Neural Radiance Fields with Unknown Camera Poses

Yuchen Luo under the supervision of Dr. Liuhua Peng
School of Mathematics and Statistics, The University of Melbourne

Introduction

Traditional 3D modelling methods: triangular mesh, point cloud, occupancy field and signed distance field.

- Disadvantage: require huge storage space and computational resources.
- Expensive to do high quality volume rendering.

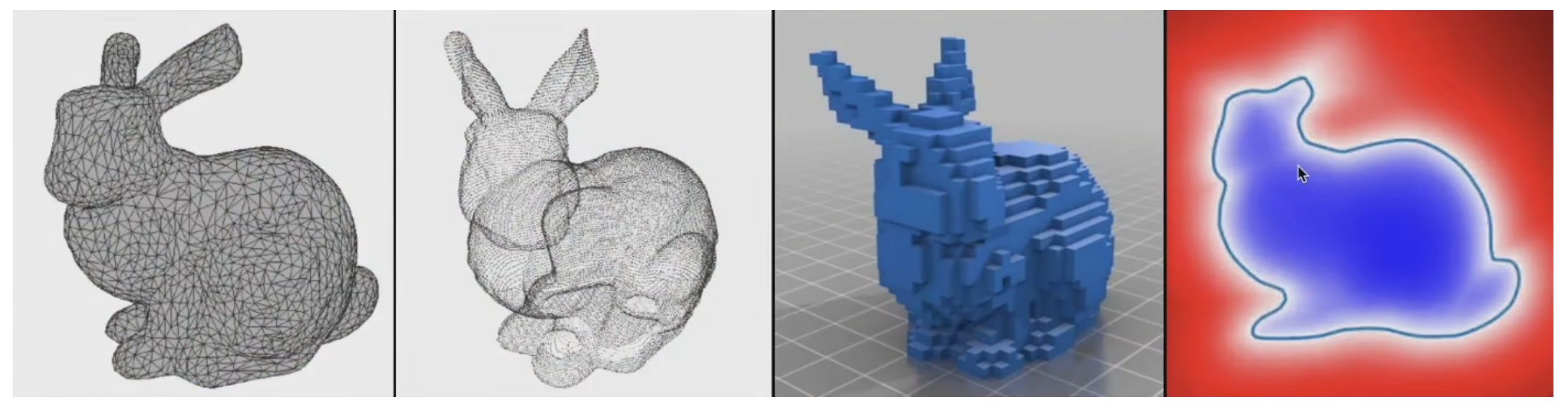


Fig. 1: Mesh, Point Cloud, Occupancy field, Signed distance field

NeRF [2] is introduced as a new method to achieve state-of-art results for synthesizing novel views for complex scenes.

- implicit-representation which allows a more easy and cheaper way to extract 3D contents from images.
- The original NeRF requires camera pose data.
- This project mainly focuses on implementing the NeRF model with no information on camera while making inferences on the poses.

NeRF Model

- A set of images $\mathcal{I} = \{I_1, I_2, I_3, ..., I_N\}$ taken from N sparse view points
- Corresponding camera poses $\mathcal{C} = \{C_1, C_2, C_3, ..., C_n\}$

The implicit representation of the neural field is modelled through a 5D continuous function $\mathbf{F} : (\mathbf{x}, \mathbf{d}) \to (\mathbf{c}, \sigma)$, following:

- $\mathbf{x} = (x, y, z)$ represents the 3D real world coordinate
- $\mathbf{d} = (\theta, \phi)$ represents the viewing direction
- $\mathbf{c} = (R, G, B)$ represents the radiance color
- σ represents the volume density

To render a pixel on the output image, we emit a ray through, shooting from the camera position and accumulate the radiance along the ray according to the rendering function [1] for expected color C:

$$C(\mathbf{r}) = \int_0^{+\infty} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})\, dt,$$

where $T(t) = \exp(-\int_0^t \sigma(\mathbf{r}(s))\, ds)$ stands for the aggregated transmittance along the ray $\mathbf{r}$.

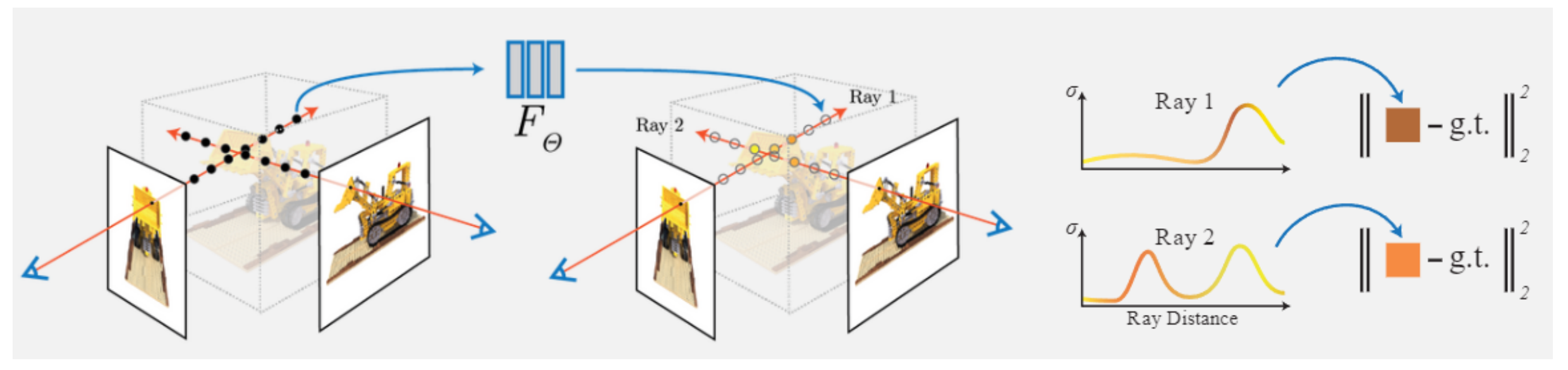


Fig. 2: NeRF

The Loss function is adopted as the Mean Square Error (MSE) to optimise:

$$\mathcal{L} = \sum_{\mathbf{r}} \|\hat{C}(\mathbf{r}) - C(\mathbf{r})\|_2^2$$

Pose Estimation

To estimate cameras poses, we consider two types of camera-world transformations:

- 3D Rotation group $\mathcal{R}$
- Translation $\vec{\mathbf{t}} \in \mathbb{R}^3$

The rotation matrix is expressed using Euler Parameters: four real numbers $(a, (b, c, d)) = (a, \vec{\omega})$ such that $a^2 + b^2 + c^2 + d^2 = 1$.

The new position of point $\vec{p}$ after rotation: $(a, (b, c, d)) = (a, \vec{\omega})$:

$$\vec{p}' = \vec{p} + 2a(\vec{\omega} \times \vec{p}) + 2(\vec{\omega} \times (\vec{\omega} \times \vec{p}))$$

To rotate an angle of ψ along the unit axis $\vec{k} = (k_x, k_y, k_z)$, the Euler's parameters will be as follows:

$$a = cos(\frac{\psi}{2}), \vec{\omega} = (b, c, d) = sin(\frac{\psi}{2})\vec{k} = (k_x sin(\frac{\psi}{2}), k_y sin(\frac{\psi}{2}), k_z sin(\frac{\psi}{2}))$$

We could simply put a 4D array $\vec{\mathbf{e}} = (\psi, k_x, k_y, k_z)$ into the neural network to train for the rotation matrix. The translation vector $\vec{\mathbf{t}}$ is also trainable as is defined in Euclidean Space. Therefore the estimation of camera poses will be composed of two networks with $\vec{\mathbf{t}}$ and $\vec{\mathbf{e}}$ being the input respectively.

Positional Encoding

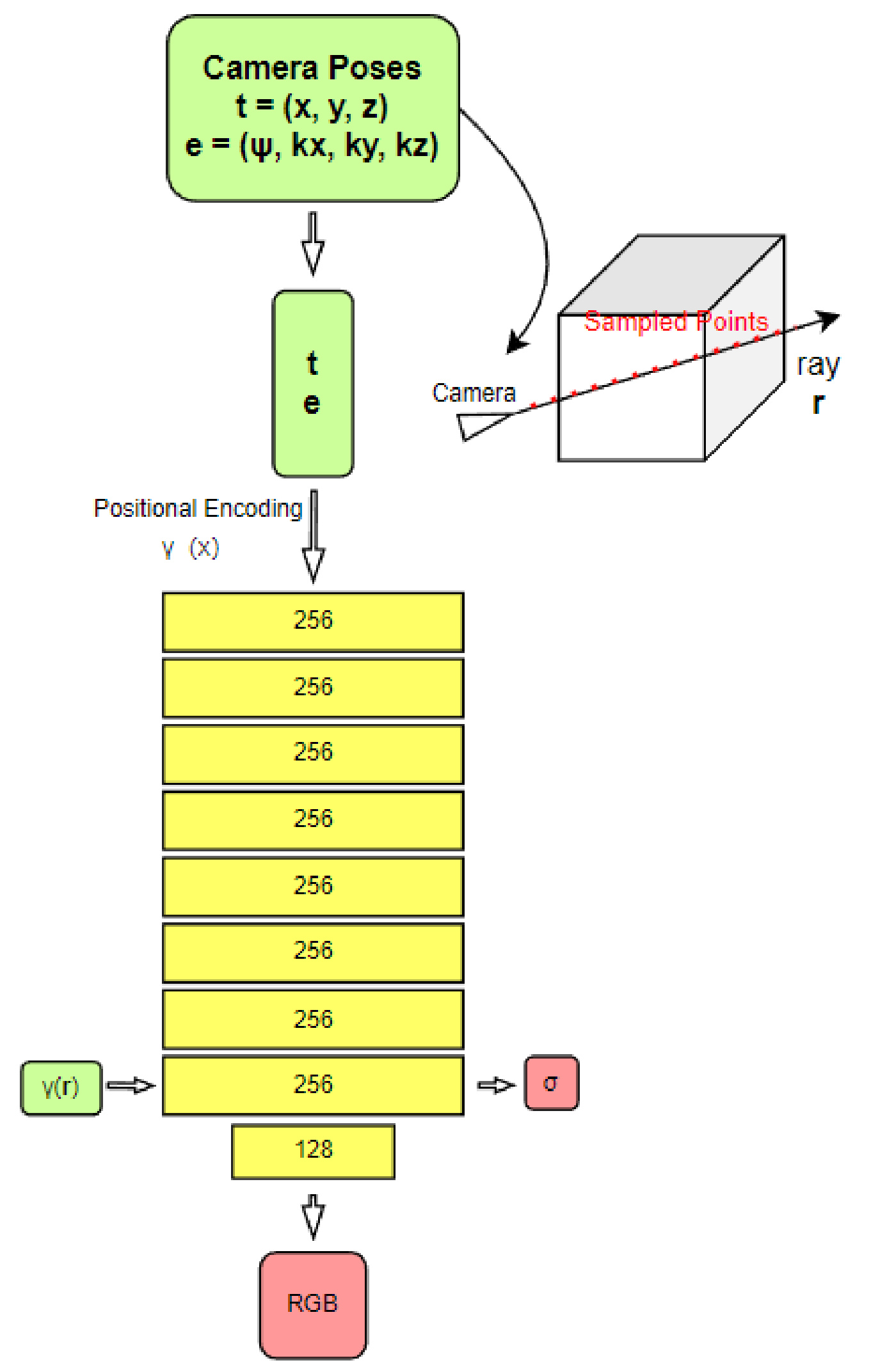
As result in [3], neural networks are often biased against high frequency function, the use of positional encoding could transfer the pixel data into high-dimensional space to strengthen the learning:

$$\mathbf{F} = \mathbf{F}' \circ \gamma, \gamma(p) = (sin(2^0\pi p), cos(2^0\pi p), sin(2^1\pi p), cos(2^1\pi p), ..., sin(2^L\pi p), cos(2^L\pi p))$$

In practical, $L = 10$ for 3D coordinate and $L = 4$ for viewing direction.

Methods

A deep neural network with 8 fully-connected ReLU layers, each with 256 channels is constructed. The Pipeline of the network and the algorithms are as shown below.



```
Algorithm 1: NeRF with Poses estimation
  Data: $N$ images $\mathcal{I} = \{I_1, I_2, ..., I_N\}$
  Result: NeRF model $\mathbf{F}$, camera poses $[\hat{\mathbf{t}}_i]$, $[\hat{\mathbf{e}}_i]$
nn ← torch.nn;
$[\hat{\mathbf{t}}_i]$ ← nn.Parameter(shape = (N, 3));
$[\hat{\mathbf{e}}_i]$ ← nn.Parameter(shape = (N, 4));        /* Initialisation */
$\mathbf{F}$ = NeRF_Module();                                  /* NeRF Structure */
for $i \leftarrow 1$ to $N$ do
    /* loop through pixels */
    for $j \leftarrow 1$ to $M$ do
        $\hat{\mathbf{r}}_{i,j}$ ← construct_ray($\hat{\mathbf{t}}_i, \hat{\mathbf{e}}_i, \hat{\mathbf{k}}_{i,m}$);
        /* Sample Points */
        for $k \leftarrow k_n$ to $k_f$ do
            $\mathbf{x}_k$ ← sample_point($\hat{\mathbf{r}}_{i,j}, \hat{\mathbf{t}}_i, k$);
            $\mathbf{c}_k, \sigma_k \leftarrow \mathbf{F}(\mathbf{x}_k, \hat{\mathbf{r}}_{i,j})$;   /* Forward NeRF Module */
        end
        $\mathcal{L} \leftarrow \sum_{\mathbf{r}} \|\hat{C}(\mathbf{r}) - C(\mathbf{r})\|_2^2$;
        $\mathcal{L}$.backward();
        NeRF.update($\hat{\mathbf{t}}_i, \hat{\mathbf{e}}_i, F$);
    end
end
```

Results and Analysis

By comparing the novel rendered view with the ground level truth of the image of the scene, our result demonstrates high quality of view synthesis. Quantitative Evaluation for 10k epoches training:

Scene	PSNR	Trainging Time	Inference Time	Pose Estimation
Bear Bag	29	171 mins	87 sec	316 sec
Rubbish Bin	33	142 mins	73 sec	305 sec
Flower	30	115 mins	77 sec	294 sec

The final result is presented below with the visualized poses on the right side. A key assumption we made is that the cameras are placed on a vertical 2D x-y plane. Further improvements could be made by extending a third axis of the camera position which would require higher computational ability and time consumption.

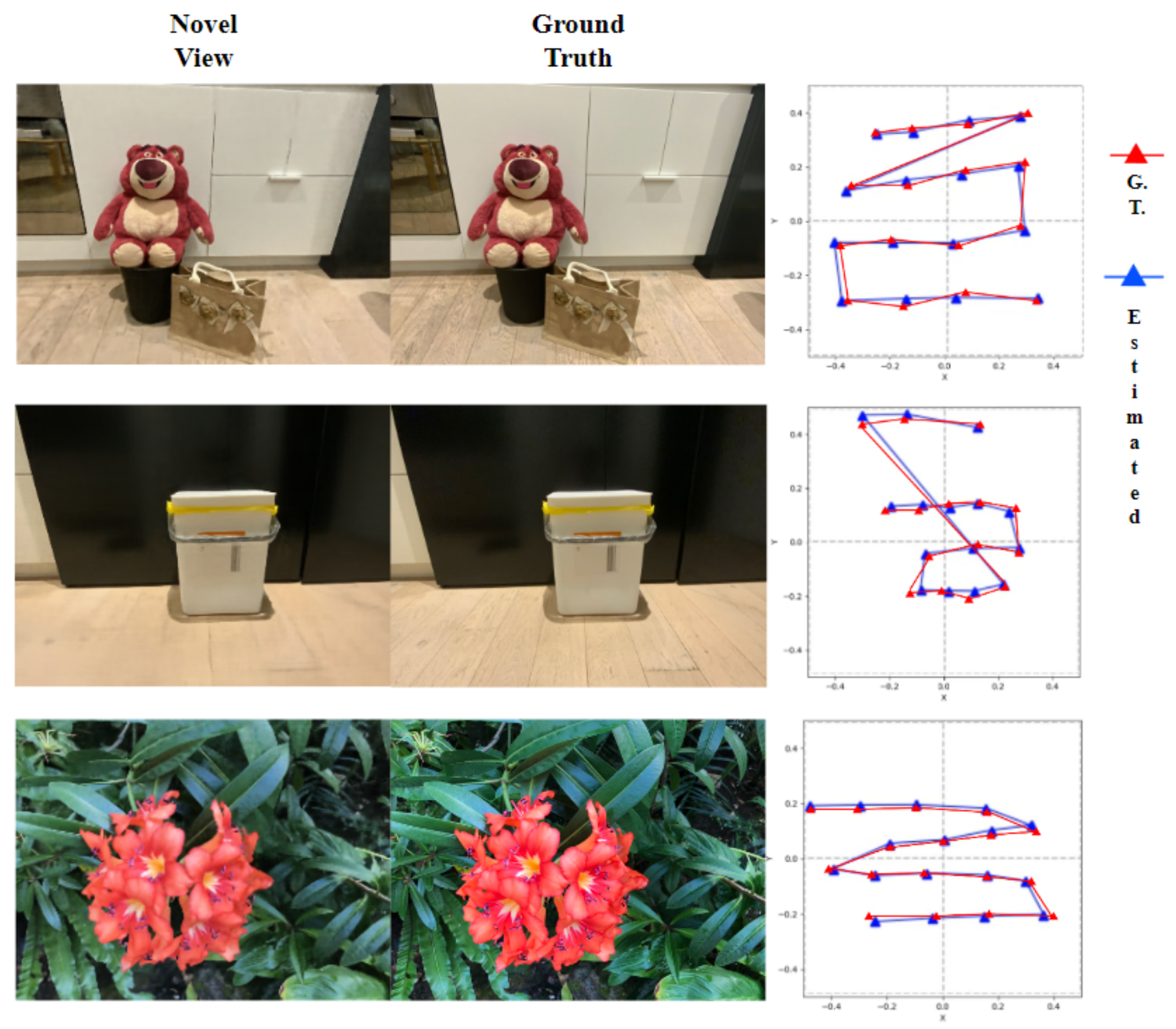


Fig. 4: Final Result

Acknowledgements

I would like to express my special thanks of gratitude to my supervisor Dr Liuhua Peng for introducing me to the interesting and modern techniques in the research area of deep learning and computer vision. Thanks for all the patient guidance, and expert explanations throughout the project, without which I could never achieve the final goal and the amazing outcomes.
I would also like to thank the School of Mathematics and Statistics at the University of Melbourne for offering this wonderful and invaluable opportunity to attend the project. This will be the most impressive experience as it provides me the insights of the amazing world of research.

References

[1] Nelson Max. "Optical Models for Direct Volume Rendering". In: (1995).
[2] Ben Mildenhall et al. "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis". In: *ECCV*. 2020.
[3] Matthew Tancik et al. "Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains". In: *NeurIPS* (2020).