CSC2547 Project Midterm Report

Haoping Xu, Tao Wu, Zhoujie Zhao

1 Introduction

3D shape estimation is an area of growing interest in past decades. Recent advances in RGB-D cameras allow us to automate the construction of 3D shapes and scenes modeling the real world. Nevertheless, the performance of standard 3D sensors falls short for transparent materials such as plastic bottles and glass mugs which are very common in everyday life. Many classic stereo vision algorithms fail to fit transparent objects due to their special visual property. Hence, generating accurate depth estimates for transparent objects is a challenging task.

In this project, we propose a novel automated dataset collection and annotation pipeline, and through that, create a transparent object depth estimation dataset, FrankaScan, consisting 4500 RGB-D images with ground truth depth, mask and boundary. Later on, we tests our dataset on ClearGrasp [1], one of the existing method capable to correct transparent object depth. We also propose to convert the depth estimation task to point cloud completion task via depth deprojection. Using GRNet [2], the STOA method for point cloud completion, various tricks on three different real-world datasets are compared for their effectiveness when applying to transfer learn GRNet pretrained on synthetic ShapeNet to real world point clouds from depth.

2 Related work

2.1 ClearGrasp

ClearGrasp [1] aims to fix the distorted and incomplete raw depth due to transparent object. It is based on the deep depth completion network [3], where the surface normal and occlusion boundary are predicted from single RGB image. The raw depth map from the RGBD sensor is then combine with the estimated depth to do global optimization. This process is aimed to minimize a weighted error between raw depth, estimated depth, neighboring pixel depth and the predicted normal. On top of that, cleargrasp includes a segment head to predict the mask for transparent objects, and remove their corresponding depth information during global optimization. The masking operation utilize the capability of deep depth completion network's ability to complete missing depth map.

ClearGrasp uses synthetic dataset generated by blender as the training set and collects the validation and testing dataset from the real world using matched transparent and opaque objects and manually place them at same location.

2.2 Keypose

Keypose network [4] can predict a set of key points from a transparent object which can be used to determining its pose for downstream applications, i.e. robot manipulation. Unlike ClearGrasp [1], keypose use stereo images as input, and predict each key point using a 2D image location as well as the disparity between left and right image to obtain the 3D location. This paper proposes two variations to compute them, for early fusion model, it computes the disparity map and one 2D location, whereas the later fusion network estimates a pair of 2D location from each stereo pair and compute the disparity using the matched key points.

Keypose collects its dataset using a eye-in-hand robot arm and tracking system, Apriltag 2[5]. The camera path is calculated using Apriltag, and along the scan trajectory, several frames' keypoints are

Preprint. Under review.

manually labelled, where the rest are labeled based on the trajectory. Also, authors also collect the depth info using a similar replacing method as CleatGrasp [1].

3 Method

3.1 Depth correction

Due to the optical proprieties of transparent objects, the depth information from common RGBD sensors are usually distorted and incomplete. We propose to segment out the incorrect depth map via the mask predicted from color image, similar to ClearGrasp [1]. But instead of training the segment network only on generated data, the model will be trained on real world dataset collected and annotated by us called FrankaScan.

With the isolated depth for transparent objects, Raw geometry to true geometry conversion can be treated as 3D point cloud completion task. Where the raw depth will be projected to a distorted and incomplete point cloud of the transparent object, and the actual point cloud sampled from the object's mesh is the target. We propose to use the SOTA model in this task, GRNet [2] and train it using FrankaScan dataset collected by us. With the completed point cloud, we can test its performance by combining it with robot grip pose estimation and actual robot picking transparent objects.

3.2 GRNet

In the GRNet, the incomplete point cloud is firstly sent to a Gridding layer, which computes weighted vertices of 3D grid cells that points lie in. An interpolation function is used here to measure the geometric relations of the point cloud. Then a 3D convolutional neural network with U-net connections is adopted to learn the features that are necessary for point cloud completion. The following is the Gridding reverse layer. It back projects each 3D grid cell to a new point whose coordinate is the weighted sum of the eight vertices. Before forward output cloud point to a multilayer perceptron that create the final completed point cloud, features from the feature map are added, which is generated by the first three transposed convolutional layers in 3D CNN.
Except the new algorithms above, a new loss is proposed in GRNet, called Gridding loss. It calculates the L1 distance between the regular 3D grids representations produced from the predicted points and ground truth.

3.3 Automate data collection

We will use an eye-in-hand Franka Emika Panda with Inteal realsense RGBD camera to collect the dataset.To fully automate the annotation is using the Apriltag 2, and placed the object to a fixed location and pose respect to the tags, which can be used later generate the ground truth with the object mesh. The Franka Panda robot eye-in-hand system is built and controllers for the robot as well as the realsense camera are programmed using FrankaPy [6]. In terms of automating annotation, the Apriltag 2 [5] approach turns to be easier both in calculating the relative 6DoF pose of the transparent object as well as writing the script. The detail dataset collection process is the following:

- An array of Apriltag2 are placed around a transparent object, each tag is from the same family with different index. And the offset from the tag and object is known using a printed template.
- Multiple random positions and viewing angles are selected. For every viewpoint, the robot aligns the camera with it and captures the RGB-D images.
- For automatic annotation, we first detect the Apriltag pose using the color input. The object pose can be extracted via tag's pose and using the 3D model of the object, the segmentation mask, ground truth depth map can be computed.
- Every viewpoint in the dataset will be consists of color image, raw depth map, ground truth depth, segmentation mask as well as the camera 6DoF pose relative to the object. The camera pose will be useful for recreating the scan scene when rendering additional synthetic dataset.

In total, the Franka Panda robot collects 4500 images of scenes consist of up to three glass beakers in two different backgrounds. Using the controlling script and automated annotation pipeline, FrankaScan dataset is created within three days, with minimal human interference. Additionally, apart from the ground truth annotations reuqired by ClearGrasp(semantic mask, occlusion and contact boundary and true depth), FrankaScan also consists of complete point cloud fro each transparent object in the scene. The complete point cloud is used in trained GRNet and compared performance with the counterpart trained on ground truth depth.

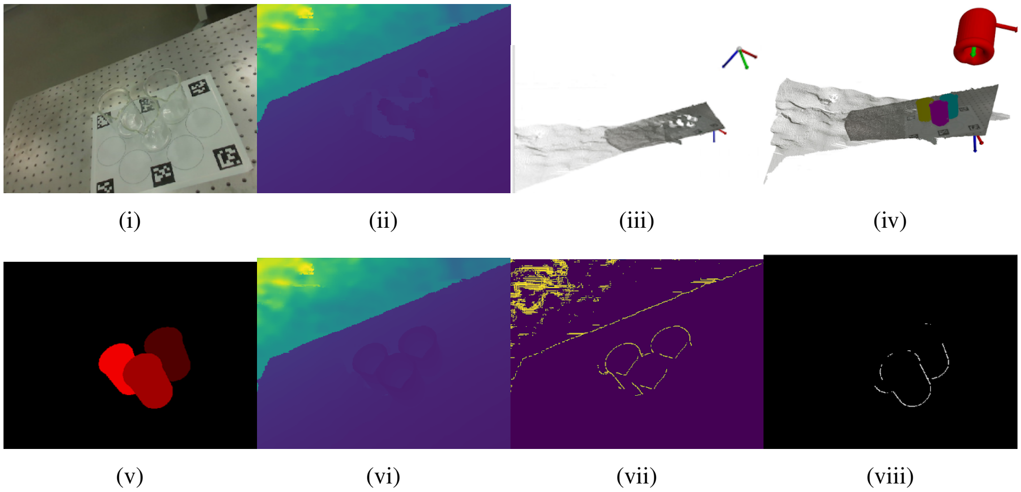


Figure 1: Dataset capture and annotation: (i) RBG image (ii) depth from sensor (iii) deprojected depth point cloud with camera pose and Apriltag pose (iv) Point clound sampled from mesh(red) and rigid transforms to object pose based on tag (yellow, blue and purple) (v) instance segment mask (vi) ground truth depth (vii) occlusion boundary (viii) contact boundary

4 Experiments and Discussion

In this section, we describe the experiments we have done on FrankaScan dataset that we collected. We evaluate the effectiveness and robustness of the baseline ClearGrasp model [1] with FrankaScan dataset. We then investigate the performance of GRNet [2] regarding the point cloud auto-completion task, and demonstrate its application in transparent object recognition.

4.1 ClearGrasp Model

We evaluate the original ClearGrasp [1] algorithm's ability to estimate the depth of transparent object through transfer learning and training from scratch on our dataset collected in Section 3.3. The sub-networks for surface normal estimation and occlusion boundary detection fail on our dataset. In Figure 2, we can see that our data set is quite different from that of ClearGrasp. Besides the boundary of the target object, we have boundaries for other objects in the scene, and we have some images only display half of our objects. Whereas in ClearGrasp's synthetic training dataset, only limited amount of objects are placed in a scene. These reasons lead to ClearGrasp's algorithm not working well on our real world dataset. The algorithm works even worse when we apply transfer learning. At least we can detect some boundaries without transfer learning, but there is nothing if we choose continue learning on the pretrained model (Figure 3).

4.2 GRNet Model

Transfer learning is applied in GRNet training. We start with a pretrained model provided by the authors [2] which was trained on ShapeNet dataset for 3D point cloud completion tasks. The model is then continue trained through a number of approaches to learn to recognize the real shape of transparent objects from raw point clouds that are deprojected from primitive depth images captured by the camera.

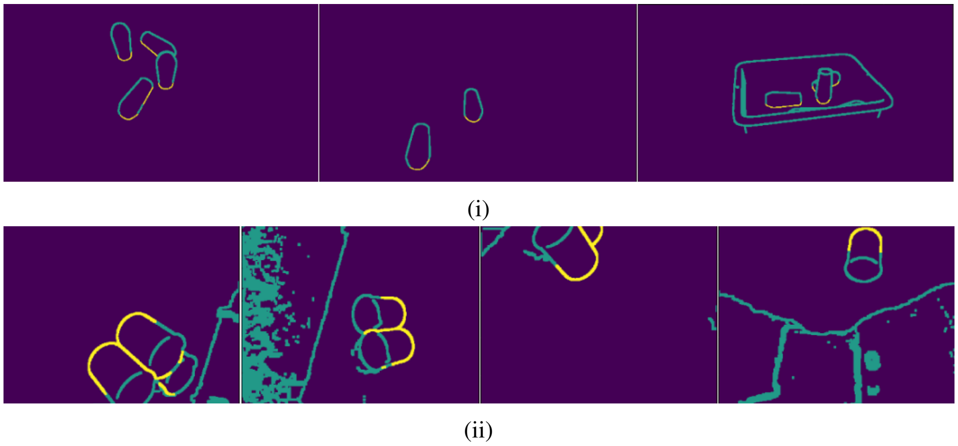


Figure 2: Combined boundary difference between ClearGrasp(i) and FrankaScan(ii)

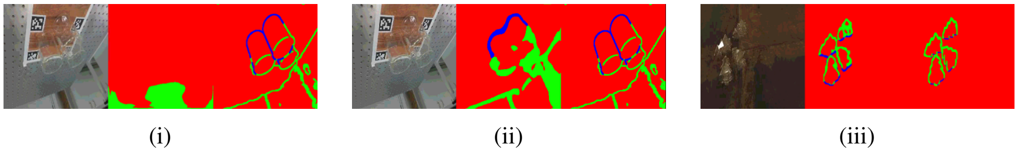


Figure 3: Output of ClearGrasp's boundary detector (i) Training on FrankaScan with transfer learning (ii) Training on FrankaScan without transfer learning (iii) ClearGrasp's pretrained model

4.2.1 Transfer Learning with ClearGrasp Dataset

We start transfer learning a pretrained GRNet model with real scan images in ClearGrasp dataset. Training and testing sets are manually separated to make sure that the object shapes in each set never appear in the other. We use Gridding Loss to train GRNet as suggested in its paper, and evaluate the model performance with two metrics: F-score and Chamfer distance. Furthermore, we observe that there is a variant distance from the camera to the background in different images, which may pose additional instability to the training procedure. Hence, we propose a point cloud normalization step by rescaling every point as a ratio to the farthest point in each input ground truth pair.

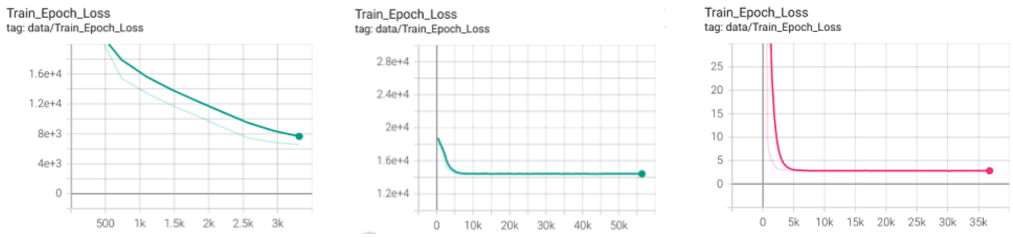
Figure 5 illustrates the losses of the dense and the coarse point clouds predicted by GRNet. Training with normalized point clouds appears to be more stable than training without normalization. It is also noticeable that both models converge to the same degree and there is no significant difference between their performance. This is due in part to the fact that ClearGrasp dataset contains only close scenes and thus the effect of normalization is not overt.

4.2.2 Transfer Learning with FrankaScan Dataset

In addition to ClearGrasp dataset, we train our model on a much larger real-world dataset, FrankaScan, which collects more than four thousand images with a greater diversity in object positions and depth of fields. We perform point cloud normalization in all following experiments.

It is worth remarking that in FrankaScan the objects do not necessarily appear near the center of the projected point clouds. On the contrary, they are often far from the origin. Observing this, we introduce an approach which we call re-centering, to encourage our model to focus on learning the shapes rather than positions of the transparent objects. When projecting the input depth image to a raw point cloud, we compute the mean point in 3D and re-center both raw and ground truth point clouds at the mean point.

The target point clouds in aforementioned models are all projected from the ground truth depth images. However, FrankaScan also provides a complete ground truth point cloud for each object captured. A complete point cloud displays the whole 3D shape of the object, while the projection from a depth image contains only partial points that can be seen from a certain angle of the camera. The last two images in the first row in Figure 7 visualize the difference between these two types of



(i) (ii) (iii)

Figure 4: Training loss of ClearGrasp (i) mask subnet (ii) boundary subnet (iii) surface normal subnet

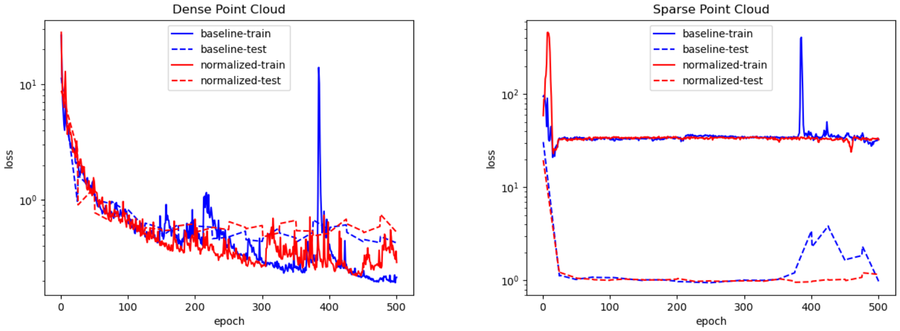


Figure 5: Gridding losses trained on ClearGrasp real scan test and validation dataset

ground truths. In the next model, we employ complete point clouds as the targets for training, and include the normalization and the re-centering stages as well.

Figure 6 displays the training curves and evaluation metrics in our experiments on FrankaScan dataset. It will be noticed that the network trained with re-centered data (drawn in red lines), comparing to the one trained with original data (drawn in blue lines), obtains a significantly smaller dense-output loss and a much stabler sparse-output loss. The Chamfer distance estimated on the test set also aligns with the loss curves - training with re-centered data leads to closer predictions. Another interesting point about re-centering trick is the sparse point cloud loss, notice that the uncentered dataset loss bounces back during training, whereas other centered datasets do not. Such difference likely suggests that GRNet coarse prediction network lack the expressive power to handle both point cloud location, size as well as shape all together. And the re-centering trick ensures all targets are in similar size and centered around the origin, and allow GRNet to focus on the shape alone.

When evaluating our third model which uses complete object point clouds as ground truths (drawn in green lines), we notice that its score falls short in either evaluation metric. We infer that the complete point cloud is a more complex target making it hard to reach a high f-score or a small Chamfer distance. To assess the performance of these models in a more accurate way, we visualize the output point clouds and quantify the results by computing the root mean squared errors of the predicted depth images, as will be discussed in detail later.

4.2.3 Experiment Results

In this section, we denote: the GRNet model trained on ClearGrasp dataset as CG; the model trained on FrankaScan dataset without re-centering as FS; the model trained with the re-centering step on FrankaScan as FS-center; and the model trained with complete clouds from object mesh as FS-comp.

Table 1 shows the performance of GRNet models trained on different datasets. They are evaluated on FrankaScan test set where point clouds extracted from depth images and object meshes are used respectively as the ground truth. We analyze the accuracy of our models on depth estimation by computing the root-mean-squared error of the predicted depth images. Moreover, we visualize the model performance on the 3D shape reconstruction task in Figure 7. In our evaluation, models

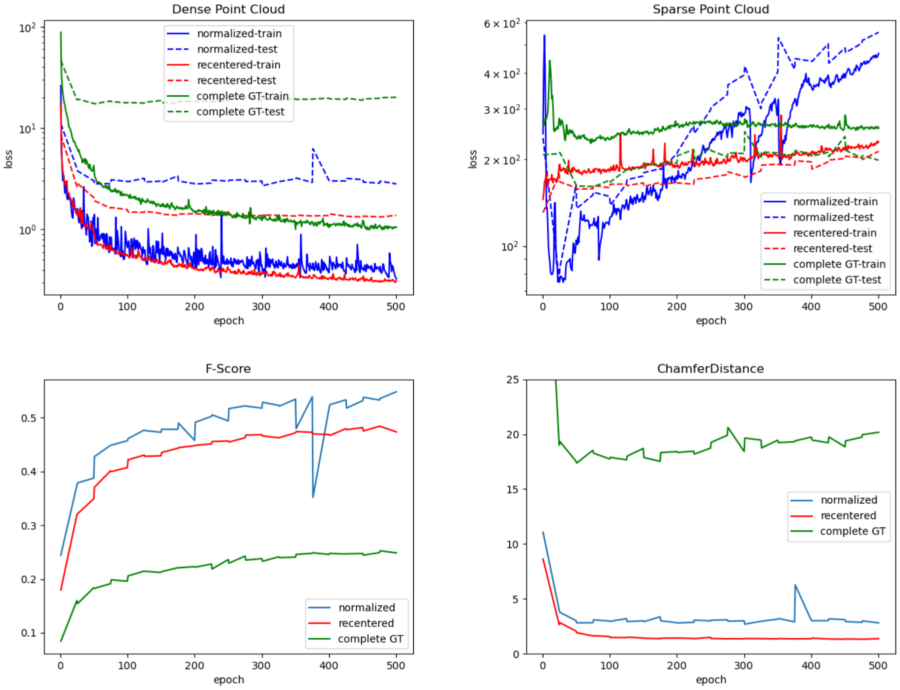


Figure 6: Gridding losses trained on FrankaScan dataset and evaluations

Training Dataset	GT clouds from depth		GT clouds from mesh		Depth image
	F-Score	Chamfer Distance	F-score	Chamfer Distance	RMS
ClearGrasp	0.1831	45.1627	0.0757	190.8023	4.8808E-2
FrankaScan	0.5482	2.8240	0.0423	173.2437	6.4542E-3
FrankaScan (center)	0.4738	1.3755	0.0450	196.5862	3.5364E-3
FrankaScan (complete)	0.0650	50.7028	0.2496	19.8709	9.5801E-3

Table 1: GRNet trained on 4 different datasets generated from real scans of RGB-D sensor, and their performances are evaluated on FrankaScan datasets (ground truth extracted from depth and object mesh) with rescaling and centering

trained on FrankaScan dataset outperform the baseline ClearGrasp in both quantitative metrics and 3D visualizations.

Inspection reveals that the variation of target clouds (from depth or from mesh) results in a significant difference in model outputs. FS-center gives the best Chamger distance when testing with target clouds projected from depth images. However, its performance drops dramatically when complete point clouds are set as the ground truth. In contrast, FS-comp does not behave well when evaluated on partial clouds, but outperforms all other models to a large degree on full-cloud completion. For the depth completion task, the three FS models generates similar depth images that are hard to assess by human eye, as illustrated in Figure 7. By quantifying the prediction accuracy using RMS error, we can see that FS-center performs relatively better than other models.

All in all, FS-comp aims to construct the whole shape, while FS-center focuses on producing accurate partial clouds that are visible from a certain camera angle. They are tailored for different tasks - 3D shape recognition and 2D depth estimation.

5 Future work

Given the promising result of FrankaScan when training GRNet and the efficiency of creating real world dataset, we are planning to further expand FrankaScan dataset to more object types,

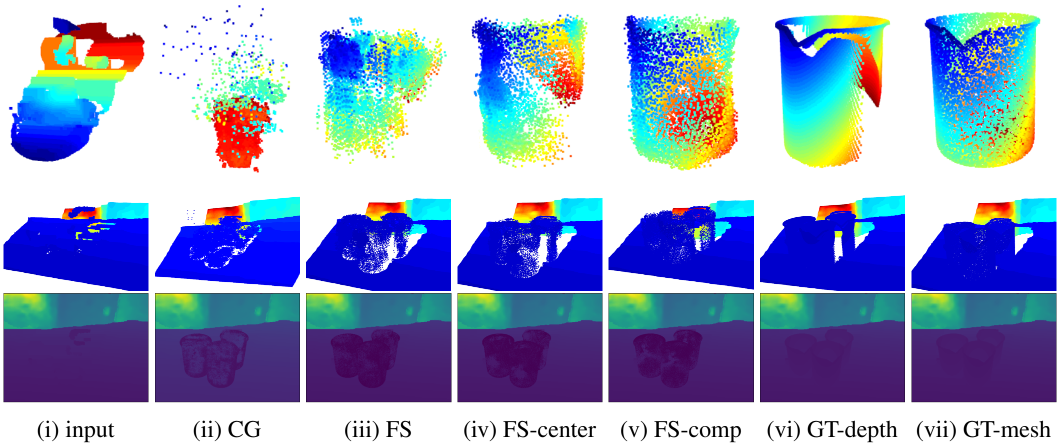

(i) input (ii) CG (iii) FS (iv) FS-center (v) FS-comp (vi) GT-depth (vii) GT-mesh

Figure 7: Results

backgrounds and lighting conditions. Currently all GRNet is trained on ground truth instance segmentation mask, we will also try Labpic[7] predicted masks.

Also we will develop the robot control pipeline to test the effectiveness of GRNet by manipulating transparent objects based on corrected depth.For future integration of GRNet with robotic control, we think FS-comp has a unique advantage comparing to other depth deprojected dataset. As, for ClearGrasp and FS-center, the completed depth image needs to processed by additional robot grip pose estimator and further increase the complicity of the perception based control pipeline. Whereas FS-center directly output a complete object point cloud with correct pose, the difficulty of grip pose prediction is mitigated as the model is directly given a 3D scene instead of depth image. Also more improvements for GRNet will be tested, currently, the output depth for GRNet consists of zero depth pixels due the point cloud nature of GRNet. Although increasing the points count in dense prediction can eventually solve this problem, the associated computational cost is not ideal. One possible fix is applying Global Optimization[3] to propagate the predicted depth pixels to its empty neighbors and further refined the corrected depth.

References

[1] Shreeyak S. Sajjan, Matthew Moore, Mike Pan, Ganesh Nagaraja, Johnny Lee, Andy Zeng, and Shuran Song. Cleargrasp: 3d shape estimation of transparent objects for manipulation. *CoRR*, abs/1910.02550, 2019.

[2] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. Grnet: Gridding residual network for dense point cloud completion, 2020.

[3] Yinda Zhang and Thomas Funkhouser. Deep depth completion of a single rgb-d image, 2018.

[4] Xingyu Liu, Rico Jonschkowski, Anelia Angelova, and Kurt Konolige. Keypose: Multi-view 3d labeling and keypoint estimation for transparent objects. *CoRR*, abs/1912.02805, 2019.

[5] J. Wang and E. Olson. Apriltag 2: Efficient and robust fiducial detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4193–4198, 2016.

[6] Kevin Zhang, Mohit Sharma, Jacky Liang, and Oliver Kroemer. A modular robotic arm control stack for research: Franka-interface and frankapy, 2020.

[7] Sagi Eppel, Haoping Xu, Mor Bismuth, and Alan Aspuru-Guzik. Computer vision for recognition of materials and vessels in chemistry lab settings and the vector-labpics data set. *ACS Central Science*, 6(10):1743–1752, 2020.