Flowed Flight Fields: Dynamic View Synthesis and Time-of-Flight Corrections Under Motion

by
Marc Mapeke

A Thesis submitted in partial fulfillment of the requirements for Honors in the Department of Computer Science at Brown University

Providence, Rhode Island
May 2023
This thesis by Marc Mapeke is accepted in its present form by
the Department of Computer Science as satisfying the research requirement
for the awardment of Honors.

Date ______________  ____________________________________
                  James Tompkin, Reader

Date ______________  ____________________________________
                  Srinath Sridhar, Reader
Acknowledgements

I would like to extend my heartfelt appreciation to James Tompkin for his guidance, support, and expertise throughout the completion of this thesis. Thank you for teaching me how to do research and encouraging my curiosity over these last two years. I would like to also thank Srinath Sridhar for being my second reader.

I am also thankful to all of my collaborators for their attention to detail and mentorship, which has enriched my knowledge in visual computing research and guided this project. Thank you to Mikhail Okunev for his constant guidance. I would also like to thank Benjamin Attal, Matthew O’Toole, and Christian Richardt.

I am grateful to my friends and classmates for their support, especially for all my peers in Mosaic+. Lastly, I want to express my deepest gratitude to my family for their love and support. Their belief in my abilities has been a constant source of motivation.

Thank you all for your invaluable contributions to this thesis and my time at Brown.
## Contents

1 Introduction .................................................. 1

2 Related Work .................................................. 3
   2.1 Dynamic Novel View Synthesis .......................... 3
   2.2 Time-of-Flight Correction .............................. 4

3 Background .................................................... 5
   3.1 Neural Radiance Fields ................................. 5
   3.2 Time-of-Flight Imaging ................................. 7
   3.3 Time-of-Flight Radiance Fields ....................... 9

4 Dataset ......................................................... 10

5 Method .......................................................... 12
   5.1 Raw Time Of Flight Model ............................. 12
      5.1.1 Loss Function ................................... 13
   5.2 Augmented Dataset .................................... 14
   5.3 Scene Flow ............................................... 16
      5.3.1 Temporal Consistency and Priors ............... 16
      5.3.2 Scene Flow Regularizations ................... 18
      5.3.3 Additional Details .............................. 19
      5.3.4 Loss Function ................................... 19

6 Experiments ................................................... 21
   6.1 Quantitative Evaluation ............................... 21
   6.2 Qualitative Evaluation ............................... 22

7 Conclusion .................................................... 28

Bibliography .................................................... 29
Chapter 1

Introduction

Depth estimation and novel view synthesis (NVS) are two related and increasingly important areas of computer vision and computer graphics research that have a wide range of applications in different fields. Depth estimation refers to the process of inferring the distance of objects in a scene from a given image or video. This task is crucial for many applications such as 3D reconstruction, autonomous driving, augmented reality, and virtual reality. Accurate depth estimation can provide strong information about the spatial layout of a scene, which is essential for many computer vision tasks.

On the other hand, novel view synthesis involves the generation of new views of a scene from a limited set of input views, by extrapolating and interpolating the scene’s appearance and geometry. Novel view synthesis is a challenging task because it requires synthesizing views that are not present in the original input, while preserving the consistency and visual quality of the scene’s geometry and appearance. This task has numerous applications in areas such as visual effects, virtual and augmented reality, and telepresence.

Dynamic novel view synthesis extends this problem to reconstructing dynamic scenes. Techniques for dynamic NVS from a monocular video sequence have demonstrated compelling results, though they can suffer from various visual artifacts due to the ill-posed nature of this problem and often require introducing priors on the dynamic scene’s depth and motion.

Separately, mobile and consumer devices now have camera systems with both color and depth sensors, including Microsoft’s Kinect or some models of the iPhone and iPad Pro. A common hardware approach to depth sensing is using time-of-flight systems that measure the time it takes for an emitted light to travel to an object and back again. By measuring the time it takes for a signal to make this round trip, we can recover accurate depth. TörRF is one neural method that is conditioned on raw time-of-flight data [1].

However, ToF reconstruction is not accurate for objects that are quickly moving. A single ToF reconstruction requires sequentially captured measurements leading to reconstruction error for objects that move in between these measurements.

To improve NVS performance in dynamic environments with fast motion, we propose this method
as a neural representation to represent and correct raw time-of-flight measurements for dynamic scenes. We evaluate our approach on custom synthetic scenes and show that it visually improves scene reconstruction results. Our method also allows us to recover raw ToF measurements at arbitrary time steps. Overall, our work demonstrates progress in improving two separate problems 1) ToF motion correction and 2) dynamic novel view synthesis, and we do this by combining insights and observations from both tasks. Addressing current limitations in both tasks will encourage further downstream usage and applications across various fields.
Chapter 2

Related Work

2.1 Dynamic Novel View Synthesis

Novel view synthesis is a technique to generate a new image of a scene or object from a different viewpoint than the original images. This can be accomplished by combining multiple images of the scene or object taken from different viewpoints, or by using a 3D model of the scene or object to render new views. Dynamic novel view synthesis extends this to dynamic scenes, where we can interpolate in both space and time.

Recently deep learning methods have shown that implicit neural representations have the ability to represent scenes with impressive view synthesis results. Our method is based on an extension of neural radiance fields or NeRF [2]. Neural fields capture a continuous scene representation that can be sampled anywhere, and demonstrate low memory requirements when compared to traditional computer graphics data structures (voxel grids, triangle meshes, etc.).

Applying neural radiance fields naively to dynamic scenes does not produce plausible results. Camera rays across different timesteps may be observing different objects and geometry in the dynamic scenes.

Some methods for representing dynamic scenes through neural networks assume that there are multiple time synchronized cameras capturing a scene. On the other hand, this work focuses on monocular video, where we have one camera view per timestep. The monocular video setting is better suited for real world applications and casual capture. This is a high-ill posed problem, but there are several recent approaches that show the ability to perform dynamic novel view synthesis with a single camera [3, 1, 4]. TöRF is a radiance field method that is conditioned on raw time-of-flight data. Neural Scene Flow Fields is a dynamic radiance field method that is conditioned on scene priors, such as depth and optical flow, and also models an explicit scene flow field to represent motion. Our method combines ideas from both of these approaches to represent dynamic scenes with fast motion.
2.2 Time-of-Flight Correction

Time-of-Flight (ToF) sensors can determine depth information by using various measurement modes to capture multiple raw measurements. The integration of raw images allows us to compute depth. However, current ToF systems rely on multiple consecutive measurements, because they cannot perform all measurements simultaneously. These consecutive measurements can result in motion artifacts that decrease the quality of the reconstructed scene depth. Several studies propose methods to detect, reduce, and/or correct these motion artifacts in ToF sensors. These methods vary substantially in approaches, but in this section we overview a few of the key ideas and approaches in these methods and address their limitations.

Several methods, such as Linder and Kolb, compensate for the motion artifacts based on optical flow [5]. Raw ToF measurements do not fulfill brightness constancy requirements expected for optical flow, so normalization tricks are typically applied. Estimating optical flow can be expensive, and is only a 2D representation of scene motion.

Some methods constrain the motion artifact corrections to certain types of motion. For example, Hussman et al.’s method is restricted to small (less than 1 meter) linear motion along conveyor belts [6]. Other methods, such as Hoegg et al. restrict the motion artifacts to only blurred areas [7].

Schmidt’s method detects and corrects motion artifacts in Time-of-Flight measurements based solely on temporal relations and temporal derivatives in the raw ToF measurements [8]. This method is simple to implement, and avoids expensive spatial operations. However, spatial information (whether it is from color images, or from the raw ToF measurements) captures additional information that can exploited in ToF corrections. Also, this method recovers 2 depth maps, instead of $n$ depth maps, where $n$ is the number of raw measurements integrated by the ToF system.

Other methods to correct for motion artifacts in ToF reconstruction rely on novel hardware constructions. This limits their applicability to existing ToF systems and hardware.
Chapter 3

Background

3.1 Neural Radiance Fields

Figure 3.1: Neural radiance fields optimize a volumetric radiance field to render a scene from new views. This figure demonstrates the pipeline for training a neural radiance field or NeRF. NeRF takes in a set of posed input images, and regresses a continuous 5D radiance field that fits the scene. This radiance field can then be rendered from novel viewpoints. This figure is borrowed from [2].
Table 3.1: Mathematical symbol legend for the following equations and explanations.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>A point ( \in \mathbb{R}^3 ).</td>
</tr>
<tr>
<td>$\omega$</td>
<td>A direction; unit vector ( \in S^2 ).</td>
</tr>
<tr>
<td>$x_t$</td>
<td>A point ( t ) units along a direction ( \omega ), ( x_t = x + \omega t ).</td>
</tr>
<tr>
<td>$\omega_i$</td>
<td>A direction incoming to a point.</td>
</tr>
<tr>
<td>$\omega_o$</td>
<td>A direction outgoing from a point.</td>
</tr>
<tr>
<td>$r_i$</td>
<td>A camera ray at time ( i ).</td>
</tr>
</tbody>
</table>

$L(x, \omega)$ or $L_{RGB}$  
Radiance measured by a camera at point \( x \) in direction \( \omega \).

$L_{ToF}(x, \omega)$  
Phasor radiance measured by a C-ToF camera.

$L_o(x, \omega)$  
Raw measurements captured by a C-ToF camera. Each measurement corresponds to a different phase offset.

$L_s(x, \omega)$  
Reflected radiance scattered from a point \( x \) in direction \( \omega \).

$I$  
Radiant intensity of a point light source.

$I_s(x, \omega)$  
Reflected radiant intensity scattered from a point \( x \) in direction \( \omega \) due to a light source collocated with the camera.

$I_s(x, \omega)$  
Vector of reflected radiant intensities scattered from a point \( x \) in direction \( \omega \) due to a light source collocated with the camera.

$\sigma(x_t)$  
Radiance field density at a point.

$T(x, x_t)$  
Transmittance function.

$F_i(x_t) = (f_{i-\rightarrow i+1}, f_{i-\rightarrow i-1})$  
Forward and backward 3D scene flow for a 3D point at time \( i \).

$\hat{F}_i(r_i)$  
Volume rendered scene flow for a ray at time \( i \).

$W_i(x_t) = (w_{i-\rightarrow i+1}, w_{i-\rightarrow i-1})$  
Predicted disocclusion weights for a 3D point at time \( i \).

$\hat{W}_i(r_i)$  
Volume rendered disocclusion weights for a ray at time \( i \).

$X_i(r_i)$  
Expected 3D point location based on depth/expected ray termination.

A neural radiance field (NeRF) is a neural network that is regressed to fit a scene by predicting a set of posed input images [2]. Assuming a static scene, the neural network $F_\theta : (x_t, \omega_o) \rightarrow (\sigma(x_t), L_s(x_t, \omega_o))$ with parameters $\theta$ takes as input a position $x_t$ and a direction $\omega_o$, and outputs both the density $\sigma(x_t)$ at point $x_t$ and the radiance $L_s(x_t, \omega_o)$ of a light ray passing through $x_t$ in direction $\omega_o$. The inputs are mapped to higher dimensions using a fourier feature mapping, which has been shown to aid multilayer perceptrons in learning high frequency functions and details [9].

The volume density function $\sigma(x_t)$ controls the opacity at every point, and allows NeRF’s to represent a wide range of 3D structures. The radiance function $L_s(x_t, \omega_o)$ represents the light scattered at a point $x_t$ in direction $\omega_o$, and describes the visual appearance of different materials (e.g., shiny or matte).

Supervising a 3D radiance field through posed 2D images requires the use of volume rendering. Volume rendering allows us to synthesize a 2D image from our 3D radiance field, by tracing camera rays through the neural volume and querying the density and radiance at multiple points along a camera ray.

Volume rendering along a ray allows us to compute the expected color of a camera ray:
Figure 3.2: Time-of-flight camera systems estimate scene depth by measuring the time it takes for an emitted light wave to hit an object in the scene and then reflect back towards the camera. This figure is borrowed from [10].

\[
L_{\text{RGB}}(x, \omega_o) = \int_{t_n}^{t_f} T(x, x_t) \sigma(x_t, \tau) L_n(x_t, \omega_o) \, dt, \tag{3.1}
\]

where \( T(x, x_t) = e^{-\int_{t_n}^{t_f} \sigma(x-x_s, s) \, ds} \) computes the transmittance of light from \( x \) to \( x_t \) along a ray from the near bound \( t_n \) to the far bound \( t_f \).

In practice, these continuous integrals are evaluated through quadrature and stratified random sampling. These functions allow us to render images of a scene from any given camera pose.

### 3.2 Time-of-Flight Imaging

Time-of-Flight technology provides 3D imaging at a low-cost with its high accuracy and performance making it useful for a variety of applications.

In general, time-of-flight (ToF) cameras work by actively illuminating the scene with a light source, and then observing the reflected light. The phase shift between the reflected and emitted signal can be measured and converted to distance (Fig. 3.2).

To compute the phase shift between the emitted signal and the reflected signal, the light source on the ToF system is either pulsed or sends a continuous wave of light. Both ToRF and this method focus on modelling continuous wave ToF systems. ToF cameras using the pulse based approach are not easily available, and require extremely precise timing controls [10].
Figure 3.3: Continuous wave ToF cameras estimate scene depth using four sequential measurements. Each of the raw 4 ToF samples are acquired by correlating the reflected signal with emitted signal phase stepped by different amounts. This figure is borrowed from [10].

The received light wave is modified by multiple factors (such as ambient light, attenuation, and phase shift), so the signal returned is not equivalent to our source signal. Therefore, to produce a single measurement that corresponds to phase shift and depth, the continuous wave method acquires 4 samples. Each of these samples correspond to correlating the reflected signal with emitted signal phase-stepped by \( \theta \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\} \) (Fig. 3.3).

With these raw ToF measurements \( Q \), you can recover the phase offset \( \phi \) between the emitted and reflected signals, and also the scene depth \( d \) using these equations:

\[
\phi = \arctan \left( \frac{Q_3 - Q_4}{Q_1 - Q_2} \right),
\]

\[
d = \frac{c}{4\pi f}\phi,
\]

where \( c \) is the speed of light and \( f \) is the modulation frequency of our emitted signal [10]. These raw ToF measurements can be represented in a single phasor image, where the numerator, \( Q_3 - Q_4 \), represents the real component, and the denominator, \( Q_1 - Q_2 \), represents the imaginary component. The magnitude of this phasor image corresponds to amplitude.

The 4 samples \( Q_1, Q_2, Q_3, Q_4 \) are acquired sequentially in time. Motion blur in the ToF reconstruction is a form of non-systematic depth error caused by camera or object motion. Motion between the raw samples causes misalignment, which results in incorrect phase offset estimation for moving objects. In color cameras, motion blur appears as smooth color transitions between the
foreground and background. On the other hand, motion blur in ToF cameras is non-smooth because of the non-linear transformations needed to convert the four raw measurements into depth [11].

### 3.3 Time-of-Flight Radiance Fields

Time-of-Flight Radiance Fields or TöRF extend the neural volume rendering procedure of NeRF to support ToF cameras [1]. ToF cameras recover depth through active illumination, so TöRF’s image formation model captures how the scene’s lighting changes with the position of the camera, and extends the volume rendering equation to model raw phasor images from a ToF camera.

The additional volume rendering equation introduced in TöRF to supervise ToF measurements is:

\[
L_\theta(x, \omega_o) = \int_{t_n}^{t_l} \frac{T(x, x_t)^2}{\|x - x_t\|^2} \sigma(x_t) I_s(x_t, \omega_o) W(2 \|x - x_t\|) dt, \tag{3.4}
\]

and similar to NeRF’s formulation this integral is evaluated through quadrature and stratified random sampling. This additional image formation model allow us to render and supervise raw ToF phasor images of a scene from any given camera pose.

An important detail here is that TöRF directly models the ToF phasor images, and not the ToF derived depth. Supervising the ToF phasor images allows TöRF to correct some systematic and non-systematic depth errors in the ToF system.
Chapter 4

Dataset

To study the errors in ToF reconstruction for fast moving objects, we created a custom Blender pipeline to render synthetic sequences with raw ToF data by modifying the physically-based render PBRT [12]. This allows us to extract ground truth depth, scene flow, camera poses, etc. This also allows us to extract raw ToF measurements at each timestep, which is not available for real world ToF cameras.

These scenes in the dataset exhibit different types of motion and geometry that will be challenging for ToF reconstruction and classical or neural scene flow estimation. These scenes feature a single static camera used to capture the scene. Reconstructing scene motion and depth from a static camera is highly ill-posed, and there are multiple incorrect solutions that may fit the training views, but provide implausible results for view synthesis. In Figure 4.1, the reconstructed ToF phase and color image are shown at one timestep for 3 fast moving cubes. We can see ghosting in the recovered phase, because standard ToF cameras do not account for motion. These motion errors in the ToF reconstruction propagate themselves into the predicted scene depth, and present issues when performing NVS.
Figure 4.1: Fast scene motion causes blurring in ToF reconstructions. In this scene, the bottom-most cube is moving the fastest from left to right, and the reconstructed geometry is the most inaccurate.
Chapter 5

Method

5.1 Raw Time Of Flight Model

<table>
<thead>
<tr>
<th></th>
<th>( t )</th>
<th>( t + 0.25 )</th>
<th>( t + 0.50 )</th>
<th>( t + 0.75 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToF_0</td>
<td>⬠</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ToF_1</td>
<td></td>
<td>⬠</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ToF_2</td>
<td></td>
<td></td>
<td>⬠</td>
<td></td>
</tr>
<tr>
<td>ToF_3</td>
<td></td>
<td></td>
<td></td>
<td>⬠</td>
</tr>
<tr>
<td>Color</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Ground Truth, From Camera Capture*

Figure 5.1: Continuous wave ToF camera systems only provide one raw ToF measurement per timestep. After swapping the model to represent the raw ToF images at different phase offsets, we only have partial supervision for the model.

This work focuses on dynamic scenes, so we pass in \( 4t \) as an additional time parameter. The radiance field is supervised by the rendering the phasor image from the raw ToF captures of the C-TOF camera. The camera’s ToF module is typically 120Hz, and the RGB image is captured at 30hz, or one-fourth the temporal frequency of the ToF module.

Supervising this model with the rendered phasor image does not capture information about how each raw ToF sampled is captured at a different \( t \) timestep.
Therefore, the first modification to this model was to predict and supervise on raw ToF measurements at every timestep, instead of the phasor image which integrates measurements across 4 timesteps. This allows the model to leverage information from the raw ToF measurements, and creates the opportunity to interpolate the raw ToF measurements (that are not captured by the C-TOF camera) using the neural network. This also increases the temporal resolution of our model by a factor of 4.

In TöRF, the dynamic neural network $F_\theta$ is: $\langle x_t, \omega_o, 4t \rangle \rightarrow \langle \sigma(x_t), L_s(x, \omega_o), I_s(x, \omega_o) \rangle$. In our work, we modify this so that the model outputs predictions for the raw ToF measurements. The updated dynamic neural network is $F_\theta$ is: $\langle x_t, \omega_o, t \rangle \rightarrow \langle \sigma(x_t), L_s(x, \omega_o), I_s(x, \omega_o) \rangle$.

$I_s(x, \omega_o)$ is a vector of four raw ToF measurements with each corresponding to a different phase offset $\theta \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$.

Naively representing dynamic scenes with this model is not sufficient. The information needed for ToF reconstruction is critically undersampled (Fig. 5.1). One ToF measurement at each $t$ timestep can map to an incorrect scene depth. Through experiments, we found that although the model can correctly overfit to one raw ToF measurement, the remaining measurements will be incorrect.

Similar to TöRF, there is still no explicit representation for scene motion. To interpolate motion, we must interpolate the timestep $t$ as opposed to interpolating across explicit models of motion, such as scene flow.

5.1.1 Loss Function

Swapping to a raw ToF model results in small changes to TöRF’s image formation models. The static and dynamic geometry are represented by separate networks, so these are composited in the image formation model through predicted blending weights.

This produces density $\sigma^{\text{blend}}$, radiance $L_s^{\text{blend}}$, and radiant intensity $I_s^{\text{blend}}$ values to pass into our image formation models:

$$L_{\text{RGB}}(x, \omega_o, \tau) = \int_{t_n}^{t_f} T^{\text{blend}}(x, x_t) \sigma^{\text{blend}}(x_t, \tau) L_s^{\text{blend}}(x_t, \omega_o, \tau) \, dt$$  \hspace{1cm} (5.1)

$$L_\theta(x, \omega_o, \tau) = \int_{t_n}^{t_f} \frac{T^{\text{blend}}(x, x_t)^2}{\|x - x_t\|^2} \sigma^{\text{blend}}(x_t, \tau) I_s^{\text{blend}}(x_t, \omega_o, \tau) W(2 \|x - x_t\|) \, dt$$  \hspace{1cm} (5.2)

$T(x, x_t) = e^{-\int_{t_n}^{t_f} \sigma(x - \omega_o, \tau) ds}$ computes the transmittance of light from $x$ to $x_t$ along a ray from the near bound $t_n$ to the far bound $t_f$.

Given a set of color images and raw ToF measurements captured of a scene at different time instances, we sample a set of camera rays from the set of all pixels, and minimize the following total squared error between the rendered images and measured pixel values:

$$\mathcal{L} = \sum_{(x, \omega_o, \tau)} \left( \|L_{\text{RGB}}(x, \omega_o, \tau) - \hat{L}_{\text{RGB}}(x, \omega_o, \tau)\|^2 + \lambda \sum_\theta M \odot \|L_\theta(x, \omega_o, \tau) - \hat{L}_\theta(x, \omega_o, \tau)\|^2 \right),$$ \hspace{1cm} (5.3)
Figure 5.2: Augmenting the dataset through estimated optical flow produces artifacts in both the warped measurements and estimated disocclusion masks. Here, we visualize an example of our augmented data for the raw ToF measurement corresponding to phase offset $\theta = 0$. The top row visualizes the warped measurement and the bottom row shows the estimated disocclusion mask.

where the scalar $\lambda \geq 0$ controls the relative contribution of both loss terms, $\hat{L}_{\text{RGB}}(x, \omega_o, \tau)$ represents the measurements of a color camera, and $\hat{L}_{\text{ToF}}(x, \omega_o, \tau)$ represents the raw measurements of a C-ToF camera. $M$ represents a masking operation, so that we only supervise the model on ToF measurements that are provided as ground truth at each $t$ timestep.

## 5.2 Augmented Dataset

To address the undersampling in space and time represented in Figure 5.1, we began with an approach to augment the dataset. Through data augmentation, we can supervise the model with 4 raw ToF measurements at each timestep.

Optical flow between frames $t$ and $t + 1.0$ is an image of 2D vectors $O_t : R^2 \rightarrow R^2$ that maps 2D pixels between frames according to their motion. Under this assumption pixels from one frame should be mapped to the pixels of the same brightness in the next frame. This allows us to formulate a following constraint that should be minimized: if we apply a flow $O_t$ to frame $I_t$, the warped image should be the same as the image $I_{t+1}$ in the non-occluded regions.

$$||\text{warp}(O_t, I_t) - I_{t+1}) \odot M_t||_2^2$$

(5.4)

Using RAFT, we computed the forward and backward optical flow between color images [13]. These predicted flows were used to warp both color image and raw ToF measurement intensities, and to also predict disocclusion masks $M_t$ for previously occluded regions (Fig. 5.2).
Figure 5.3: Data augmentation (and other model improvements described in later sections) allow supervision on all four raw ToF measurements for each time step. Supervising four raw ToF measurements constrains the reconstructed dynamic scene. In this chart, we visualize which inputs are augmented and which are provided by camera capture.

Directly estimating optical flow between corresponding raw ToF measurements (i.e. ToF<sub>0</sub> at time t and t + 1.0) is not possible because optical flow models rely on color correspondence and brightness constancy across frames, so the predictions were not plausible. Raw ToF measurements vary in pixel intensity with changes in depth. Raw ToF measurements are also single-channel, and don’t have the same range of pixel intensities within an image that a typical color image would have in the real world. Therefore, predicting optical flow between corresponding raw ToF measurements is out of distribution for neural optical flow methods.

Our goal is to minimize the total squared error between rendered images from our radiance field and measured values, so the loss equation remains the same as equation 5.3 in this formulation. To ignore errors in regions of disocclusion, we replace M in equation 5.3 with our computed disocclusion masks. Figure 5.3 demonstrates where the augmented dataset allows us to supervise the model versus Figure 5.1.

Although this was a simple extension, we are still limited by the radiance field not having an explicit representation for scene motion. There are artifacts in the augmented data because we are relying on predicted optical flow across large motions without any further optimization on the estimated motion. Warping the raw measurements in pixel space produces artifacts in the warped image and the predicted disocclusion masks. The model is sensitive to these errors. Warping in 2D also does not capture how raw ToF measurements vary in color intensity with changes in depth.

These limitations inspired us to explicitly model scene flow in our radiance field to better represent the relationships between scene motion and raw ToF measurements.

<table>
<thead>
<tr>
<th></th>
<th>t</th>
<th>t + 0.25</th>
<th>t + 0.50</th>
<th>t + 0.75</th>
<th>t+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToF&lt;sub&gt;0&lt;/sub&gt;</td>
<td>✧</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>✧</td>
</tr>
<tr>
<td>ToF&lt;sub&gt;1&lt;/sub&gt;</td>
<td>→</td>
<td>✧</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>ToF&lt;sub&gt;2&lt;/sub&gt;</td>
<td>→</td>
<td>→</td>
<td>✧</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>ToF&lt;sub&gt;3&lt;/sub&gt;</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>✧</td>
<td>→</td>
</tr>
<tr>
<td>Color</td>
<td>✧</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
</tbody>
</table>

✧ Ground Truth, From Camera Capture

→ Augmented Data with Estimated Optical Flow
5.3 Scene Flow

We can model motion in our radiance field by also predicting a scene flow field. Scene flow between frames is a 6D vector field \( F_t(x_t) : \mathbb{R}^3 \rightarrow \mathbb{R}^6 \) that maps 3D points between forward and backward frames according to their motion. In practice, we predict a velocity field that is integrated over time to produce a scene flow or displacement, and motion is assumed to be linear between frames. We also predict disocclusion weights for the 3D motion.

Our updated dynamic model is now \( F_\theta \) is : \((x_t, \omega_o, t) \rightarrow (\sigma(x_t), L_s(x, \omega_o), I_s(x, \omega_o), F_i(x_t), W_i(x_t))\).

To enforce consistency between the predicted scene flow field and the radiance field at neighboring times, we introduce several losses.

5.3.1 Temporal Consistency and Priors

The temporal photometric consistency and temporal ToF consistency loss enforce that the scene should be consistent at neighboring times when we account for motion.

To accomplish this, we volume render rays that are warped from time \( j \) to \( i \) by the scene flow. This should undo the relative motion between time moments, and we can supervise the warped ray with the ground truth at time \( i \) (Fig. 5.4).

\[
C_{j\rightarrow i}(r_i) = \int_{t_n}^{t_f} T^{\text{blend}}(x, x_t)\sigma^{\text{blend}}(r_{i\rightarrow j}(t), j)I_s^{\text{blend}}(r_{i\rightarrow j}(t), \omega_o, j) dt
\]

where \( r_{i\rightarrow j}(t) = r_i(t) + f_{i\rightarrow j}(r_i(t)) \). (5.5)

\[
\mathcal{L}_{\text{pho}} = \sum_{r_i} \sum_{j \in i\pm 1} ||C_{j\rightarrow i}(r_i) - \hat{\mathcal{L}}_{\text{RGB}}(x, \omega_o, i)||^2_2
\]

For the temporal photometric consistency loss \( \mathcal{L}_{\text{pho}} \), our current implementation computes this only across time moments where the color images are captured.

We can define our temporal ToF consistency loss similarly to above.

\[
L_{\theta, j\rightarrow i}(r_i) = \int_{t_n}^{t_f} \frac{T^{\text{blend}}(x, x_t)^2}{\|x - x_t\|^2} \sigma^{\text{blend}}(r_{i\rightarrow j}(t), j)I_s^{\text{blend}}(r_{i\rightarrow j}(t), \omega_o, j)W(2\|x - x_t\|) dt
\]

where \( r_{i\rightarrow j}(t) = r_i(t) + f_{i\rightarrow j}(r_i(t)) \). (5.8)

\[
\mathcal{L}_{\text{tot}} = \sum_{r_i} \sum_{j \in i\pm 0.25, 0.50, 0.75} \sum_{\theta} M \odot ||L_{\theta, j\rightarrow i}(r_i) - \hat{L}_{\theta}(r_i, i)||^2_2
\]

For the temporal ToF consistency loss, our current implementation supervises the raw ToF measurements at the non-fractional time moments by warping into reference frames where we have ground truth for the raw ToF measurement. \( M \) represents a masking operation for raw measurements that do not have a camera-captured ground truth.
17

Figure 5.4: Warping and rendering rays displaced by predicted scene flow allows us to supervise the model across different timesteps. This formulation allows us to supervise the warped ray queried in the radiance field at time $j$ with a rendered ray from the radiance field at time $i$. This figure is borrowed from [3].

In this current formulation, there are two important issues that we address by converting our scene flow representation into integrated velocity over time:

1) MLP unrolling is an expensive operation, and predicting scene flow at fractional time moments to warp each fractional time moment into a reference color frame would be extremely memory intensive.

2) Estimating scene flow between raw ToF measurements is non-trivial, especially because the intensity of raw ToF measurements changes with depth. On the other hand, estimating scene flow between color images is better constrained (and can be supervised with priors).

To address this, the scene flow when warping between time $i$ and a fractional time $j$ (ex. $i + 0.25, i + 0.50, i + 0.75$) is computed as the scene flow from $i \rightarrow i \pm 1$ scaled by the factor $|i - j|$. Therefore, instead of expensive MLP unrolling, we are instead performing multiplication.

In our problem setting, we have a single camera view per timestep, making this an ill-posed monocular reconstruction problem. Similar to Li et al., we introduce a data-driven optical flow prior to avoid convergence to sub-optimal solutions [3].

The geometric consistency prior minimizes the difference between predicted 2D optical flow from neural methods, such as Teed and Deng, and our reprojected scene flow field [13].
\[
F_{i \rightarrow j}(r_i) = \int_{t_n}^{t_f} T^{\text{blend}}(x_t, i) \sigma^{\text{blend}}(x_t, i)f_{i \rightarrow j}(x_t)dt
\tag{5.11}
\]

\[
X_i(r_i) = \int_{t_n}^{t_f} T^{\text{blend}}(x_t, i) \sigma^{\text{blend}}(x_t, i)x_i(r_i(t))tdt
\tag{5.12}
\]

To supervise and visualize the scene flow in 2D, we must find the perspective projection of the warped 3D point. \(p_{i \rightarrow j}(r_i)\) represents the corresponding 2D location in the neighboring frame \(j\) from frame \(i\):

\[
p_{i \rightarrow j}(r_i) = \pi(K(R^j(X_i(r_i)) + F_{i \rightarrow j}(r_i)) + t^j))
\tag{5.13}
\]

\(K\) is the camera intrinsic matrix, which is known for our synthetic data, but can be estimated for real-world data. \(\pi\) is a perspective operation. \((R^j, t^j) \in SE(3)\) represent the transformations to go from the world coordinate system to the camera coordinate system at time \(j\). Using these, we can formulate our geometric consistency loss \(L_{\text{geo}}\), where \(\hat{p}_{i \rightarrow j}(r_i)\) is the output of [13].

\[
L_{\text{geo}} = \sum_{r_i} \sum_{r_j \in i \pm 1} ||\hat{p}_{i \rightarrow j}(r_i) - p_{i \rightarrow j}(r_i)||_1
\tag{5.14}
\]

\(L_{\text{geo}}\) is only computed between computed across color frames, or every \(4t\) time steps, where \(t \in \mathbb{Z}_{\geq 0}\). As explained in Section 5.2, this is because neural optical flow methods do not predict plausible optical flow between raw ToF measurements.

Unlike Neural Scene Flow Fields (NSFF), we do not use the output of a monocular depth estimation network as an additional prior [3]. The ToF signal provides exact information for the radiance field’s depth (ignoring errors in ToF depth reconstruction from errors, such as multi-path interference). The priors on scene flow can be noisy, so they are intended for initialization, and the weights on the geometric prior are linearly decayed to 0 over the course of training.

### 5.3.2 Scene Flow Regularizations

Taking further inspiration from NSFF, we implemented several regularizations on the predicted scene flow in the radiance field [3]. In our experiments, we have noticed that scene flow tends to diverge or converge to unreasonable solutions without these regularizations.

Cycle consistency regularizes the predicted scene flow by minimizing the difference between forward and backward scene flow for corresponding points across time. Cycle consistency is ambiguous in disocclusion regions, so this loss is tweaked by predicted disocclusion weights:

\[
L_{\text{cyc}} = \sum_{r_i} \sum_{r_j \in i \pm 1} w_{i \rightarrow j} ||f_{i \rightarrow j}(x_i) + f_{j \rightarrow i}(x_i) ||_1
\tag{5.15}
\]
Spatial smoothness in scene flow encourages scene flows sampled at neighboring points along a ray in the volume to be consistent:

\[
L_{sp} = \sum_{x_i} \sum_{y_i \in N(x_i)} \sum_{j \in i \pm 1} ||f_{i \rightarrow j}(x_i) - f_{i \rightarrow j}(y_i)||_1, \quad (5.16)
\]

and \(N(x_i)\) is the set of neighboring points along the ray \(r_i\).

Temporal smoothness in the scene flow encourages the scene to be piece-wise linear by minimizing the summation of forward and backward scene flow for points in the volume:

\[
L_{temp} = \sum_{x_i} ||f_{i \rightarrow i+1}(x_i) + f_{i \rightarrow i-1}(x_i)||_2^2 \quad (5.17)
\]

Minimal scene flow is an L1 regularization on the predicted scene flow in the volume to encourage minimal values:

\[
L_{min} = \sum_{x_i} \sum_{j \in \{i \pm 1\}} ||f_{i \rightarrow j}(x_i)||_1 \quad (5.18)
\]

### 5.3.3 Additional Details

In Section 5.1, we found that supervising the model initially with one raw ToF measurement per timestep often converges to local minima with incorrect scene depth. Often times, the reconstructed radiance field would encourage the scene depth to approach 0 or the minimum scene bounds. Introducing scene flow and temporal warping increased the model’s instability during training.

To address this issue, we supervise all the raw ToF measurements at each timestep with the nearest ground truth raw ToF measurement. This measurements are not aligned to the scene’s motion, so this is used for initialization and to avoid local minima. This prior was linearly decayed to 0 for the first 100k iterations of training. As this prior is decayed, the learned scene motion allows the network to correct for motion errors properly.

### 5.3.4 Loss Function

Our final loss function for our model is a weighted summation of our original loss functions in equation 5.3 and our other defined losses.

\[
\mathcal{L} + \lambda_{pho}\mathcal{L}_{pho} + \lambda_{tof}\mathcal{L}_{tof} + \lambda_{cyc}\mathcal{L}_{cyc} + \lambda_{geo}\mathcal{L}_{geo} + \lambda_{min}\mathcal{L}_{min} + \lambda_{temp}\mathcal{L}_{temp} + \lambda_{sp}\mathcal{L}_{sp} \quad (5.19)
\]

For the results and experiments shown in the next section, we set the hyperparameters \(\lambda_{pho} = \lambda_{geo} = 0.01, \lambda_{tof} = \lambda_{min} = \lambda_{temp} = 0.0001, \lambda_{cyc} = 0.00001,\) and \(\lambda_{sp} = 0\). In our experiments,
the scene flow spatial smoothness regularization did not always have noticeable effects on the final results. However, when used $\lambda_{sp} = 0.0001$ showed the best results.

The following experiments are trained for 200k iterations with a batch size of 256 camera rays per training iteration. It is possible that results improve with longer training runs, but we were limited by the time of experiments.
Chapter 6

Experiments

6.1 Quantitative Evaluation

<table>
<thead>
<tr>
<th></th>
<th>PSNR</th>
<th>Depth MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>TôRF</td>
<td>45.94</td>
<td>0.64</td>
</tr>
<tr>
<td>Ours</td>
<td>42.88</td>
<td>1.40</td>
</tr>
<tr>
<td>TôRF, Dynamic Region</td>
<td>46.63</td>
<td>0.65</td>
</tr>
<tr>
<td>Ours, Dynamic Region</td>
<td>44.42</td>
<td>1.44</td>
</tr>
</tbody>
</table>

Table 6.1: Quantitative comparisons between our method and TôRF. We compute metrics on the scene’s dynamic objects and also on the entire scene. This is because the scene mostly composed of a static background, and this can influence metrics.

The main quantitative metrics used for evaluation are PSNR for image quality when compared to the ground truth and mean squared error for the depth measurements. The quantitative metrics don’t fully reflect the visual quality in the results, and extensive visual results can be found in the next section. From the quantitative experiments, we the baseline method TôRF appears to have better image quality and depth estimation for the camera poses seen during training time. The qualitative section shows results for novel poses and more information about the reconstructed depth maps, which is the focus of this paper.

The mathematical formula for mean squared error (MSE) is:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2,$$

(6.1)

where $n$ is the number of samples, $y_i$ is the true value of the $i$th sample, and $\hat{y}_i$ is the predicted value of the $i$th sample. The MSE measures the average squared difference between the predicted and true values.

The mathematical formula for Peak signal-to-noise ratio (PSNR) for images is:
\[
PSNR = -10 \log_{10}(MSE).
\]  

(6.2)

This ratio is typically used as an image quality metric between an original image and a reconstructed image. Higher ratios correspond to better image quality.

### 6.2 Qualitative Evaluation

The results for this project are best demonstrated through pictures. The next few pages demonstrate the improved results of this method for one scene, where a cube moves from the back left of the scene to the front right. We demonstrate results on both depth estimation and novel view synthesis.
Figure 6.1: Our method is able to recover plausible scene flow. From top to bottom the visualizations are: 1) the 3D scene flow field projected to the camera and visualized based on the Middlebury flow color code, 2) a normalized map of the projected scene flow magnitudes, and 3) the scene depth along warped camera rays. The first two rows demonstrate that the dynamic cube’s movement is captured in the scene flow, while the background has minimal scene flow. Here, we also visualize the depth along warped rays. The depth along warped rays is important for proper constraints on the temporal ToF consistency loss in equation 5.10.
Baseline (TöRF) Our Results

Figure 6.2: Modeling scene motion and warping raw ToF measurement improves ghosting in fast moving objects. These are training view comparisons of the predicted color images. The baseline results show sharp color images with ghosting around some edges. These artifacts are most present in the bottom right edges of the cube. Our results demonstrate lower visual quality and less high frequency details. Because we are dealing with a more complex optimization problem, our approach may need more iterations to reach similar levels of visual quality.
Baseline (TöRF) Our Results

Figure 6.3: Our method removes depth discontinuities in reconstructed dynamic objects. These are training view comparisons of the predicted depth images. The baseline results show ghosting in the depth maps. We would expect to see a smooth gradient in the depth map of the cube, but depth discontinuities are present visually. Our results show smoother depth maps in the dynamic cube.
Figure 6.4: Modeling scene motion allows our method to recover missing ToF measurements. These are comparisons on the predicted raw ToF measurements at time $t$ versus the ground truth raw ToF measurements from camera capture. Our method is able to produce plausible results for the missing raw ToF measurements because of the ToF warping used as a constraint for the radiance field. $L_0$ is directly supervised by ground truth, but the other raw ToF measurements do not have any direct supervision. ToF warping allows us to model the relationship between raw ToF measurements and scene depth. This allows us to recover 4 raw ToF measurements at any time moment in the field.
Figure 6.5: Training view comparisons of the predicted depth images for novel view synthesis. The original dataset had a static camera, but in these results, we perform novel view synthesis by generating a spiral around the original camera pose. The depth errors due to motion propagate themselves during NVS, and introduce non-existent geometry. Our method compensates for motion errors in ToF depth estimation and improves depth estimation during training and novel poses. Accurate depth reconstruction is important in aiding dynamic NVS.
Chapter 7

Conclusion

Dynamic view synthesis from monocular camera systems is an extremely ill-posed problem, and when using a static camera, we lose information about geometry that is implicitly encoded by varying camera poses. Separately, time-of-flight cameras are efficient and simple systems that have found increasing usage in the real world, despite the reconstruction errors due to motion. Our work builds on TöRF, and proposes an approach to improve on both of these tasks within one framework. Our model extends TöRF to reconstruct scene flow, and using the learned scene flow, we formulate losses to express constraints in the dynamic scene. We demonstrate plausible interpolate results for the missing raw ToF measurements, and show that depth maps that are corrected for motion error improve NVS.

Although we have made improvements in correcting time-of-flight motion errors for the task of dynamic novel view synthesis, there is more progress to be made. Dynamic NVS with a monocular and static camera is extremely ill-posed and extensions to our method can improve both accuracy and stability during training.

Better static and dynamic separation in the scene and modeling disocclusion masks for fractional time warping will improve accuracy. These changes would allow the model to better allocate the capacity of the static and dynamic networks, and improve both color and depth accuracy (especially around edges). Regarding training stability, through experiments we have noticed that the model often diverges or converges to incorrect scene flow. Additional constraints and priors on the raw ToF measurements can improve the training stability. We also plan to explore directly computing some loss functions, such as photometric and ToF loss, on raw points in the radiance field instead of volume integrated rays. Swapping to windowed positional encoding may allow the model to first better optimize low frequency representations of the scene flow.
Bibliography


