ClearGrasp: 3D Shape Estimation of Transparent Objects for Manipulation

Shreeyak S. Sajjan[1] Matthew Moore[1] Mike Pan[1] Ganesh Nagaraja[1]
Johnny Lee[2] Andy Zeng[2] Shuran Song[2,3]
[1] Synthesis.ai [2] Google [3] Columbia University

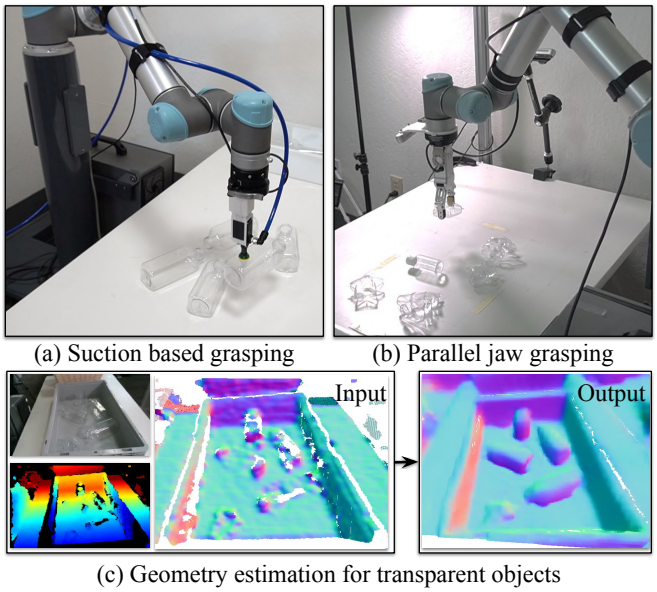
***Abstract*— Transparent objects are a common part of everyday life, yet they possess unique visual properties that make them incredibly difficult for standard 3D sensors to produce accurate depth estimates for. In many cases, they often appear as noisy or distorted approximations of the surfaces that lie behind them. To address these challenges, we present ClearGrasp – a deep learning approach for estimating accurate 3D geometry of transparent objects from a single RGB-D image for robotic manipulation. Given a single RGB-D image of transparent objects, ClearGrasp uses deep convolutional networks to infer surface normals, masks of transparent surfaces, and occlusion boundaries. It then uses these outputs to refine the initial depth estimates for all transparent surfaces in the scene. To train and test ClearGrasp, we construct a large-scale synthetic dataset of over 50,000 RGB-D images, as well as a real-world test benchmark with 286 RGB-D images of transparent objects and their ground truth geometries. The experiments demonstrate that ClearGrasp is substantially better than monocular depth estimation baselines and is capable of generalizing to real-world images and novel objects. We also demonstrate that ClearGrasp can be applied out-of-the-box to improve grasping algorithms' performance on transparent objects. Code, data, and benchmarks will be released. Supplementary materials: https://sites.google.com/view/cleargrasp**[1]

I. Introduction

Transparent objects are a common part of everyday life, from reading glasses to plastic bottles – yet they possess unique visual properties that make them incredibly difficult for machines to perceive and manipulate. In particular, transparent materials (which are both refractive and specular) do not adhere to the geometric light path assumptions made in classic stereo vision algorithms. This makes it challenging for standard 3D sensors to produce accurate depth estimates for transparent objects, which often appear as noisy or distorted approximations of the surfaces that lie behind them. Hence, while considerable research has been devoted to robotic manipulation of objects using 3D data (*e.g.* RGB-D images, point clouds) [49, 55, 34], many of these algorithms cannot be immediately applied to transparent objects – which remain critical for applications like dish washing or sorting/cleaning plastic containers.

In this work, we present **ClearGrasp**, an algorithm that leverages deep learning with synthetic training data to infer accurate 3D geometry of transparent objects for robotic manipulation. The design of ClearGrasp is driven by the following three key ideas:

- Commodity RGB-D cameras often provide good depth estimates for typical non-transparent surfaces. Therefore, rather than directly estimating all geometry from scratch, we conjecture that correcting initial depth estimates from RGB-D cameras is more practical: enabling us to use the depth from the non-transparent surfaces to inform the depth of transparent surfaces. For this to work reliably, we propose to predict pixel-wise masks of transparent surfaces (to detect and remove unreliable depth), as well as occlusion and contact edges between transparent surfaces and the background (to extend reliable depth).
- The refractive and specular patterns appearing on transparent objects provide stronger visual cues for their curvature (*e.g.* surface normals) than their absolute depth. This motivates using deep networks to infer surface normal information from RGB data, which we find to be substantially more reliable than directly inferring depth values.
- While real-world ground truth 3D training data for transparent objects is difficult to obtain, we show that it is possible use high-quality rendered synthetic images with domain randomization as training data to obtain reasonable



(a) Suction based grasping (b) Parallel jaw grasping

(c) Geometry estimation for transparent objects

Fig. 1. **ClearGrasp** leverages deep learning with synthetic training data to infer accurate 3D geometry of transparent objects from a single RGB-D image. The estimated geometry can be directly used for downstream robotic manipulation tasks (*e.g.* suction and parallel-jaw grasping).

[1]We would like to thank Ryan Hickman for managerial support, Ivan Krasin and Stefan Welker for fruitful technical discussions, Cameron (@camfoxmusic) for sharing 3D models of his potion bottles and Sharat Sajjan for helping on webpage design.

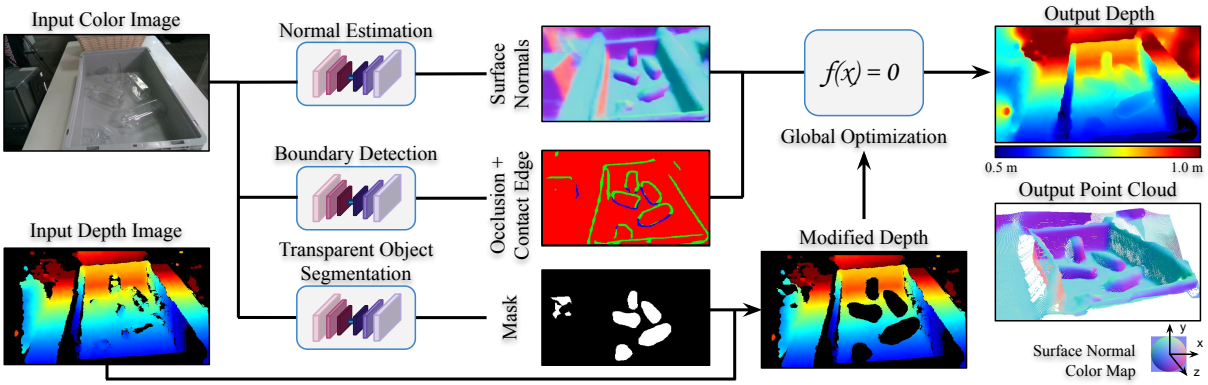


Fig. 2. **Overview.** Given an RGB-D image of a scene with transparent objects, ClearGrasp uses three networks to infer 1) surface normals, 2) masks of transparent surfaces, where depth is unreliable, and 3) occlusion and contact edges between the transparent surfaces and the rest of the scene. These outputs are then combined and used as input to a global optimization, which returns an adjusted depth map that corrects and completes the input depth.

results on real-world data. Interestingly, we also find that by mixing synthetic training data with real-world out-of-domain data (*e.g.* images without transparent objects), our model is able to generalize better to both real-world images and novel transparent objects unseen during training.

Our primary contributions are twofold. First, we propose an algorithm for estimating accurate 3D geometry of transparent objects from RGB-D images. Second, we construct a large-scale synthetic dataset of over 50,000 RGB-D images as well as a real-world test benchmark with 286 RGB-D images of transparent objects and their ground truth geometries. Our experiments demonstrate that ClearGrasp is capable of generalizing not only to transparent objects in the real-world, but also to novel objects unseen in training. ClearGrasp is substantially better than monocular depth estimation baselines, and our ablative studies show the importance of critical design decisions. We also demonstrate that ClearGrasp can be applied out-of-the-box with state-of-the-art manipulation algorithms to achieve 86% and 72% picking success rates with suction and parallel-jaw grasping respectively on a real-world robot platform. Code, data, pre-trained models, and benchmarks will be released.

II. Related Work

Estimating geometry from color images. Surface normal estimation is a popular problem tackled by deep convolutional networks [51, 13, 60, 57]. While predicted surface normals are useful for tasks like shading [23], 2D-3D alignment [3], and face morphing [28], it alone is insufficient to describe an object's complete 3D geometry, making it difficult to be directly used by manipulation algorithms that require 3D data (*e.g.* depth images, point clouds) [49, 55, 34]. More recent works study how to obtain 3D data from color images by directly inferring depth images [14, 30, 10, 41, 22, 52, 16], or filling in missing depth values in RGB-D images captured by commodity 3D cameras [21, 17, 59]. However, none of these works explicitly handle transparent objects, for which ground truth 3D data is very difficult to obtain – data from commodity 3D stereo cameras often have inaccurate or missing depth estimates for transparent surfaces.

Recognizing transparent objects. Transparent objects have plagued computer vision since the inception of the field. Due to their refractive and reflective nature, their appearance can vary drastically according to background and illumination conditions. Classic methods for detecting transparent objects mostly relied on idiosyncrasies such as specular reflections or local characteristics of edges due to refraction [38, 15, 36, 35]. Later methods rely on deep learning models like SSD [26] or RCNN [29] to predict bounding boxes enclosing transparent objects. Seib et al. [43] proposed a method to exploit sensor failures in depth images for transparent object localization using convolutional networks. Wang et al. [50] proposed localizing glass objects using a Markov Random Field to predict glass boundary and region jointly from multiple modalities from an RGB-D camera. Based on the localization, they recover depth readings by a piece-wise planar model. However, our method not only detects transparent objects, but also recovers detailed non-planar geometries, which are critical for manipulation algorithms.

Estimating geometry of transparent objects. Works on estimating transparent object geometry are often studied in a constrained environment: For example, the work might assume a specific capturing procedure [18, 24, 1], known background pattern [40, 19], sensor type [44] or known object 3D model [39, 32, 27]. Lysenkov et al. [33] propose a method for the recognition and pose estimation of rigid transparent objects using a Kinect sensor. Using a segmentation mask of the transparent objects, 3D models of objects created at the training stage are fitted to extracted edges. Our approach is able to generalize to objects not seen during training and does not require prior knowledge of the 3D model of the objects or camera position.

Learning from synthetic data. Synthetic data has proven to be useful in various tasks such as depth estimation [42], 3D semantic scene completion [45], hand pose estimation [12], robotic grasping [34], automatic shading of sketches [23], and person re-identification for tracking [4]. However, very few synthetic datasets support planar reflectors [48], let alone transparent objects. In our trials, we find that very high quality rendering and 3D models are required to

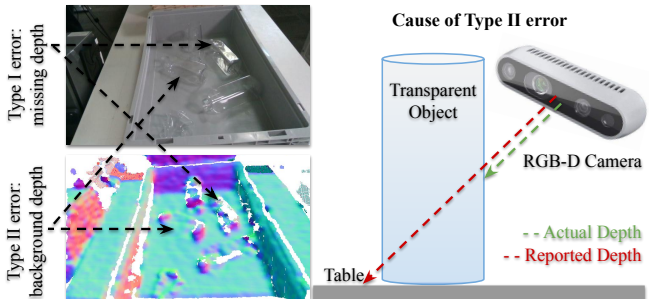


Fig. 3. **Errors in depth for transparent objects:** Type I errors, missing depth, is often caused by specular reflections on the surface. Type II errors, inaccurate depth estimates (returns background depth instead of the object depth), is caused by transparency of the surface material.

synthesize representative imagery of transparent objects and their related artifacts *e.g.* specular highlights and caustics. Datasets that do contain transparent objects have been used to study refractive flow estimation Chen et al. [8], semantic segmentation [47], or relative depth [46]. These datasets are generated in a simplified setting (*e.g.* rendered transparent objects in front of random images from COCO [31]). On the contrary, our method targets reconstructing detailed absolute depth of transparent objects within realistic environments.

III. Method

Given a single RGB-D image of transparent objects, ClearGrasp first uses the color image as input to deep convolutional networks to infer a set of information: surface normals, masks of transparent surfaces, and occlusion boundaries. ClearGrasp then uses this information and the initial depth image as input to a global optimization, which outputs a new depth image that refines the initial depth estimates from the sensor for all transparent surfaces in the scene (Sec. III-A). For training and testing, we construct a synthetic dataset and a real-world benchmark for transparent objects (Sec. III-B, III-C). In Sec. III-B, we demonstrate the application of ClearGrasp to a real-word robotic pick-and-place system.

A. Estimating 3D Geometry of Transparent Object

We adopt the depth completion pipeline proposed by Zhang and Funkhouser [59] with a few critical modifications to address the unique challenges presented by transparent objects. First, instead of only filling in the missing depth regions, we train an additional network to predict a pixel-wise mask for transparent surfaces and use it to remove unreliable depth measurements from the depth camera, see Fig. 3. Second, instead of predicting only occlusion edges (discontinuities in depth), we propose to predict both occlusion and contact edges (boundaries of objects in contact with other surfaces) so that the network can distinguish different type of edges and predict more accurate depth discontinuity boundaries, which is critical for the global optimization step, see Fig. 7. Fig. 2 shows an overview of our approach. The following paragraphs provide details on each module.

Transparent object segmentation. Due to the reflective and refractive nature of transparent objects, they cause erroneous readings in commodity RGB-D sensors. Fig. 3 explains 2 types of errors. Type I error refers to missing depth, commonly caused by specular highlights. Type II error occurs when the light is refracted through the transparent material, only to reflect back from the surface behind the object. This causes the sensor to report the depth of surfaces behind the object instead of the object itself. These inaccurate non-zero depth estimates are difficult to detect using standard depth completion, which would only propagate the inaccurate depth and result in corrupted reconstructions. To address this issue, we predict the pixel-wise masks of transparent objects using a Deeplabv3+ [9] with a DRN-D-54 backbone [53] to remove all depth pixels corresponding to transparent surfaces.

Surface normal estimation. This module predicts pixel-wise surface normals for the input RGB color image using Deeplabv3+ with DRN-D-54. The last convolutional layer is modified to have 3 output classes. To ensure that estimated normals are unit vectors, the output is L2 normalized.

Boundary detection. This module labels each pixel of the input color image as one of three classes: (a) Non-Edge, (b) Occlusion Boundary (depth discontinuity) (c) Contact Edges (points of contact between 2 objects). Contact edges, while not directly used by the optimization step, is very important because it helps the network better distinguish between different types of edges observed in color images, and therefore results in more accurate predictions of depth discontinuity boundaries. This significantly decreases chances of the model predicting a boundary around an entire object, which would prevent the global optimization step from solving back its depth using predicted surface normals. We use the same Deeplabv3+ model with a DRN-D-54 backbone. Since the pixel ratio of boundaries to background is low, we use a weighted cross-entropy loss with boundary pixels weighing 5x more than background pixels.

Global optimization for depth. Using the depth image (with all pixels corresponding to transparent surface removed) and predictions of surface normals and occlusion and contact edges, ClearGrasp reconstructs the 3D surfaces of transparent objects (missing depth region) via the global optimization algorithm proposed by Zhang and Funkhouser [59]. The optimization algorithm fills in the removed depth using the predicted normals to guide the shape of the reconstruction, while observing the depth discontinuities indicated by the occlusion boundaries. It solves a system of equations with the goal of minimizing the weighted sum of squared errors of three terms: $E = \lambda_D E_D + \lambda_S E_S + \lambda_N E_N B$, where E_D measures the distance between the estimated depth and the observed raw depth, E_S measures difference between the depths of neighboring pixels and E_N measures the consistency between estimated depth and predicted surface normal. B down-weights the normal terms based on the predicted probability that a pixel is on an occlusion boundary. In our experiments: $\lambda_D = 1000$, $\lambda_S = 0.001$ and $\lambda_N = 1.0$.

B. Synthetic Training data generation

We selected Synthesis AI's platform to generate our synthetic data, using Blender's physics engine[6], as well as the

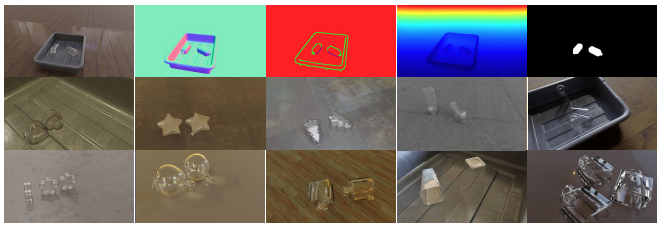


Fig. 4. **Synthetic data.** Top row is the rendered image and its groundtruth (surface normal, boundary, depth and mask). Bottom two rows are rendering of different objects.

Fig. 5. **Real-world benchmark.** From left to right: the data capturing process, screenshot of GUI showing the process of replacing transparent bottle with opaque bottle, RGB-D images of transparent objects, RGB-D images of replaced spray-painted objects.

physically-based, ray-tracing Blender Cycles [5] rendering engine. We selected this because it is highly configurable, and is able to simulate important effects for transparent objects like refraction and reflection through multiple surfaces, as well as soft shadows.

The dataset consists of 9 CAD models modeled after real-world transparent plastic objects, in which we hold out 4 of the objects during training to test the algorithm generalization ability. Additionally, one gray tote box is used as an background object. We employed 33 HDRI lighting environments and 65 textures for the ground plane underneath the transparent objects. Camera intrinsics were set to that of the Intel RealSense D415 camera. To generate each scene, between 1 and 5 CAD model objects were created above the plane surface, with or without a gray tote box and the CAD model objects were dropped so they would come to rest according to physics. Then, a random selection of HDRI lighting environments and ground plane surface textures would be applied to each scene as well.

For each scene, the ground truth data includes: (1) monocular RGB render, (2) aligned depth in meters, (3) semantic segmentation of all transparent objects, (4) pose of the camera (5) pose of each CAD object, and (6) surface normals of the scene. Fig. 4 shows example rendered images and their corresponding ground truth geometry. The final training dataset consists of over 13,000 images of 3 objects each and 5000 images each of another 2 objects. 100 images of each were kept aside as a validation set. For the test set, we rendered around 100 images of each of the 4 testing objects.

C. Real-World Benchmark

To test the ability of our model to generalize to real-world images, we create a dataset of real-world transparent objects. The setup consists of a photography background cloth or wooden laminate spread across a flat surface kept against a wall. Five unique wooden laminates and five different background cloths were used. The scene was lit with ambient lighting to avoid sharp caustics. The camera was mounted on a tripod at a distance of 40-100cm from the objects.

To capture the depth of transparent objects, we separated the objects into 2 equal sets and spray painted one set with a rough stone texture, which gives much better depth than a flat color. A GUI app was developed that could overlay 2 frames read from the camera, as shown in Fig. 5. First the transparent objects were placed in the scene along with various random opaque objects like cardboard boxes, decorative mantelpieces and fruits. After capturing and freezing that frame, each object was replaced with an identical spray-painted instance. Subsequent frames would be overlaid on the frozen frame so that the overlap between the spray painted objects and the transparent objects they were replacing could be observed. With high resolution images, sub-millimeter accuracy can be achieved in the positioning of the objects.

The validation dataset consists of 173 images of 5 known objects used in synthetic training data. The testing set consist of 113 images of 5 novel objects, including 3 new glass objects not present in the synthetic dataset. Each image contains 1-6 objects, with an average of 2 objects per image.

D. Grasp planning

By integrating ClearGrasp into a robotic picking system, we can investigate its benefits for downstream manipulation tasks. We adapted a state-of-the-art grasping algorithm for our experiment [55, 54, 56], which consists of a convolutional network that predicts the probability of picking success for a scripted grasping primitive across a dense pixel-wise sampling of end effector locations and orientations across the completed depth images from ClearGrasp. Specifically, it uses an 18-layer fully convolutional residual network [20] with dilated convolutions [53] and ReLU activations [37], interleaved with 2 layers of max pooling, 2 layers of spatial bilinear $2x$ upsampling. The network takes as input a 4 channel image – the surface normal map (3 channels) concatenated channel-wise with the completed depth image (1 channel) inferred from ClearGrasp – and outputs a probability map with the same size and resolution as that of the input image. The picking system assumes that the 3D camera is calibrated with respect to robot coordinates using the calibration procedure in [54] – hence each pixel in the depth image maps to a 3D location. The robot executes a top-down parallel-jaw grasp or suction where the tip of the end effector is centered at the 3D location of the pixel with the highest predicted probability from the network.

For our experiments with parallel-jaw grasping, as in [55] we account for different grasping angles by constructing top-down orthographic heightmaps from ClearGrasp depth images, rotating the input heightmaps by 16 orientations (multiples of $22.5°$), and feeding each heightmap through the network for a total of 16 forward passes. The pixel and the corresponding rotation with the highest predicted probability among all 16 maps determines the respective grasping angle. The network is trained end-to-end using the binary cross-entropy error from predictions of grasp success

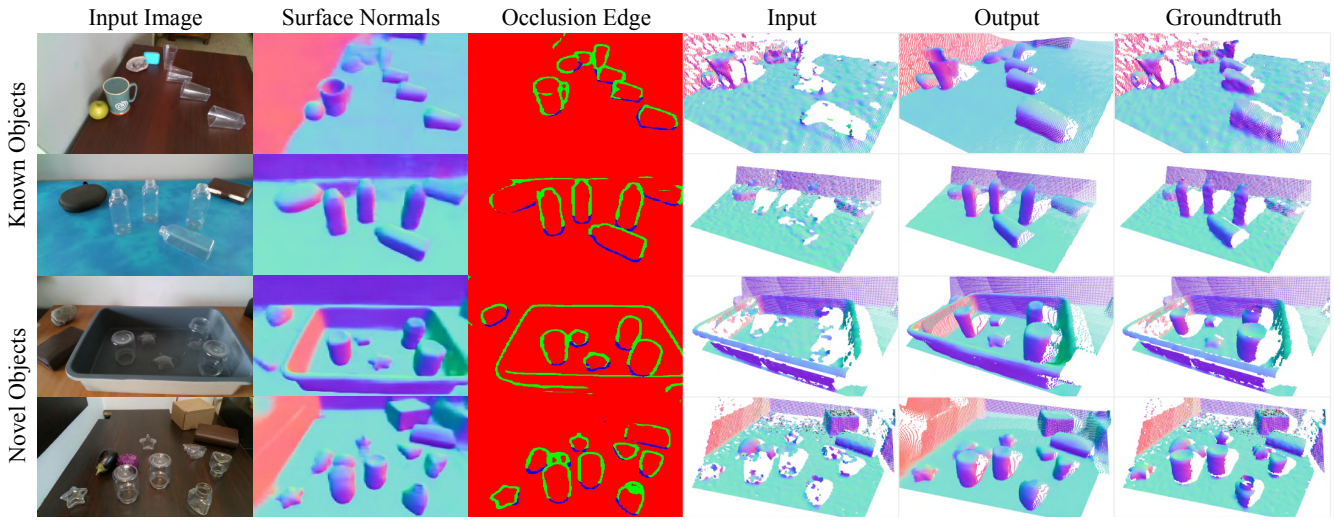


Fig. 6. **Qualitative results** on real-world benchmark with known objects (rows 1-2) and novel objects (rows 3-4). More results can be found in the supplementary material website.

against the binary ground truth success labels. We pass gradients only through the single pixel on which the grasping primitive was executed. Since each pixel-wise prediction shares convolutional features for all grasping locations and orientations, the network is sample-efficient and trains within a few hundred trial-and-errors.

IV. Evaluation

We evaluate ClearGrasp's ability to estimate transparent object geometry on both synthetic and real-world benchmarks, then apply it to a real-world robotic picking system.

Datasets used to train and test our algorithm:

- Syn-train: Synthetic training set with 5 objects as described in Sec. III-B.
- Syn-known: Synthetic validation set for training objects.
- Syn-novel: Synthetic test set of 4 novel objects.
- MP+SN: Out-of-domain real-world RGB-D datasets of indoor scenes that do not contain transparent objects' depth (Matterport3D [7] and ScanNet [11]).
- Real-known: Real-world test set for all 5 of the training objects. Sec. III-C describes the capturing procedure.
- Real-novel: Real world test set of 5 novel objects, including 3 not present in synthetic data.

Metrics: For surface normal estimations, we calculate the mean and median errors (in degrees) and the percentages of pixels with estimated normals less than thresholds of 11.25, 22.5, and 30 degrees. For depth estimation, we use metrics standard among previous works [14]: the Root Mean Squared Error in meters (RMSE), the median error relative to the depth (Rel) and percentages of pixels with predicted depths falling within an interval ($[\delta = |predicted - true|/true]$, where δ is 1.05, 1.10 or 1.25). Depth is evaluated by resizing the images and ground truth to 144x256p resolution. For mask prediction, we use pixel-wise intersection over union for evaluation as well as true positive rate. Unless specified, metrics are calculated on the real-known dataset only over the pixels belonging to transparent objects.

Generalization: real-world images. Table I also shows results on an experiment to test the cross-domain performance of our models. Despite never being trained on real-world transparent objects, we find our models are able to adapt well to the real-world domain achieving very similar RMSE and Rel scores on known objects across domains. However, the surface normal prediction accuracy decreases on real images. We observe large errors in surface normal estimations when transparent object occlude novel opaque objects. Surprisingly, the metrics of Real-novel objects are better than Syn-novel. We attribute this to the 3 new glass objects used in real-world images which show more evident refraction characteristics due to their thicker material as compared to the thin plastic material of all other objects.

Generalization: novel object shapes. We inspect the ability of our algorithm to generalize to previously unseen object shapes. Table I shows the results of depth estimation on novel objects, conducted on both synthetic data and real-world data. We see that it is able to generalize remarkably well in both cases, achieving better results than on the known objects. This is likely due to the smaller size of novel objects, which cause a relatively smaller error in depth reconstruction.

Comparison with Monocular Depth Estimation. We compare our approach with DenseDepth [2], a monocular depth estimation method that has state-of-the-art performance. DenseDepth uses a deep neural network to directly predict the depth value from the color image. We train DenseDepth with the same training data as our approach. The results in Table I show that our model outperforms the monocular depth estimation methods by a large factor.

Effect of mask prediction. We test the effectiveness of cleaning the input depth by removing all pixels belonging to transparent objects, as shown in Table II. By not removing the initial noisy depth values, we notice a significant increase in the final depth estimation error. Table I reports the accuracy of the mask prediction in both intersection over union and true positive rate. In our approach, having a high true positive rate ($> 95\%$) is critical for removing all the

TABLE I

GENERALIZATION. CLEARGRASP GENERALIZES TO BOTH REAL IMAGES AND NOVEL TRANSPARENT OBJECTS UNSEEN IN TRAINING.

Testset		Depth Estimation						Surface Normal Estimation					Mask	
Type	Object	RMSE↓	REL↓	MAE↓	$\delta_{1.05}$ ↑	$\delta_{1.10}$ ↑	$\delta_{1.25}$ ↑	mean↓	med.↓	11.25° ↑	22.5° ↑	30° ↑	IoU	TP
Synthetic	Known	0.044	0.047	0.033	71.23	92.60	98.24	15.64	10.62	53.71	78.28	85.83	0.93	95.90
Synthetic	Novel	0.040	0.071	0.035	42.95	80.04	98.10	25.32	20.53	24.04	55.88	69.73	0.94	97.58
Real	Known	0.039	0.053	0.029	70.23	86.98	97.25	21.93	18.72	32.82	64.39	76.05	0.63	96.30
Real	Novel	0.028	0.040	0.022	79.18	92.46	98.19	22.29	18.09	31.63	63.44	76.06	0.58	96.95

TABLE II

BASELINE COMPARISONS AND ABLATION STUDY.

	RMSE↓	REL↓	MAE↓	$\delta_{1.05}$ ↑	$\delta_{1.10}$ ↑	$\delta_{1.25}$ ↑
DenseDepth [2]	0.270	0.428	0.259	18.67	34.34	58.29
DeepCompletion [59]	0.054	0.081	0.045	44.53	69.71	95.77
- Mask	0.054	0.080	0.044	44.46	69.73	96.06
- Contact Edge	0.061	0.096	0.054	36.64	65.11	92.38
- Edge Weights	0.049	0.075	0.042	51.77	73.70	95.59
Full	**0.038**	**0.048**	**0.027**	**72.94**	**87.88**	**97.17**

incorrect initial depth values.

Effect of contact edges and edge weights Table II shows the effect of using a weighted loss function and the impact of adding the additional class of contact edges to our occlusion boundary estimation model. Both of these methods contribute significant improvement in depth completion results.

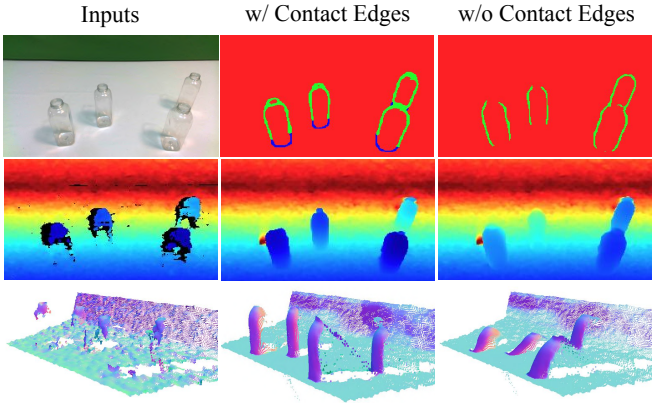


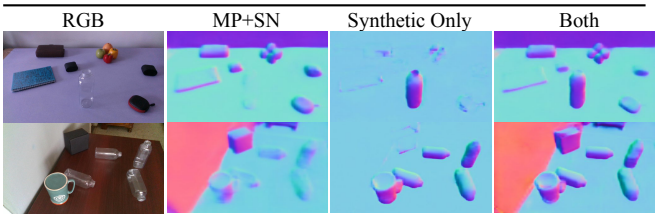
Fig. 7. **Effects of contact edges.** By training our boundary estimation model with contact edges (middle column), ClearGrasp predicts better depth for transparent objects than without contact edges (right column).

Effect of training data. Our main training dataset consists of synthetic images of transparent objects. Since it is expensive to capture real-world data with accurate geometry ground truth for transparent objects, we propose to mix in typical real-world RGB-D indoor scene images in training to reduce the domain gap. Table III shows the model performance under different training procedure: with/without pre-training on out-of-domain real-world data (80k images from the Scannet and Matterport datasets) and with/without in-domain synthetic data fine-tuning. Fig. 6 additionally shows the qualitative results of surface normal estimation for all the above cases. We see that a model trained on out-of-domain real-world data is not able to pick up transparent objects. However, pre-training with such data improves results, especially for real-world test sets.

TABLE III

TRAINING DATA. NORMAL ESTIMATION PERFORMANCE UNDER DIFFERENT TRAINING PROCEDURES: WITH/WITHOUT OUT-OF-DOMAIN REALWORLD DATA (MP+SN) AND IN-DOMAIN SYNTHETIC DATA (SYN).

Pretrain	Train	Mean↓	Median↓	11.25 ↑	22.5 ↑	30 ↑
MP+SN	-	43.92	45.31	9.51	22.69	32.03
-	Syn	**21.59**	24.74	24.74	55.97	70.40
MP+SN	Syn	21.93	**18.72**	**32.82**	**64.39**	**76.05**



Robot manipulation. We also incorporate ClearGrasp as part of a real-world robotic picking system to observe how it influences the overall grasping performance of transparent objects. In this experiment, a pile of 3 to 5 transparent objects are presented on a table within the robot's workspace, of which RGB-D images are captured using a calibrated RealSense camera. The goal of the robot is to pick up objects from the table using a state-of-the-art grasping algorithm described in Sec. III-D. Fig. 1 shows the setup. We test the algorithm using two end-effectors: suction and a parallel-jaw gripper. For each end-effector type, with and without ClearGrasp, we train a grasping algorithm using 500 trial and error grasping attempts, then test it with 50 attempts. We compute the average grasping success rate $= \frac{\#\text{ successful picks}}{\#\text{ picking attempts}}$ [54] as the evaluation metric. With both end-effectors, we observe that ClearGrasp significantly improves the grasping success rate of transparent objects: it improves the grasping success from 64% to 86% for suction, and 12% to 72% for parallel-jaw grasping.

V. CONCLUSION AND FUTURE WORK

We present ClearGrasp, an algorithm that leverages deep learning with synthetic training data and multiple sensor modalities (color and depth) to infer accurate 3D geometry of transparent objects for manipulation. However, the proposed system is still far from perfect. Possible future directions may include: explicitly leveraging lighting information during the inference step to improve the algorithm's accuracy under different lighting conditions, improving the algorithm robustness in cluttered environments where predicting accurate occlusion and contact edges is more challenging and making the algorithm robust to sharp caustics and shadows.

References

[1] Sven Albrecht and Stephen Marsland. Seeing the unseen: Simple reconstruction of transparent objects from point cloud data. In *Robotics: Science and Systems*, 2013.

[2] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *arXiv e-prints*, abs/1812.11941:arXiv:1812.11941, 2018.

[3] Aayush Bansal, Bryan Russell, and Abhinav Gupta. Marr revisited: 2d-3d alignment via surface normal prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5965–5974, 2016.

[4] Igor Barros Barbosa, Marco Cristani, Barbara Caputo, Aleksander Rognhaugen, and Theoharis Theoharis. Looking Beyond Appearances: Synthetic training data for deep cnns in re-identification. *Comput. Vis. Image Underst.*, 2018.

[5] Blender. Blender Cycles, 2019. URL https://docs.blender.org/manual/en/2.80/render/cycles/introduction.html.

[6] Blender. Blender Physics Engine, 2019. URL https://docs.blender.org/manual/en/latest/physics/index.html.

[7] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[8] Guanying Chen, Kai Han, and Kwan-Yee K. Wong. Tom-net: Learning transparent object matting from a single image. In *CVPR*, 2018.

[9] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–818, 2018.

[10] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. *CoRR*, abs/1604.03901, 2016.

[11] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[12] Xiaoming Deng, Shuo Yang, Yinda Zhang, Ping Tan, Liang Chang, and Hongan Wang. Hand3d: Hand pose estimation using 3d neural network. *arXiv preprint arXiv:1704.02224*, 2017.

[13] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.

[14] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *CoRR*, abs/1406.2283, 2014.

[15] Mario Fritz, Gary Bradski, Sergey Karayev, Trevor Darrell, and Michael J Black. An additive latent feature model for transparent object recognition. In *Advances in Neural Information Processing Systems*, pages 558–566, 2009.

[16] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018.

[17] Xiaojin Gong, Junyi Liu, Wenhui Zhou, and Jilin Liu. Guided depth enhancement via a fast marching method. *Image and Vision Computing*, 31(10):695–703, 2013.

[18] Chen Guo-Hua, Wang Jun-Yi, and Zhang Ai-Jun. Transparent object detection and location based on RGB-D camera. In *Journal of Physics: Conference Series*, volume 1183, page 012011. IOP Publishing, 2019.

[19] Kai Han, Kwan-Yee K Wong, and Miaomiao Liu. A fixed viewpoint approach for dense reconstruction of transparent objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4001–4008, 2015.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[21] Daniel Herrera, Juho Kannala, Janne Heikkilä, et al. Depth map inpainting under a second-order smoothness prior. In *Scandinavian Conference on Image Analysis*, pages 555–566. Springer, 2013.

[22] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. Revisiting single image depth estimation: toward higher resolution maps with accurate object boundaries. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1043–1051. IEEE, 2019.

[23] Matis Hudon, Mairead Grogan, Rafael Pages, and Aljosa Smolic. Deep normal estimation for automatic shading of hand-drawn characters. In *The European Conference on Computer Vision (ECCV) Workshops*, September 2018.

[24] Yijun Ji, Qing Xia, and Zhijiang Zhang. Fusing depth and silhouette for scanning transparent object with RGB-D sensor. *International Journal of Optics*, 2017, 2017.

[25] Alexander B. Jung, Kentaro Wada, Jon Crall, Satoshi Tanaka, Jake Graving, Sarthak Yadav, Joy Banerjee, Gbor Vecsei, Adam Kraft, Jirka Borovec, Christian Vallentin, Semen Zhydenko, Kilian Pfeiffer, Ben Cook, Ismael Fernndez, Weng Chi-Hung, Abner Ayala-Acevedo, Raphael Meudec, Matias Laporte, et al. imgaug. https://github.com/aleju/imgaug, 2019. Online; accessed 25-Sept-2019.

[26] May Phyo Khaing and Mukunoki Masayuki. Transpar-

ent object detection using convolutional neural network. In *International Conference on Big Data Analysis and Deep Learning Applications*, pages 86–93. Springer, 2018.

[27] Ulrich Klank, Daniel Carton, and Michael Beetz. Transparent object detection and reconstruction on a mobile platform. In *2011 IEEE International Conference on Robotics and Automation*, pages 5971–5978. IEEE, 2011.

[28] I Kokkinos, S Zafeiriou, et al. Face normals in-the-wild using fully convolutional networks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[29] Po-Jen Lai and Chiou-Shann Fuh. Transparent object detection using regions with convolutional neural network. In *IPPR Conference on Computer Vision, Graphics, and Image Processing*, pages 1–8, 2015.

[30] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. *CoRR*, abs/1606.00373, 2016. URL http://arxiv.org/abs/1606.00373.

[31] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[32] Ilya Lysenkov and Vincent Rabaud. Pose estimation of rigid transparent objects in transparent clutter. In *2013 IEEE International Conference on Robotics and Automation*, pages 162–169. IEEE, 2013.

[33] Ilya Lysenkov, Victor Eruhimov, and Gary Bradski. Recognition and pose estimation of rigid transparent objects with a kinect sensor. *Robotics*, 273, 2013.

[34] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv preprint arXiv:1703.09312*, 2017.

[35] K. McHenry, J. Ponce, and D. Forsyth. Finding glass. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, 2005.

[36] Kenton McHenry and Jean Ponce. A geodesic active contour framework for finding glass. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 1038–1044. IEEE, 2006.

[37] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[38] Cody J Phillips, Konstantinos G Derpanis, and Kostas Daniilidis. A novel stereoscopic cue for figure-ground segregation of semi-transparent objects. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1100–1107. IEEE, 2011.

[39] Cody J Phillips, Matthieu Lecce, and Kostas Daniilidis. Seeing glassware: from edge detection to pose estimation and shape recovery. In *Robotics: Science and Systems*, volume 3, 2016.

[40] Yiming Qian, Minglun Gong, and Yee Hong Yang. 3d reconstruction of transparent objects with position-normal consistency. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[41] Michael Ramamonjisoa and Vincent Lepetit. Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation. *arXiv preprint arXiv:1905.08598*, 2019.

[42] Konstantinos Rematas, Ira Kemelmacher-Shlizerman, Brian Curless, and Steve Seitz. Soccer on your tabletop. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4738–4747, 2018.

[43] Viktor Seib, Andreas Barthen, Philipp Marohn, and Dietrich Paulus. Friend or foe: exploiting sensor failures for transparent object localization and classification. In *2016 International Conference on Robotics and Machine Vision*, volume 10253, page 102530I. International Society for Optics and Photonics, 2017.

[44] Seongjong Song and Hyunjung Shim. Depth reconstruction of translucent objects from a single time-of-flight camera using deep residual networks. *arXiv preprint arXiv:1809.10917*, 2018.

[45] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[46] Jonathan Stets, Zhengqin Li, Jeppe Revall Frisvad, and Manmohan Chandraker. Single-shot analysis of refractive shape using convolutional neural networks. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 995–1003. IEEE, 2019.

[47] Jonathan Dyssel Stets, Rasmus Ahrenkiel Lyngby, Jeppe Revall Frisvad, and Anders Bjorholm Dahl. Material-based segmentation of objects. In *Scandinavian Conference on Image Analysis*, pages 152–163. Springer, 2019. ISBN 978-3-030-20204-0. doi: 10.1007/978-3-030-20205-7_13.

[48] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[49] Andreas ten Pas and Robert Platt. Using geometry

to detect grasp poses in 3d point clouds. In *Robotics Research*, pages 307–324. Springer, 2018.

[50] Tao Wang, Xuming He, and Nick Barnes. Glass object localization by joint inference of boundary and depth. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 3783–3786. IEEE, 2012.

[51] X. Wang, D. F. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 539–547, 2015.

[52] Wofk, Diana and Ma, Fangchang and Yang, Tien-Ju and Karaman, Sertac and Sze, Vivienne. FastDepth: Fast Monocular Depth Estimation on Embedded Systems. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[53] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[54] Andy Zeng, Shuran Song, Stefan Welker, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4238–4245. IEEE, 2018.

[55] Andy Zeng, Shuran Song, Kuan-Ting Yu, Elliott Donlon, Francois R Hogan, Maria Bauza, Daolin Ma, Orion Taylor, Melody Liu, Eudald Romo, et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

[56] Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Tossingbot: Learning to throw arbitrary objects with residual physics. *arXiv preprint arXiv:1903.11239*, 2019.

[57] Jin Zeng, Yanfeng Tong, Yunmu Huang, Qiong Yan, Wenxiu Sun, Jing Chen, and Yongtian Wang. Deep surface normal estimation with hierarchical rgb-d fusion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[58] Jianfeng Zhang. Pytorch re-implementation of deeplabv3+, 2018. URL https://github.com/jfzhang95/pytorch-deeplab-xception. Online; accessed 25-Sept-2019.

[59] Yinda Zhang and Thomas Funkhouser. Deep depth completion of a single rgb-d image. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[60] Yinda Zhang, Shuran Song, Ersin Yumer, Manolis Savva, Joon-Young Lee, Hailin Jin, and Thomas Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

APPENDIX

The appendix consists of additional system details, analysis, and experimental results.

A. Additional Details on Dataset

Fig. 8 and Fig. 9 showcase the objects used within our synthetic and real-world datasets. In the real-world dataset, all images of known objects are taken with a RealSense D435 camera and 80% of the images of novel objects were taken with a RealSense D415 camera instead.

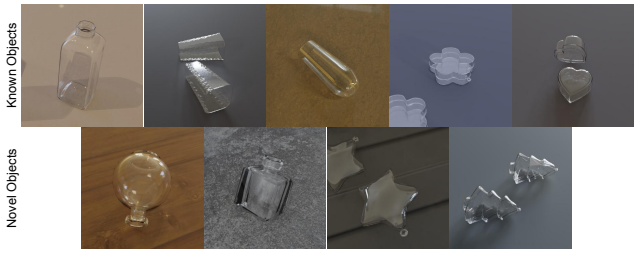


Fig. 8. **Known and novel objects in Synthetic dataset.** We have 5 known objects for training and 4 novel objects for testing. The test objects are all challenging: 2 are of thick glass (a different material from our plastic known objects) and 2 are of complex shapes.

Fig. 9. **Known and novel objects in Real-World benchmark dataset.** We have 5 known objects and 5 novel objects. All 5 known objects and 2 of the novel objects were used to model the synthetic objects. 2 other novel objects are made of glass instead of plastic.

B. Limitations and Failure Cases

We detail some of the failure cases of our models. Four examples are shown in Fig. 10

- The biggest limitation of our approach is that it is not always possible to reconstruct depth from surface normals directly [59] - if a region is completely enclosed by an occlusion boundary, its depth is left indeterminate from the rest of the scene. In Case I, we see a bottle that is partially occluded, with its contact edges not visible. In Case II, we see the mouth of the glasses cause the inner portions to be (correctly) completely enclosed by an occlusion boundary. In both cases, the depths of such regions become indeterministic and can be assigned random values.
- Cluttered scenes are challenging. In cases where multiple transparent objects are partially or completely occluding each other, it becomes challenging to correctly predict surface normals and occlusion boundaries, which leads to errors in the output depth. Case III highlights such a scenario. Another situation which our models find challenging is when the background seen behind a transparent object is not constant - such as when a bottle is at the edge of a table or when its partially occluding an opaque object.
- As seen in Case IV, bright directional lighting and its associated caustics cause our model to mistakenly identify shadows of transparent objects as transparent objects. Our models seems to pick up on cues like specular highlights to identify transparent objects and may be confusing the caustics on shadows with specular highlights - hence detecting the shadow as a transparent object. Since our synthetic dataset does not contain accurate caustics due to the limitations of the Cycles rendering engine, our model is particularly susceptible to this problem.

C. Additional Training Details

We make use of Deeplabv3+ with a DRN-D-54 backbone [58] in Pytorch for all 3 of our neural networks - surface normal estimation, occlusion boundary prediction and segmentation of transparent surfaces. For all 3 networks, we start with a model pre-trained for semantic segmentation on the COCO dataset and use the same hyperparameters: SGD Optimizer with constant learning late of 1e-6, momentum 0.9 and weight decay 5e-4. We used a GCP server with 8x Nvidia V100 GPUs enabling a batch size up to 128 at an input image size of 256x256p.

For surface normals, we initially pre-train our model on the Matterport3D (MP) and Scannet (SN) datasets by selecting a random subset of approximately 40k images from each, for a total dataset of 80k images. When training on transparent objects, we include a new random subset of 2k images from MP and 2k from SN each epoch. Our synthetic dataset contains only a flat plane and up to 5 transparent objects, lacking any other surfaces like walls and random opaque objects. Injecting MP+SN images every epoch allows the model to retain knowledge of the previous domain and predict more accurate normals for surfaces like walls. To train the model more quickly on the different task of surface normal estimation, we adopted a staggered training approach: First, we trained a small subset of 100 images at a reduced resolution of 128x128p. Second, we took an early checkpoint before the model starts to stabilize and train on a larger subset of our data. This step was repeated on subsequently larger subsets. Third, we repeat the procedure with the larger image size of 256x256p taking the checkpoint from the previous step.

To make the models more robust, the following data augmentations were utilized from the imgaug [25] library:

- Flip Up-Down
- Flip Left-Right (not used for surface normals)
- Rotate 90 degrees (not used for surface normals)
- Color Space Augmentations: Add (RGB), Multiply (RGB), Add Hue, Add Saturation, Contrast Normalization, Grayspace
- Blur: Motion Blur or Gaussian Blue
- Noise: Add Element-wise, Multiply Element-wise, Additive Gaussian Noise, Additive Laplacian Noise, Dropout
- Large Patches:

– Channel-Wise Coarse Dropout up to 1/4th the size of the image. This makes the model more robust to varying backgrounds seen behind a transparent object.
– Bright White Patches: We use a Simplex Noise blended with a white image to generate random white patches which are overlaid with transparency on our images. Since our synthetic images do not contain significant caustics, this augmentation attempts to make the model more robust to bright patches of light due to caustics or directional lights.

Using this data augmentation strategy, we noticed a significant improvement in scenes with a patterned background cloth, bright caustics or directional lights and cases where the background behind a transparent object varied (like when it partially occludes an opaque object).

D. Experiment on Network Architectures

During our trials, we noticed that surface normal estimation was better on bottles that were kept further away from the camera. This led us to hypothesize that a larger receptive field might be helpful for transparent objects. Table IV shows the results of experimenting with different models and input image sizes. We try Deeplabv3+ with 2 different backbones: Resnet-101 and DRN-54 (Dilated Residual Network). We also experiment with different input image sizes. The results indicate that a smaller input image size, which effectively increases receptive field width, performs better. Further, replacing Resnet with DRN, which increases the receptive field of the network [53], improves results even more - hence validating our hypothesis.

Backbone	Input Size	Mean↓	Median↓	11.25 ↑	22.5 ↑	30 ↑
Resnet101	512	34.9	32.8	16.0	42.6	56.8
Resnet101	256	25.3	22.2	25.7	56.9	70.0
DRN-54 [ours]	256	**22.5**	**19.4**	**28.6**	**61.5**	**75.5**

TABLE IV
NETWORK ARCHITECTURES FOR SURFACE NORMAL.

E. Additional Qualitative Results

Finally, we present some qualitative results on our ablation study and comparison with baselines (Ref to Table II). Fig. 11 shows the performance of our approach a) without masks, b) without contact edges and c) without weighted loss terms for the contact edges. Fig. 12 shows the qualitative results of our method in comparison with DeepCompletion and DenseDepth. For more qualitative results please visit our website.

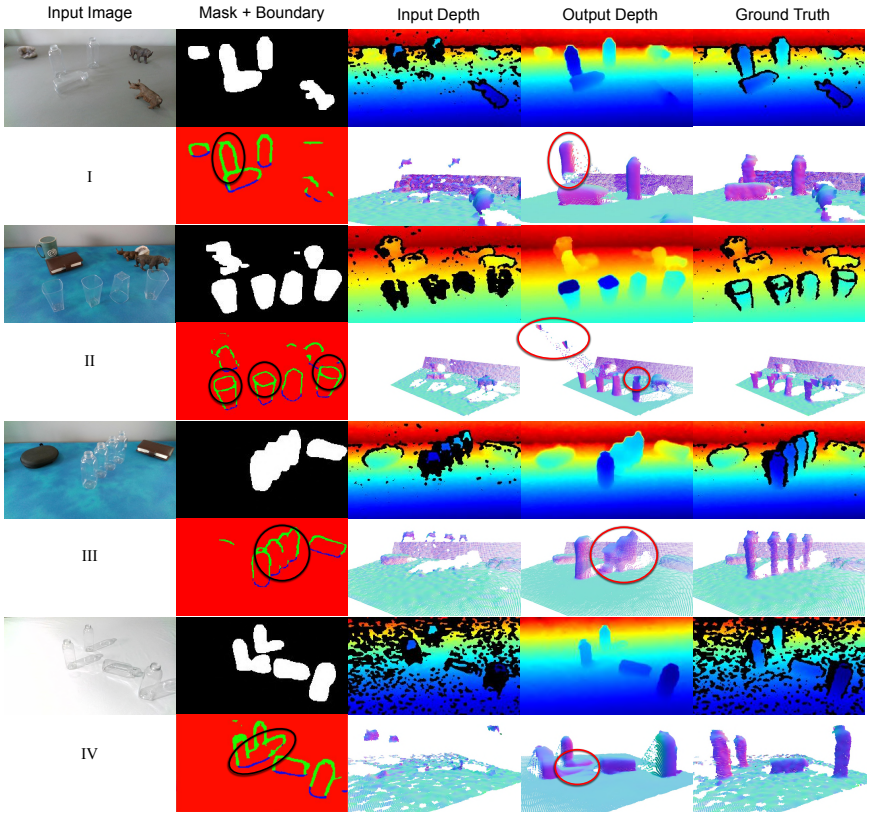


Fig. 10. **Failure Cases.** Most of the errors in output depth (highlighted in red) are due to the errors in occlusion boundary prediction (highlighted in black) - either erroneous outputs or surfaces with no contact edges due to occlusion.

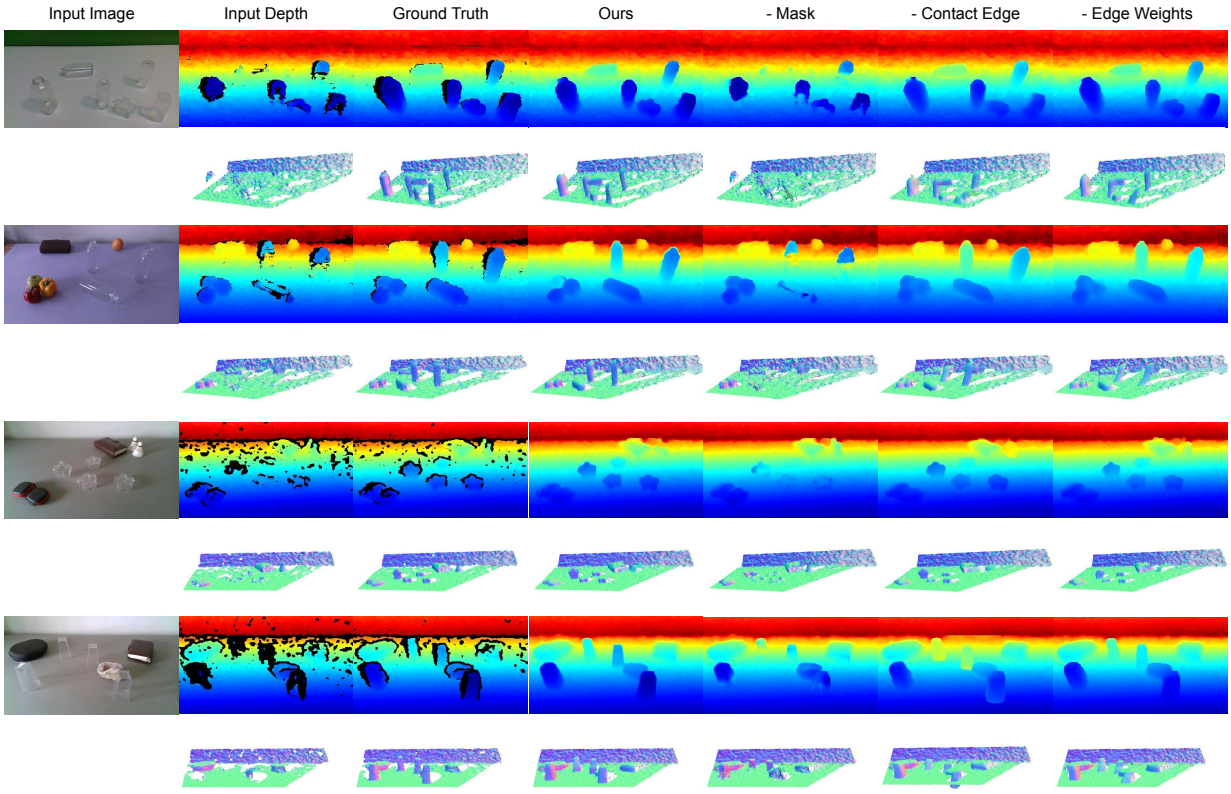


Fig. 11. **Qualitative results - Ablation Study**

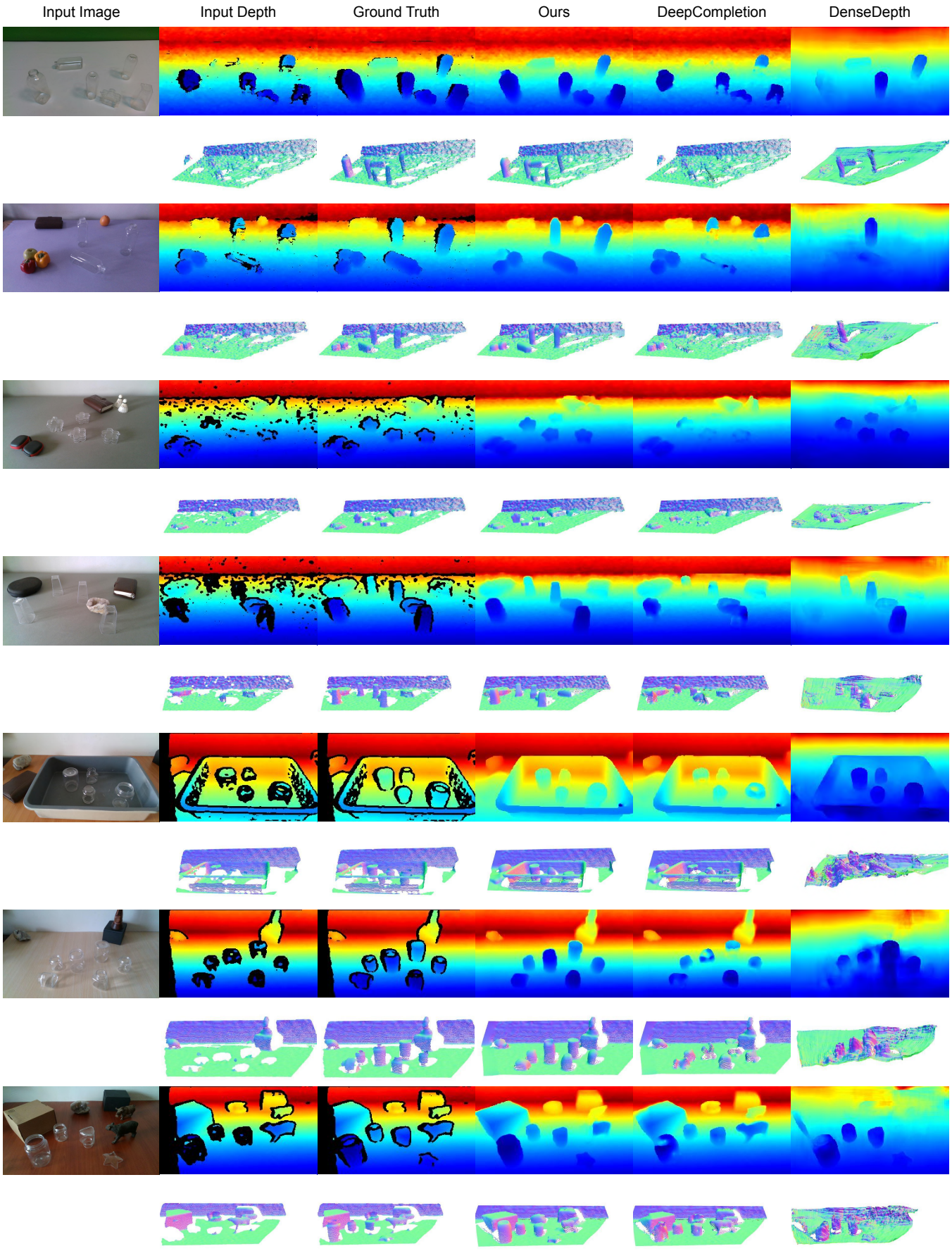


Fig. 12. **Qualitative results - Comparison with baselines**