Transparent Liquid Segmentation for Robotic Pouring

Anonymous Author(s)
Affiliation
Address
email

Abstract

Liquid state estimation is important for robotics tasks such as pouring; however, estimating the state of transparent liquids is a challenging problem. We propose a novel segmentation pipeline that can segment transparent liquids such as water from a static, RGB image without requiring any manual annotations or heating of the liquid for training. Instead, we use a generative model that is capable of translating unpaired images of colored liquids into synthetically generated transparent liquid images. Segmentation labels of colored liquids are obtained automatically using background subtraction. We use paired samples of synthetically generated transparent liquid images and background subtraction for our segmentation pipeline. Our experiments show that we are able to accurately predict a segmentation mask for transparent liquids without requiring any manual annotations. We demonstrate the utility of transparent liquid segmentation in a robotic pouring task that controls pouring by perceiving liquid height in a transparent cup. Accompanying video and supplementary information can be found at https://sites.google.com/view/roboticliquidpouring

1 Introduction

Robots that could pour liquids would enable us to automate tasks such as cooking, pouring medicines into vials in pharmacies, or watering our plants. However, transparent liquids are difficult to perceive in images; the only visual signals a perfectly transparent liquid can provide are the refraction of light passing through the liquid. Obtaining depth measurements for liquids is similarly difficult since the liquid will refract the projected infrared light.

Previous works have explored robotic pouring in various settings [4, 5, 9, 10, 16], but all require significant compromises in the environment or method for data collection. For example, several methods for transparent liquid segmentation require heating up the liquid during training to obtain ground-truth labels when viewed by a thermal camera [19, 21, 20]; however, heating up the liquid for training is a tedious process that limits how much training data can be easily collected. Other approaches require observing the liquid from multiple viewpoints [4], checkerboard backgrounds [10], weight measurements [10], or liquid motion [29, 25, 12, 3]; these requirements on the environment restrict the applicability of these methods.

In this work, we propose a method for perceiving transparent liquid (such as water) inside transparent containers. Our method requires significantly fewer restrictions on the operational domain than previous methods. Specifically, our method operates on individual images (we do not require liquid motion or multiple frames), and requires no manual annotations or heating of liquids during training. To accomplish this, our method uses a generative model that learns to translate images of colored liquid into synthetically generated images of transparent liquid. Because images of colored liquid are easy to segment, this automatically provides us with ground-truth labels for the synthetically generated

Submitted to NeurIPS 2021 Workshop on DGMs and Downstream Applications. Do not distribute.

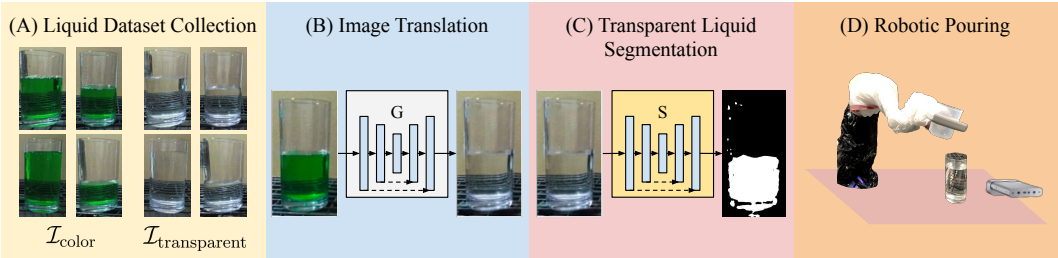


Figure 1: We introduce a simple method for learning to segment transparent liquids in transparent containers. Our approach consists of four steps: (A) collect two datasets (unpaired) of colored and transparent liquid in containers; (B) create synthetic segmentation labels for transparent liquids using image translation (C) train a transparent liquid segmentation model using the generated labels; (D) closed-loop robotic pouring of a specific amount of liquid using our transparent liquid segmentation.

images of transparent liquid. Finally, we demonstrate the utility of our method for transparent liquid segmentation on a robot pouring task.

2 Related Work

2.0.1 Transparent object perception

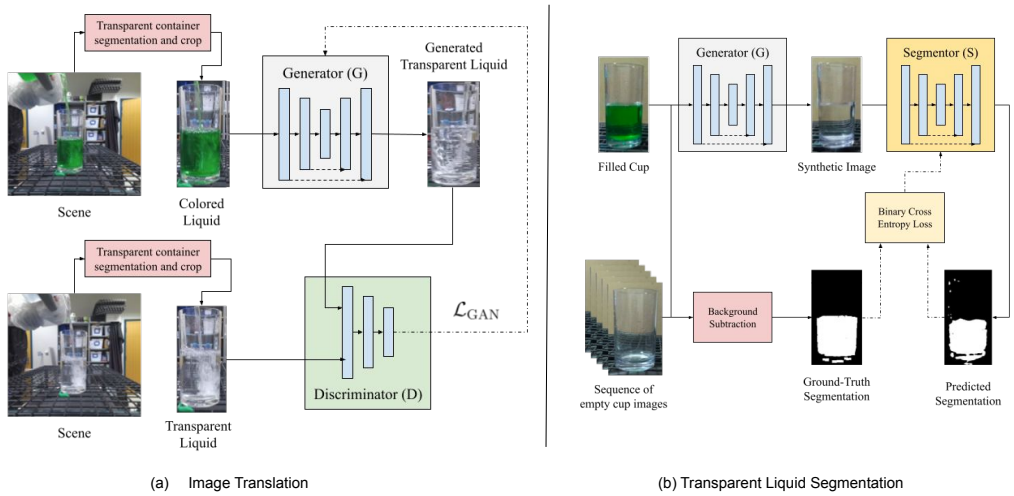
Perceiving transparent objects is particularly challenging because transparent objects can refract, reflect, and absorb light. Some previous works focus on perceiving transparent containers; methods have been developed for transparent object segmentation [26, 13], depth estimation [18, 30], key-point estimation [15], and transparent object matting [2]. Other methods for segmenting transparent objects use light field cameras [27, 28] or light polarization [8]. On the manipulation side, other recent works have been developed to directly grasp transparent objects without first estimating their 3D shape [24]. Our approach builds on [26] for transparent container segmentation; however, our focus is on segmenting the transparent liquid inside the container. Unlike previous work that uses manual annotations for training [26], our work does not rely on any manual annotations.

2.0.2 Transparent liquid perception

Perception of liquid is more challenging than perception of object due to the lack of a fixed shape or geometry. While perception of colored liquid can sometimes be done using background subtraction [9], it does not work for transparent liquid. One approach to transparent liquid perception is to use heated liquid observed by a thermal camera to obtain ground-truth labels for liquid [19, 21, 20]. However, the requirement to heat the liquid before recording the ground-truth is a tedious process; our method does not require heated liquid. To segment liquid while it is being poured, one can use optical flow [29] or audio signals [25, 12]. Our method can segment static liquid, which is important for liquid state estimation before initiating a pouring task. Some methods reason about the refraction of the infrared light emitted by a depth sensor [4, 3], multiple noisy readings from different viewpoints [4], or from different time points during pouring [3], integrated probabilistically. In contrast, our method can segment the liquid from just a single RGB image. Another approach is to use a depth sensor to estimate the height of the liquid surface [5]; however, such depth readings are inaccurate for transparent liquids. A different strategy is to pour liquid in front of a checkerboard background or to use weight readings from a scale [10]. Our method does not require a checkerboard background or a scale. Finally, some approaches forgo a separate module for transparent liquid perception and learn an end-to-end policy for pouring transparent liquid [14]. However, so far such approaches have only been shown to work in simulation due to the sample complexity of learning a sensori-motor policy.

3 Method

We describe our method for transparent liquid segmentation when liquids are placed within transparent containers (see Figure 2 for an overview). First, we collect a dataset of colored liquid and another



(a) Image Translation (b) Transparent Liquid Segmentation

Figure 2: Our vision pipeline for training a segmentation network that can be used to segment transparent liquids. In (a), we use the losses described in Section 3.1 to train a generator G which transforms images of colored liquids in $\mathcal{D}_{\text{color}}$ to look like images in $\mathcal{D}_{\text{transparent}}$. In (b), we use the generator G trained via CUT to translate images in $\mathcal{D}_{\text{color}}$ into transparent images, and generate masks M_i via the background subtraction method described in the supplementary material. Finally, we train a segmentation model S on this synthetic supervised dataset using a standard binary cross-entropy loss.

(unpaired) dataset of transparent liquid. We then use an image translation method to learn to translate an image of colored liquid into a synthetically generated image of transparent liquid that is identical to the input image, except that the liquid is now transparent. Next, we use background subtraction to find the colored liquid pixels in the colored liquid dataset. We treat the colored liquid segmentation as a ground-truth label for the synthetically generated transparent liquid. We then train a network to segment transparent liquid, using paired samples of the synthetically generated transparent liquid and colored liquid ground-truth labels.

3.1 Learning to translate colored liquid to transparent liquid

To train a model for transparent liquid segmentation, we ideally want a dataset of labeled images of transparent liquids. However, labeling such a dataset is tedious. Instead, we make use of an image translation method to synthetically generate the desired labels.

We collect one dataset of colored liquids in transparent containers $\mathcal{D}_{\text{color}}$ and a second dataset of transparent liquids in transparent containers $\mathcal{D}_{\text{transparent}}$. Given these two datasets, we learn an image translation model from colored to transparent liquids. To do so, we use Contrastive Unpaired Translation (CUT) [17], which we train to convert an image of a colored liquid into an image of a transparent liquid.

We briefly describe CUT and how we adapt it for our method. The backbone of CUT is a generator that translates an image of the source domain into an image of the target domain. To encourage this translation, three loss terms are used. First, a standard adversarial GAN loss is used to encourage the generator to output images that are visually similar to those in the target domain:

$$\mathcal{L}_{\text{GAN}} = \mathbb{E}_{y\sim Y}\left[\log D(y)\right] + \mathbb{E}_{x\sim X}\left[\log(1 - D(G(x)))\right] \tag{1}$$

where G, D, denote the generator and the discriminator respectively, and X, Y denote the source domain and the target domain, respectively. The generator G is divided into an encoder G_{enc} and a decoder G_{dec}, such that the output is $\hat{y} = G(x) = G_{\text{dec}}(G_{\text{enc}}(x))$, for an image x from the source domain.

Additionally, CUT uses a patch-wise contrastive loss [22] to encourage corresponding patches between the input and output images to be similar to each other in feature space. Specifically, given an image x from the source domain X, the image is translated into an image y of the target domain Y.

The patch-wise contrastive loss, $\mathcal{L}_{PatchNCE}(G, H, X)$, maximizes the mutual information between $H(G_{\text{enc}}(x))$ and $H(G_{\text{enc}}(y))$, where H is a small multi-layer perceptron (MLP). The generator is also trained with an identity loss $\mathcal{L}_{PatchNCE}(G, H, Y)$ to help regularize the encoder and minimize unnecessary modifications to a source image. The combined loss is:

$$\mathcal{L}_{\text{CUT}} = \mathcal{L}_{\text{GAN}} + \lambda_X \mathcal{L}_{PatchNCE}(G, H, X) + \lambda_Y \mathcal{L}_{PatchNCE}(G, H, Y) \tag{2}$$

We apply CUT directly to our two datasets of raw images, where our source domain $X = \mathcal{D}_{\text{color}}$ and target domain $Y = \mathcal{D}_{\text{transparent}}$. Thus, we use CUT to convert an image of colored liquid $x_{\text{color}} \in \mathcal{D}_{\text{color}}$ to a synthetically generated image of a transparent liquid $\hat{y}_{\text{transparent}} = G(x_{\text{color}})$.

Importantly, the patch-wise contrastive loss encourages the object parts in the image x_{color} to be in the same location as the parts in the translated image $\hat{y}_{\text{transparent}}$. For example, in Figure 2, the cup, liquid, surface, and even shadows are in the same locations between the image of colored liquid x_{color} that is input to the generator G, compared to the synthetic image of transparent liquid $\hat{y}_{\text{transparent}}$ that is output by the generator. The primary difference between these images is that the colored liquid has changed to become transparent; the liquid is in the same location as in the input.

This property (that the liquid is in the same location in the generator input x_{color} as in the output $\hat{y}_{\text{transparent}}$) is crucial to the success of our proposed segmentation method. If we assume that the only property that has changed as a result of the translation is the liquid color, then we can directly use the segmentation masks from the colored liquid x_{color} as pseudo-ground truth for the generated transparent liquid $\hat{y}_{\text{transparent}}$. Because it is simple to segment an image of colored liquid using color thresholding or background subtraction (see supplementary materials for details), we can then easily generate segmentation labels for the synthetic transparent images $\hat{y}_{\text{transparent}}$ without requiring human annotations. Formally, given an image of colored liquid $x^{(i)}_{\text{color}}$ with corresponding segmentation mask $M^{(i)}$, we generate a synthetic image of transparent liquid $\hat{y}^{(i)}_{\text{transparent}} = G(x^{(i)}_{\text{color}})$ to which we associate the colored image's segmentation mask M_i as a pseudo-ground truth segmentation label. This creates a synthetic labeled dataset: $\{\hat{y}^{(i)}_{\text{transparent}}, M^{(i)}\}$ which we use to train our transparent liquid segmentation model, as described below.

3.2 Learning transparent liquid segmentation

We can use the aforementioned synthetic labeled dataset to train a transparent liquid segmentation model S using the Binary Cross Entropy loss between the predicted liquid segmentation mask and the pseudo-ground truth M_i (described above). Architectural, hyper-parameter, and implementation details are described in the supplementary materials. Our image translation and segmentation modules are trained on one Nvidia 2080 Ti GPU.

3.3 Robot liquid pouring

In this section, we describe the details of the robotic pouring system we designed to demonstrate the utility of our transparent liquid segmentation model in pouring tasks. While other works have explored sophisticated and flexible robotic pouring methods, we emphasize that our robotic system design is a simple testbed for our perception. Our system consists of two stages: a visual postprocessing stage that converts a liquid segmentation into an estimate of a scalar fill level in a container $\hat{l}$ and a pouring controller which drives the system to reach a target fill level l_{target}. See Figure 3.

3.3.1 Task Design

For our robotic pouring experiments, we use a Franka-Emika Panda 7-DOF robotic arm, with a custom-built end-effector that rigidly secures a source container for pouring [1].

We place a Microsoft Azure Kinect RGB camera directly in front of the robot's workspace. We place the target cup directly in front of the camera at a known location. We then drive the end-effector to a known location directly above the target cup. We ensure that the source cup in the robot's end-effector is not included in the image recorded by the Kinect camera. We fill the source cup with liquid, which will be poured into the target cup.

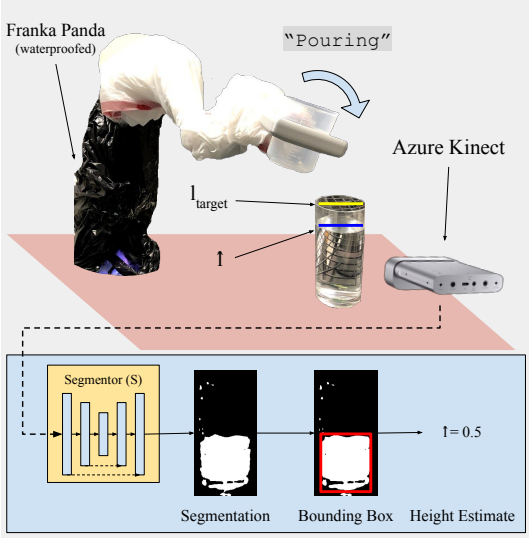


Figure 3: Our robotic pouring system. For each RGB image captured, we use our learned segmentation model S to output a segmentation of the liquid in the cup, and detect the bounding box of the cup using TransLab [26]. We compute a liquid level estimate $\hat{l}$, and pour at a fixed angle until the perceived liquid level is within a small threshold.

3.3.2 Visual postprocessing

To minimize any potential errors caused by the segmentation model, we first use an off-the-shelf method TransLab [26] model to detect the location of the cup in the scene. Next, we crop the image region around the detected cup location. Then, we use our segmentation model to segment the transparent liquid pixels from within this crop. Finally, we perform a filtering step that removes noise as well as small water particles in motion (i.e. during pouring). The remaining segmented points are from liquid inside the target cup.

It is difficult to design a control system for pouring that operates directly on a current and target segmentation mask. Instead, we define a target fill height h_{target} from the camera perspective; we then define the process variable as the fraction of the cup that is filled: $l_{\text{target}} = \frac{h}{h_{\text{cup}}}$, where h is the liquid height and h_{cup} is the cup height (as seen from the camera perspective). This task was inspired by the task of a robot waiter, which must refill everyone's cups to a certain fraction of the cup height. Further, keeping the process variable grounded in 2D visual features avoids complexities of estimating the 3D geometry of the scene.

To estimate the current fill level $\hat{h}$, we compute a vertically-aligned bounding box of the segmented pixels. We then use the upper edge of the box as the estimated height $\hat{h}$ of the liquid. As mentioned previously, we use previous work to segment the transparent cup in the scene [26]; we use a similar approach as above to fit a bounding box and find the height of the cup h_{cup}. From these computations, we can estimate the process variable $\hat{l} = \frac{\hat{h}}{h_{\text{cup}}}$.

3.3.3 Control

With the robot arm positioned directly above the cup, we restrict the control space to be one of two states: $\{\texttt{NotPouring}, \texttt{Pouring}\}$. The `NotPouring` state corresponds to the starting postion, where the source cup is completely vertical (upright). The `Pouring` state corresponds to a 60 rotation about the central axis of the cup (parallel to the table). When the control signal changes between the two states, the end-effector rotates at a constant angular velocity until the desired control state is reached. The system initially commands `Pouring` until the estimated height is within a margin ϵ of the target; afterwards, the system commands `NotPouring`.

Because the vision system is a simple neural network with simple post-processing, it is very low latency. Therefore, we can operate our control loop at roughly 10Hz. Because of the responsive

system, bang-bang control of the pouring into the target cup is effective. Finally, to compensate for some errors that occur in the perception system when the target cup is empty, we always begin the control with roughly 1s of initial pouring.

Further details about our data capture and background subtraction can be found in the supplementary materials.

3.4 Diverse Backgrounds

Previous work has achieved background generalization for object segmentation by synthetically pasting a foreground segment on top of a random background scene [11, 6, 7]. However, this technique is infeasible for learning to segment transparent liquids inside of transparent containers, since the container and liquid will refract the background, which cannot be easily imitated synthetically. Instead, to generate a diverse set images with physically-accurate refraction characteristics, we set up a large flatscreen LED display behind the pouring scene. We then played videos containing a diverse set of indoor and outdoor scenes during the data collection procedure. This allowed us to create datasets with a high degree of background diversity, with natural patterns of light refracting through water. While these procedures could be included as part of the regular data collection procedure, for simplicity, they were only performed for the results in Section 4.3 (Ablation 3).

4 Results

4.1 Dataset Description

To train our method, we collected 4 distinct pouring videos each of green-colored water and clear water. This resulted in datasets with 2231 and 2237 RGB frames, respectively. We also collected a test set of 133 images of transparent liquids in the same scene.

4.2 Segmentation performance

To evaluate the segmentation performance of the method for transparent liquids, we create a test set of 65 images of transparent containers that have varying amounts of transparent liquids (water) in them, placed at different distances from the camera. We manually annotate the location of the transparent liquids for the test set using [23]; however, we do not use such annotations for training.

Method	Low	Medium	High	All
Ours	0.56	0.78	0.84	0.72
Supervised (10%)	0.61	0.91	0.86	0.79
Supervised (1%)	0.52	0.54	0.38	0.50
Ours (10%)	0.56	0.80	0.78	0.71
Ours (1%)	0.38	0.60	0.53	0.51
Opaque Dataset	0.02	0.04	0.06	0.03

Table 1: Average Intersection over Union (IoU) scores on a test set of transparent liquid images, each filled with water to varying amounts. We show the performance for subsets of images with varying amounts of liquid in the cup (Low, Medium, and High) as well as an average over all images.

Our results for transparent liquid segmentation can be found in Table 1 as well as in Fig 4 (bottom row). Our method generally succeeds at segmenting the liquid pixels in the image, achieving high IoU scores across the test set. However, our method experiences a small drop in performance on images in which the cup is filled to a high level (i.e. right-most image in "Ours" in Figure 4). We hypothesize that such examples have regions of liquid that are distant from a liquid-cup or liquid-air boundary; thus these cases are harder to classify when there are no refractive patterns to indicate the presence of liquid.

4.3 Pouring Performance

To assess the utility and effectiveness of transparent liquid segmentation in our real robotic pouring system, we conduct a series of pouring trials. We choose four different initial fill-levels and target

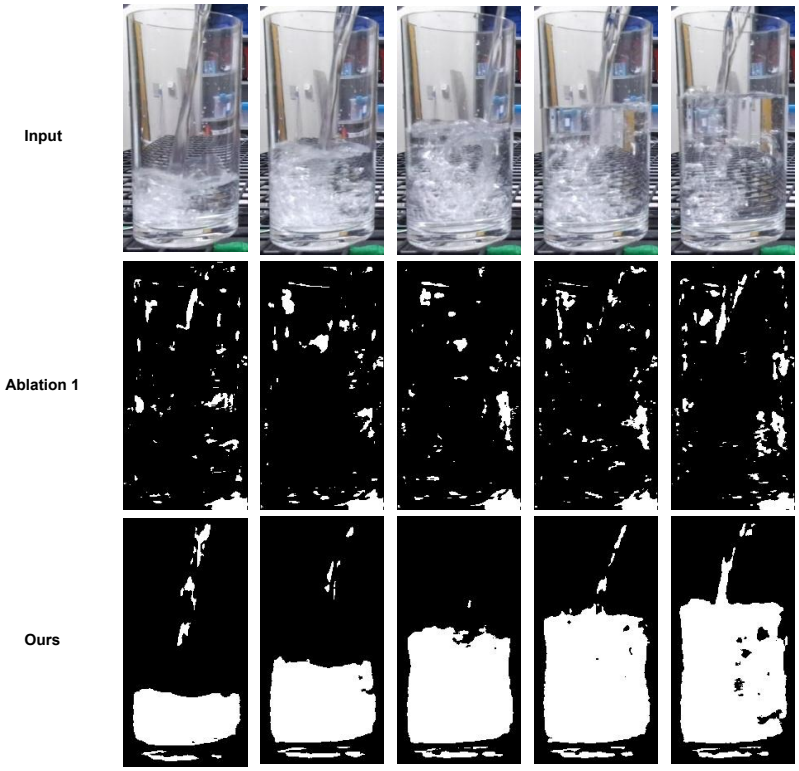


Figure 4: Representative segmentations of our method compared to a model trained on images of colored liquid with segmentation labels obtained through background subtraction ("Ablation 1"), which is unable to accurately segment the liquid.

	Transparent Liquid Ours (RMSE)	Colored Liquid Background Subtraction (RMSE)
$0\% \rightarrow 25\%$	$1.00 \pm 0.43\%$	$8.46 \pm 2.14\%$
$0\% \rightarrow 50\%$	$0.82 \pm 0.67\%$	$1.86 \pm 0.71\%$
$0\% \rightarrow 75\%$	$1.18 \pm 0.74\%$	$1.57 \pm 0.50\%$
$25\% \rightarrow 75\%$	$0.75 \pm 0.49\%$	$1.61 \pm 0.65\%$
All	$0.94 \pm 0.61\%$	$3.38 \pm 3.18\%$

Table 2: Percent error of the pouring system on both transparent and colored liquids. For transparent liquid pouring, we use our learned segmentation model; for colored liquid pouring, we use background subtraction. Contrary to expectations, our system performs better with transparent segmentation than with colored segmentation.

fill-levels (see Table 2), and conduct 20 pouring trials for each fill-level. We measure the level of liquid achieved upon the controller reaching the final `NotPouring` state, and report the average error across each fill-level, as well as across all trials.

We conduct two sets of trials: first with transparent water, using our transparent liquid segmentation system; second with green-colored water, using the background-subtraction method described in the supplementary materials. Given the high accuracy of background subtraction, we consider this second task to represent how the system would perform with near-perfect segmentation. We opted not to evaluate using segmentation methods with poor performance, due to the high risk of spilling liquid in the workspace. Results can be found in Table 2.

In the case of pouring transparent liquids, we are able to achieve the desired ratio l_{target} with an average accuracy of 0.94%, which corresponds to a roughly 0.13cm error on average. Surprisingly, in the case of colored liquid (where background subtraction yields high-fidelity segmentation), accuracy

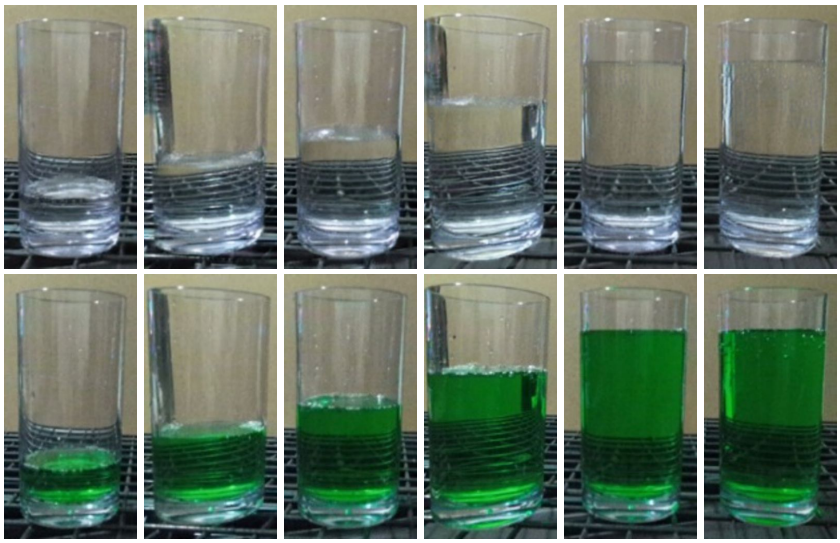


Figure 5: Image translation from colored liquid to transparent liquid by our trained model; **Top Row:** Real world colored liquid images from $\mathcal{D}_{\text{color}}$; **Bottom Row:** Generated transparent liquid images

is worse, with an average error of 3.38% or 0.47cm error. However, when observing the system, we noticed that the bounding box computation is sensitive to segmentations with splashing/sloshing: in these cases, the bounding box overestimates the amount of liquid in the cup, and terminates pouring earlier than it should. Segmentations from our model pick up less of this transient liquid, and thus outperform the "ground-truth" segmentation. With a more sophisticated postprocessing step and controller, these effects could potentially be mitigated.

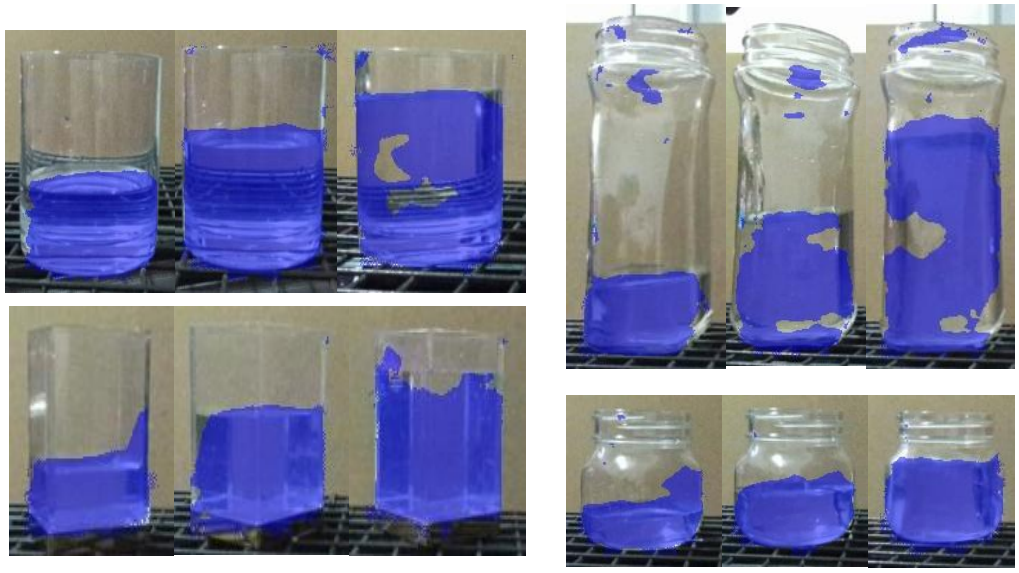
4.4 Visual Translation

We show representative examples of the image translation achieved by our CUT-based model in Fig 5. We observe that the network is able to translate the input image containing green-colored water pixels into images of clear water while still capturing the same background and refraction patterns as that in the source image. Most importantly, the transparent liquid in the synthetic images is in the same location as the colored liquid in the original images. This property allows us to apply the background mask from the colored liquids as the ground-truth label for the synthetic images of transparent liquids.

4.5 Visual Ablations

Ablation 1: What is the benefit of training the segmentation model on synthetically generated transparent (instead of colored) liquid? To answer this question, we train a segmentation model on colored liquid with color jitter and evaluate it on transparent liquid. We jitter the brightness, contrast and hue to obtain the input image for training the segmentation model. This ablation explores whether such color augmentation is sufficient to train a model for transparent liquid segmentation. Quantitative results of this ablation ("Opaque Dataset") are shown in Table 1. Qualitative results are shown in Fig 4. The model fails to detect the correct liquid height during our evaluation. This analysis shows that using color jitter on a colored liquid image is not sufficient domain randomization to capture the texture and patterns required to segment transparent liquids, supporting the idea that the segmentation model should be trained directly on images of transparent liquid.

Ablation 2: What is the benefit of the translation approach over using a small amount of manually-labeled data? Since our method uses a standard supervised loss function to train the model on synthetic data, a natural comparison is to evaluate performance when the model is trained on images of transparent images with manual segmentation annotations. Since annotation is very time-consuming ($\sim$1min per image), we annotate 10% of images in the training domain, and observe how well the model can learn to segment on two supervised subsets: 10% labeled and 1% labeled.



(a) Segmentation generalization to novel containers

(b) Segmentation generalization to novel backgrounds

Figure 6: Generalization of our transparent liquid segmentation model. In (a), we show that when our segmentation model is trained on one specific kind of cup in a scene (top row), it is able to generalize reasonably to unseen transparent cups in the same scene. In (b), we show that when our segmentation model is trained on a diverse set of background images, it is able to effectively generalize to novel backgrounds.

Additionally, we train our method on same fraction of images from our synthetic dataset. Results can be found in the second section of Table 1.

We draw two conclusions from these results. First, both our method and the fully-supervised ablation require substantially less data to reach their peak performance on the test set than is available: "Ours (10%)" shows the same performance as our full method ("Ours"), and "Supervised (10%)" outperforms our method. This is unsurprising, given that our test images are visually quite similar to the training set. Second, with only 1% of manually-annotated labels available, the supervised baseline ("Supervised (1%)") performs substantially worse than our full method. While our dataset is rather small – we were able to manually label 10% of the data – our data collection method is extremely scalable. In large datasets it is reasonable to expect one might only have access to manually-annotated labels for 1% of a full dataset. This provides evidence that our method would scale well to liquid datasets where a small amount of supervised data would not provide sufficient generalization.

Ablation 3: Does the approach work if there are diverse cups and backgrounds? To evaluate our segmentation model's capacity to generalize to diverse cups and backgrounds, we conduct two studies, shown in Figure 6. We first train a segmentation model on an image dataset consisting of a single cup in a single scene with varying heights of liquids. We then evaluate on transparent cups of different shapes with different fill levels of transparent liquid. As can be seen in Figure 6, the model generalizes reasonably to other cups, indicating that it has learned to detect relevant liquid features invariant a specific container. Next, we train a model on a cup in front of a diverse set of backgrounds and various water heights, with a dataset collected as described in Section 3.4. As can be seen in Figure 6, the model learns a segmentation function that is invariant to novel unseen backgrounds. These two experiments demonstrate the potential of our approach to scale to a general set of scenes, provided a sufficiently diverse dataset generated by our method.

5 Conclusion

In this paper, we propose a method to segment transparent liquid placed inside transparent containers using static RGB images. A generative model is used to translate colored liquids to transparent texture. We show that an encoder - decoder network can be used to predict the segmentation mask directly from RGB images of transparent liquids without using any additional input modalities. We use background subtraction on colored liquids to obtain the ground truth for training the segmentation model and we do not require any manual annotations. Our method shows good results for most test cases of transparent liquid segmentation. Finally, we demonstrate the utility of transparent liquid segmentation on a real robotic pouring task. We hope that our method paves the way for more flexible and robust perception of transparent liquids for robot pouring.

References

[1] Dan Bronstein, Kelvin Kang, Ben Kolligs, Jacqueline Liao, and Corinne Alini. Franka intelligent water pouring, 2021.

[2] Guanying Chen, Kai Han, and Kwan-Yee K Wong. Tom-net: Learning transparent object matting from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9233–9241, 2018.

[3] Chau Do and Wolfram Burgard. Accurate pouring with an autonomous robot using an rgb-d camera. In *International Conference on Intelligent Autonomous Systems*, pages 210–221. Springer, 2018.

[4] Chau Do, Tobias Schubert, and Wolfram Burgard. A probabilistic approach to liquid level detection in cups using an rgb-d camera. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2075–2080, 2016. doi: 10.1109/IROS.2016.7759326.

[5] Chenyu Dong, Masaru Takizawa, Shunsuke Kudoh, and Takashi Suehiro. Precision pouring into unknown containers by service robots. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5875–5882. IEEE, 2019.

[6] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 364–380, 2018.

[7] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. On the importance of visual context for data augmentation in scene understanding. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[8] Agastya Kalra, Vage Taamazyan, Supreeth Krishna Rao, Kartik Venkataraman, Ramesh Raskar, and Achuta Kadambi. Deep polarization cues for transparent object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8602–8611, 2020.

[9] Monroe Kennedy, Kendall Queen, Dinesh Thakur, Kostas Daniilidis, and Vijay Kumar. Precise dispensing of liquids using visual feedback. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1260–1266. IEEE, 2017.

[10] Monroe Kennedy, Karl Schmeckpeper, Dinesh Thakur, Chenfanfu Jiang, Vijay Kumar, and Kostas Daniilidis. Autonomous precision pouring from unknown containers. *IEEE Robotics and Automation Letters*, 4(3):2317–2324, 2019.

[11] Mate Kisantal, Zbigniew Wojna, Jakub Murawski, Jacek Naruniec, and Kyunghyun Cho. Augmentation for small object detection. *arXiv preprint arXiv:1902.07296*, 2019.

[12] Hongzhuo Liang, Shuang Li, Xiaojian Ma, Norman Hendrich, Timo Gerkmann, Fuchun Sun, and Jianwei Zhang. Making sense of audio vibration for liquid height estimation in robotic pouring. *arXiv preprint arXiv:1903.00650*, 2019.

[13] Jie Liao, Yanping Fu, Qingan Yan, and Chunxia Xiao. Transparent object segmentation from casually captured videos. *Computer Animation and Virtual Worlds*, 31(4-5):e1950, 2020.

[14] Xingyu Lin, Yufei Wang, Jake Olkin, and David Held. Softgym: Benchmarking deep reinforcement learning for deformable object manipulation. 2020.

[15] Xingyu Liu, Rico Jonschkowski, Anelia Angelova, and Kurt Konolige. Keypose: Multi-view 3d labeling and keypoint estimation for transparent objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11602–11610, 2020.

[16] Carolyn Matl, Yashraj Narang, Ruzena Bajcsy, Fabio Ramos, and Dieter Fox. Inferring the material properties of granular media for robotic tasks. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2770–2777. IEEE, 2020.

[17] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*, 2020.

[18] Shreeyak Sajjan, Matthew Moore, Mike Pan, Ganesh Nagaraja, Johnny Lee, Andy Zeng, and Shuran Song. Clear grasp: 3d shape estimation of transparent objects for manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3634–3642. IEEE, 2020.

[19] Connor Schenck and Dieter Fox. Towards learning to perceive and reason about liquids. In *International Symposium on Experimental Robotics*, pages 488–501. Springer, 2016.

[20] Connor Schenck and Dieter Fox. Reasoning about liquids via closed-loop simulation. *arXiv preprint arXiv:1703.01656*, 2017.

[21] Connor Schenck and Dieter Fox. Visual closed-loop control for pouring liquids. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2629–2636. IEEE, 2017.

[22] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019.

[23] Kentaro Wada. labelme: Image Polygonal Annotation with Python. `https://github.com/wkentaro/labelme`, 2016.

[24] Thomas Weng, Amith Pallankize, Yimin Tang, Oliver Kroemer, and David Held. Multi-modal transfer learning for grasping transparent and specular objects. *IEEE Robotics and Automation Letters*, 5(3):3791–3798, 2020.

[25] Justin Wilson, Auston Sterling, and Ming Lin. Analyzing liquid pouring sequences via audio-visual neural networks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2019*, pages 7702–7709, 11 2019. doi: 10.1109/IROS40897.2019.8968118.

[26] Enze Xie, Wenjia Wang, Wenhai Wang, Mingyu Ding, Chunhua Shen, and Ping Luo. Segmenting transparent objects in the wild. *arXiv preprint arXiv:2003.13948*, 2020.

[27] Yichao Xu, Hajime Nagahara, Atsushi Shimada, and Rin-ichiro Taniguchi. Transcut: Transparent object segmentation from a light-field image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3442–3450, 2015.

[28] Yichao Xu, Hajime Nagahara, Atsushi Shimada, and Rin-ichiro Taniguchi. Transcut2: Transparent object segmentation from a light-field image. *IEEE Transactions on Computational Imaging*, 5(3):465–477, 2019.

[29] Akihiko Yamaguchi and Christopher G Atkeson. Stereo vision of liquid and particle flow for robot pouring. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 1173–1180. IEEE, 2016.

[30] Luyang Zhu, Arsalan Mousavian, Yu Xiang, Hammad Mazhar, Jozef van Eenbergen, Shoubhik Debnath, and Dieter Fox. Rgb-d local implicit function for depth completion of transparent objects. *arXiv preprint arXiv:2104.00622*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] Please see Section 4
   (b) Did you describe the limitations of your work? [Yes] Please see Section 4.2
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The abstract contains a link to the supplementary material: `https://sites.google.com/view/roboticliquidpouring`
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please see Section 3, 4 for experimental details
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Please see 3.2
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] MIT License
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We have included a URL in the abstract of this paper.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] Only open source materials have been used
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] No personally identifiable information or offensive content present
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]