

이화여자대학교 대학원
2020학년도
석사학위 청구논문

Toward Autonomous Robotic
Arrangement of Objects using Deep
Image Manipulation

인공지능·소프트웨어학부
Siyeon Kim
2021

Toward Autonomous Robotic Arrangement of Objects using Deep Image Manipulation

이 논문을 석사학위 논문으로 제출함

2020 년 12 월

이화여자대학교 대학원

인공지능 · 소프트웨어 학부 Siyeon Kim

Siyeon Kim의 석사학위 논문을 인준함

지도교수 김 영 준 _____

심사위원 민 동 보 _____

오 유 란 _____

김 영 준 _____

이화여자대학교 대학원

Table of contents

Table of Contents	1
List of Figures	1
Abstract	1
I. Introduction	1
A. Motivation.....	1
B. Research Goal.....	2
C. Main Contribution.....	2
D. Organization.....	3
II. Related Works	5
A. Photorealistic Synthetic Dataset.....	5
B. Object Position Translation	7
1. Image-to-Image Translation.....	7
2. Deep Image Manipulation via Latent Space Exploration	8
C. Object Arrangement	10
III. System Overview	12
IV. Organized Scene Generation	15
A. Preparing a dataset	15
B. Object Arrangement using image-to-image translation.....	19
1. The Pix2PixHD Baseline	20
2. Swapping Autoencoder Baseline.....	22
3. The Proposed Model for Object Position Translation.....	24
C. Texture Reconstruction.....	28
V. Object Pose Estimation	30
VI. Experimental Results	31

A. Implementation details	31
B. Robot Simulation.....	32
1. Simulation Setup.....	32
2. Simulation Results.....	34
C. Comparison against baselines.....	37
D. Ablation Study.....	39
E. Discussion.....	41
VII. Conclusion.....	43
Bibliography	45
국문초록.....	51

List of Figures

Figure 1. Photorealistic synthetic datasets	6
Figure 2. YCB object and model set [3]	6
Figure 3. Cycle-consistency loss [21].	8
Figure 4. Style-based generator [26].....	9
Figure 5. Object Arrangement using user positive data	11
Figure 6. System Overview.....	14
Figure 7. The 10 YCB objects and their appearance count in the entire scene.....	16
Figure 8. Our custom dataset using YCB object models [3].....	18
Figure 9. The network architecture of pix2pixHD [1]	21
Figure 10. Swapping Autoencoder Model [2].....	23
Figure 11. The results of semantic segmentation using pix2pixHD model [1].....	25
Figure 12. Our network architecture for object arrangement.....	26
Figure 13. Object pose alignment at the semantic level	27
Figure 14. Results of texture reconstruction	29
Figure 15. Fetch robot spawned in a Gazebo environment.....	32
Figure 16. Gazebo simulation (top left) and displaying the camera topics (bottom) published from the head camera using visualization tool RViz (top right).....	33
Figure 17. Results of the arranged scene and the robot simulation in Gazebo	36

Figure 18. Comparison of our model against other baselines.....38

Figure 19. Results of object location translation with and without using structural
loss ...39

Summary

In this dissertation, we propose a framework that enables robots to arrange objects in a chaotic scene autonomously. Previous research showed that robots need to be provided with a given goal or a guided plan from users to perform manipulation tasks, such as packing objects into boxes and carrying objects around complex obstacles.

Without the provision of a goal in the form of a human command, a robot can generate an aligned scene by itself. To attain arranged scenes, we obtain semantic masks and translate object poses from an initial scene at a semantic level using the existing image-to-image translation model [1]. Our model also utilizes disentanglement and code swapping approaches [2] to reconstruct the RGB textures from the aligned semantic images. To obtain our goal, it is necessary to select a proper dataset; however, the pre-existing datasets are not appropriate for our object arrangement task. Thus, we construct a photorealistic synthetic dataset for the arrangement task, which consists of YCB object models [3].

After accomplishing the goal scenes, we enable the manipulator to organize objects using sample-based motion planning algorithms. We test and demonstrate that the robot can autonomously set goals and successfully carry out object arrangement tasks on the table.

I. Introduction

A. Motivation

Recently, there have been increasing concerns about autonomous robots that perform various high-level tasks under complex situations. To become more intelligent, robots need to cope with their surrounding environments and read the context of various settings. For these reasons, robots must be able to process visual data from their environments and analyze the relationships between objects.

With the desire to develop robots with autonomy, many researchers have focused on robot manipulation utilizing robot learning approaches. For robot grasping, the existing research has adopted deep learning techniques to establish a robust grasping algorithm [2, 3, 4]. For instance, 6-DOF GraspNet from Mousavian et al. [4] utilized variational autoencoders for sampling a set of grasps and for assessing and refining the sampled 6D grasp poses. Moreover, the previous research on 6 DoF object-pose estimation [5, 6, 7], which is crucial for real-world robot applications, has introduced deep neural networks for pose estimation to enhance performance in the presence of lighting variation and occlusion.

Although the robots can perceive the environments, the goals to carry out the manipulation tasks must be provided through human commands. However, it is inconvenient and even burdensome for the user to deliver the guides and sequential goals to intelligent agents every time. Therefore, we focus on the question of *what if robots can generate goals by themselves and carry out feasible actions without having to be guided by a human?* Until

relatively recently, there have been just a few existing studies [10, 11] that have answered this question. Specifically, Kang et al. [11] proposed a method to automatically arrange objects using task and motion planning. However, in previous works, the robots needed to be guided by some positive user examples and, thus, did not cover many general cases. Therefore, rather than being provided with goals given by humans, we aim to accomplish an autonomous robotic object arrangement in the robot simulation.

B. Research Goal

We propose a novel framework for the robot’s autonomous goal generation, in particular, to perform object arrangement tasks. In our framework, the robot can automatically suggest target-aligned scenes without human commands using a novel combination of image manipulation, such as image-to-image translation and style transferring based on deep learning, object pose estimation, and task and motion planning. Unlike the previous robot object arrangement studies, our work enables a robot to carry out feasible actions without human guidance or interference.

C. Main Contributions

Thus, our main contributions are as follows:

- A mobile manipulator generates a target-aligned goal from an initial messy scene by itself.
- To focus on the object classes, we perform the object position translation on a semantic level and use a structural loss by comparing the structures between the generated and the ground truth of aligned scenes.
- The robot successfully performs object arrangement tasks from an initial scene to its autonomously-generated scene using task and motion planning.

We utilized the pix2pixHD model from Wang et al. [1] with image-to-image translation and applied the disentanglement method from Park et al. [2] to decompose images into the texture and structure codes. Translating images to their target domains enabled the obtaining of semantic masks from the input RGB images and the organization of object poses. Also, decomposing images into their components made the model adjust each component and enhanced the accuracy of synthesized results. We also performed the robot simulation in which the robot could successfully carry out a series of feasible actions and achieve the goals for object arrangement.

D. Organization

The rest of the dissertation is organized as follows. Section II presents the previous research on a photorealistic dataset, image-to-image translation, and object arrangement. Section III briefly shows the overview of our system, and Section IV explains the baselines and our model about semantic segmentation, object position translation, and texture reconstruction. Section V demonstrates the approach to estimating the object poses from the RGB images. Then, in Section VI, our framework is compared with different models, and the ablation test is conducted without using the structural loss to show the performance and robustness of our model. Also, we demonstrate the robot simulation in the Gazebo environment. Finally, we give conclusions in Section VII.

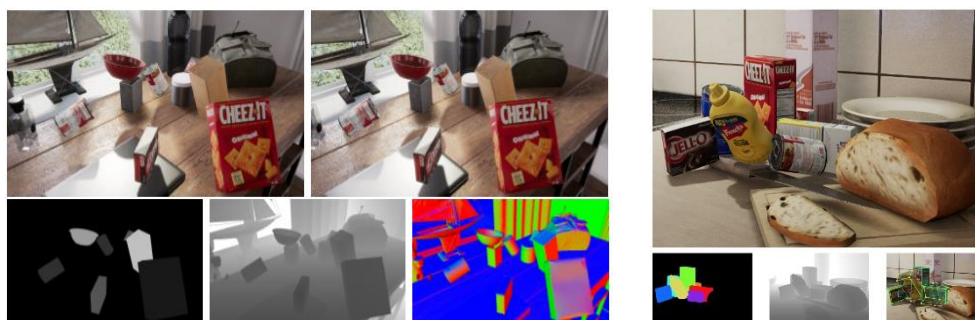
II. Related Works

For objects' position translation, our framework is primarily based on image-to-image translation works. Specifically, we used the conditional GAN framework from pix2pixHD [1] to translate from a messy scene to an ordered one because it showed the best performance at this task compared to other previous methods.

A. Photorealistic Synthetic Dataset

Recently, there have been rising demands to utilize synthetic data for training deep neural networks due to its unlimited amount of pre-labeled training data and prevention from overfitting. In particular, using a photorealistic synthetic dataset lessens the burdensome annotation and even enhances its accuracy. The use of synthetic data for training deep neural networks has gained in popularity, as we can see in the following dataset: SIDOD [12], Falling Things (FAT) Dataset [13], SceneNet RGB-D [14], and others (shown in Figure 1).

However, there was no appropriate photorealistic dataset for our object arrangement task. As in the existing works, we proposed our own dataset, which consisted of the YCB dataset [3], a benchmark for robot manipulation. The YCB dataset [3] (shown in Figure 2) is constituted of several household objects and is widely used to test and evaluate robot tasks such as grasping and domain adaption.



(a) SIDOD dataset [14]

(b) FAT dataset [15]



(c) SceneNet RGB-D dataset [14]

Figure 1. Photorealistic synthetic datasets



Figure 2. YCB object and model set [3]

B. Object Position Translation

Since Goodfellow et al. [15] proposed a revolutionary framework, Generative Adversarial Networks (GANs) have widely applied semantic segmentation, local and global image editing, and image style transfer to various image-to-image translation tasks. We used a conditional generative model for object position translation and an autoencoder for texture reconstruction to successfully arrange objects in images.

1. Image-to-Image Translation

Since Isola et al. [16] proposed a pix2pix framework using an adversarial method that translates images from the input domain to the output domain, image-to-image translation using the adversarial loss [15] rather than the L1 loss has become a broadly treated problem [7, 8, 9]. The pix2pix framework [16] mapped input images to output images through learning conditional GANs and utilized this method for various tasks such as generating cat photos from user sketches. However, it is difficult to apply this method for high-resolution images (such as 1,024 x 1,024). The pix2pixHD model [1] extended this method to facilitate the image-to-image translation with high-resolution images by adopting a multi-scale discriminator and coarse-to-fine generator. Although these frameworks produced sufficient performances, they required a large amount of labeled data.

Recently, to overcome the limitation, there has been an increased need for unsupervised learning for image-to-image translation [10, 11, 12]. Specifically, a CycleGAN from Zhu et al. [21] adopted a cycle-consistency loss (as shown in Figure 3) to transfer image

styles without using a labeled dataset.

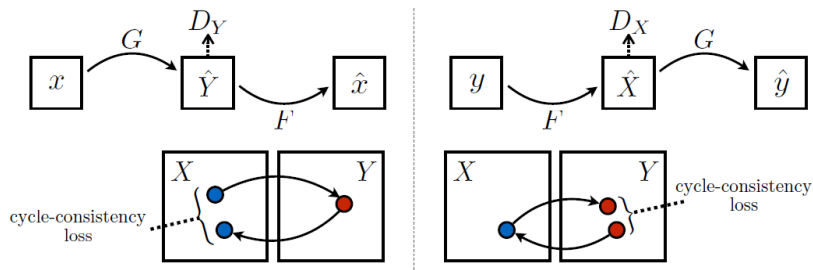


Figure 3. Cycle-consistency loss [21].

Even though the unsupervised learning methods have advantages over the supervised ones, our system required the high performances of output aligned images and, thus, referred to the pix2pixHD model.

2. Deep Image Manipulation via Latent Space Exploration

Disentanglement [13, 14, 15, 16] that separates inputs into independent latent matrices or vectors enables a profound understanding of image manipulation and even the subtle adjustment of those components for generating many realistic and reliable outputs. Karras et al. [26] also proposed a generator architecture StyleGAN, which learns high-level attributes such as the pose and identity of human faces. To leverage the performance, they embedded the input latent code into an intermediate latent space. Also, StyleGAN2 [27] and Image2StyleGAN [28] extended the embedding latent space used in StyleGAN to reconstruct the images with a style-based generator (shown in Figure 4).

Rather than sampling the latent codes from a fixed distribution such as a Gaussian distribution, the swapping autoencoder model from Park et al. [2] learned the latent code space and thus decomposed the images into structure and texture codes, which are its representative components. Then, it shuffled those codes between two images to manipulate images locally and globally, like other previous code-swapping approaches [16, 21]. We adopted Park et al.’s model [2] because it is more promising for the reconstruction of our texture reconstruction task compared to other previous methods.

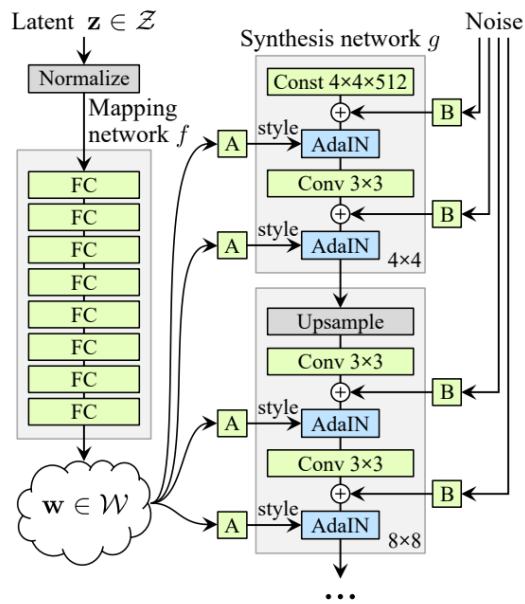


Figure 4. Style-based generator [26]

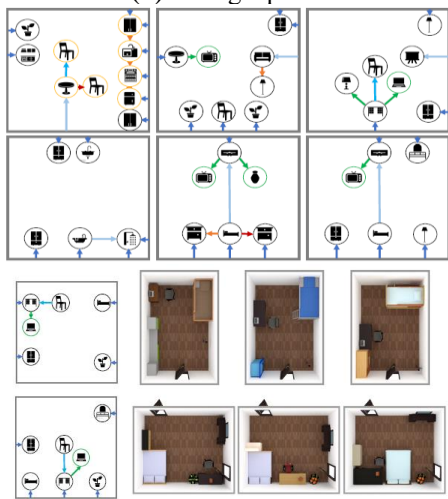
C. Object Arrangement

Fisher et al. [30] proposed a method for synthesizing 3D object arrangements from a few user-provided examples. They classified object groups into contextual categories in the scenes, trained a probabilistic model using a small number of user preferences, and utilized the model as a guide to organizing the scenes. Wang et al. [31] generated an object relation graph by encoding objects as nodes and spatial object relationships as edges using a deep graph convolutional generative model for indoor scene synthesis. Also, other previous works [31, 32, 33] used deep generative models and convolutional neural networks to generate synthesized indoor scenes by collecting reference scenes from users.

In Abdo et al. [10], the robot performed object arrangement in tidy-up tasks such as organizing a shelf or sorting objects in boxes through collaborative filtering. Like the existing works [29, 30], Abdo et al. grasped user preferences and performed object grouping using collected and crowdsourced data. Kang et al. [11] suggested an approach to automatic object arrangement using task and motion planning. They also extracted object relationships of target scenes by collecting user-preferrable examples. Although our framework is inspired by this work [11], we decided to automatically generate goal scenes by utilizing deep learning approaches rather than establishing object relationship graphs.



(a) Using a probabilistic model and contextual object categories [30]



(b) A deep graph convolutional generative model [31]



(c) A CNN-based model [32]

Figure 5. Object Arrangement using user positive data

III. System Overview

To accomplish our goals, we divided our system into three steps, as shown in Figure 6: organized scene generation (step 1), object pose estimation (step 2), and task and motion planning (step 3). In the first step, we aligned the object poses from input messy scenes by utilizing the image-to-image translation approach. We used our custom dataset for our object arrangement task, which consisted of the messy scene images and their corresponding arranged images. We estimated the individual object’s poses from both input and goal scenes in step 2 and carried out motion planning from the initial object poses to the goal poses in the robotic action in step 3.

In order to organize the scenes, we needed to create a proper dataset, as the existing indoor scene dataset was not appropriate for arrangement tasks. Thus, we constructed a dataset that consisted of unkempt scene images and their corresponding clean scene images using Unreal Engine 4 and a customized plugin, the Nvidia deep learning dataset synthesizer (NDDS) [35] provided by NVIDIA. During dataset generation, we considered physical conditions such as gravity. The dataset also contained semantic masks of both organized and unorganized scenes to facilitate object position translation. These image sets were used to perform semantic segmentation, object arrangement, and texture reconstruction.

After the dataset construction, at step 1, as shown in Figure 6, we utilized the image-to-image translation works, such as the pix2pixHD model [1], to accomplish the arrangement task. To translate object location, we decomposed the problem into three stages: semantic segmentation, object pose arrangement, and texture reconstruction. While they may require additional processes to obtain, semantic masks allow the training model to concentrate on the

information of object position by removing the effects of texture information from images. Since pix2pix and pix2pixHD models were devised to target a segmentation task and its inverse, we adopted the pix2pixHD model for semantic segmentation and semantic level object arrangement. We needed to adopt additional loss terms to enhance the accuracy of object arrangement tasks because the model was not designed for them. Therefore, we introduced a simple loss called latent loss that compared the structure codes between generated images and the ground truths. Through comparing these, we enhanced the accuracies of the model.

Finally, the aligned semantic images should recover their textures to detect object poses because the current object pose estimation techniques require RGB textures to predict their exact geometries. The swapping autoencoder model [2] proposed that the system manipulates images using an encoder and generative adversarial networks. It separated input images into a structure code and a texture code using the encoder. Then, it shuffled those latent codes between two images to effectively produce a realistic manipulated image. Our model integrated the swapping model into our system to recover RGB textures from semantic images. For texture recovery, the aligned semantic images were used as a structure code, while the chaotic RGB images were used as a texture code. Then, we shuffled both components extracted from those images and successfully achieved goal scenes, the aligned RGB images.

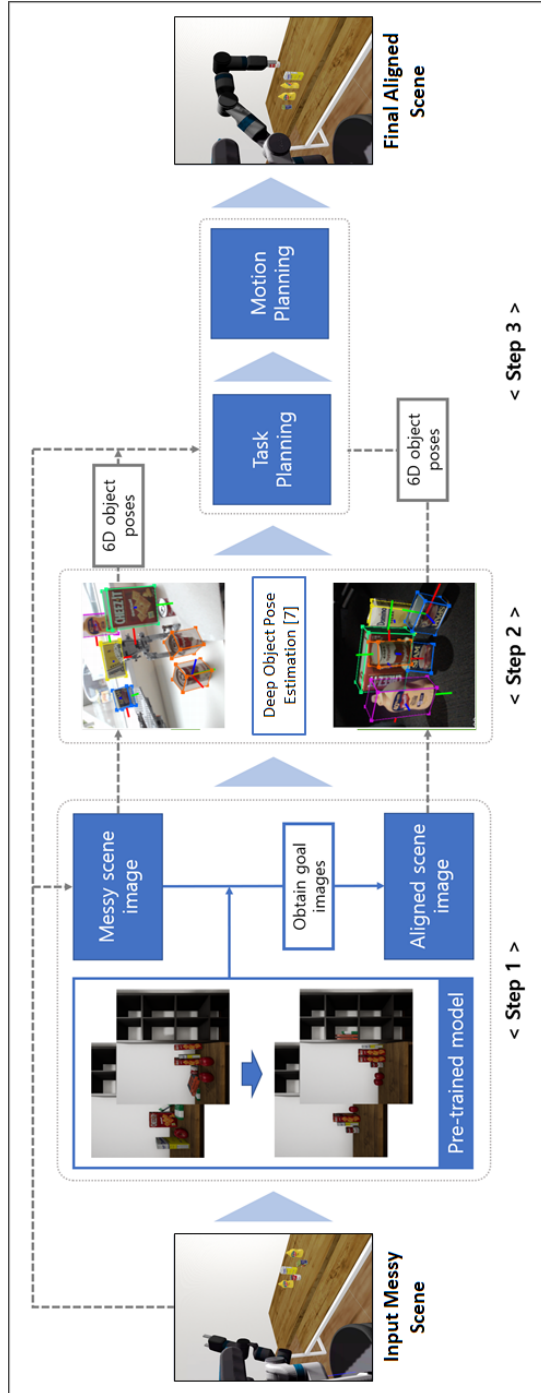


Figure 6. System Overview

IV. Organized Scene Generation

A. Preparing a Dataset

Before conducting autonomous goal generation, we created our dataset devised for the object arrangement task. To construct the dataset, we imported ten household objects from the YCB dataset [3] such as a tomato soup can, an apple, or books (shown in Figure 7), which are widely used benchmarks in robot manipulation, into a virtual environment within Unreal Engine 4 (UE4) (shown in Figure 8). Multiple YCB objects were randomly placed on the table or added to the bookshelf in a messy scene. For generating reliable chaotic scenes, the objects were dropped at random orientations and positions by gravity. The aligned scene contained the same number and classes as the messy scene. We manually arranged the scene by clustering the objects with the same classes. After deploying objects, the images from the organized and disorganized scenes were generated by a custom UE4 plugin [35] offered by NVIDIA.

To prevent the training model from overfitting issues, various scene images were captured by four different camera positions; each pair of messy and aligned scenes shared the same camera's location. We trained the object arrangement model with an image resolution of 640 x 480 pixels based on the pix2pixHD model. Note that the dataset was constituted of 1,000 pairs of the original RGB and its corresponding semantic mask images for both the chaotic and arranged scenes.

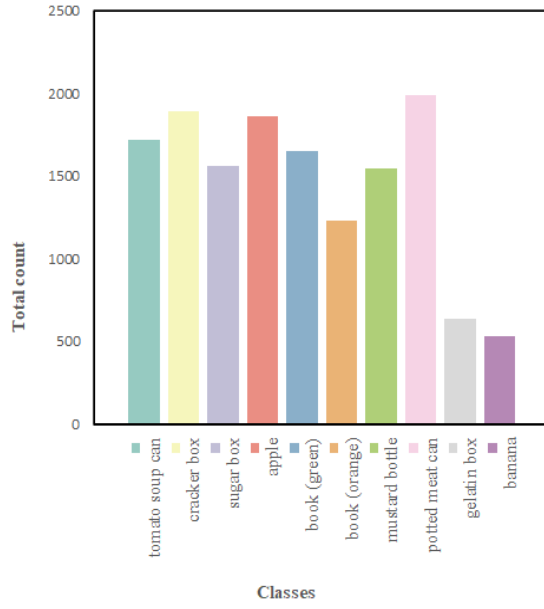
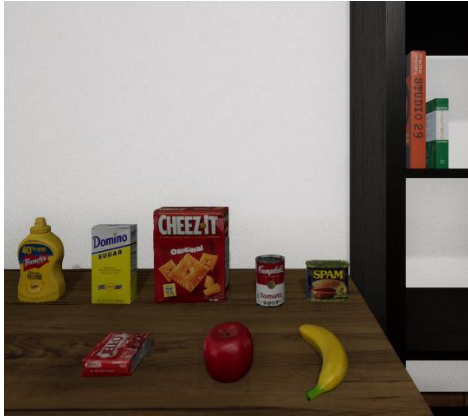
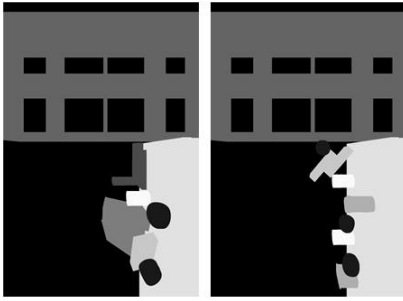
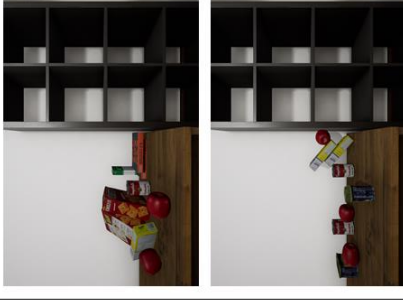
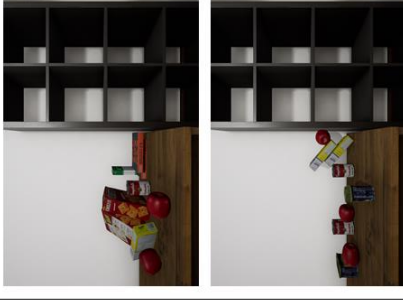

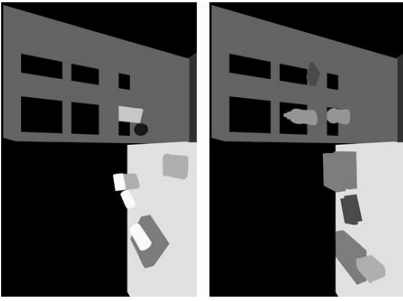





Figure 7. The 10 YCB objects and their appearance count in the entire scene.

	Messy Scene	Aligned Scene
View 1		
		
View 2		
		

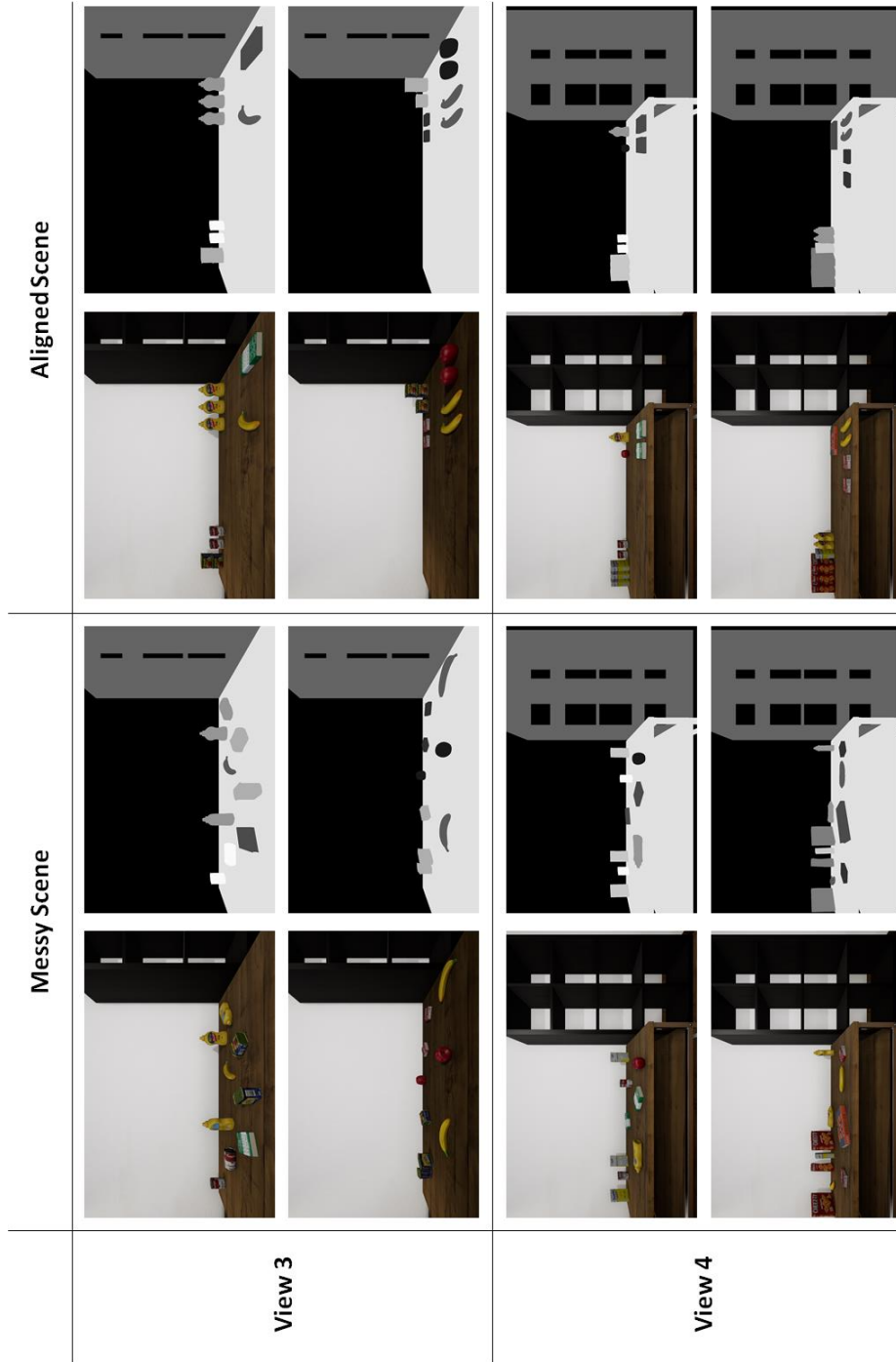


Figure 8. Our custom dataset using YCB object models [3]

B. Object Arrangement using image-to-image translation

For object position translation, our framework was largely based on pix2pixHD [1]. We used the conditional GAN framework from pix2pixHD to translate from a messy scene to an ordered one because it shows the best performance at this task compared to other previous methods. However, those existing cGAN strategies were not appropriate for our arrangement task to a large extent because those were devised to treat local or global editing or style transferring. To generate much more accurate scenes, we thus introduced structural loss into the discriminator and used semantic masks to concentrate on object class information.

Our model took a 3D tensor (size $3 \times 512 \times 512$) of RGB image as an input. Then, it sequentially learned to generate its corresponding semantic masks and to translate object pose at the semantic level. Translating object poses at the semantic level may enable the model to avoid being distracted by unnecessary information under abundant textures and, thus, concentrate on their location information. During training the object pose translation, the latent loss was adopted to organize the object positions in semantic masks. Afterward, we used the autoencoder from Park et al. [2] to apply RGB textures into semantic masks. The encoder separated input images into both the texture and structure code, which represented both RGB texture information and skeletal structure component from the images. Then, it swapped the texture codes from messy RGB images and structure codes from the aligned semantic masks to create the goal images combining both the original RGB textures and the semantic structures.

1. The Pix2PixHD Baseline

We adopted a pix2pixHD model from Wang et al. [1] as the baseline of our model for semantic segmentation and object position translation. The pix2pixHD model is a supervised learning framework based on the pix2pix [16] model, which is constituted of a generator G and a discriminator D . It receives an input image x_i and a corresponding target output image y_i as input training data. It uses an objective function of a conditional GAN represented as:

$$L_{GAN}(G, D) = E(x, y)[\log D(x, y)] + E_x[\log(1 - D(x, G(x)))] \quad (1)$$

Thus, it strives to find the ideal generator which minimizes the loss function, while the discriminator tries to maximize it:

$$\min_G \max_D L_{GAN}(G, D) \quad (2)$$

In addition, it uses a coarse-to-fine generator, a multi-scale discriminator, and an additional adversarial loss because it aims to enhance the quality of the model for high-resolution images ($> 512 \times 512$).

To deal with high-resolution images, the generator was decomposed into two sub-networks: the global generator G_1 and the local generator G_2 (as shown in Figure 9). The global generator operated the original images, while the local generator concentrated on local areas of interest through increasing the image size.

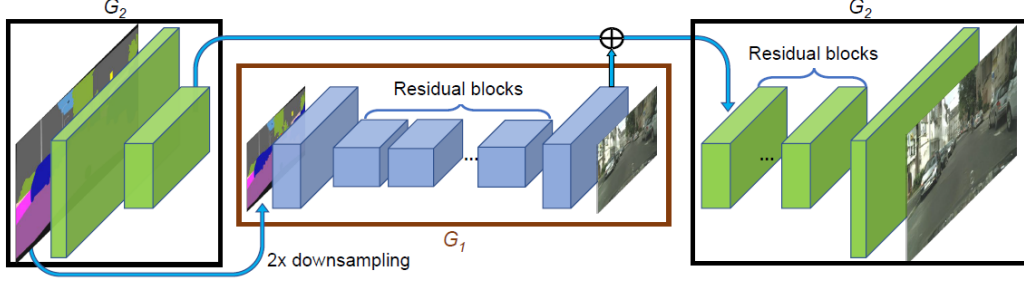


Figure 9. The network architecture of pix2pixHD [1]

Also, Wang et al. [1] proposed using multi-scale discriminators at different image scales to cope with both the network capacity and overfitting at the same time. The three different discriminators D_1, D_2, D_3 were used in the network structure to distinguish between the original and synthesized images at three different levels. Thus, it was able to adjust from the coarse-scale to the fine-scale of the image and even assist the coarse-to-fine generator. Thus, the equation (1) was transformed into the equation (3) below:

$$\min_G \max_{D_1, D_2, D_3} \sum_{k=1,2,3} L_{GAN}(G, D_k) \quad (3)$$

Finally, Wang et al. enhanced the GAN loss by matching intermediate features between real and generated images. The feature mating loss is expressed as:

$$L_{FM}(G, D_k) = E_{(x,y)} \sum_{i=1}^T \frac{1}{N_i} [\|D_k^{(i)}(x, y) - D_k^{(i)}(x, G(y))\|_1] \quad (4)$$

where $D_k^{(i)}$ is an i th-layer discriminator, T is the total number of layers, and N_i is the number of elements in i th-layer. By incorporating the equation (3) and (4), the full objective function is represented as:

$$\min_G \left(\left(\max_{D_1, D_2, D_3} \sum_{k=1,2,3} L_{GAN}(G, D_k) \right) + \lambda_1 \sum_{k=1,2,3} L_{FM}(G, D_k) \right) \quad (5)$$

2. Swapping Autoencoder Baseline

Because recent object pose estimation techniques require RGB textures to detect 6D poses, it was necessary to reconstruct RGB textures of images rather than use semantic mask images. Thus, we adopted a deep image manipulation method [2], which decomposed an image into a structure and a texture and enforced to swap the latent components between two different images.

Park et al. encoded each image into two latent codes using an encoder E and reconstructed both latent components into the original images with a generator G (as shown in Figure 10). The image reconstruction loss $L_{rec}(E, G)$ and the non-saturating adversarial loss $L_{GAN,rec}(E, G, D)$ were used to confirm whether both codes are successfully separated:

$$L_{rec}(E, G) = E_{x \sim X} \|x - G(E(x))\|_1 \quad (6)$$

$$L_{GAN,rec}(E, G, D) = E_{x \sim X} [-\log(D(G(E(x))))] \quad (7)$$

, where a discriminator D is applied to make the images more realistic.

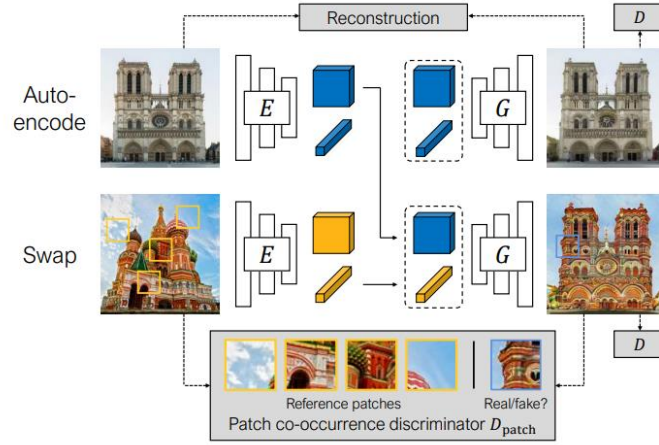


Figure 10. Swapping Autoencoder Model [2].

Furthermore, latent space \mathbf{Z} of images were separated into two latent components $\mathbf{z} = (\mathbf{z}_s, \mathbf{z}_t)$, which consisted of a structure component \mathbf{z}_s and a texture component \mathbf{z}_t . Then, Park et al. enforced the latent components from the hybrid images to produce realistic images using the GAN loss. With randomly sampled two images x^1 and x^2 , the adversarial loss on the hybrid images is calculated as:

$$L_{GAN,swap}(E, G, D) = \mathbb{E}_{x^1, x^2 \sim X, x^1 \neq x^2} [-\log(D(G(\mathbf{z}_t^1, \mathbf{z}_s^2)))] \quad (8)$$

Additionally, the co-occurrent loss that compared the textures between the chaotic input image M_{rgb} and the synthesized RGB image O_{rgb} was added. According to Park et al., both images have a consistency of textures and, therefore, the loss term makes their textures

indistinguishable. They randomly extracted patches from the image and compared those patches between images rather than the whole image.

$$L_{CooccurGAN}(E, G, D_{patch}) = \mathbb{E}[-\log(D_{patch}(crop(G(\mathbf{z}_t^1, \mathbf{z}_s^2)), crops(M_{rgb})))] \quad (9)$$

, where the crop is randomly selected between 1/8 to 1/4 of the original images.

Thus, the total objective function for texture reconstruction is written as $L_{total} = L_{rec} + 0.5L_{GAN,rec} + 0.5L_{GAN,swap} + L_{CooccurGAN}$.

3. The Proposed Model for Object Position Translation

Although a natural RGB image contains texture-rich information, it hindered our model from successfully translating object position because it is challenging to infer exact class information of every pixel due to the shadows and various colors of an object. Therefore, we conducted semantic segmentation (as shown in Figure 11) and used a semantic mask to focus on each object’s class information. It prevented our model from being affected by noises and enabled it to translate the poses effectively.

The existing frameworks [15, 18, 20] were devised for local or global image style editing and texture transfer, not for translating object position. Even though the previous model has a possibility of performing our object arrangement task to some extent, it was necessary to introduce an additional loss term to enhance the performance of the model.

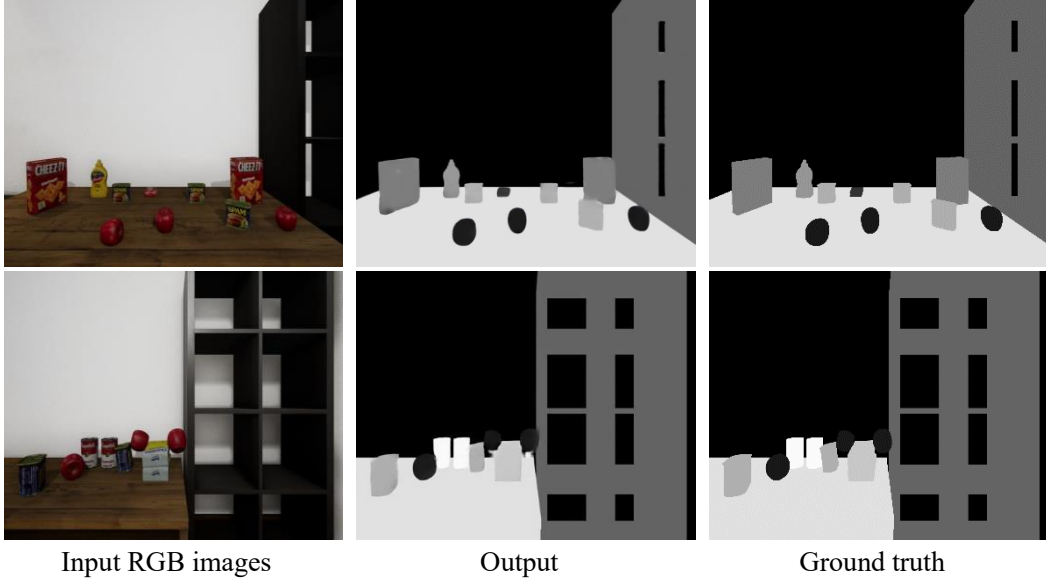


Figure 11. The results of semantic segmentation using pix2pixHD model [1]

Thus, like Park et al. [2], we separated images into two latent components $\mathbf{z} = (\mathbf{z}_s, \mathbf{z}_t)$ using encoder E and confirmed whether the encoder successfully decomposes both latent components by using the decoder and its discriminator. Then, we introduced a structural loss, which compared the structural components between an arranged image and its ground truth. The latent loss comparing both structure codes is expressed below:

$$L_{LATEENT}(G, D) = E(\mathbf{z}_s^y, \mathbf{z}_s^x)[\log D(\mathbf{z}_s^y, \mathbf{z}_s^x)] + E_{\mathbf{z}_s^y}[\log(1 - D(\mathbf{z}_s^y, G(\mathbf{z}_s^y)))] \quad (10)$$

where \mathbf{z}_s^y is a structure code of the synthesized image and \mathbf{z}_s^x is that of the ground truth.

The objective function for object position translation can be transformed as:

$$\min_G \left(\left(\max_{D_1, D_2, D_3} \sum_{k=1,2,3} L_{GAN} + \lambda_2 L_{LATEENT} \right) + \lambda_1 \sum_{k=1,2,3} L_{FM}(G, D_k) \right) \quad (11)$$

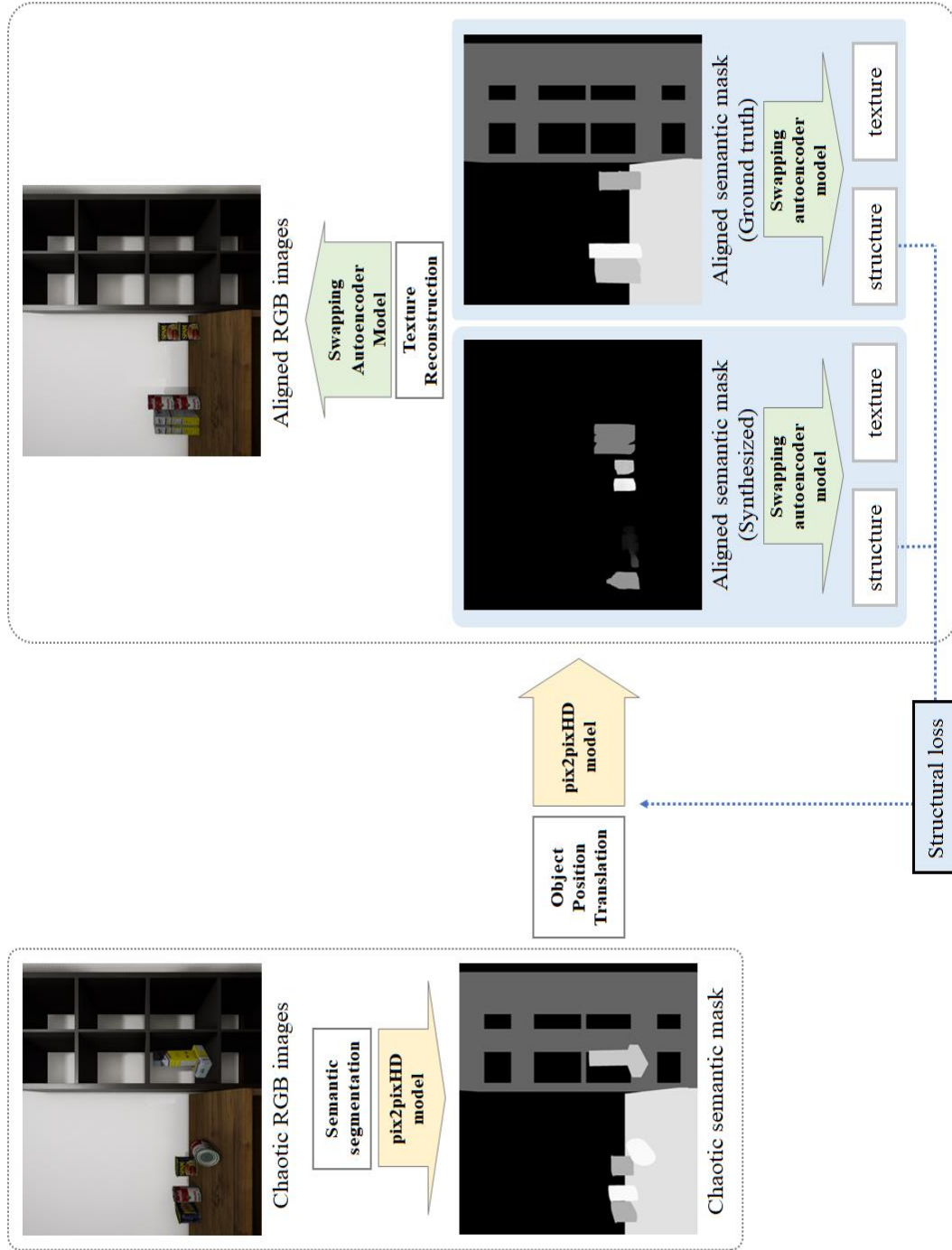
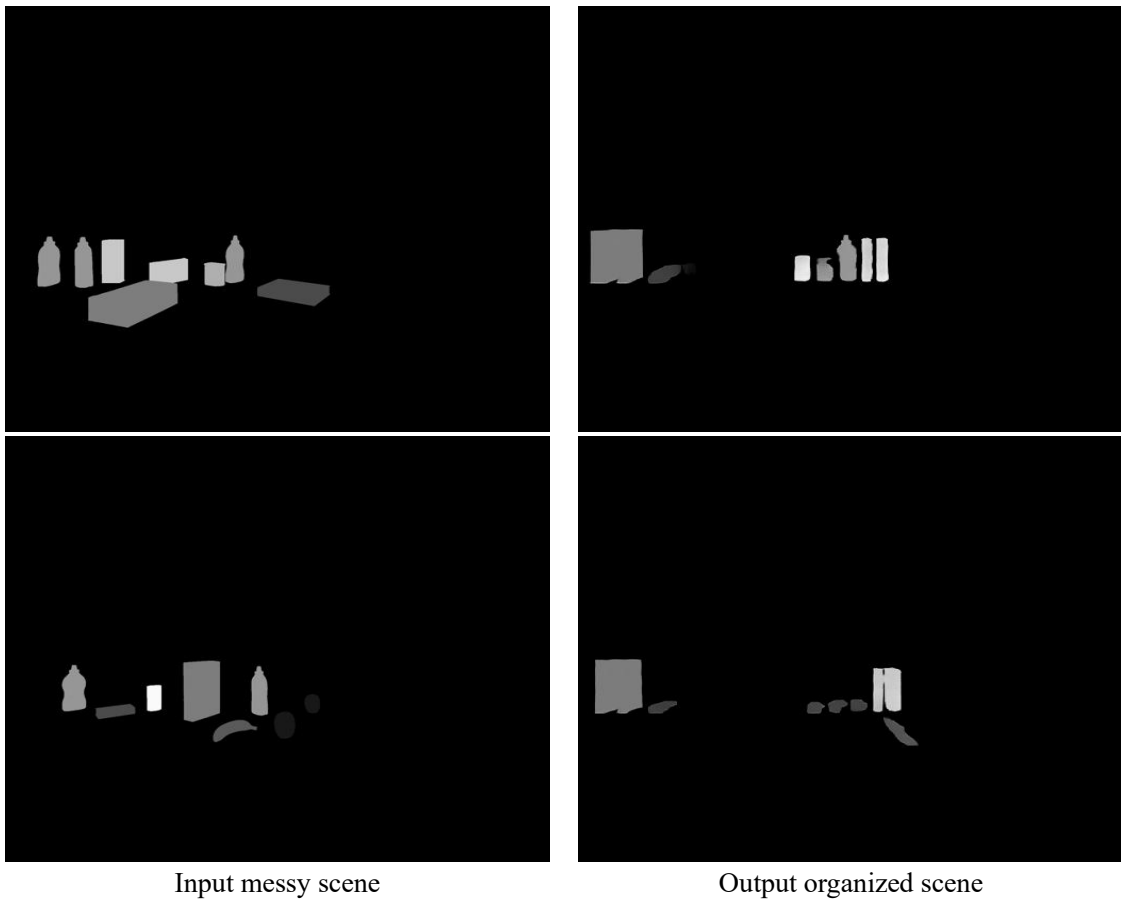


Figure 12. Our network architecture for object arrangement

Therefore, our system can be represented as shown in Figure 12 and the total objective function can be expressed with $L_{total} = L_{rec} + 0.5L_{GAN,rec} + 0.5L_{GAN,swap} + L_{CooccurGAN} + L_{GAN} + L_{LATENT} + L_{FM}$, which integrates the pix2pixHD losses, swapping autoencoder losses, and our structural loss. The model learned the direction of minimizing the total objective function and successfully obtained the aligned scenes at the semantic mask level (as shown in Figure 13).



Input messy scene

Output organized scene

Figure 13. Object pose alignment at the semantic level

C. Texture Reconstruction

While the swapping autoencoder used two natural RGB images as inputs, we used a chaotic RGB image M_{rgb} and an organized semantic image O_{seg} generated by object position translation. As explained above, both images were decomposed into two latent components: $\mathbf{Z}_{M_{rgb}} = (\mathbf{z}_s^{M_{rgb}}, \mathbf{z}_t^{M_{rgb}})$ and $\mathbf{Z}_{O_{seg}} = (\mathbf{z}_s^{O_{seg}}, \mathbf{z}_t^{O_{seg}})$. Then, we shuffled the texture code of the messy RGB image $\mathbf{z}_t^{M_{rgb}}$ and the structure code of the semantic image $\mathbf{z}_s^{O_{seg}}$. Due to the consistency of the number of objects and their classes between input images, it was possible to rebuild the object and background textures of the generated images. Then, the loss term $L_{GAN,swap}$ to generate the hybrid images more realistically is re-written as:

$$L_{GAN,swap}(E, G, D) = \mathbb{E}_{M_{rgb}, O_{seg} \sim X, M_{rgb} \neq O_{seg}} \left[-\log \left(D \left(G \left(\mathbf{z}_t^{M_{rgb}}, \mathbf{z}_s^{O_{seg}} \right) \right) \right) \right]. \quad (12)$$

Also, the co-occurrent loss is expressed as:

$$L_{CooccurGAN}(E, G, D_{patch}) = \mathbb{E} \left[-\log \left(D_{patch} \left(crop(G(\mathbf{z}_t^{M_{rgb}}, \mathbf{z}_s^{O_{seg}})), crops(M_{rgb}) \right) \right) \right]. \quad (13)$$

By converting the input RGB images with the chaotic RGB image and the aligned semantic masks, we successfully obtained the aligned RGB image (as shown in Figure 14) by using the existing networks from the swapping autoencoder [2].

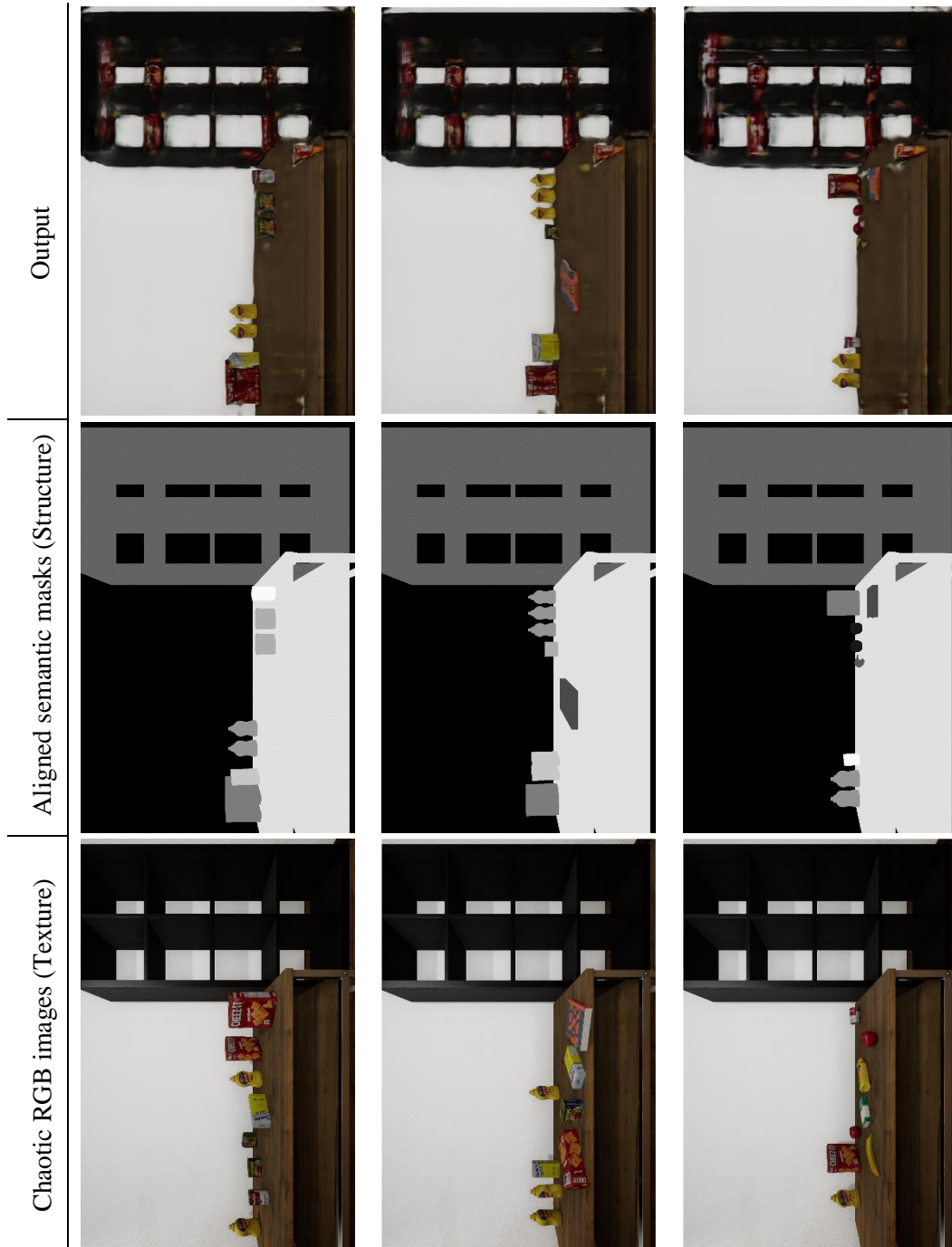


Figure 14. Results of texture reconstruction

V. Object Pose Estimation

We chose the deep object pose estimation (DOPE) framework [7], which detects and estimates the 6-DoF household object poses from a single RGB image. Tremblay et al. [7] used a fully convolutional deep neural network to estimate nine belief maps of 2D keypoints for the projected eight vertices of the 3D bounding boxes and one centroid. After the vertices and centroid of the bounding box have been determined, it utilized a PnP algorithm [36] to extract objects' 3D translation and rotation with respect to the camera.

Like the DOPE system, we used the pre-trained YCB object models [3], such as potted-meat cans and cracker boxes. By integrating the DOPE node with the Gazebo simulation, we identify the 6-DoF geometries of objects in unkempt scenes. For estimating object poses under an arranged scene, we implemented a ROS image publisher node that published the generated organized images and camera information like a fake camera. By offering camera intrinsic parameters and providing 3D coordinate frames, it successfully published the images as if they were observed by the mobile manipulator. Then, the node was linked with the DOPE node for detecting the object poses in the arranged scenes.

VI. Experimental Results

In section VI.A, we explain the experimental setup, specifically, the custom dataset in section VI.B.1, the implementation details in section VI.B.2, the task and motion planning in section VI.C, and the baselines for comparisons in section VI.D. In section VI.E, we show our system results and compare the performance of our model and other benchmarks.

A. Implementation Details

We adopted a variety of techniques to prevent overfitting issues during training: weight decay, data augmentation, and dropout. First, we used the default weight decay parameter from pix2pixHD, which was initially set as 0.0002 during the first epochs and linearly decreased to zero at the last 100 epochs. Second, data augmentation such as flipping was arbitrarily applied to both input images. Third, we randomly dropped out the neurons in the layers and decreased the number of generator filters to 32. The entire network was trained from scratch, using Adam optimizer [37]. We trained all our models on two NVIDIA Titan RTX GPU with 24GB GPU memory. During training image-to-image translation and texture reconstruction, we resized the images with the resolution of 512 x 512 to reduce the total training time. The training time took 4 hours for semantic segmentation, about 13 hours for object position translation, and about a week for texture reconstruction.

B. Robot Simulation

1. Simulation Setup

For our purposes, the ultimate test determines whether our framework is sufficient for the robot arrangement task. We performed a robot simulation in the Gazebo environment under the ROS system. A Fetch robot was initially spawned into the environment (shown in Figure 15) and objects were placed with respect to various scenarios. We evaluated the framework and tested robots in several scenarios.

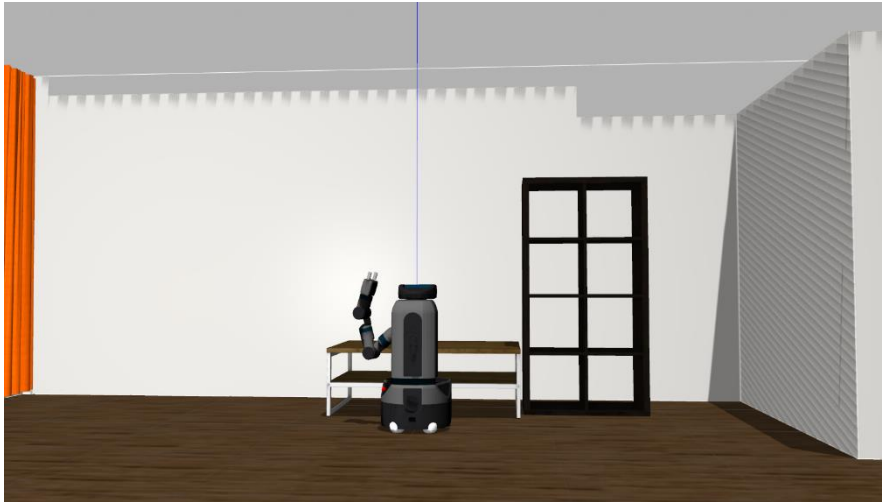


Figure 15. Fetch robot spawned in a Gazebo environment.

During the execution of the feasible actions, we attached objects in a manipulator to prevent them from falling. We then executed motion planning using MoveIt! [38], a motion

planning framework that supports several sampling-based algorithms such as RRT and RRT-Connect. Specifically, we utilized the RRT-Connect algorithm and confirmed that the Fetch robot could successfully perform the tasks. During the simulation, we used the perception and pick-and-place package provided by Fetch robotics to enable the robot to detect the initial object poses for grasping and perform sequential actions. Due to the limited reachable space of the Fetch robot, it was necessary to confine the scenarios and object positions on the table. To check the reachable space, we visualized the head camera topics published from the camera using a visualization tool RVIZ (shown in Figure 16).

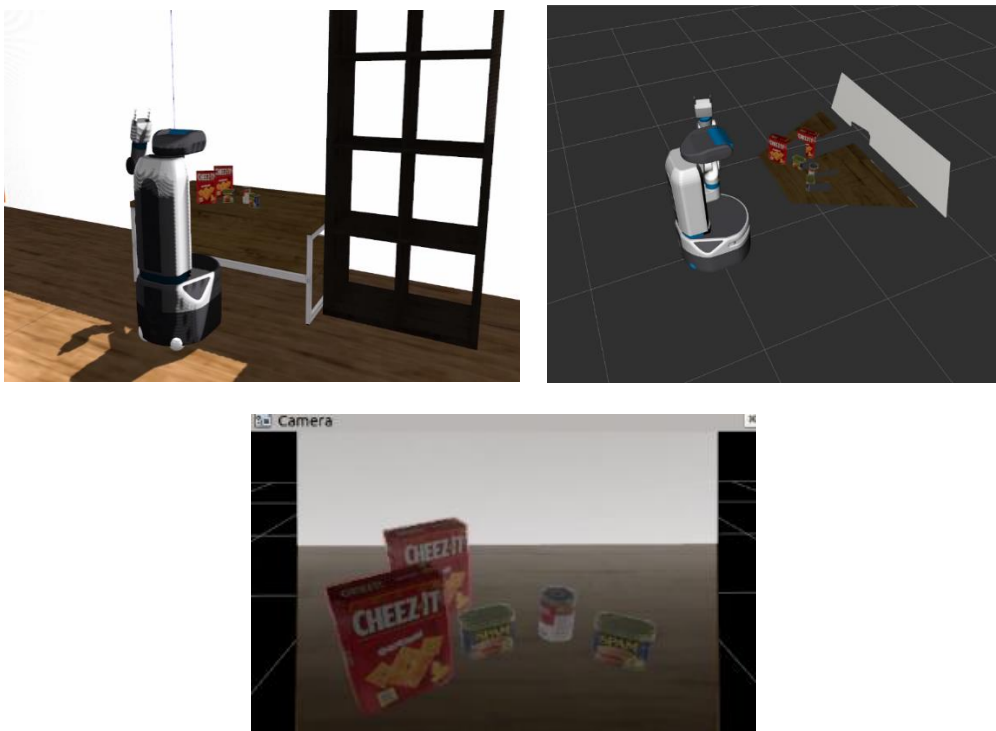
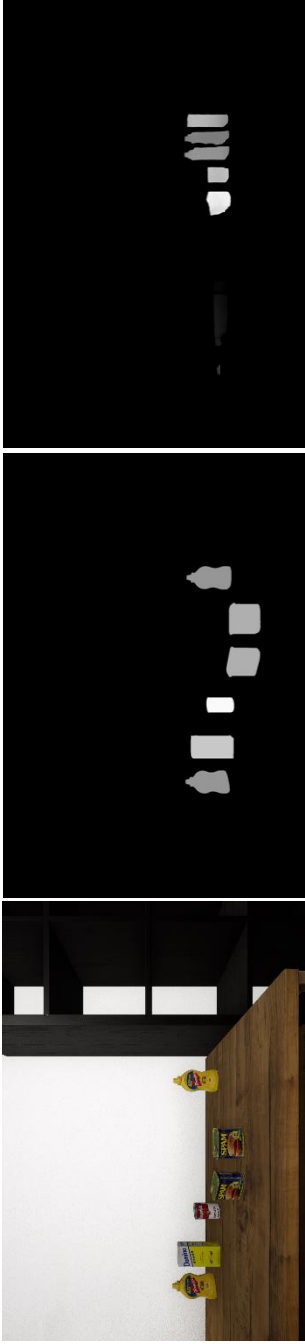


Figure 16. Gazebo simulation (top left) and displaying the camera topics (bottom) published from the head camera using visualization tool RViz (top right).

2. Simulation Results

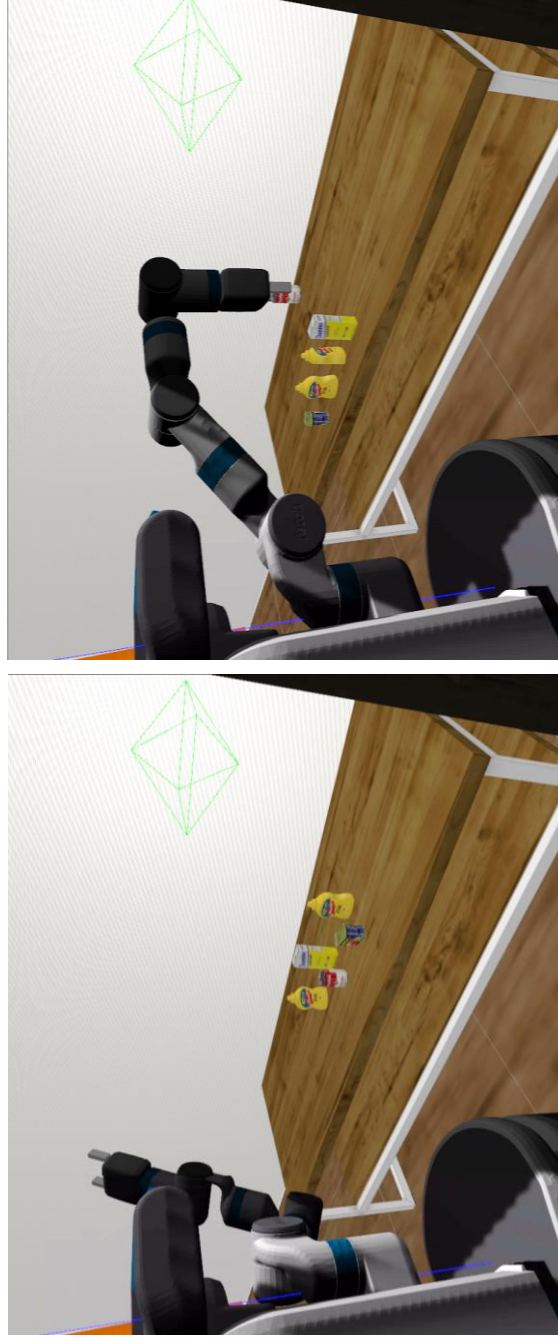
We conducted a robot simulation and checked that the robot could generate an aligned goal from a chaotic input scene and perform the motion planning regarding the following scenarios. Due to the grasping issues, we needed to exclude bulky or flat objects such as a cracker box and a gelatin box. As shown in Figure 17, the robot successfully created an arranged scene and accomplished it by carrying out the sequences of actions.



Input scene

Chaotic semantic mask

Synthesized goal



Input Chaotic scene in Gazebo

Aligned result scene in Gazebo

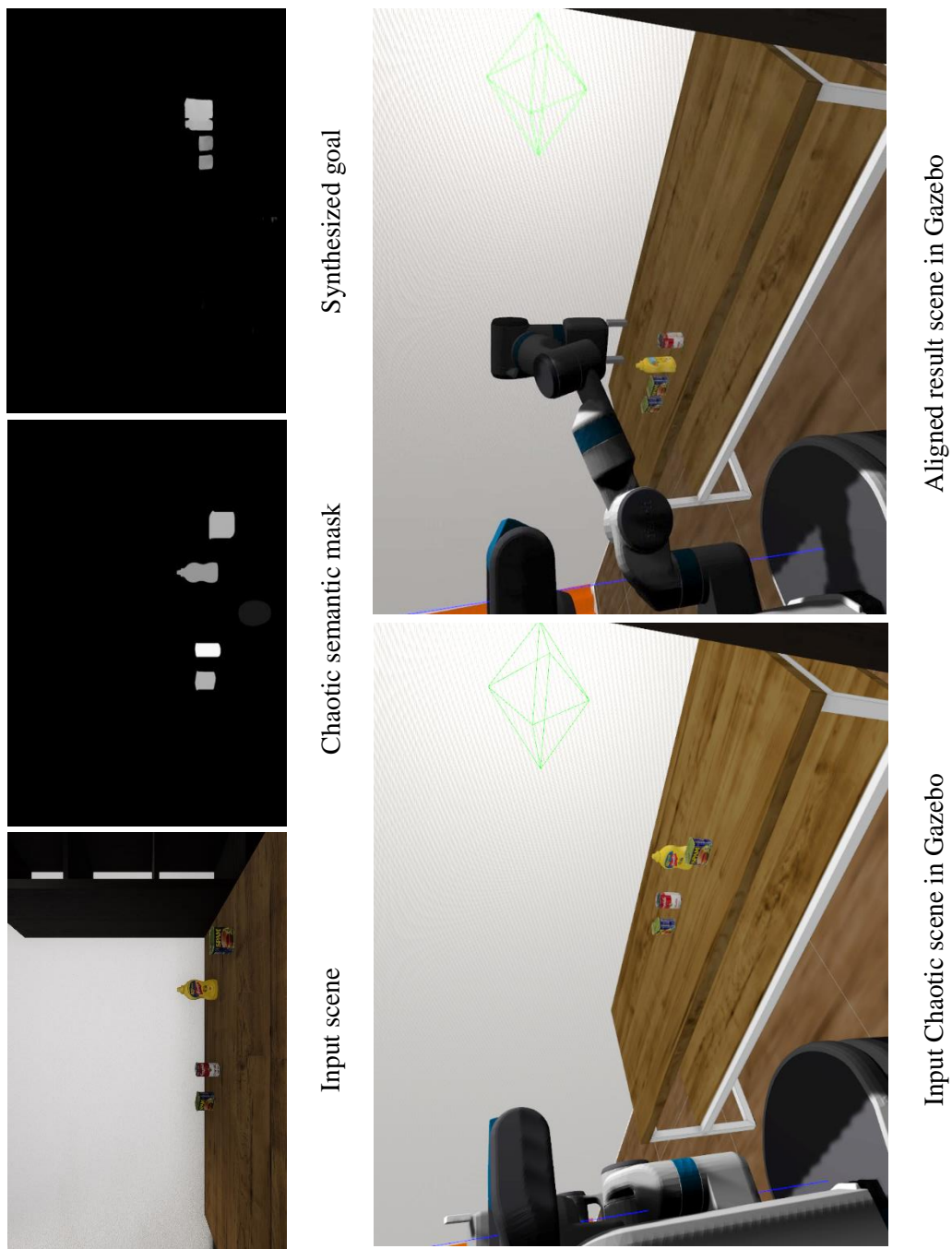


Figure 17. Results of the arranged scene and the robot simulation in Gazebo

C. Comparison against baselines

We compared our framework with two baselines for object location translation: pix2pix [16] and cycleGAN [21]. We trained both models with the same image size, 512 x 512. While conducting the evaluation, we used the default settings from each model and the same details as our model.

Pix2Pix [16] — This method trained on the paired dataset based on the conditional GAN (cGAN). The conditional GANs learn a generator $G: \{x, z\} \rightarrow y$, which is trained to generate a mapping from input image x and random noise vector z to output image y .

cycleGAN [21] — We also compared our model against cycleGAN, which learns a mapping from an input domain X to an output domain Y , $G: X \rightarrow Y$. It proposed an unsupervised approach that coupled it with an inverse mapping $F: Y \rightarrow X$ and introduced a cycle consistency loss aimed to find a generator satisfying $F(G(X)) \approx X$ and vice versa.

As demonstrated in Figure 18, we confirmed that our model using pix2pixHD [1] showed better results than the previous works, pix2pix [16] and cycleGAN [21]. The pix2pix model rarely accomplished the object placement and almost failed to generate the goals. Also, the results using the cycleGAN model were slightly changed, but the object positions were not aligned and remained similar to the input images.

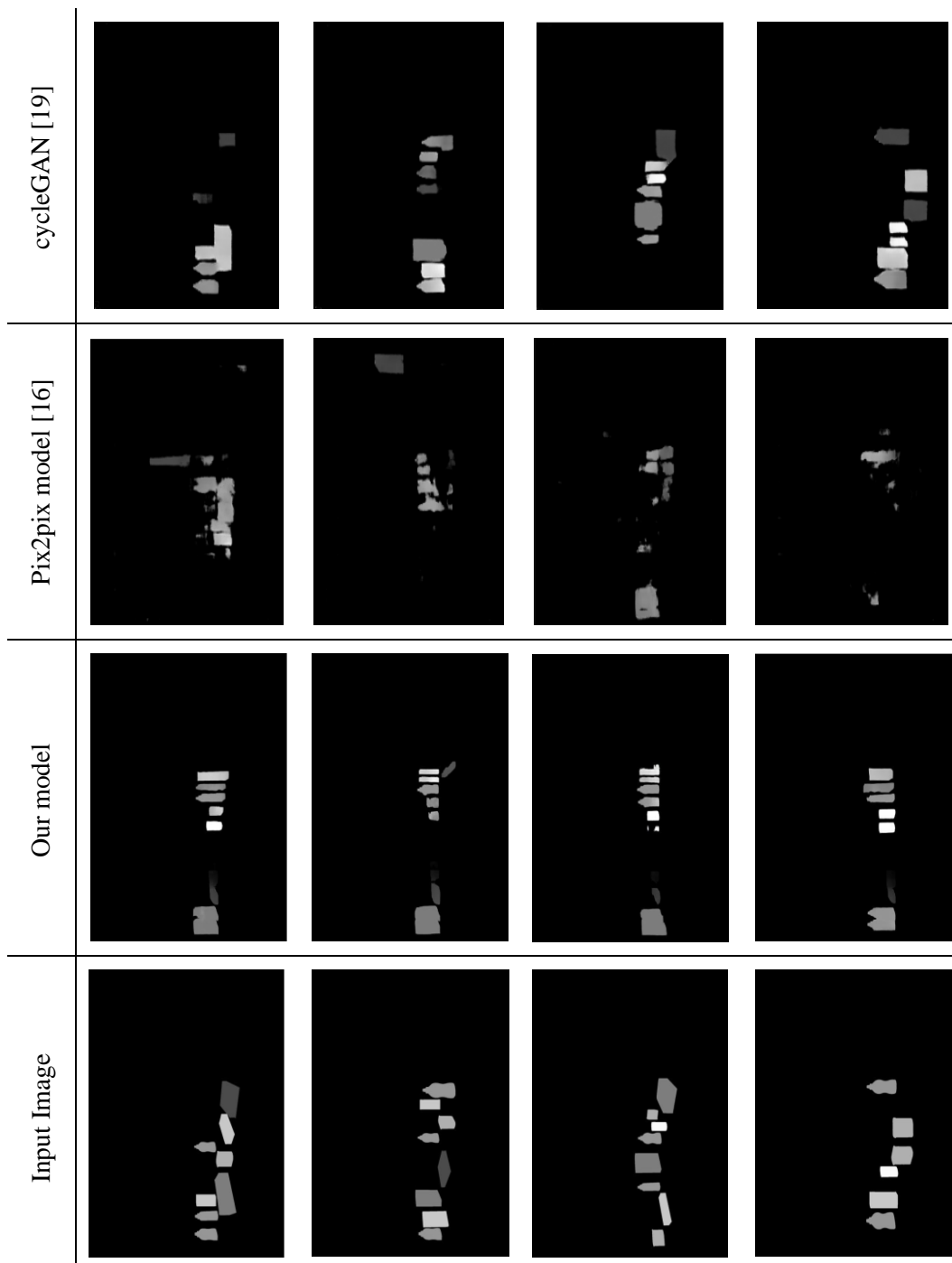


Figure 18. Comparison of our model against other baselines

D. Ablation Study

To show the effect of the structural loss, we perform an ablation study on several scenarios for object position translation at the semantic level. In our framework, the original images were separated into their latent components, and the structure codes from both generated and ground truth images were compared to enhance the result images. To verify the contribution of the structural loss, the model for object location translation was trained without using the structural loss.

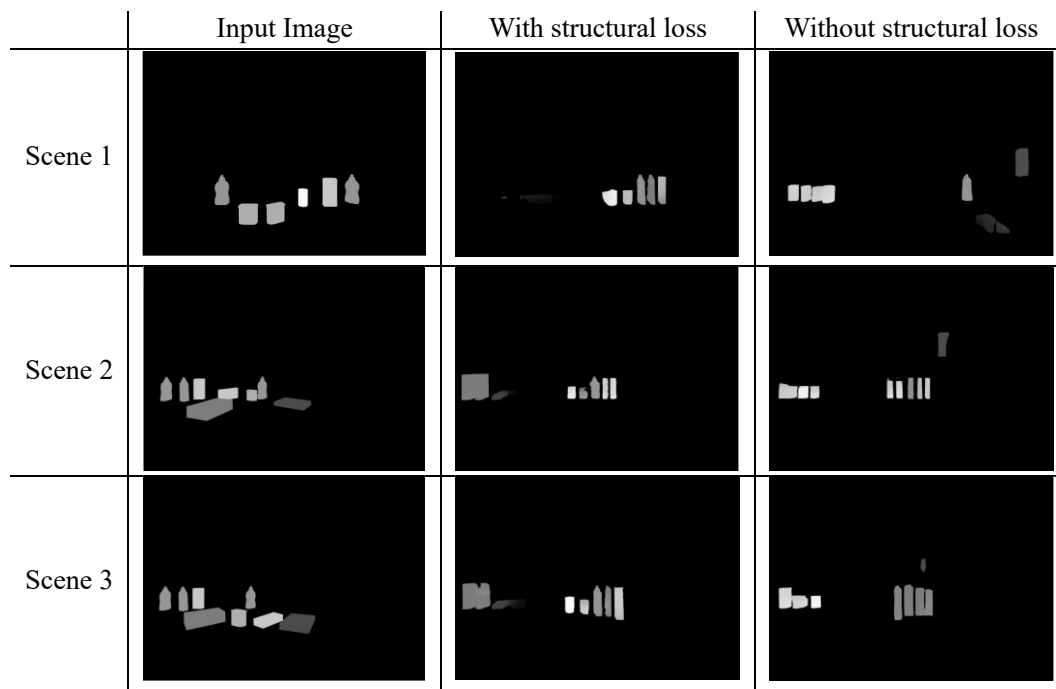


Figure 19. Results of object location translation with and without using structural loss

As shown in Figure 19, the model presented better outcomes with structural loss than without it. In general, the results with structural loss were consistent with the number of objects and their classes. However, the results without the loss showed several defects, such as wrong object placement and object class that did not exist in the input images.

E. Discussion

We implemented object location translation and a Gazebo simulation for object pose estimation and robot planning. As shown in Figure 13, the objects were arranged at a semantic mask level. However, some objects were removed and added, and their classes were sometimes hard to distinguish. It is assumed that the training images were too broad to cover the scenarios. Thus, we will enhance the object arrangement step by amending our dataset.

As shown in Figure 18, we verified that our framework is better at object location translation using the pix2pixHD model [1] than the pix2pix [16] and the cycleGAN [21]. For generating high-resolution aligned images, the former method is proper than other methods. Also, we could confirm that the structural loss is helpful to enhance the accuracy of aligned image synthesis by the ablation test. The results revealed that the model is improved by comparing the images at the latent level.

Also, we executed object pose estimation and motion planning under the ROS system. Due to the low accuracy of generated goal scenes at the RGB level, it was difficult to detect the objects from the images. Therefore, we manually constructed the initial and goal scenes, then executed both approaches to carry out the robot simulation. We performed motion planning successfully, but collisions and planning-failed cases still occurred. We are going to amend this planning issue, as well.

As a limitation, we tested all individual steps; however, we could not integrate the entire system into one complete system. The main reason was that the results of the synthesized images were not as good as suitable for object pose estimation. Thus, we decided to manually put objects and construct the initial and goal scenes that are similar to the synthesized images.

Due to the addition and deletion of objects from generated semantic masks, it was difficult to recover the RGB textures and obtain the final aligned images for detecting object poses. Also, during the texture recovery, the textures were often distorted because there are some pose changes between objects from input images and from the generated images. Thus, it was hard to represent the realistic textures of the results. To successfully estimate individual object poses from the generated RGB images, we still need to improve our system for realistic scene synthesis by using other texture reconstruction approaches.

VII. Conclusion

In this thesis, we demonstrate that our framework enables a robot to carry out object arrangement by generating the goals by themselves without needing to have goals given as human commands. Providing the goals to robots is burdensome; if robots can create the goals autonomously, it might lessen the difficulty and be more convenient for people.

To achieve our goals, we performed the object arrangement using semantic segmentation, object pose translation, and texture reconstruction. To focus on the object class information, it was necessary to go down to the semantic level, which was revealed to be helpful. To perform a simulation, we generated a similar Gazebo simulation, using a Fetch robot, with the Unreal Engine 4.

As a result, we confirmed that the objects are successfully arranged by our framework. The robot can create an aligned scene from an input scene by applying the pre-trained model. After the goal generation, it conducts proper pick-and-place motions and organizes objects like the given goal images.

As we mentioned in the discussion, during the object pose translation and the texture reconstruction, the results of the synthesized images were not as good as we expected, and, therefore, we could not complete the whole system into one as we planned. The aligned RGB scene images were still inaccurate for detecting individual object poses using the existing methods.

Due to the robot perception and safety issues in real robot hardware, it is difficult to operate robot manipulation in the real-world. Thus, in this work, we only tested our framework in the robot simulation. However, because of the limitation of the gazebo simulation, it is

difficult to encompass more general scenarios because grasping fails easily owing to the inaccurate approximation of the physical properties of objects. Also, in this work, we did not consider the navigation of the robot's base, and, therefore, the manipulator could reach only a small range of the table.

As for future work, we would like to enhance the accuracy of the object pose translation and texture reconstruction by adopting additional loss terms into the object position translation, and other texture recovery approaches. It is necessary to improve it to detect object poses from 2D RGB images to carry out further systems. Furthermore, we plan to amend the simulation to successfully carry out motion planning and complete our framework. Eventually, we will expand our system and test it with a real robot system.

Bibliography

- [1] T. C. Wang, M. Y. Liu, J. Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, “High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8798-8807, 2018
- [2] T. Park, J. Y. Zhu, O. Wang, J. Lu, e. Shechtman, A. A. Efros, and R. Zhang, “Swapping Autoencoder for Deep Image Manipulation,” *arXiv preprint arXiv:2007.00653*, 2020.
- [3] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, “The YCB object and Model set: Towards common benchmarks for manipulation research,” in *2015 International Conference on Advanced Robotics (ICAR)*, pp. 510-517 2015, pp. 510–517, 2015.
- [4] A. Mousavian, C. Eppner, and D. Fox, “6-DOF GraspNet: Variational Grasp Generation for Object Manipulation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2901-2910, 2019.
- [5] M. Liu, Z. Pan, K. Xu, K. Ganguly, and D. Manocha, “Generating Grasp Poses for a High-DOF Gripper Using Neural Networks,” *arXiv preprint arXiv:1903.00425*, 2019.
- [6] L. Karacan, Z. Akata, A. Erdem, and E. Erdem, “Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts,” *arXiv preprint arXiv:1612.00215*, 2016.
- [7] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, “Deep

- Object Pose Estimation for Semantic Robotic Grasping of Household Objects,” *arXiv preprint arXiv:1809.10790*, 2018.
- [8] C. Wang, D. Xu, Y. Zhu, R. Martin-Martin, C. Lu, L. Fei-Fei, and S. Savarese, “DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3343-3352, 2019.
- [9] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, “PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes,” *arXiv preprint arXiv:1711.00199*, 2017.
- [10] N. Abdo, C. Stachniss, L. Spinello, and W. Burgard, “Robot, organize my shelves! Tidying up objects by predicting user preferences,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1557-1564, 2015.
- [11] M. Kang, Y. Kwon, and S. E. Yoon, “Automated task planning using object arrangement optimization,” in *2018 15th International Conference on Ubiquitous Robots (UR)*, pp. 334–341, 2018.
- [12] M. Jalal, J. Spjut, B. Boudaoud, and M. Betke, “SIDOD: A Synthetic Image Dataset for 3D Object Pose Recognition With Distractors,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 475–477, 2019.
- [13] J. Tremblay, T. To, and S. Birchfield, “Falling Things: A Synthetic Dataset for 3D Object Detection and Pose Estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2038-2041, 2018.
- [14] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison, “SceneNet RGB-D: 5M

- Photorealistic Images of Synthetic Indoor Trajectories with Ground Truth,” *arXiv preprint arXiv:1612.05079*, 2016.
- [15] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, d. Warde-Farley, S. Ozair, Courville, Y. Bengio, “Generative Adversarial Networks,” *Advances in neural information processing systems (NIPS)*, vol. 27, pp. 2672-2680, 2014.
- [16] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-Image Translation with Conditional Adversarial Networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.1125-1134, 2017.
- [17] X. Wang and A. Gupta, “Generative Image Modeling using Style and Structure Adversarial Networks,” in *European conference on computer vision (ECCV)*, pp. 318-335, 2016.
- [18] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays, “Scribbler: Controlling Deep Image Synthesis with Sketch and Color,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5400-5409, 2017.
- [20] Y. Taigman, A. Polyak, and L. Wolf, “Unsupervised Cross-Domain Image Generation,” *arXiv preprint arXiv:1611.02200*, 2016
- [21] M. Y. Liu, T. Breuel, and J. Kautz, “Unsupervised Image-to-Image Translation Networks,” *Advances in neural information processing systems (NIPS)*, pp. 700-708, 2017.
- [19] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2223-2232,

- 2017.
- [22] A. Achille and S. Soatto, “Emergence of Invariance and Disentanglement in Deep Representations,” *The Journal of Machine Learning Research*, vol. 19, no. 1, pp. 1947-1980, 2018.
- [23] R. T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud, “Isolating Sources of Disentanglement in Variational Autoencoders,” *Advances in neural information processing systems (NIPS)*, pp. 2610-2620, 2018.
- [24] C. Eastwood and C. K. I. Williams, “A framework for the quantitative evaluation of disentangled representations,” in *International Conference on Learning Representations (ICLR)*, 2018.
- [25] M. Mathieu, J. Zhao, P. Sprechmann, A. Ramesh, and Y. LeCun, “Disentangling factors of variation in deep representations using adversarial training,” *Advances in neural information processing systems (NIPS)*, pp. 5040-5048, 2016.
- [26] T. Karras, S. Laine, and T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 4401-4410, 2019.
- [27] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and Improving the Image Quality of StyleGAN,” in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 8110-8119, 2020.
- [28] R. Abdal, Y. Qin, and P. Wonka, “Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space?,” in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 4432-4441, 2019.

- [29] Q. Hu, A. Szabó, T. Portenier, M. Zwicker, and P. Favaro, “Disentangling Factors of Variation by Mixing Them,” in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 3399-3407, 2018.
- [30] M. Fisher, M. Savva, D. Ritchie, T. Funkhouser, and P. Hanrahan, “Example-based Synthesis of 3D Object Arrangements,” *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, pp. 1-11, 2012.
- [31] K. Wang, Y. A. Lin, B. Weissmann, M. Savva, A. X. Chang, and D. Ritchie, “PlanIT: planning and instantiating indoor scenes with relation graph and spatial prior networks,” *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–15, 2019.
- [32] Z. Zhang, Z. Yang, C. Ma, L. Luo, A. Huth, E. Vouga, and Q. Huang, “Deep Generative Modeling for Scene Synthesis via Hybrid Representations,” *ACM Transactions on Graphics (TOG)*, pp. 1-21, 2020.
- [33] D. Ritchie, K. Wang, and Y. Lin, “Fast and Flexible Indoor Scene Synthesis via Deep Convolutional Generative Models,” in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 6182-6190, 2019.
- [34] M. Li, A. G. Patil, K. Xu, S. Chaudhuri, O. Khan, A. Shamir, and H. Zhang, “GRAINS: Generative Recursive Autoencoders for INdoor Scenes,” *ACM Transactions on Graphics (TOG)*, pp. 1-16, 2019.
- [35] *NVIDIA/Dataset_Synthesizer*. NVIDIA Corporation, 2020.
- [36] V. Lepetit, F. Moreno-Noguer, and P. Fua, “EPnP: An Accurate $O(n)$ Solution to the PnP Problem,” in *International journal of computer vision (IJCV)*, vol. 81, no. 2, pp. 155–166, 2009.

- [37] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” arXiv preprint arXiv:1412.6980, 2014.
- [38] D. Coleman, I. Sucan, S. Chitta, and N. Correll, “Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study,” arXiv preprint arXiv:1404.3785, 2014.

국문초록

김시연

컴퓨터공학과

이화여자대학교 대학원

이 논문에서 우리는 정렬되지 않은 장면에서 물체를 정렬된 장면으로 로봇이 스스로 정리할 수 있는 프레임 워크를 제안했다. 이전 연구들에서는 로봇이 정리나 다른 작업을 수행하기 위해서는 사용자로부터 주어진 목표에 대한 정보나 또는 안내 계획을 제공되어야 했다. 하지만, 이렇게 사용자가 목표 정보에 대해서 매번 제공을 하는 것은 번거롭기 때문에, 로봇이 스스로 목표를 생성함으로써 사용자가 조금 더 편하고 힘들지 않도록 하는 것이 우리 연구의 목표이다.

본 연구의 목표인 물체 정리를 달성하기 위해, 정렬되지 않은 장면과 정렬된 장면의 짝으로 이루어진 데이터 세트의 필요성이 있었다. 따라서, 단순히 여러 물체들이 놓인 기존의 데이터 세트로는 위의 목표를 달성하기에 부적절하여, 본 연구에서는 먼저, YCB 객체 모델 [1]로 구성된 사실적인 합성 데이터 세트를 구성하여 활용하였다. 이렇게 생성된 데이터 셋을 활용하여, 기존의 이미지 변환 모델을 활용하여 물체들이 정렬되도록 배치하였다. 이를 시뮬레이션에서 로봇이 정리할 수 있도록 모션 계획 알고리즘을 사용하여 목표를 달성함으로써 로봇이 자율적으로 목표를 세우고 물체 배치 작업을 성공적으로 수행할 수 있음을 보여주었다.