend to be computationally expensive or require controlled environments. While methods like SimLiquid [32] have demonstrated the feasibility of liquid volume estimation from RGB-D data using a YOLO-based model, they rely on a model-based learning approach that requires a sufficiently large dataset of similar containers to generalize effectively. Moreover, they do not account for the complexities of cluttered scenes, challenging real-world lighting conditions, diverse vessel types, and patterned surfaces.

Other approaches, such as in Zhu et al. [33], employ multimodal sensing with visual and tactile inputs to estimate fluid volumes, increasing system complexity and hardware requirements. Additionally, while datasets like ClearPose [14] contain some examples of filled glasses, they are limited in visual diversity and environmental realism.

III. Methodology

This work introduces SPILL: Size, Pose, and Internal Liquid Level estimation of unknown transparent objects from a single RGB-D image to enable autonomous robotic pouring. Unlike prior approaches that rely on multi-view setups or depth completion, SPILL leverages lightweight, semantically meaningful keypoints. The SPILL pipeline, as illustrated in Fig. 2, consists of two main perception modules: (1) an off-the-shelf YOLOv8-based object detector [15] to identify transparent glassware, and (2) a MaxViT-UNet-based keypoint detection model [16], [17] trained to predict following semantic 2D keypoints (cf. Fig. 2):

- *bottom-front*: The front center of the base of the glass from the perspective of the camera,
- *top-left, top-right, top-front*: The left, right, and front of the glass's rim from the perspective of the camera (for locating the pouring target),
- *liquid-level*: The liquid level, corresponding to where the liquid surface meets the front of the glass in the image (for estimating how full the glass is).

By focusing on semantically meaningful and task-relevant features, SPILL avoids dense reconstructions required by depth completion methods, maintaining low computational overhead while generalizing effectively across diverse glass types. The pipeline is also highly flexible, facilitating the estimation of task-relevant geometric properties, e.g., the fullness of a glass is estimated via the addition of a single keypoint at the liquid surface (*liquid-level*). This lightweight representation is well suited for pouring, where relative fullness is sufficient.

To recover 3D geometry, SPILL leverages geometric reasoning on top of these 2D keypoints. A support plane, typically corresponding to the tabletop surface on which the glassware rests, is first extracted from the depth map. Then, the keypoints are lifted to 3D to recover the glass's size, pose, and internal liquid level under a tapered-cylinder model.

TABLE I

L2 DISTANCE ERRORS (IN PIXELS) OF THE KEYPOINT DETECTION MODEL ON THE VALIDATION SET OF GLASSES-IN-THE-WILD.

Keypoint	Mean Error ± Std (px)	Median Error (px)
bottom-front	4.17 ± 6.71	2.24
top-front	2.65 ± 5.43	2.00
top-left	1.79 ± 2.76	1.41
top-right	2.18 ± 5.30	1.41
fluid-level	7.37 ± 16.25	2.83
Overall	3.63	1.98

The remainder of this section details (i) the architecture and training details of this keypoint detection model, (ii) the glass size, pose, and internal liquid level estimation process from 2D keypoints, and (iii) the datasets used to train the keypoint detection model.

A. Keypoint Detection Module

The keypoint detection model follows a UNet-like encoder-decoder architecture [17], similar to the pipeline used by Lips et al. [34]. Keypoint detection is formulated as a heatmap regression task, where each semantic keypoint category (e.g., base, rim, fluid level) is mapped to a distinct heatmap. The encoder uses pretrained MaxViT [16] weights, while the decoder is trained from scratch. Similar to UNet [17], the architecture incorporates skip connections at multiple spatial resolutions, allowing finer spatial details from early encoder layers to guide the decoder's predictions.

The model is trained on a combination of the ClearPose dataset and a subset of *Glasses-in-the-Wild*. The training set includes all 550 images from ClearPose and 690 from *Glasses-in-the-Wild*, totalling 1240 images. The validation set contains 310 images from distinct *Glasses-in-the-Wild* participants to prevent train/validation data leakage, yielding an 80%/20% split. Images are cropped using YOLOv8-predicted bounding boxes with padding and resized to 256×256, preserving spatial context and consistency.

Training is performed for up to 500 epochs with early stopping, a batch size of 16, and random data augmentation. Heatmap regression uses Gaussian blobs (std. dev. 8 px) to model spatial ambiguity of the semantic keypoints. Full details are available in the code.

The validation set is used to select the best-performing checkpoint, with localization errors (L2 distance between predicted keypoint and the ground truth keypoint) detailed in Table I. Notably, *bottom-front* and *fluid-level* show higher errors, and the large standard deviation of the *fluid-level* keypoint reflects occasional failure cases when transparent liquid is visually mistaken for an empty glass.

B. Size, Pose, and Internal Liquid Level Estimation

SPILL makes three simplifying assumptions tailored to the pouring task to enable single RGB-D image estimation:

- **Rotational symmetry**: Most drinking glassware is approximately rotationally symmetric. SPILL ignores features like handles, which are irrelevant to pouring.
- **Tapered-cylinder model**: SPILL models glasses as tapered cylinders, which is sufficient for pouring and volume estimation. Non-conforming shapes like wine or cocktail glasses with stems are excluded.
- **Support plane assumption**: SPILL assumes glasses are placed upright on a flat surface, such as a table.

These assumptions make SPILL lightweight by eliminating the need for depth completion or multi-view triangulation, suitable for real-time use in human-centered robotics.

To estimate the 3D geometry of the glass, the 2D keypoints are lifted into 3D space using the camera intrinsics and the estimated support plane. The support plane is modeled as $\pi : ax + by + cz + d = 0$ in the camera frame, with normal vector $\mathbf{n} = [a, b, c]^\top$. The plane parameters are estimated from the full point cloud, using RANSAC to deal with outliers (including the points corresponding to the glasses).

The *bottom-front* keypoint $p_{\text{bf}} = (u_{\text{bf}},\ v_{\text{bf}})$ is lifted into 3D by assuming it lies on the estimated support plane π. A projection ray is cast from the camera's optical center through the 2D keypoint via the camera intrinsics matrix $\mathbf{K}$:

$$\mathbf{p}_{\text{bf}} = \mathbf{K}^{-1}[\, u_{\text{bf}}, v_{\text{bf}}, 1\,]^\top,\ \mathbf{x}_{\text{bf}} = -\frac{d}{\mathbf{n}^\top \mathbf{p}_{\text{bf}}} \cdot \mathbf{p}_{\text{bf}},$$

where the second equation gives the 3D intersection of the ray with the support plane. Intuitively, this "lifts" the 2D keypoint onto the plane where the glass stands. A local coordinate frame is then defined on the support plane:

$$\mathbf{Z} = \mathbf{n}, \quad \mathbf{X} = \frac{\mathbf{x}_{\text{bf}} - (\mathbf{n}^\top \mathbf{x}_{\text{bf}})\mathbf{n}}{\|\mathbf{x}_{\text{bf}} - (\mathbf{n}^\top \mathbf{x}_{\text{bf}})\mathbf{n}\|}, \quad \mathbf{Y} = \mathbf{Z} \times \mathbf{X}.$$

Here, $\mathbf{Z}$ is the plane normal, $\mathbf{X}$ lies in the plane and points from the glass base toward the camera, and $\mathbf{Y}$ completes the orthogonal basis.

To estimate the glass radius r and height h, we first measure the pixel width and height from the 2D keypoints:

$$w_{\text{2D}} = \|p_{\text{tl}} - p_{\text{tr}}\|, \quad h_{\text{2D}} = \|p_{\text{bf}} - p_{\text{tf}}\|.$$

We then project these into 3D, compensating for perspective and foreshortening:

$$r = \frac{w_{\text{2D}}}{2} \cdot \frac{\|\mathbf{x}_{\text{bf}} + h\mathbf{n} - r\mathbf{X}\|}{f},\ h = h_{\text{2D}} \cdot \frac{\left\|\mathbf{x}_{\text{bf}} + \frac{h}{2}\mathbf{n}\right\|}{f \cdot \sqrt{1 - \left(\frac{\mathbf{n}^\top \mathbf{x}_{\text{bf}}}{\|\mathbf{x}_{\text{bf}}\|}\right)^2}}.$$

where $f = \frac{f_x + f_y}{2}$ is the average focal length of the camera.

These implicit equations relate the observed 2D keypoint distances to the true 3D dimensions of the glass under projection. They are solved via fixed-point iteration, initialized with $r = h = 0$, which converges rapidly in practice

To account for noise in the 2D keypoint detections, the parameters of the cylindrical model (h and r) are further refined via nonlinear optimization. Specifically, we minimize the reprojection error e between the 3D keypoints $\mathbf{x} = [X, Y, Z]^\top$ (as predicted by the 3D cylinder model) and their corresponding 2D observations $\mathbf{p} = (u, v)^\top$:

$$e = \|\text{projection} - \text{observation}\| = \left\|(\tfrac{1}{Z}\mathbf{K}\mathbf{x})_{0:1} - \mathbf{p}\right\|,$$

where $(\cdot)_{0:1}$ extracts the 2D image coordinates.

This step also estimates a tapering angle θ, modeling a linear change in radius from base to rim, to account for glasses that deviate from perfect cylinders. The tapering angle θ and height h are optimized jointly to minimize the error between the observed 2D keypoints and the re-projection of their 3D estimates.

The full SPILL pipeline runs at an inference time average 25.3 ms of per frame on an Intel® Core™ Ultra 7 Processor 155H CPU with NVIDIA® GeForce RTX™ 4060 GPU, including 8.49 ms $\pm$ 1.52 ms for YOLOv8 inference, 14.9 ms $\pm$ 1.52 ms for keypoint detection, and 1.87 ms $\pm$ 0.15 ms for 2D-to-3D lifting and geometric optimization.

C. Dataset Details

To train the keypoint detector, we use the ClearPose dataset [14], supplemented with our own *Glasses-in-the-Wild* dataset, which captures a wide range of real-world conditions. This dataset expands the diversity of glass shapes, sizes, colors, and contents, as well as environmental factors such as lighting, reflections, occlusions, and backgrounds. The following sections describe these datasets in more detail.

1) ***ClearPose Dataset****:* The ClearPose dataset [14] was used as the primary source for training the keypoint detection model. This dataset is divided into nine sets, each containing images of transparent objects under varying levels of clutter, lighting, and occlusion. Glasses were detected and extracted using the YOLOv8m object detection model, followed by manual confirmation through visual inspection.

Sets 1 through 3 of ClearPose were excluded due to irrelevance and high clutter levels; Dataset 1 focused solely on chemical equipment, and Datasets 2 and 3 presented excessive occlusions. The remaining subsets (Sets 4 through 9) were further filtered, yielding 550 samples comprising 14 distinct types of glasses across 25 unique scenes. To ensure a balanced representation of glasses with and without liquid, additional samples were curated from Datasets 8 and 9, which included a higher prevalence of glasses containing liquids. The final curated dataset included 210 empty glasses and 340 glasses containing varying liquid levels.

Despite its utility, ClearPose has notable limitations: it covers only 14 glass types, includes only fully transparent glassware without colored or patterned variants, and provides few liquid-containing samples despite targeted sampling.

2) ***Glasses-in-the-Wild Dataset****:* To address these shortcomings, we crowdsourced the *Glasses-in-the-Wild* dataset. Because SPILL's keypoint detector is trained solely on RGB images, scalable data collection with ordinary cameras was possible. This dataset comprises 1,000 images contributed by 11 participants, covering 93 unique glasses in 60 diverse real-world scenarios with varying backgrounds, lighting conditions, reflections, occlusions, and distractors.

The dataset covers a wide range of liquid levels, from nearly empty to almost full, ensuring robust training for liquid level estimation. 24.3% of the glasses are empty, while 75.7% contain liquid, with a mean fill of 48%. This diversity in glass geometry, liquid levels, and environmental conditions provides essential variability for training keypoint-based models under realistic conditions.

IV. Experiments & Results

To evaluate the real-world performance of SPILL for robotic pouring, we conducted 500 autonomous pouring trials and measured end-to-end task performance. Beyond controlled experiments, we also deployed the system in live public demonstrations to assess robustness in unstructured settings. This section describes (i) the experimental design, (ii) quantitative results, (iii) and insights from public events.

A. Experimental Design

The robot setup, shown in Fig. 1, consists of a UR5e equipped with a wrist-mounted Intel RealSense D435 RGB-D camera for perception. A rack of bottles is mounted at fixed positions beside the robot, allowing it to autonomously grasp and return bottles during operation.

At the start of each trial (a single tabletop configuration), the robot captures an RGB-D image and runs the SPILL pipeline to detect all visible glasses and estimate their size, 6D pose, and fill level. For each detected glass, a trajectory is planned using the recovered 3D cylinder models for collision avoidance. Pouring is executed via a pre-recorded wrist-tilt motion aligned with the estimated glass opening. The amount of liquid poured is controlled by modulating the extent of the pouring motion, based on the predicted available volume.

We conducted 500 individual pouring attempts in which the robot had to fill a glass for 75% with water. We tested the robot in three distinct scenes and created a large number of different glass configurations for each scene, each containing a varying number of glasses (from 1 to 10). Fig. 3 illustrates three representative trials illustrating variations in environmental conditions, scene complexity, and glass types. The selected scenes differ in table height, surface texture, and lighting to ensure diverse coverage.

Glassware was selected from a set of 20 unique glasses, varying in shape, color, opacity, and volume (5-170 cl). A round-robin assignment strategy ensured even usage across trials. Importantly, these glasses were excluded from the training and validation sets used for the keypoint detector to ensure a fair assessment of generalization.

During each trial, we placed a subset of glasses on the table at random, introducing variability in object arrangement and partial occlusions. A subset of glasses was pre-filled to varying degrees to assess the system's ability to detect fluid levels and ignore full containers. This design allowed simultaneous evaluation of the size, pose, and internal liquid level estimation of the system under realistic conditions.

B. Main Results

Each pouring outcome is categorized in Figure 4, summarizing all 500 trial outcomes. The robot was tasked with filling each glass to 75% capacity and achieved a 93.6% overall success rate, defined as the combined total of the following four categories:

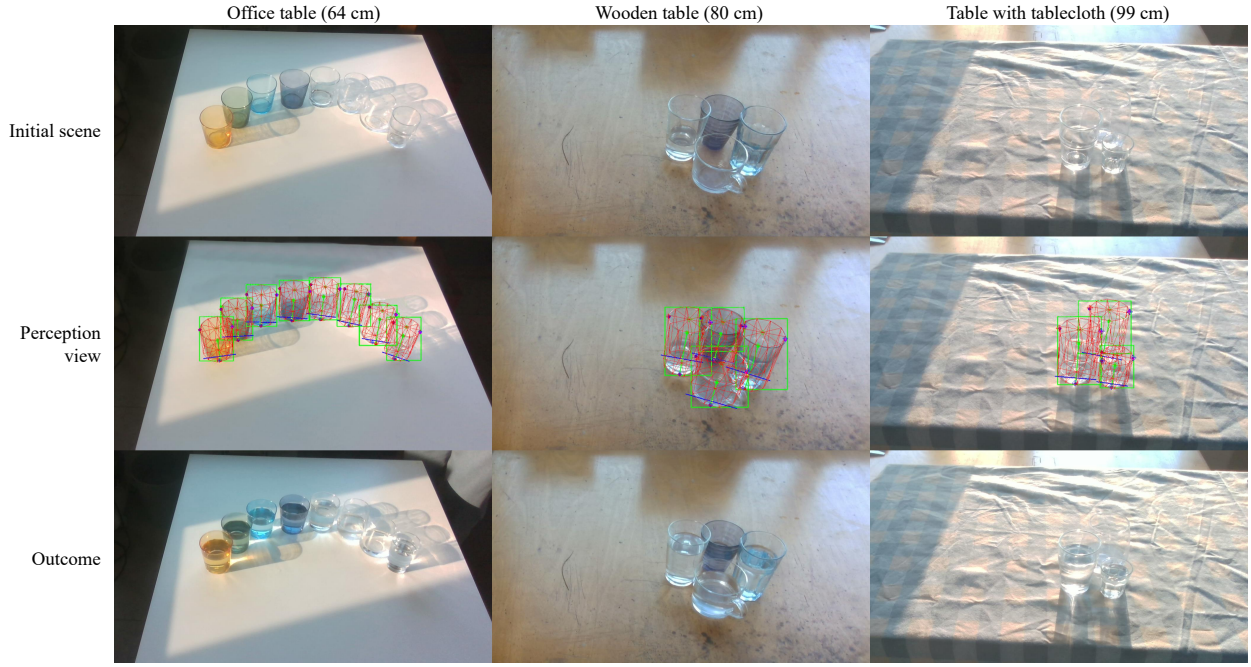

Fig. 3. Qualitative examples from three different trials illustrating the diversity in backgrounds, lighting conditions, and glassware. Each column corresponds to a single trial. The top row shows the initial scene setup; the middle row displays the annotated scene with estimated glass dimensions and fluid volume predictions; the bottom row shows the outcome after pouring.

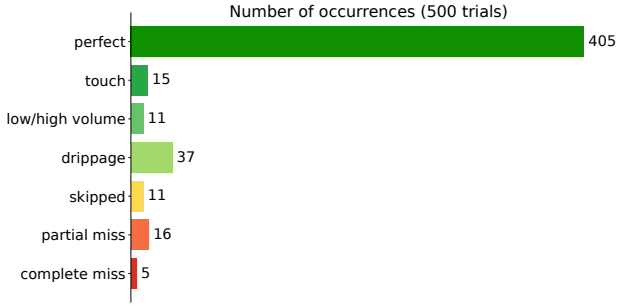

Fig. 4. Experimental outcomes across 500 robot pouring trials. The system achieved a 93.6% (*perfect*, *touch*, *low/high volume*, and *drippage*) overall success rate, with most failures stemming from minor misalignments or intentionally skipped pours (2.2%). Only 4.2% of trials (*partial* and *complete misses*) resulted in actual spillage.

- **Perfect success (81.0%)**: the pour was precisely centered, with no spillage or contact with the glass.
- **Touch (3.0%)**: the robot made minor contact with the glass, but pouring proceeded cleanly without liquid loss.
- **Low/high volume (2.2%)**: the final fill level was below 50% (underfill) or exceeded 90% (overfill).
- **Drippage (7.4%)**: a few small droplets were spilled, but did not require cleanup or intervention.

Failures accounted for 6.4% of trials, comprising:

- **Skipped (2.2%)**: the robot avoided pouring, either due to perceiving an empty glass as full, or due to planning constraints such as collisions or unreachable poses.
- **Partial miss (3.2%)**: the pour was off-center causing some spillage, but most liquid still entered the glass.
- **Complete miss (1.0%)**: the pour entirely missed the glass, typically requiring human intervention.

C. Detailed Performance Analysis

We now provide additional analysis of the results of the experiments described in Section IV-A, focusing on SPILL's ability to generalize across diverse scenes, varying number of glasses in the scene, and to previously unseen glasses. We therefore break down the success rates of our experiments along the variations on these three aspects. The results are given in Fig. 5 and are discussed in more detail below.

Scene Variation Fig. 5 (left) shows the success rate grouped by scene. Despite differences in lighting, table height, and surface texture, the system maintained consistent performance across all three setups, demonstrating SPILL's robustness to environmental variation.

Number of Glasses Fig. 5 (middle) shows performance as a function of the number of glasses in the scene. Trials with more than three glasses show a modest increase in failures, mainly due to increased occlusions and planning complexity. Glasses were placed within a fixed $60\,\text{cm} \times 60\,\text{cm}$ tabletop area, constrained by the robot's workspace. However, the failure rate remains stable beyond three glasses, suggesting that SPILL handles spatial complexity well as long as glasses are sufficiently spread out, i.e., performance is more sensitive to occlusion than to the absolute number of glasses.

Per-Glass Analysis Fig. 5 (right) reports results grouped by glass type. Simple, cylindrical glasses exhibited the highest success rates. More challenging designs, such as curved (e.g., storsint), ribbed (e.g., glasmal), or patterned (e.g., pokal) glasses, had increased error rates due to visually deceptive geometries and less predictable geometry. These

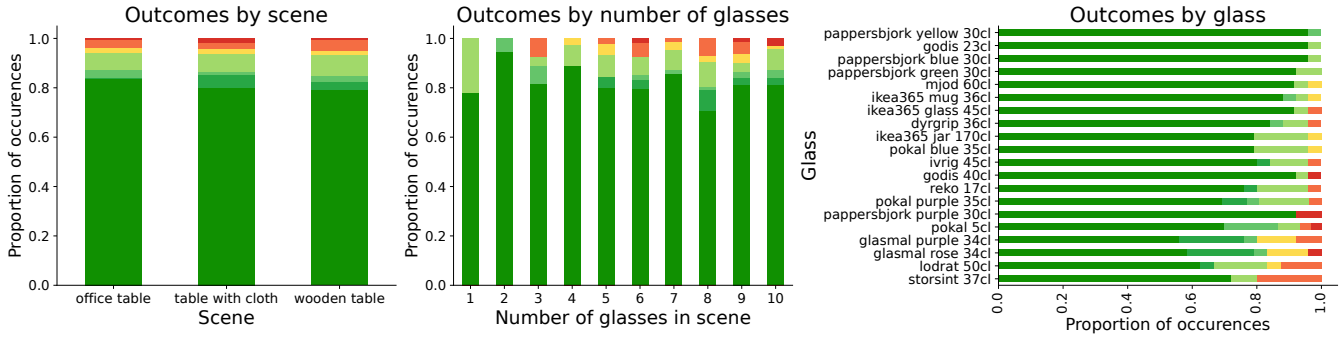

Fig. 5. Detailed performance analysis across 500 pouring trials. (**Left**) Outcomes grouped by scene; (**Middle**) Outcomes versus the number of glasses in the scene; (**Right**) Outcomes per glass type, sorted by success rate.

results highlight the influence of object shape and appearance on perception and pose estimation accuracy.

D. Public Demonstrations

To further assess the practical viability of SPILL, we deployed the system in four live public demonstrations, where the robot was tasked with pouring juice into glasses placed freely on a table by visitors. Unlike controlled experiments, there was no control over glass type, placement, or lighting; participants were simply instructed to place a glass on the table and not touch it. The robot operated continuously using the SPILL pipeline, without any environment-specific calibration. Fig. 6 shows the robot during these events, illustrating the real-world, human-centered environment.

Throughout the demonstrations, the robot served over 200 drinks for non-recruited visitors. While setup constraints prevented formal evaluation during the first three events, the final deployment included a 360° camera setup to evaluate performance. Of the recorded pouring attempts, 61 were successful and one failed in a cluttered scene. Two additional pours violated system assumptions, where glasses were unexpectedly moved mid-pour, highlighting the limitation of assuming static placement. Excluding these, the effective success rate was 98.3%. These results align with our controlled experiments and further confirm SPILL's robustness in real-world deployment.

V. Discussion

These results confirm SPILL's capability to operate in unstructured real-world conditions with high reliability, even in multi-glass scenarios with occlusion, clutter, and visual complexity. However, they also reveal the remaining challenges, and how future work might address them.

Most critical failures (4.2%) were due to pose estimation errors from heavy occlusion, resulting in misaligned pours. These cases showed unstable, inconsistent perception across frames, suggesting detectability. Future systems could flag low-confidence predictions via temporal checks and resolve ambiguity by adjusting the viewpoint, repositioning closely spaced glasses, or ignoring occluded glasses altogether.

These insights point to a key advantage of our approach: modularity and interpretability. Rather than relying on opaque end-to-end pipelines, our method provides explicit geometric cues (e.g., liquid level), which can be reasoned about and manipulated by higher-level agents.

Fig. 6. The robot deployed in public events, autonomously detecting glasses and serving juice. Across multiple events, it successfully served over 200 drinks with 98.3% success in a 62-pour session.

In our view, systems like SPILL can complement end-to-end learning from raw pixels. Intermediate, interpretable representations, such as SPILL's compact semantic keypoints for glass geometry and liquid level, can provide useful structure for downstream reasoning, improving generalization and reducing required data, a direction explored by recent work such as KALM [35], a promising path forward.

While our implementation of SPILL focuses on tabletop pouring, its modular design, interpretable keypoints, and scene-aware geometric reasoning, generalize naturally to a broader range of tasks like transferring liquids, checking fill levels before grasping, or deciding which glass to fill based on partial observation.

VI. Conclusion and Future Work

We present SPILL, a lightweight perception pipeline for Size, Pose, and Internal Liquid Level estimation of unknown transparent glassware using a single RGB-D image. Leveraging semantic keypoint detection, SPILL avoids object-specific models or depth completion, effectively addressing the challenges posed by transparent objects.

SPILL achieved a 93.6% success rate across extensive experiments, demonstrating reliable performance in cluttered, unstructured environments with diverse glass types and scenes. Real-world deployments further validated this, achieving a 98.3% success rate during public demonstrations, highlighting its readiness for deployment in service robotics.

SPILL's modular, interpretable design offers key advantages: It supports integration into broader robotic systems and enables explicit reasoning over object geometry and fill level. We believe such compact representations could complement end-to-end learning from pixels to increase generalization.

Despite generalizing across diverse glass types and scenes, SPILL has limitations: it does not yet support containers such as stemmed glasses, assumes glasses remain stationary, and lacks awareness of human presence. Liquid-level estimation remains coarse; future work could address this through richer training data, surface-level keypoints, and geometric models for curved vessels. Furthermore, integrating SPILL with human-robot interaction pipelines could enable safer and more intuitive collaboration.

Overall, SPILL advances transparent object perception by enabling single-view 3D estimation without object-specific models, demonstrating reliable and generalizable performance for autonomous manipulation in service settings.

REFERENCES

[1] J. Jiang, G. Cao, J. Deng, T.-T. Do, and S. Luo, "Robotic perception of transparent objects: A review," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 6, pp. 2547–2567, 2024.

[2] T. Tang, J. Liu, J. Zhang, H. Fu, W. Xu, and C. Lu, "RFTrans: Leveraging Refractive Flow of Transparent Objects for Surface Normal Estimation and Manipulation," *IEEE Robotics and Automation Letters*, vol. 9, no. 4, pp. 3735–3742, 2024.

[3] J. Lee, S. M. Kim, Y. Lee, and Y. M. Kim, "NFL: Normal Field Learning for 6-DoF Grasping of Transparent Objects," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 819–826, 2024.

[4] A. Ummadisingu, J. Choi, K. Yamane, S. Masuda, N. Fukaya, and K. Takahashi, "Said-nerf: Segmentation-aided nerf for depth completion of transparent objects," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 7535–7542.

[5] B. P. Duisterhof, Y. Mao, S. H. Teng, and J. Ichnowski, "Residual-NeRF: Learning Residual NeRFs for Transparent Object Manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 13 918–13 924.

[6] H. Fang, H.-S. Fang, S. Xu, and C. Lu, "TransCG: A Large-Scale Real-World Dataset for Transparent Object Depth Completion and a Grasping Baseline," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7383–7390, 2022.

[7] X. Chen, Z. Zhou, Z. Deng, O. Ghasemalizadeh, M. Sun, C.-H. Kuo, and A. Sen, "Tabletop Transparent Scene Reconstruction via Epipolar-Guided Optical Flow with Monocular Depth Completion Prior," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, 2023, pp. 1–8.

[8] Y. Zhou, W. Peng, Z. Yang, H. Liu, and Y. Sun, "Transparent object depth completion," 2024, arXiv:2405.15299.

[9] C. Guo, H. Zhang, Y. Jiang, B. Chen, W. Chen, and L. Yang, "TCG: Transparent Object Grasping Method Based on Depth Completion in Dense Clutter," in *International Conference on Advanced Robotics and Mechatronics (ICARM)*, 2024, pp. 586–592.

[10] Z. Zhou, Z. Sui, and O. C. Jenkins, "Plenoptic Monte Carlo Object Localization for Robot Grasping Under Layered Translucency," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–8.

[11] H. Yu, S. Li, H. Liu, C. Xia, W. Ding, and B. Liang, "TGF-Net: Sim2Real Transparent Object 6D Pose Estimation Based on Geometric Fusion," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3868–3875, 2023.

[12] M. Byambaa, G. Koutaki, and L. Choimaa, "6D Pose Estimation of Transparent Object From Single RGB Image for Robotic Manipulation," *IEEE Access*, vol. 10, pp. 114 897–114 906, 2022.

[13] I. Lysenkov, V. Eruhimov, and G. Bradski, "Recognition and Pose Estimation of Rigid Transparent Objects with a Kinect Sensor," in *Robotics: Science and Systems VIII*, 2013, pp. 273–280.

[14] X. Chen, H. Zhang, Z. Yu, A. Opipari, and O. C. Jenkins, "Clearpose: Large-scale transparent object dataset and benchmark," in *European Conference on Computer Vision*, 2022.

[15] G. Jocher, J. Qiu, and A. Chaurasia, "Ultralytics YOLO," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[16] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik, and Y. Li, "MaxViT: Multi-Axis Vision Transformer," 2022, arXiv:2204.01697.

[17] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," 2015, arXiv:1505.04597.

[18] I. Lysenkov and V. Rabaud, "Pose estimation of rigid transparent objects in transparent clutter," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 162–169, iSSN: 1050-4729.

[19] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity Invariant CNNs," in *2017 International Conference on 3D Vision (3DV)*, 2017, pp. 11–20, iSSN: 2475-7888.

[20] S. Sajjan, M. Moore, M. Pan, G. Nagaraja, J. Lee, A. Zeng, and S. Song, "Clear grasp: 3d shape estimation of transparent objects for manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3634–3642.

[21] T. Li, Z. Chen, H. Liu, and C. Wang, "FDCT: Fast Depth Completion for Transparent Objects," *IEEE Robotics and Automation Letters*, vol. 8, no. 9, pp. 5823–5830, 2023.

[22] H. Zhang, A. Opipari, X. Chen, J. Zhu, Z. Yu, and O. C. Jenkins, "TransNet: Category-Level Transparent Object Pose Estimation," 2022, arXiv:2208.10002.

[23] X. Liu, R. Jonschkowski, A. Angelova, and K. Konolige, "KeyPose: Multi-View 3D Labeling and Keypoint Estimation for Transparent Objects," 2020, arXiv:1912.02805.

[24] J. Chang, M. Kim, S. Kang, H. Han, S. Hong, K. Jang, and S. Kang, "GhostPose: Multi-view Pose Estimation of Transparent Objects for Robot Hand Grasping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 5749–5755.

[25] K. Bai, H. Zeng, L. Zhang, Y. Liu, H. Xu, Z. Chen, and J. Zhang, "ClearDepth: Enhanced Stereo Perception of Transparent Objects for Robotic Manipulation," 2024, arXiv:2409.08926.

[26] L. Manuelli, W. Gao, P. Florence, and R. Tedrake, "kPAM: KeyPoint Affordances for Category-Level Robotic Manipulation," 2019, arXiv:1903.06684.

[27] Z. Qin, K. Fang, Y. Zhu, L. Fei-Fei, and S. Savarese, "KETO: Learning Keypoint Representations for Tool Manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 7278–7285.

[28] A. Yamaguchi and C. G. Atkeson, "Stereo vision of liquid and particle flow for robot pouring," in *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 1173–1180.

[29] C. Do and W. Burgard, "Accurate Pouring with an Autonomous Robot Using an RGB-D Camera," in *Intelligent Autonomous Systems 15*, 2019, pp. 210–221.

[30] C. Schenck and D. Fox, "Perceiving and reasoning about liquids using fully convolutional networks," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 452–471, 2018.

[31] G. Narasimhan, K. Zhang, B. Eisner, X. Lin, and D. Held, "Self-supervised Transparent Liquid Segmentation for Robotic Pouring," in *International Conference on Robotics and Automation (ICRA)*, 2022, pp. 4555–4561.

[32] Y. Huang, J. Zhang, R. Yu, S. Li, and W. Ding, "SimLiquid: A Simulation-Based Liquid Perception Pipeline for Robot Liquid Manipulation," *Journal of Field Robotics*, 2025.

[33] F. Zhu, R. Jia, L. Yang, Y. Yan, Z. Wang, J. Pan, and W. Wang, "Visual-Tactile Sensing for Real-time Liquid Volume Estimation in Grasping," 2022, arXiv:2202.11503.

[34] T. Lips, V.-L. De Gusseme, and F. wyffels, "Learning Keypoints for Robotic Cloth Manipulation Using Synthetic Data," *IEEE Robotics and Automation Letters*, vol. 9, no. 7, pp. 6528–6535, 2024.

[35] X. Fang, B.-R. Huang, J. Mao, J. Shone, J. B. Tenenbaum, T. Lozano-Pérez, and L. P. Kaelbling, "Keypoint abstraction using large models for object-relative imitation learning," *arXiv preprint arXiv:2410.23254*, 2024.