Transfusion: A Novel SLAM Method Focused on Transparent Objects

Yifan Zhu
Nankai University
zhuyifan@mail.nankai.edu.cn

Jiaxiong Qiu
Nankai University
qiujiaxiong727@gmail.com

Bo Ren[†]
Nankai University
rb@nankai.edu.cn

Abstract

Recently RGB-D sensors have become very popular in the area of Simultaneous Localisation and Mapping (SLAM). The RGB-D SLAM approach relies heavily on the accuracy of the input depth map. However, refraction and reflection of transparent objects will result in false depth input of RGB-D cameras, which makes the traditional RGB-D SLAM algorithm unable to work correctly in the presence of transparent objects. In this paper, we propose a novel SLAM approach called transfusion that allows transparent object existence and recovery in the video input. Our method is composed of two parts. Transparent Objects Cut Iterative Closest Points (TC-ICP)is first used to recover camera pose, detecting and removing transparent objects from input to reduce the trajectory errors. Then Transparent Objects Reconstruction (TO-Reconstruction) is used to reconstruct the transparent objects and opaque objects separately. The opaque objects are reconstructed with the traditional method, and the transparent objects are reconstructed with the visual hull-based method. To evaluate our algorithm, we construct a new RGB-D SLAM database containing 25 video sequences. Each sequence has at least one transparent object. Experiments show that our approach can work adequately in scenes contain transparent objects while the existing approach can not handle them. Our approach significantly improves the accuracy of the camera trajectory and the quality of environment reconstruction.

1. Introduction

Nowadays, the interests in visual SLAM has been significantly increased due to its variety of usages on many other computer vision applications, such as augmented reality, autonomous driving car, and robotics navigation. There are rich types of sensors utilized for SLAM algorithm, such as lidar, monocular camera, RGB-D camera, etc [7, 3, 2]. Among them, the RGB-D camera has been widely considered for the visual SLAM since it can provide a rich source of 3D information at a relatively low cost.

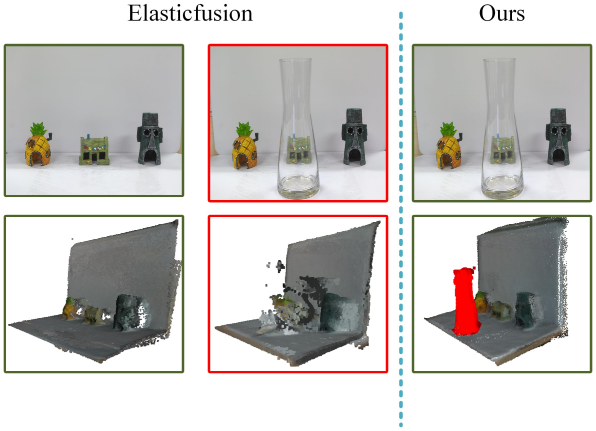
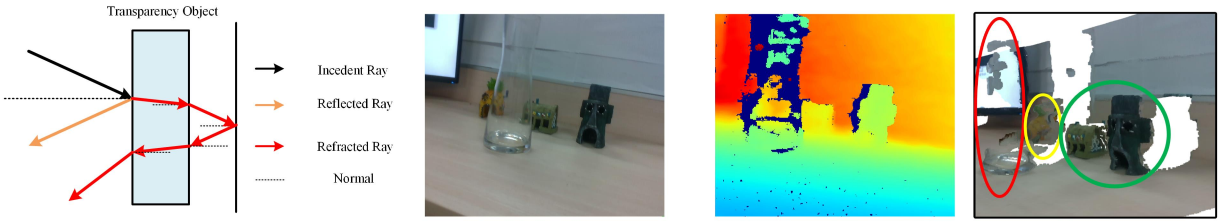


Figure 1. **Reconstruction result of two scene with our method and Elasticfusion[31].** Due to the transparent object's characteristic, the passed RGB-D slam method can not reconstruct the scene contains transparent objects properly. The first column is the reconstruction result of Elasticfusion without transparent objects; in the second column is the reconstruction result of Elasticfusion with transparent objects; in the third column is the reconstruction result of our algorithm with transparent objects.

There is a lot of excellent SLAM methods that take RGB-D as perception modules [31, 24, 5, 16]. They can get more refined reconstruction results than the monocular methods and can eliminate the scale drifting problem. However, they are all developed in an environment where there are all opaque objects, and as shown in Fig. 1, they cannot properly function when the environment contains transparent objects. But, transparent objects made of glass, such as bottles, windows, are ubiquitous around us, so it is essential to study the RGB-D SLAM algorithms that contain transparent objects in the environment model.

RGB-D cameras usually equip infrared light emission and receivers to measure the depth, such as Microsoft's Kinect and Intel's RealSense. Unlike opaque objects, transparent materials do not satisfy the classic geometric light path assumptions of stereo vision algorithms. When in-

[†]Bo Ren is the corresponding author



(a) (b) (c) (d)

Figure 2. **Affect of transparent object to the depth map**. (a) When infrared rays are emitted to transparent objects, both reflection and refraction occur on their surface.(b) and c) are the RGB image and depth map of the scenes. (d) The 2.5D view by combining the depth map with RGB image. The area in the red circle is where the transparent object locates. The yellow circle is the area affect by the transparent. The green circle is the opaque object

frared rays (IR) are emitted to transparent objects, both reflection and refraction occur on their surface. As shown in Fig. 2, some rays reflect on the surface of a transparent object or reflect on the surface of the object behind the transparent object. Thus when RGB-D cameras are used to acquire the environment's depth, there is a high probability of getting erroneous results due to these objects. The transparent objects often show up as noisy or distorted surfaces in the depth inputs. Meanwhile, the object behind transparent materials will be distorted due to their occlusion, making it challenging for RGB-D cameras to produce accurate depth estimates in the area occluded by transparent objects.

On the other hand, the RGB-D SLAM approach relies heavily on the accuracy of the environmental depth obtained, which is in nature of the Iterative Closest Points (ICP) algorithm. The RGB-D SLAM systems often leverage the ICP algorithm in the tracking stage. ICP algorithm takes the depth map as input then estimates camera pose through 3D points registering. Inaccurate depth will lead the ICP algorithm to get the wrong camera pose.

This leads to another challenge of recovering the concrete shape of transparent objects in the reconstruction phase because the RGB-D camera cannot get its correct depths. Moreover, the reconstruction of objects behind them will also be affected because transparent objects also distort their depth. As illustrated in Fig. 2, wrong depth destroyed the system's reconstruction results.

In this paper, we propose a novel RGB-D SLAM approach that can handle scenes include transparent objects. Our method can simultaneously reduce tracking errors and recover the whole scene's correct shape.

Firstly, Transparent Objects Cut Iterative Closest Points (TC-ICP) algorithm is used to estimate the camera pose. TC-ICP can detect transparent objects' location and remove them from the input, which will reduce the depth error caused by transparent objects. Then Transparent Objects Reconstruction (TO-Reconstruction) algorithm is used to recover the whole scene's 3D model. Transparent object reconstruction only takes advantage of the RGB image, mask, and camera pose information, which can be readily integrated into the traditional RGB-D methods.

Since our main task is making the SLAM algorithm work adequately in scenes containing transparent objects, and there is no existing database dedicated to this task. To evaluate our algorithm, we construct a new database with 25 video sequences. In every sequence, there is at least one transparent object.

In summary, the main contributions of our paper are:

- We propose a novel RGB-D SLAM approach called Transfusion, which shows excellent performance in camera pose estimation and scene reconstruction in environments containing transparent objects.
- We construct a new RGB-D database Trans-SLAM for the research of SLAM algorithms in the environment that contains transparent objects.

2. Related Works

2.1. Visual SLAM

SLAM algorithm is an attractive research topic in computer vision and robotic fields. They can be roughly divided into two classes by the sensor they take. The MonoSLAM [6] system is the first real-time monocular reconstruction system, which uses extended Kalman filtering (EKF) as the back-end. Later, PTAM [17] is proposed as a new framework for parallel tracking and mapping, which can reduce map updates' computational complexity. Then LSD-SLAM [8] introduces a direct tracking method based on Lie algebra, which improves the trajectory accuracy. However, the monocular SLAM method suffers from the scale drifting problem and can hardly generate dense reconstruction results. The other methods are the RGB methods.

The other one is the RGB-D SLAM. Newcombe et al. propose KinectFusion [24], which preprocesses the raw input depth image by a bilateral filter to reduce noise.Then DVO-SLAM [16] combines dense visual odometry with pose SLAM. ElasticFusion [31] models the scene as a set of

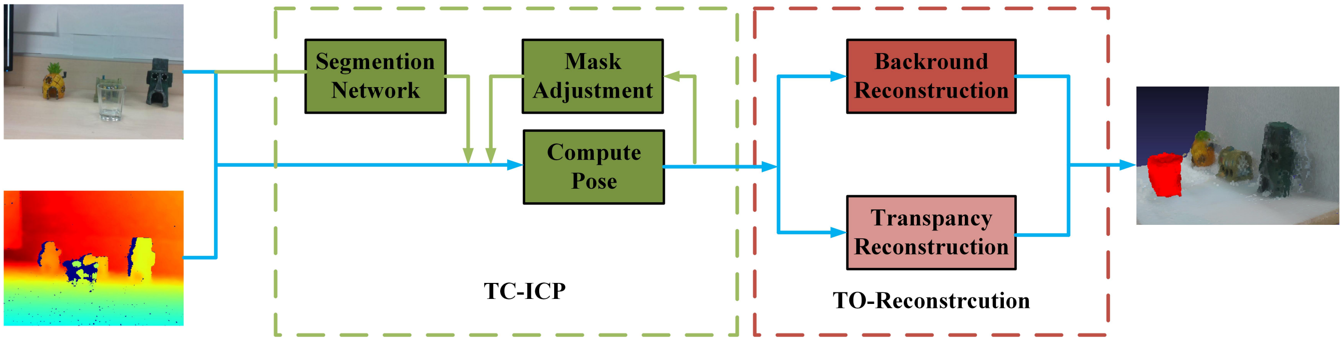


Figure 3. **System Overview**. Our system consists of two main parts: TC-ICP and TO-Reconstruction. The TC-ICP aims to estimate the camera pose without the influence of transparent objects, and the TO-Reconstruction is to recover the correct shape of the environment containing transparent objects.

surfels deformed to accommodate loop closures. Compared with the monocular method, they can reconstruct a dense point cloud map of the environment. Moreover, these methods can get the scale factor from the input, eliminating the scale drifting problem. However, all these methods apply to cases containing opaque objects only.

2.2. Transparent Objects Detection

Since the emergence of the computer vision field, transparent object detection has been a challenging problem. 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 [25, 22, 21, 11]. Later methods rely on deep learning models like SSD [20] or Fast-RCNN [13] to predict bounding boxes enclosing transparent objects. Seib et al. [28] propose a method to exploit sensor failures in the depth map for transparent object localization using convolutional networks. Wang et al. [30]propose localizing glass objects using a Markov Random Field to predict glass boundary and region jointly from multiple modalities from an RGB-D camera. These methods are only tested on small datasets and do not work properly in the wild.

2.3. Transparent object reconstruction

The reconstruction of transparent objects is a challenging problem in the field of computer vision. Murase first proposes the 3D reconstruction of the refractive surface [23], using the optical flow method and pixel tracking method to restore the water surface model. Masaki Yamazaki et al. propose a transparent object reconstruction method [33] based on the phase shift method using a stereo camera. The reconstruction result is finally obtained by analyzing the distortion of the stripe pattern on the transparent object photographed by the camera. On this basis, Qian et al. propose a transparent object reconstruction method [26] based on directed light measurement. The method solves the reconstruction by using the position-normal consistency as a constraint. However, this algorithm cannot generate complete object reconstruction results. N. Alt et. al [9] proposed an method to reconstruct the transparent objects via depth camera, however their method needs first to estimate the background model using a video sequence that excludes transparent objects. S. Albrecht et. al [1] use the noisy point cloud data to recover the shape of transparent objects, but they assume that there is a planar surface underneath or behind the transparent object. Zheng proposes a transparent object reconstruction method [19] based on deep learning. This method uses a visual hull for initialization, then an automatic encoder is designed to learn the propagation path of light in a transparent object, and finally generate a point cloud by PointNet.This method can recover the whole shape of transparent objects, but they require the information of the environment and this method is time-consuming.

3. Methods

3.1. Transfusion System

In this paper, we propose a novel RGB-D SLAM approach called Transfusion that can work adequately in scenes containing transparent objects. Our Transfusion system can significantly improve the performance in both pose establishment and scene reconstruction.

We adopt the state-of-the-art ElasticFusion system as our base system, which provides excellent reconstruction and camera pose estimation results by combining the color and brightness information with the ICP algorithm. ElasticFusion works well when all objects in the environment are opaque. However, as shown in Fig. 1, if there are transparent objects, the whole system will get the wrong results.

Our improved framework consists of two main parts: TC-ICP and TO-Reconstruction. The TC-ICP in Sec. 3.2 aims to eliminate the adverse influence of transparent objects and get the accurate camera pose, and the TO-Reconstruction in Sec. 3.3 aims to reconstruct the correct geometry structure of the scene.

3.2. Transparent Cut Iterative Closest Points

To eliminate the influence of the incorrect depth information of transparent objects on the pose calculation, we design a novel ICP algorithm called TC-ICP to replace the traditional ICP part used in Elasticfusion.

First, we will recap the traditional ICP algorithm. ICP algorithm estimates camera pose through 3D point cloud registration, which can be converted to a optimization problem by Lie algebra as follow:

$$E_{icp} = \sum_k \left(\left(\mathbf{v}^k - \exp(\hat{\boldsymbol{\xi}}) \mathbf{T} \mathbf{v}_t^k \right) \cdot \mathbf{n}^k \right)^2 \tag{1}$$

where v^{k_t} is the back-projection of the $k-th$ vertex in depth map, v^k and n_k are the corresponding vertex and normal represented in the previous frame. $\mathbf{T}$ is the current estimate of the transformation from the previous camera pose to the current one, and $exp(\hat{\xi})$ is the matrix exponential that maps a member of the Lie algebra $\mathfrak{se}_3$ to a member of the corresponding Lie group $\mathbb{SE}_3$. The Gauss-Newton non-linear least-squares method is used to optimise this function.

The traditional RGB-D methods cannot handle transparent objects because they take all the depth information given by the RGB-D camera for pose estimation, which contains the distorted depth caused by the transparent object.

In our TC-ICP approach, we first segment the transparent from the input image to get the rough segmentation result $\mathcal{M}_r$, which is done by applying a pre-trained transparent objects segmentation network.

The results of the segmentation network usually have some defects when used in world situations due to polytropic illumination and perspective. Consequently, a mask adjustment procedure is essential to get the final result $\mathcal{M}_f$ (Sec. 3.2.2)

Next, the $\mathcal{M}_f$ is processed by the equation bellow to get the processed depth map:

$$\mathcal{D}_p = \mathcal{D} \cap \mathcal{M}_f \tag{2}$$

where $\mathcal{D}_p$ is the processed depth map which has less wrong information.

$$E_{TC\_ICP} = \sum_k \left(\left(\mathbf{v}_p^k - \exp(\hat{\boldsymbol{\xi}}) \mathbf{T} \mathbf{v}_{pt}^k \right) \cdot \mathbf{n}_p^k \right)^2 \tag{3}$$

Then we leverage the same optimizer in traditional ICP to estimate the camera pose.

3.2.1 Transparent Objects Segmentation Nework

To segment transparent objects from the surroundings, we propose a novel transparent object segmentation network based on SINet [10], which takes the RGB image as input to find the coverage object in the surroundings.

Unlike SINet, our model is composed of two parallel branches: segmentation branch and boundary branch. The segmentation branch is for transparent object segmentation, while the boundary branch is for boundary prediction. ResNet50 [14] is used as the backbone network to extract a set of features $\{\mathcal{X}_k\}_{k=0}^4$.

The network agriculture is shown in Fig. 4. For the segment branch, the Receptive Field (RF) components are used to expand the receptive field. As shown in Fig. 4, first we concatenation the low-level features $\{\mathcal{X}_0, \mathcal{X}_1\}$, and then the fused feature is downsampled twice. Then the RF component is employed to generate rf_s^4 features. After combining the three levels of features, a set of enhanced features $\{rfs_k, k = 1, 2, 3, 4\}$ are acquired.

Recent evidences have shown that low-level features in shallow layers preserve more spatial details, so in the boundary branch, we directly use the low-level feature $\{\mathcal{X}_0, \mathcal{X}_1\}$ generate from ResNet50. Then the Atrous Spatial Pyramid Pooling module (ASPP) is employed to enlarge the receive field in $\{\mathcal{X}_4\}$.Then integrate three feature maps mentioned above into the input feature map $\{f_k^b, k = 1, 2, 3\}$ for the boundary decoder.

Fig. 4 illustrates the detailed structure of the decoder we used in the segmentation network. For the segment branch, Partial Decoder Component (PDC) integrating four levels of features $\{rfs_k, k = 1, 2, 3, 4\}$ is used to get the segment result. For the boundary branch, a series of up sampling and convolution operations are performed on the feature map $\{f_k^b, k = 1, 2, 3\}$ to get the boundary prediction. We define our training loss function as follows:

$$L = L_s + L_b \tag{4}$$

where L_s and L_b represent the losses for the segmentation text and boundary, respectively. Here L_s and L_b are the standard Cross-Entropy (CE) loss.

3.2.2 Mask Adjustment

The segmentation results obtained by segmentation network still have deviations in practical applications. As shown in Fig. 5, the algorithm will recognize parts of opaque objects as transparent objects. Besides, because of the gap between the data we captured and the data used for training, the network often generates undersized masks.

Accordingly, we propose mask adjustment to address these issues, which is interleaved performed with pose estimation. Mask adjustment can detect the wrong segmentation results and optimize the size of the mask through a scale factor s. Its core idea is that when the number of error depth points dropped, the ICP algorithm's error is reduced.

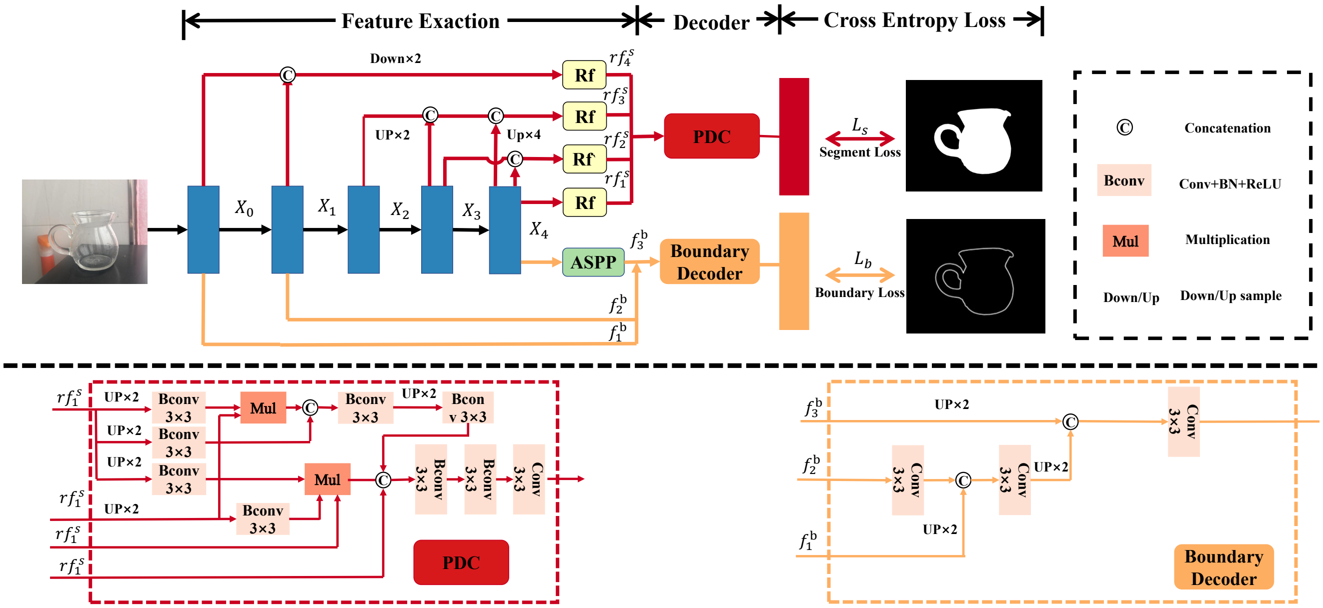


Figure 4. **Network Architecture.** Our model is composed of two parallel branches: segmentation branch and boundary branch. Above the dotted line is the overall architecture of our network, and below the dotted line is the decoder structure we use in our model.

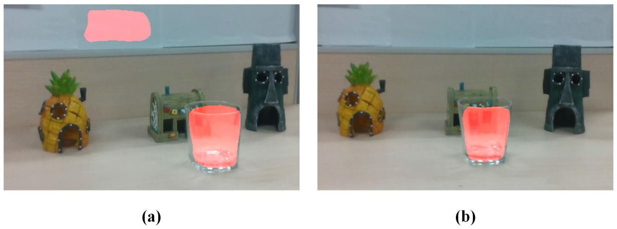


Figure 5. **Wrong segmentation**. (a)the algorithm may recognize parts of non-transparent objects as transparent objects. (b)The mask gained from segmentation generally cannot completely cover the entire transparent object.

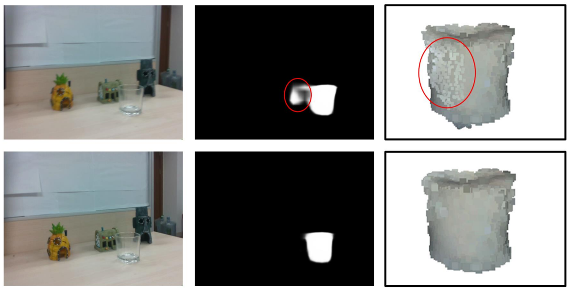


Figure 6. **The effect of motion blur on segmentation**. The segmentation results of the image with a high degree of motion blur is worse.

Our mask adjustment contains two steps. In the first step, we check each mask M_i in segmentation results in turn whether it contains incorrect segmentation results. For each mask, we can calculate the contribution of removing it to reducing the registration error. This contribution can be compute as $E_{init} - E_m$. E_m is the registration error without the area covered by M_i computed by ICP algorithm. E_{init} is the registration error with all depth input. If the contribution is less than $5e-04$, this mask will be regarded as the wrong segmentation. Then remove it from the initial segmentation result and repeat the above operation until the optimized segmentation result $\mathcal{M}_d$ is acquired.

In the second step, we optimize the size of masks in $\mathcal{M}_d$ through scale factor s. As is shown in Fig. 5, the size of the transparent object's mask is often smaller than its real size, so let each mask in $\mathcal{M}_d$ be multiplied by a scale factor s to increase its size appropriately. In the step n, compute the registration error E_s without the area covered by sM_i through ICP algorithm. If $E_{init} - E_s < 1e-05$ or $E_s - E_m < 0$, stop the algorithm and return the $(n-1)$th s, otherwise, increase s by 0.005 and continue this algorithm.

3.3. Transparent Objects Reconstruction

TO-Reconstruction is proposed to reconstruct the environmental model that containing transparent objects, and it can be divided into two parts, traditional RGB-D method and visual hull based method. The traditional RGB-D method integrated in Elasticfusion is used to reconstruct the opaque objects, and the visual hull based method is used to reconstruct transparent objects. After the mask adjustment part described in Sec.3.2.2, the left opaque objects can be reconstructed, so next we will concentrate on the reconstruction of transparent objects in the following.

As analyzed in Sec. 1, it is hard to get the accurate depth of the transparent objects, so we can not use the traditional RGB-D reconstruction methods to reconstruct transparent objects. Inspired by [19], we propose a visual hull based method to recover the shape of transparent objects.

3.3.1 Key Frame Selection

Taking a video sequence as the algorithm input is different from taking a single image as the algorithm input, which will suffer from the motion blur problem. From Fig. 6, some frames extracted from the video sequences contains motion blur, which will directly lead to imprecise segmentation and reconstruction results.

Since the visual hull algorithm does not need all frames of video input for reconstruction, we can select key frames that contain very little or no motion blur as the input set. Laplace value is used to measure the blur of RGB image. We first compute the Laplacian response by:

$$\nabla^2 f = I * \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{5}$$

Then take the standard deviation squared of the response as the Laplacian value. In the experiments, we set the threshold to 50 and the images with Laplace value below the threshold are considered to be blurry and not suitable for reconstruction. Then we will get the input set S that can be used in the reconstruction stage.The key frame is only used to reconstruct the transparent objects. In the tracking period, we use all the frames to do frame-to-frame tracking.

3.3.2 Transparent Object Reconstruction

We propose our transparent object reconstruction method based on visual hull. Traditional visual hull based algorithms require cameras to be fixed or slightly moving in Z-axis, but the pose of the input frame usually changes within a relatively large range of motion. So we transform the Z-axis range of the obtained pose to -0.5m to 0.5m, and perform affine transformation on the corresponding mask.

From the key frame selection stage, we can get the key frame set S. To recover the shape of transparent objects, we first compute viewing cone of each image in S with its mask and camera pose.Then the intersection of viewing cones is determined as the initial volume $\mathcal{V}$. After that, $\mathcal{V}$ is optimized by the space carving method proposed in [27].

The viewing cone of input images can be calculated by projecting the mask in images to the 3D space. We use the equation 6 to project the pixel in mask to the 3D space.

$$v(u) = TK^{-1}u \tag{6}$$

where u is pixels in the mask. K and T are the intrinsic matrix and transformation matrix of camera, respectively, and $v(u)$ is the project result. With given K, T and the mask, a generalized cone called viewing cone can be calculated that contains the actual object as shown in Fig. 7. The intersection of these cones of the images in S can be regarded as the initial volume $\mathcal{V}$ for the next volume optimize stage.

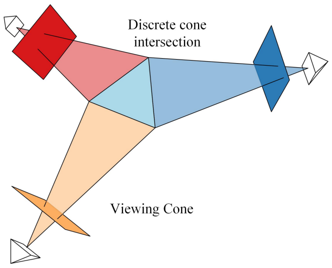


Figure 7. **Viewing cone**. Along with the camera viewing parameters, the mask defines a back-projected generalized cone that contains the actual object. This cone is called a viewing cone.

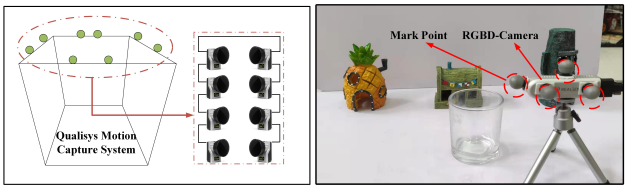


Figure 8. **Equipments used in data construction**.

Then initial volume $\mathcal{V}$ will be optimized by space carving, which employs the photo-consistency measure to decide if voxels v in $\mathcal{V}$ should be carved away or retained. A voxel is considered to be photo-consistent when its colors that can be seen by all the cameras appear to be similar.

We take the criterion in [27] to determine the photo-consistency of v, which can deal with the textureless regions and specular highlights. If there is a non-photo-consistent v^n in V, we will set $\mathcal{V} = V - v^n$ and repeat this operation until all the v is photo-consistent. Then the final $\mathcal{V}$ is set as the optimal volume $\mathcal{V}^*$, which is the final result.

4. Experiments

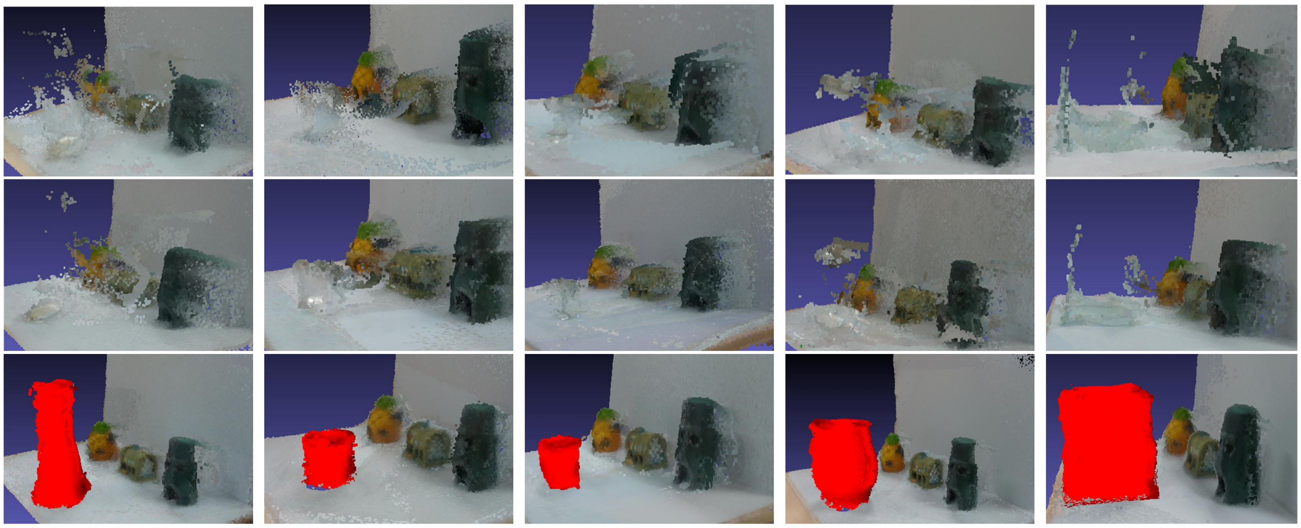
4.1. Database Construction

There are many database [29, 4, 12] published for SLAM research. They provide RGB images, depth images, and ground truth trajectory. However, all the scene in these databases exclude transparent objects. To evaluate the SLAM algorithm in the scene containing transparent objects, we collect a new RGB-D database called Trans-SLAM, which contains 25 video sequences and every sequence contains at least one transparent object.

The RGB image and depth map is recorded by a RealSense D435i RGB-D camera at the frame rate of 30 Hz and sensor resolution of 640x480. Every video sequence has about 2000 images, 60 seconds. Each scene comprises three opaque objects and at least one transparent object in

Scene	ATE ↓			R.RPE ↓			T.RPE ↓		
	RGB-D SLAM V2	Elasticfusion	ours	RGB-D SLAM V2	Elasticfusion	ours	RGB-D SLAM V2	Elasticfusion	ours
Long-Necked Vase	0.446	0.334	**0.158**	0.135	0.124	**0.018**	0.220	0.192	**0.024**
Cylindrical Glass	0.351	0.288	**0.131**	0.245	0.143	**0.084**	0.253	0.132	**0.025**
Angular glass	0.349	0.350	**0.128**	0.135	0.145	**0.016**	0.223	0.253	**0.035**
Belly Glass Vase	0.465	0.544	**0.169**	0.174	0.204	**0.021**	0.374	0.404	**0.043**
Glass Fish Tank	1.921	2.375	**0.919**	0.273	0.135	**0.028**	0.825	0.544	**0.423**
fr1/desk	0.023	**0.020**	0.021	0.015	**0.010**	0.011	0.012	0.008	**0.007**
fr2/xyz	**0.008**	0.011	0.010	**0.003**	0.005	0.008	**0.002**	0.004	0.006
Mean	0.509	0.560	**0.219**	0.141	0.109	**0.027**	0.273	0.220	**0.080**

Table 1. SLAM results



Long-Necked Vase Cylindrical Glass Angular glass Belly Glass Vase Glass Fish Tank

Figure 9. **Reconstruction results**. The first row and second row are the results of Elasticfusion and RGD-SLAM V2, and the third row is the results of our method.

Methods	IOU ↑	MAE ↓	BER ↓	Acc ↑
DSC [15]	0.776	0.084	0.091	90.860
BDRAR [34]	0.759	0.081	0.087	89.860
R^3Net [18]	0.765	0.072	0.103	91.860
SINET [10]	0.765	**0.070**	0.084	92.140
TransLab [32]	**0.872**	**0.070**	0.083	92.310
Ours	0.856	0.071	**0.081**	**92.871**

Table 2. Segmentation results

front of them. The transparent objects in our database include five typically transparent objects in the real world, such as vases, glass cups, and fish tanks. All scenes are captured indoor, and the camera is moved by hand.

The reference poses for SLAM should have high accuracy in both local relative 6DoF transformations and global positioning. Thus, the ground-truth trajectory and 6DoF pose of the camera in our experiments are obtained from the Quality motion capture system, which consists of 8 high-end motion capture units that can record the high accuracy trajectory and 6DoF pose.

Fig. 8 shows the object and equipments we used in the data construction stage. We put four marks on the RGB-D camera for the motion capture system to locate the camera's global position and its 6DoF pose. And the detail of our database can be find in the supplementary martial.

4.2. Datasets and Parameter Setup

The Transfusion system is built and tested in C++. To evaluate the performance of the proposed approach in scenes containing transparent objects, we test our system on the Trans-SLAM database. The transparent object segmentation network is implemented with pytorch and trained with the Adam optimizer. The transparent object segmentation network is trained and tested on the trans10k database [32]. During the training stage, the batch size is set to 32, and the learning rate starts at 1e-4. We use 2×V100 GPU for the training and testing of the segmentation network. Furthermore, the whole Transfusion system is running on one Nvidia 3080 GPU, and the segmentation network is converted into libtoch model, which can be used in the C++ environment.

4.3. Evaluation Criteria

We use Absolute Trajectory Error (ATE) and Rotation/Translation Relative Pose Error(R./T.RPE) to evaluate

Scene	ATE ↓			R.RPE ↓			T.RPE ↓		
	No-Mask Adjustment	Use SINet	Ours	No-Mask Adjustment	Use SINet	Ours	No-Mask Adjustment	Use SINet	Ours
Long-Necked Vase	0.168	0.191	**0.158**	0.021	0.033	**0.018**	0.048	0.094	**0.024**
Cylindrical Glass	0.147	0.218	**0.131**	0.056	0.095	**0.084**	0.048	0.084	**0.025**
Angular glass	0.143	0.275	**0.128**	0.064	0.087	**0.016**	0.054	0.107	**0.035**
Belly Glass Vase	0.209	0.371	**0.169**	0.047	0.109	**0.021**	0.092	0.154	**0.043**
Glass Fish Tank	1.057	1.284	**0.919**	0.634	0.104	**0.028**	0.488	0.624	**0.423**
Mean	0.345	0.468	**0.301**	0.164	0.086	**0.034**	0.146	0.213	**0.110**

Table 3. Ablation study

	Time(s)
Segmentation	0.03913
Mask Adjustment	0.01537
TO-Reconstruction	0.6715

Table 4. Time consumption of Main Part

camera trajectory performance. Moreover, we use four metrics widely used in semantic segmentation to evaluate the segmentation result, they are Intersection over Union (IoU), Pixel Accuracy Metrics (Acc), Mean Absolute Error (MAE) metrics, and Balance Error Rate (BER).

4.4. Results And Analysis

We provide the averaged quantitative comparisons of tracking tasks on all sequences in table 1. From table 1, we can observe that our method has significantly improved all metrics' performance on the tracking task in the Trans-SLAM database. The fr1/desk and fr1/xyz are from the TUM RGB-D dataset and exclude transparent objects. The Elasticfusion, RGB-D SLAM V2 and our approach all work well when there is no transparent object. However, when there is a transparent object, their performances have decreased dramatically, while our approach can still works adequately in the environment contains transparent objects.

From table 2, we can observe that our segmentation method achieves the best performance in the trans10k database compared with the other methods.

Fig. 9 shows the reconstruction results in the Trans-SLAM database. When the environment contains transparent objects, Elasticfusion and RGB-D SLAM V2 can not recover the scene's shape correctly, especially the transparent objects and the area behind them. Corresponding, our method can recover the correct shape of the whole scene.

Table 4 represents the time consumption of each part. The whole system operates at an interactive frame rate, which can run in real-time except for transparent object reconstruction, and the transparent object reconstruction can be done off-line by the saved masks and camera trajectory.

Table 1 shows that the degree of transparent objects' influence on the SLAM system is also related to their curvature and the size of the area they occupied. Compared with angular glass, the cylindrical glass's error correspondingly greater because the ray has more distortion when it hits the transparent object with extensive curvature. Furthermore, the fish tank has the most significant error. Because fish tank occupies a larger volume, the area influenced by it is more significant.

4.5. Ablation Study

We conduct ablation studies (table 3) to investigate the individual contribution of our method's component on the Trans-SLAM. For the "No-Mask Adjustment," we remove the mask adjustment part from the algorithm. For the "Use SINet," we use the SINet as our segmentation network.

From table 3, we can observe that by cutting transparent objects from depth input, all the algorithms have improved in the pose estimation. The mask adjustment part also has contributed to reducing the error because the mask adjustment part can adjust the mask and get the most accurate result. When using SINet as our segmentation network , the algorithm also has an improvement, but it is worse than the other method listed in the table 3. This result proves that the performance of our network is better than SINet.

5. Conclusion

We propose a novel RGB-D SLAM system called Transfusion, which can correctly estimate the camera trajectory and reconstruct the scene when there are transparent objects in the environment. Transfusion uses the TC-ICP to estimate the camera pose without transparent objects' influence by cutting them out. Then use the TO-reconstruction to recover the shape of the transparent object. Experiments show that our method can improve the performance in both pose establishment and reconstruction. **Limitation.** Because the scene of database is small, the camera moves have some close loop, so the reconstruction result has some noises. Although our method achieve good performance to recover the shape of transparent objects, the quality of reconstruction results still needs to be improved. Moreover, the speed of the reconstruction period still has space for improvement for real-time reconstruction.

Acknowledgments

This work is supported by the Open Project Program of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (No.VRLAB2021A04).

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. 3

[2] Josep Aulinas, Yvan Petillot, Joaquim Salvi, and Xavier Lladó. The SLAM problem: a survey. page 9. 1

[3] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, Sept. 2006. 1

[4] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016. 6

[5] Alejo Concha and Javier Civera. Rgbdtam: A cost-effective and accurate rgb-d tracking and mapping system. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 6756–6763. IEEE, 2017. 1

[6] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, June 2007. 2

[7] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part I. *IEEE Robotics & Automation Magazine*, 13(2):99–110, June 2006. 1

[8] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, volume 8690, pages 834–849. Springer International Publishing, Cham, 2014. Series Title: Lecture Notes in Computer Science. 2

[9] N. Alt et. al. Reconstruction of transparent objects in unstructured scenes with a depth camera. In *ICIP*, 2013. 3

[10] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2777–2787, 2020. 4, 7

[11] Mario Fritz, Gary Bradski, Sergey Karayev, Trevor Darrell, and Michael Black. An additive latent feature model for transparent object recognition. *Advances in Neural Information Processing Systems*, 22:558–566, 2009. 3

[12] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 6

[13] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 3

[14] 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. 4

[15] Xiaowei Hu, Lei Zhu, Chi-Wing Fu, Jing Qin, and Pheng-Ann Heng. Direction-aware spatial context features for shadow detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7454–7462, 2018. 7

[16] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Dense visual slam for rgb-d cameras. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2100–2106. IEEE, 2013. 1, 2

[17] Georg Klein and David Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–10, Nara, Japan, Nov. 2007. IEEE. 2

[18] Qingpeng Li, Lichao Mou, Qizhi Xu, Yun Zhang, and Xiao Xiang Zhu. R3̂-net: A deep network for multi-oriented vehicle detection in aerial images and videos. *arXiv preprint arXiv:1808.05560*, 2018. 7

[19] Zhengqin Li, Yu-Ying Yeh, and Manmohan Chandraker. Through the Looking Glass: Neural 3D Reconstruction of Transparent Shapes. page 10. 3, 6

[20] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 3

[21] 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. 3

[22] Kenton McHenry, Jean Ponce, and David Forsyth. Finding glass. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 973–979. IEEE, 2005. 3

[23] H. Murase. Surface shape reconstruction of an undulating transparent object. In *[1990] Proceedings Third International Conference on Computer Vision*, pages 313–317, Osaka, Japan, 1990. IEEE Comput. Soc. Press. 3

[24] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohli, Jamie Shotton, and Andrew Fitzgibbon. KinectFusion: Real-Time Dense Surface Mapping and Tracking. page 66. 1, 2

[25] 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. 3

[26] Yiming Qian, Minglun Gong, and Yee-Hong Yang. 3D Reconstruction of Transparent Objects with Position-Normal Consistency. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4369–4377, Las Vegas, NV, USA, June 2016. IEEE. 3

[27] Ruigang Yang, Pollefeys, and Welch. Dealing with textureless regions and specular highlights - a progressive space carving scheme using a novel photo-consistency measure. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 576–584 vol.1, 2003. 6

[28] 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. 3

[29] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012. 6

[30] 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. 3

[31] Thomas Whelan, Stefan Leutenegger, Renato Salas Moreno, Ben Glocker, and Andrew Davison. ElasticFusion: Dense SLAM Without A Pose Graph. In *Robotics: Science and Systems XI*. Robotics: Science and Systems Foundation, July 2015. 1, 2

[32] Enze Xie, Wenjia Wang, Wenhai Wang, Mingyu Ding, Chunhua Shen, and Ping Luo. Segmenting transparent objects in the wild. *arXiv preprint arXiv:2003.13948*, 2020. 7

[33] Masaki Yamazaki, Sho Iwata, and Gang Xu. Dense 3D Reconstruction of Specular and Transparent Objects Using Stereo Cameras and Phase-Shift Method. In Yasushi Yagi, Sing Bing Kang, In So Kweon, and Hongbin Zha, editors, *Computer Vision C ACCV 2007*, volume 4844, pages 570–579. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. Series Title: Lecture Notes in Computer Science. 3

[34] Lei Zhu, Zijun Deng, Xiaowei Hu, Chi-Wing Fu, Xuemiao Xu, Jing Qin, and Pheng-Ann Heng. Bidirectional feature pyramid network with recurrent attention residual modules for shadow detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 121–136, 2018. 7