3D Physical State Prediction and Visualization using Deep Billboard

Kondo Naruya

Master's Program in Informatics
Degree Programs in Comprehensive Human Sciences
Graduate School of Comprehensive Human Sciences
University of Tsukuba
March 2023

3D Physical State Prediction and Visualization using Deep Billboard

Name: Kondo Naruya

Data-driven reproduction of various physical objects into virtual reality (VR) is a long-standing challenge. Even with the rapid development of deep learning technology in recent years, there are still many problems. (1) For static objects, when bringing reproduced objects to the virtual world, it is difficult to convert them into a format that can be used in VR, such as mesh, without losing the pre-computed highly accurate appearance. (2) For dynamic objects, there are almost no cases that achieved reproducing both realistic appearance and functionality without losing real-time performance. Aiming to solve these problems, we propose Deep Billboard, a technique for removing the constraints of reproducible objects and utilizing them in real-time in virtual reality. The key is that while existing technologies have aimed to reproduce 3D objects correctly in 3D space, Deep Billboard directly aims to reproduce 3D objects in a way that looks correct to the VR user, by representing objects only in a single plane (billboard) that faces the user and is re-rendered frame by frame. Our system, connecting a commercial VR headset with a server running neural rendering, allows real-time high-resolution simulation of detailed rigid objects, hairy objects, actuated dynamic objects, and more, in an interactive VR world, drastically narrowing the existing real-to-simulation (real2sim) gap while preserving smooth interactivity. To the best of our knowledge, we are the first to implement, evaluate, and open-source a functional system of high-quality neural rendering in interactive VR applications, expanding the impact of neural rendering models to interactive simulation such as gaming, shopping and robot learning beyond research demos.

Main Academic Advisor: Yoichi OCHIAI
Secondary Academic Advisor: Tatsuki FUSHIMI

Contents

1 Introduction **1**

2 Related Works **4**
2.1 Data-driven 3D reproduction 4
2.2 Generative 3D-aware Image Synthesis 5
2.3 Billboards 6

3 Preliminaries **7**
3.1 Neural Radiance Fields 7
3.2 Deep State Space Models 7

4 Deep Billboards **10**
4.1 Method 10
4.2 Static Scene 11
4.3 Dynamic Scene 12
4.4 Physical Interaction 12
4.5 System 14

5 NeRF Billboards **15**
5.1 Implementation 15
5.1.1 Real-time rendering 15
5.1.2 Trimming Learned 3D Field 15
5.2 Results 16
5.2.1 Comparison with Meshes and Point Clouds 16
5.2.2 Comparison with Mesh Optimization-based Method using NeRF . . . 17
5.2.3 Quality as a 3D object 17
5.3 Discussion 18
5.4 Summary 18

6 World Billboards **20**
6.1 Implementation 20
6.1.1 Input/Output Definitions 20
6.1.2 Photography Device 21
6.1.3 Preprocess 22
6.1.4 Deep State Space Model 23
6.2 Results 23

6.2.1 Prediction Quality 23
6.2.2 Quality as a 3D object 25
6.3 Discussion 25
6.4 Summary 26

7 Disucussion **31**
7.1 Limitations 32

8 Conclusion **33**

Acknowledgements **34**

References **35**

List of Figures

3.1 Graphical model of DSSM 8
3.2 Examples of images rendered by NeRF 9
3.3 Example of video generated by DSSM 9

4.1 Deep Billboard object on a single canvas. 10
4.2 Qualitative results of our NeRF Billboard on NeRF-Synthetic dataset. 11
4.3 Qualitative Results of World Billboard on our toy dataset. 12
4.4 Combination of NeRF Billboard and 3D mesh. 12
4.5 Deep Billboard provides basic physical interaction. 13
4.6 Deep Billboard VR system overview. 14

5.1 Qualitative comparison between mesh, point cloud and DeepBillboard. . . . 16
5.2 Qualitative comparison between nvdiffrec and DeepBillboard. 17
5.3 Interactive use of Deep Billboard. 18

6.1 Graphical model of our DSSM and definitions of inputs and outputs. 20
6.2 A look of the taking video data. 21
6.3 Chromakey-Free Video Preprocessing 22
6.4 Comparison of the results of video prediction conditioned on viewpoint series. 24
6.5 150 frames predicted by DSSM. (1 of 3) 27
6.6 150 frames predicted by DSSM. (2 of 3) 28
6.7 150 frames predicted by DSSM. (3 of 3) 29
6.8 Interactive use of World Billboard. 30

List of Tables

5.1 Quantitative comparison of NVDiffRec and DeepBillboard. 17

Chapter 1

Introduction

An aspirational goal for virtual reality (VR) is to bring in a rich diversity of real world objects losslessly. Data-driven 3D modeling is increasingly in demand from the virtual reality (VR) industry, which has been growing rapidly in recent years and it is expected to be used in VR applications such as virtual shopping, showrooms, and telecommunications that are grounded in the real world. In addition, Data-driven 3D modeling is a crucial technology for robot learning research. It enables the creation of realistic simulations of objects and environments for training and testing purposes, without the need for expensive and potentially dangerous physical experiments. This will help improve the performance and reliability of robot learning algorithms, leading to the development of more advanced and capable robots. So far, using CG software to model shapes and movements by hand has been the mainstream method. However, when trying to model the face of a specific person or the pasta of a particular restaurant, it is unrealistic to accurately reproduce the movement of facial muscles or the way sauce tangles in pasta. This is because it is impossible to completely model the physical laws at work in these situations. Thus, manual modeling methods were clearly unsuitable especially for modeling people, food, and clothing in their original appearance and function, where humans are likely to notice discomfort. Even if the objects are not as complex as those just mentioned, modeling them by hand is laborious, requiring specialized knowledge and time, so data-driven 3D modeling will play an important role in enriching realistic content and expanding the range of experiences in the virtual world by making it possible for anyone to model them easily.

With recent advances in deep learning research, data-driven modeling techniques are rapidly evolving, and at the demonstration level, it is now becoming possible to reproduce static or time-varying objects from images or videos. In practice, however, automated modeling techniques are not widely used. As for static scene reproduction, NeRF (Neural Radiance Field) [1] can obtain highly accurate color and density 3D fields from dozens of images. NeRF can predict the color and density of points on the line of sight using a neural net when a new viewpoint is entered, and by using this, NeRF can generate a high quality image from the new viewpoint that is almost indistinguishable from the real one. However it is not yet available for use in existing real-time CG engines as it is. Some studies [2, 3] have forced the conversion to mesh, etc., but it is known that the visual quality is clearly deteriorated, and the reality is unfortunately lost, so it is difficult to say that the VR experience has been greatly improved by deep 3D reproduction technologies.

As for dynamic scene reproduction, there are several studies that have achieved this by deforming simple meshes in real time (BANMo [4]). However, high resolution and detailed deformations can not been captured, and the realism of the appearance and deformation are limited by the mesh representation. Although several methods of mesh-independent spatio-temporal reconstruction using NeRF have been proposed in recent years (HyperNeRF [5]), they require recalculation of the color density in each frame, which is very computationally expensive and not real-time at all. The reason why real-time rendering was becoming possible with NeRF for static scene representation was because the they pre-calculate color and density fields of the entire 3D space with grid sampling and caching in a very light format that can be tuned by viewing direction, and objects were stored in files ranging in size from a few hundred MB to several GB. However, in the reproduction of dynamic scenes, the data size of the model becomes ridiculously large when caching a time-varying 3D field, and although the reproduction of dynamic scenes using NeRF can achieve a high-definition appearance, real-time rendering will be quite challenging to achieve even in the future. Also, as with NeRF, since these are not in mesh format, it is not trivial how they would be used in a virtual world.

Looking back on the history of the field of 3DCG and 3D games, some techniques have long been used to represent objects with complex shapes more easily and lightly, and one of these techniques is billboard. Billboard is a technique used for objects such as trees, grass, fire, smoke, etc., which does not necessarily require much change in appearance as the observer's viewpoint changes, and represent pseudo-3D objects by tilting one or a few images so that they are always facing the observer. This technique has long been used from old 3D games such as Super Mario 64 to today. What we can learn from this is that it is not necessary for a 3D object to have an exactly accurate 3D shape for VR users; rather, it is sufficient if the observed 2D representation looks correct. Looking back at recent deep learning-based 3D modeling methods based on this idea, the reasons why these methods are difficult to be put to practical use in VR may be that (1) even though perfectly accurate 3D geometry is not necessary for 3D models, they are aiming for it, (2) despite the fact that it is not suitable for direct optimization of the observation for users, for reducing computational cost while maintaining observability from arbitrary viewpoints, 3D objects have been reduced to a sort of classical representation format such as meshes.

Inspired by the classic billboard, we propose Deep Billboard, which extends billboard with deep learning techniques. Deep Billboard is based on a simple billboard, but dynamically generates the appropriate appearance for an observer based on the observer's position, time changes, and any other information that may affect the observation of the object, and re-renders it on a 2D canvas every frame and create a pseudo-3D representation of the object. This allows Deep Billboard to provide a high-quality appearance directly to the user without being restricted by meshes, etc. when actually used in the virtual world. It also allows objects to be represented at low computational cost, since there is no need to consider accurate color and density fields or deformation of the entire object, but only to generate a plausible image. On the other hand, the Billboard representation can be used only in a limited way in suddenly VR because it throws away the shape information in a naive way. Therefore, we present several systematic ideas, such as a way to realize simple interaction, and a way to realize observation from multiple viewpoints, and show that they can be fully

utilized in casual VR applications. That is, Deep Billboard offers a new alternative to the problem of 3D restoration, in which 3D objects are represented by image or video prediction rather than strictly restoring the 3D structure. This will be a breakthrough in data-driven object reproduction, as it will enable the reproduction of complex objects that have been difficult to achieve both real-time use and high quality in the past and actually used in the virtual world.

In our experiments, we show examples of static and dynamic object reproduction using DeepBillboard. We will also show interaction in VR space and a system for using DeepBillboard with VR devices. Our possible contributions are as follows. (1) We designed the first system to simulate 3D objects with 2D rendering on per-object. (2) We show simulations of static and dynamic objects using DeepBillboard, as well as physical interactions and viewing systems in VR. (3) We have open-sourced the DeepBillboard system and provided several demos.

Chapter 2

Related Works

2.1 Data-driven 3D reproduction

Data-driven reproduction methods can be divided into several categories depending on the final representation format.

Explicit Representations The voxel-based methods [6, 7, 8] and mesh-based methods [9, 10, 11, 12] are common approaches for 3D reconstruction. It is known that the former is difficult to reconstruct at high resolution, while the latter is difficult to estimate complex topology. MVS [13, 14, 15] is one of the best known practical applications of 3D mesh reconstruction from images, and is commonly used in 3D modeling work today, but it cannot capture delicate shapes such as hair, transparent areas, or view-dependent appearance, nor can it model objects with few visual features.

Implicit Representation (Static Scene) [16, 17] leverage differentiable rendering to reconstruct 3D geometry with appearance from image collections. Neural Volumes [18], NeRF [1] and followups [19, 20, 21, 22] learn color and density in 3D space using differentiable ray marching from multiple camera perspectives. NeRFs achieve state of the art accuracy in the novel view synthesis tasks. [23, 24, 25, 26] have made NeRF rendering 1000 times or faster than the original NeRF by caching and optimizing data structures, making real-time rendering possible. However, these methods only show how to utilize them in interactive 2D viewers, and none of the studies have shown how to use them directly in VR. The common way to use these methods in VR is to convert the (high expressive) trained model into a (less expressive) mesh, which is often done both at the research level [3, 27] and in practical use [2]. These converting methods are known to provide fast inference, but are known to deteriorate the appearance, especially for objects with complex shapes such as hairy objects, translucent objects or dynamic objects. In contrast, we aim to make the Neural Rendering method usable as a high quality 3D model like a regular 3D model in VR without loss of accuracy and with interactivity.

Implicit Representation (Dynamic Scene) [28, 5, 29, 30] are extensions of NeRF for spatiotemporal reconstruction of dynamic scenes, allowing viewpoint interpolation from a single video as input, under the same time evolution as the input video. However, there

are some problems, such as the limited range of possible complementary viewpoints and the inability to predict and render in real time. [31] is similar to these, but by learning dynamics itself using deep state space models [32], it can predict and generate new time evolution, independent of the time evolution of the original data. These methods are capable of generating the entire observation angle of view, but how to utilize these methods on an object-by-object basis in VR has not been discussed much. In this study, we present one solution to this problem.

Viewpoint Interpolation Viewpoint interpolation is also a powerful approach to computer graphics that allows three-dimensional objects and scenes to be visualized in a realistic way without full 3D geometric model reconstruction. Multi plane images [33, 34] and light field interpolation [35, 36, 37, 38] are such image-based approaches for novel view synthesis without 3D reconstruction. When the postures of novel viewpoints can be limited, such as forward-facing scenes, these representations are very light-weight, accurate and effective, but otherwise, their use case is limited. [39, 40, 41, 42, 43] are stereo matching based rendering methods that enable 3D reconstruction from fewer images but they suffer from view inconsistency in the presence of surface occlusions when using local features in input views to identify surfaces for novel view synthesis. [44, 45, 46, 47, 48] estimate surface using volume rendering and provide very efficient 3D reconstruction but it is extremely difficult to correctly estimate the shapes with transparent or fuzzy parts. In contrast, our goal is to bring objects that are so complex that it is difficult to estimate their surface and appropriate to use powerful methods such as NeRF, to VR to be used as 3D models.

2.2 Generative 3D-aware Image Synthesis

Recently, 3D aware image generation based on Generative Adversarial Networks [49] has become feasible. Voxel-based GAN [50, 51, 52, 53, 54] generate voxels directly using 3DCNNs, but their inefficient representation of 3D space makes it difficult to generate high resolution due to memory constraints during training. Some mesh-based GAN [55, 56] have been proposed, but they are difficult to generate with high fidelity due to lack of expressiveness. As an alternative, implicit representation networks have been proposed for 3D scene generation [57, 58, 59], however, the 3D consistency was not enough and the generated images were not as good as the 2D GAN. Several studies [60, 61, 62] have solved the previous problems by combining the NeRF mechanism with 3D GAN. Among them, EG3D [61] enables real-time rendering by streamlining the coordinate input-based 3D representation, and also enables its 3D reconstruction by optimizing the GAN seeds when given a single target image. 3D-aware GANs are very powerful techniques, but these methods only work well on people, cat faces, cars, etc., where huge data sets are available, and are not applicable to new objects. Also, as with NeRF and others, the way to utilize them as 3D models has not been explored much.

2.3 Billboards

Billboard's technique of using one or a few images to represent a 3D model has been used from the 3D video games of the 1990s, such as Super Mario 64, to today. There are several studies based on billboards. Some researches [63, 64, 65, 66, 67] use billboard for fire, smoke, grass, trees or cartoon face modeling. However, all of these methods are limited in their applicability. Several recent methods combined with machine learning have proposed new ways to use billboards. Articulated Billboards [68] generates free-viewpoint images of a soccer stadium, and does so by representing each of the players' body parts on billboards automatically generated from camera images. FreeStyleGAN [69] presents a demonstration of a post-editing of the face of a mesh-reconstructed person using 2D GAN and billboards. Rig-space Neural Rendering [70] is a research aimed at replacing ultra-high resolution 3D models with 2DGAN and billboard representations to achieve both high quality and fast rendering, but it assumes that a precise 3D model can be obtained in advance. Generally, billboards are supposed to be powerful in that they can represent 3D models plausibly, sometimes in a data-driven manner, while avoiding the difficulty of accurately reconstructing 3D structures, but in reality billboards are not widely used at all. Unlike these previous studies of billboards, in this study, we show that billboards can be used to model a wide range of real-world objects with little human effort, without limiting the scene to be modeled, and proposes a method that can be used interactively like ordinary 3D models. With this, we aim to make it easy to bring a wide range of real-world objects into VR and use them.

Chapter 3

Preliminaries

3.1 Neural Radiance Fields

Neural radiance fields (NeRF) [1] are 3D representations that can be rendered from arbitrary viewpoints, capturing continuous geometry and view-dependent appearances. The expected color of a camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ can be estimated by a finite-sample approximation $\hat{C}(\mathbf{r})$:

$$\hat{C}(\mathbf{r}) = \sum_{i=0}^{N-1} w_i \, \mathbf{c}(\mathbf{r}(t_i)) \,, \tag{3.1}$$

$$\text{where} \quad w_i = T_i(1 - \exp(-\sigma(\mathbf{r}(t_i))\delta_i)) \tag{3.2}$$

$$T_i = \exp\left(- \sum_{j=0}^{i-1} \sigma(\mathbf{r}(t_j))\delta_j \right) , \tag{3.3}$$

where t_i indicates the position of the point to be sampled from the ray, and $\delta_i = t_{t+1} - t_i$ are the distances between point samples. In NeRF, an MLP parameterizes $\mathbf{c}(\cdot)$ and $\sigma(\cdot)$ jointly and is trained to minimize the loss between the ground-truth pixel colors $C(\mathbf{r})$ in the images and the predicted colors $\hat{C}$ for a batch of rays $\mathcal{R}$:

$$\mathcal{L}_{\text{RGB}} = \sum_{\mathbf{r} \in \mathcal{R}} \left\| C(\mathbf{r}) - \hat{C}(\mathbf{r}) \right\|_2^2 \tag{3.4}$$

3.2 Deep State Space Models

Deep State Space Model (DSSM, also called Deep Karman Filters, Deep Markov Model, or simply SSM)[32, 71] is a deep learning-based prediction model for time series. DSSM assumes that the data observed at each time have latent states s_t and observations at each time o_t are generated from that states. Figure 3.1 shows the graphical model of DSSM is shown. The DSSM predicts an observation $o_{1:T}$ conditioned on an initial state s_0 and an action sequence $a_{1:T}$, as shown in equation (3.5).

$$\begin{aligned} p(o_{1:T}|a_{1:T}, s_0) &= \prod_{t=1}^{T} p(o_t|a_{1:t}, s_0) \\ &= \prod_{t=1}^{T} \int p(o_t|s_t) p(s_t|s_{t-1}, a_t) ds_t \end{aligned} \tag{3.5}$$

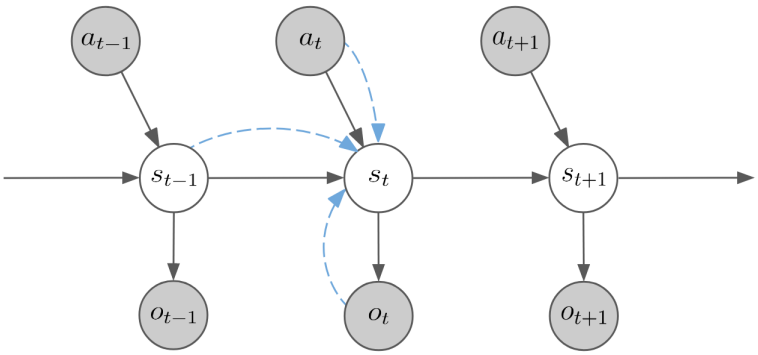


Figure 3.1: Graphical model of DSSM. The solid line represents the generator distribution, and the dotted line represents the inference distribution. Inference distributions are used only during training.

The DSSM makes the assumption that the state vector s_t at each time point follows a normal distribution. Plus, DSSM introduces an inference model $q(s_t|s_{t-1}, a_t, o_t)$ as in VAE [72], models the generative process and the inference process by a neural network, and finds their parameters by maximum likelihood estimation with the following variational lower bounds.

$$\begin{aligned}\log p(o_{1:T}|a_{1:T}) \quad \geq \quad & \sum_{t=1}^{T} \Big(\mathbb{E}_{s_t \sim q(s_t|s_{t-1},a_t,o_t)}[\log p(o_t|s_t)] \\ & - \mathbb{E}_{s_{t-1} \sim q(s_{t-1}|s_{t-2},a_{t-1},o_{t-1})}[\mathrm{D}_{\mathrm{KL}}(q(s_t|s_{t-1}, a_t, o_t) \| p(s_t|s_{t-1}, a_t))] \Big)\end{aligned} \tag{3.6}$$

DSSM can be used to predict future observations as shown in Figure 3.3.

Figure 3.2: Examples of images generated by NeRF, from [1]. These are results of object reconstruction of the NeRF Synthetic Dataset.

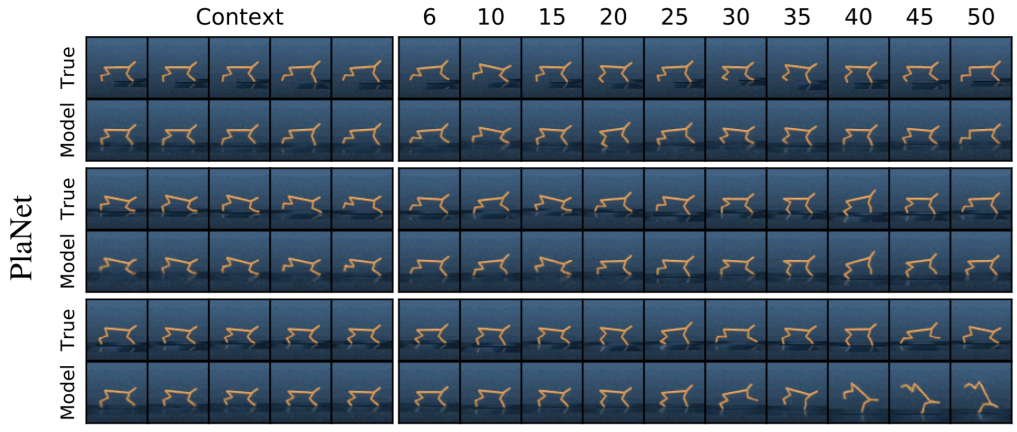


Figure 3.3: Example of video generated by DSSM, from [73]. This model learns video frames from a side view of a reinforcement learning agent.

Chapter 4

Deep Billboards

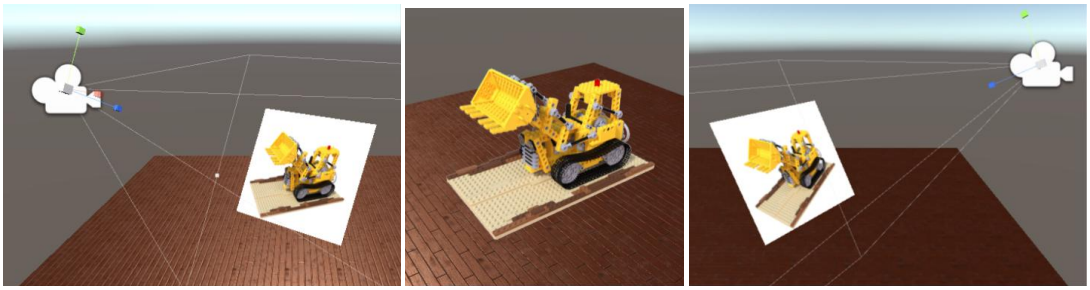


Figure 4.1: Deep Billboard object on a single canvas. Since billboard object is rendered for a single perspective, it feels flat when viewed from other angles.

4.1 Method

We propose Deep Billboard, a generalization of the classic billboard. In contrast to classic billboard, Deep Billboards textures are redrawn each frame with images generated by Neural Rendering model, conditioned by the viewer's perspective (Figure. 4.1). Compared to standard explicit 3D models such as meshes, Deep Billboards enable object rendering of much higher resolutions, drastically improving the realism of virtual experiences while preserving real-time interactivity. Any Deep image or video generative model can be combined with our Deep Billboards.

The main algorithm of Deep Billboard is described in the Algorithm.1. Deep Billboard's job is, in addition to that of a classic billboard, to update its own texture every time step by generating a new observation image according to the viewer's point of view with neural rendering model. The model is required to produce either an alpha-mapped RGBA image or an image with the non-object areas filled with a single color. The figures in the paper show images filled with a single color for clarity, but in practice the background of the image is transparent and used as a texture. Here, we described $\mathbf{d}$ and $\mathbf{x}$ as inputs, and additional inputs such as information of environmental light or user actions, for example, can be added to the input depending on the generation model. Conversely, if the image generation model does not take into account the distance to the observer, $\mathbf{x}$ may not be used as an input. Deep Billboards can utilize a wide range of objects in VR without the difficulty of explicit

Algorithm 1 Deepbillboard for viewpoints of viewer A

Require: novel view synthesis model F, billboard $\mathcal{B}$
while the billboard is seen by A **do**
  Tilt $\mathcal{B}$ to face A's viewpoint position
  Get input of F
  Generate Observation $\hat{O} = F(\mathbf{d}, \mathbf{x})$
    where $\mathbf{d}, \mathbf{x}$ = rotation of $\mathcal{B}$, viewpoint position of A
  Update the billboard texture of $\mathcal{B}$ with $\hat{O}$
end while

3D reconstruction. As examples, we show their use for static objects (NeRF Billboard) and self-moving dynamic objects (World Billboard). The scope of applicable objects will be discussed later. Figure 4.2 shows an example of the visual result.

4.2 Static Scene

Figure 4.2: Qualitative results of our NeRF Billboard on NeRF-Synthetic dataset.

As an example of Deep Billboard's use for static objects, we show its combination with NeRF. NeRF uses a neural network to implicitly parameterize emitted color and volume density functions with spatial coordinates as inputs, which is passed to a ray-tracing integration for view-conditioned image generation. While NeRF allows a much higher-resolution 3D construction that is comparable to the real objects, prior works importing NeRF into VR converting the models back to coarse meshes, due to software incompatibility, high computational demand, and slow rendering of original NeRF models. In contrast to these works, we directly utilize neural rendering without explicitly converting to a 3D model, thus creating a 3D model that looks exactly as it was learned in NeRF. For real-time neural rendering, we instead used PlenOctrees [24], an extension of NeRF to update the billboards. When NeRF and PlenOctrees are trained on images taken casually, there is a problem that the background remains in the generated image. To solve this problem, the density outside the rough bounding volume of the object was overwritten to 0 for the PlenOctrees format data. Note that, to the best of our knowledge, our system is the first to import NeRF models directly into an interactive VR application without loss of appearance.

4.3 Dynamic Scene

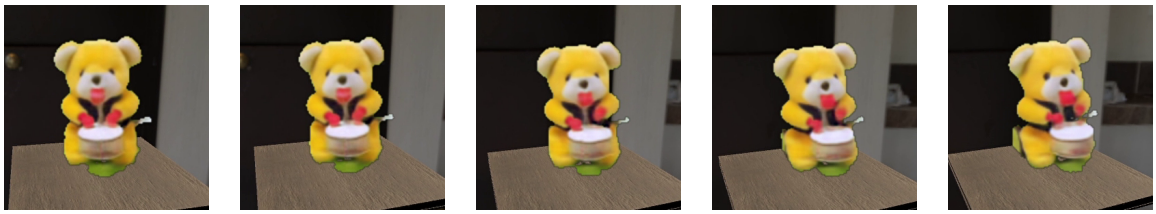


Figure 4.3: Qualitative Results of World Billboard on our toy dataset.

While NeRF allows rich rendering of complex static objects, extension to dynamic scenes is still an active area of research [1]. To show the versatility of our Deep Billboard system, we also replace our billboard updater using PlaNet [73], a neural state-space model (SSM) or a *world* model for action-conditioned video prediction in deep reinforcement learning (RL), to achieve data-driven reproduction of time-varying objects from a single video only. From a single 10-minute video labeled with the camera position and orientation, we learned video prediction conditioned on the initial viewpoint image and viewpoint series, and reproduced the time-varying 3D object through the billboard in our interactive VR system. World Billboard is also used in the same way as Algorithm.1, but unlike NeRF Billboard, the object's state s at the previous time is stored and used to generate the image at the next time, allowing for the generation of object motion. The training data for PlaNet was preprocessed from real-world video and transformed into a set of normalized 10-frame series of images. This preprocessing included frame scaling to normalize the distance between the camera and the subject, cropping the object area, and filling in the background.

4.4 Physical Interaction

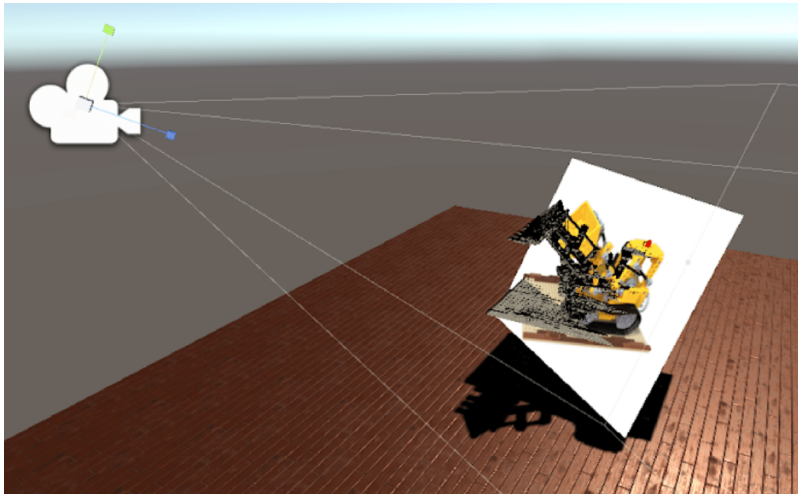
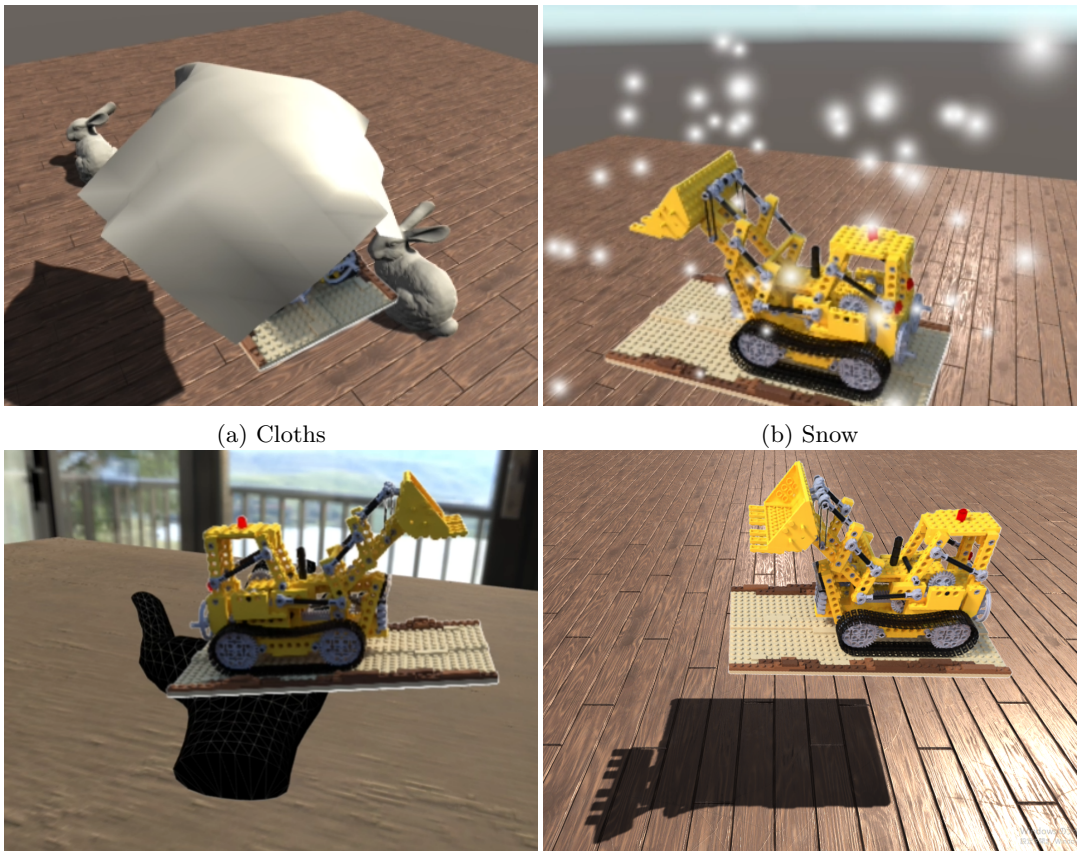


Figure 4.4: Combination of NeRF Billboard and 3D mesh.

Another disadvantage of a classic billboard is that it does not allow physical interaction. To solve this problem, we propose a system where a billboard is used for visual interaction



(a) Cloths (b) Snow

(c) Lift up (d) Shadow

Figure 4.5: Deep Billboard provides basic physical interaction.

while an invisible rough mesh for physics interaction (Figure 4.4). Figure 4.5 shows the four basic physical interactions realized with this approach. (See the supplemental videos.) In this example, we use invisible meshes acquired by following prior works [3, 27] that converts a learned neural rendering model to a mesh by marching cube algorithm. Even if a mesh is not prepared using such kind of methods, physical interactions can roughly be achieved using alternative meshes, such as a spherical mesh, for example. In Figure 4.5a, the cloth is dropped from above and placed over the object; it can be seen that the cloth deforms with the shape of the object, regardless of the shape of the Billboard 2D image itself. In Figure 4.5b the object is being snowed on from above, and the snow that hits it appears to bounce back. In Figure 4.5c shows an object being lifted by touching it, indicating that this method can also handle object movement. In Figure 4.5d, we can see that the suitable shadow appears when illuminated by a light source. From these experiments, it can be said that the combination of visible billboard and invisible mesh is able to simultaneously realize good visuals and good basic physics. The range of feasible physical interactions will be discussed later.

4.5 System

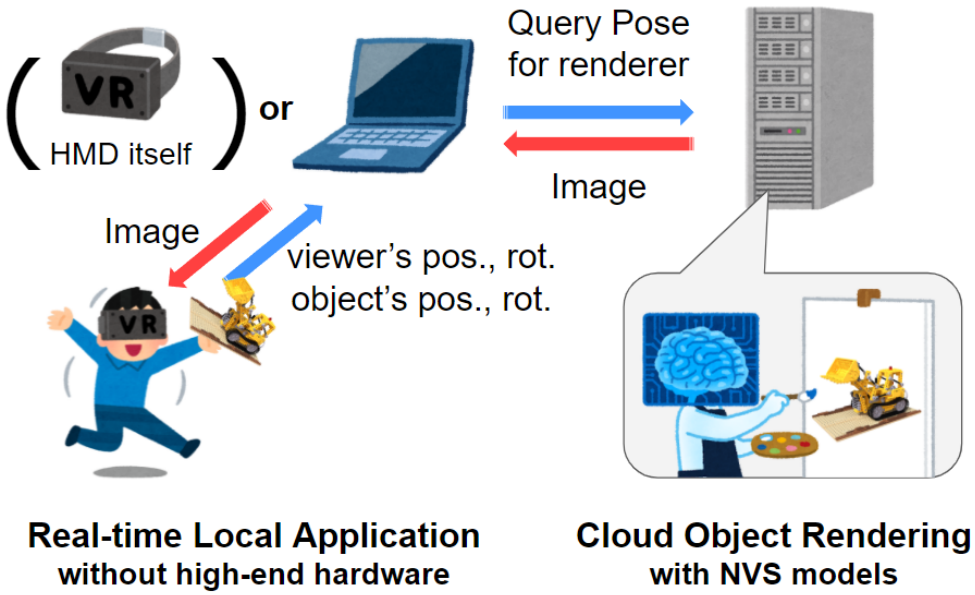


Figure 4.6: Deep Billboard VR system overview.

Neural rendering models such as NeRF or PlaNet require large computational power, so we have built a system that uses cloud rendering to enable even inexpensive VR headsets to handle highly accurate pseudo-3D models as long as they have an internet connection (Figure. 4.6). Each Deep Billboard object is rendered on a separate cloud server process, and rendered 2D frame is sent to the VR system per frame to achieve real-time interaction with minimal on-board processing. Our system operated NeRF Billboard (800x800 rendering resolution) at roughly 10 $\sim$ 15 fps and World Billboard (256x256 rendering resolution) at roughly 20 fps with a single TITAN X GPU (cloud server).

Chapter 5

NeRF Billboards

5.1 Implementation

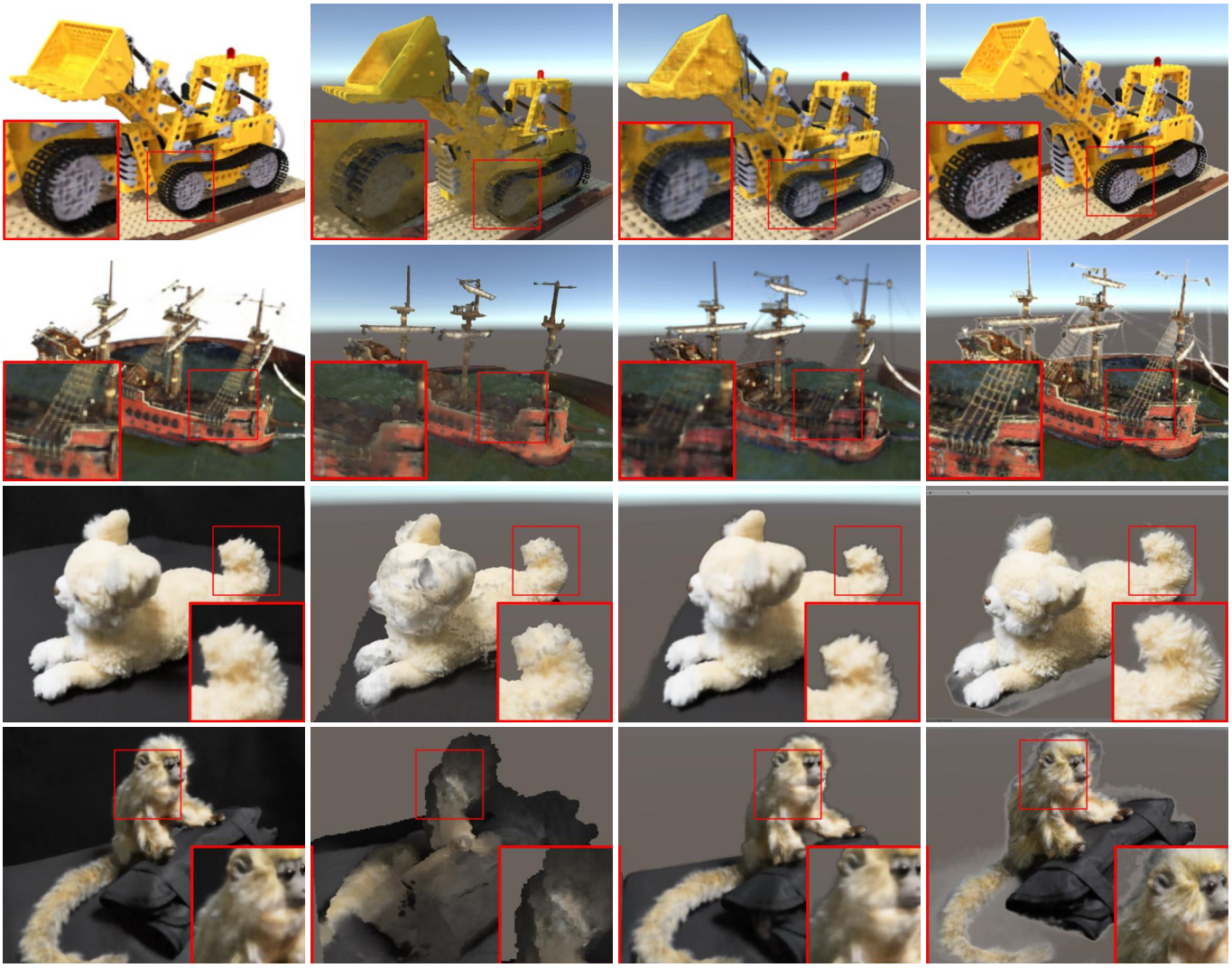
NeRF Billboard, as mentioned above, combines NeRF and DeepBillboard and uses NeRF's novel view synthesis to display static objects as if they were 3D objects in NeRF quality, without being bound by meshes or other representation formats. However, there are some problems in combining the two as they are, so we made the following modifications.

5.1.1 Real-time rendering

Since the original NeRF takes about 30 seconds to synthesize one Novel view, we use its successor, PlenOctrees, instead of NeRF for training and rendering. PlenOctrees learns the field by dividing the field into Octrees, representing each microregion as spherical harmonics, and optimizing these spherical harmonics with the gradient method. By not passing through a neural net, the rendering is 3000 times faster, while producing rendering accuracy equal to or better than the original NeRF. We trained PlenOctrees with 3 degrees of freedom for spherical harmonics and 8 octree depths (up to 512 grid divisions).

5.1.2 Trimming Learned 3D Field

In order to appear like a 3D model in the virtual world, it must be possible to give the billboards an appropriate transparency map. Therefore, if the image used to train NeRF or PlenOctrees contains a background, the generated image must be rendered with the background removed. Since the PlenOctrees representation can determine whether each octree cell is background or not by its position in space, we solved this problem by removing the background octree cells by setting a threshold on the spatial coordinates in advance. Also, the original implementation of PlenOctrees outputs only RGB, but we have changed it to output an RGBA image based on the density field obtained by training PlenOctrees and calculating transmittance in the same way as NeRF.



(a) before extracted (b) Mesh (c) Point Cloud (d) Deep Billboard

Figure 5.1: Qualitative comparison between mesh, point cloud and DeepBillboard.

5.2 Results

5.2.1 Comparison with Meshes and Point Clouds

To evaluate the performance of NeRF Billboard, we compared the object representation by Deep Billboard with that of NeRF converted to a simple mesh or point cloud in actual CG software. The point cloud was acquired by uniformly grid sampling color and density from the NeRF and adopting where the density was above a certain level. The mesh was obtained by marching cube of the point cloud.

First of all, it can be seen that the conversion to mesh or point cloud clearly deteriorates the visual accuracy. For example, the ropes on the ship in the mesh or point cloud have disappeared, the stuffed animal has turned black, and the fine irregularities have become smooth in appearance. While the Deep Billboard clearly shows every detail, the plush toy made with Deep Billboard has a noisy haze on the surface of the object, which reduces the

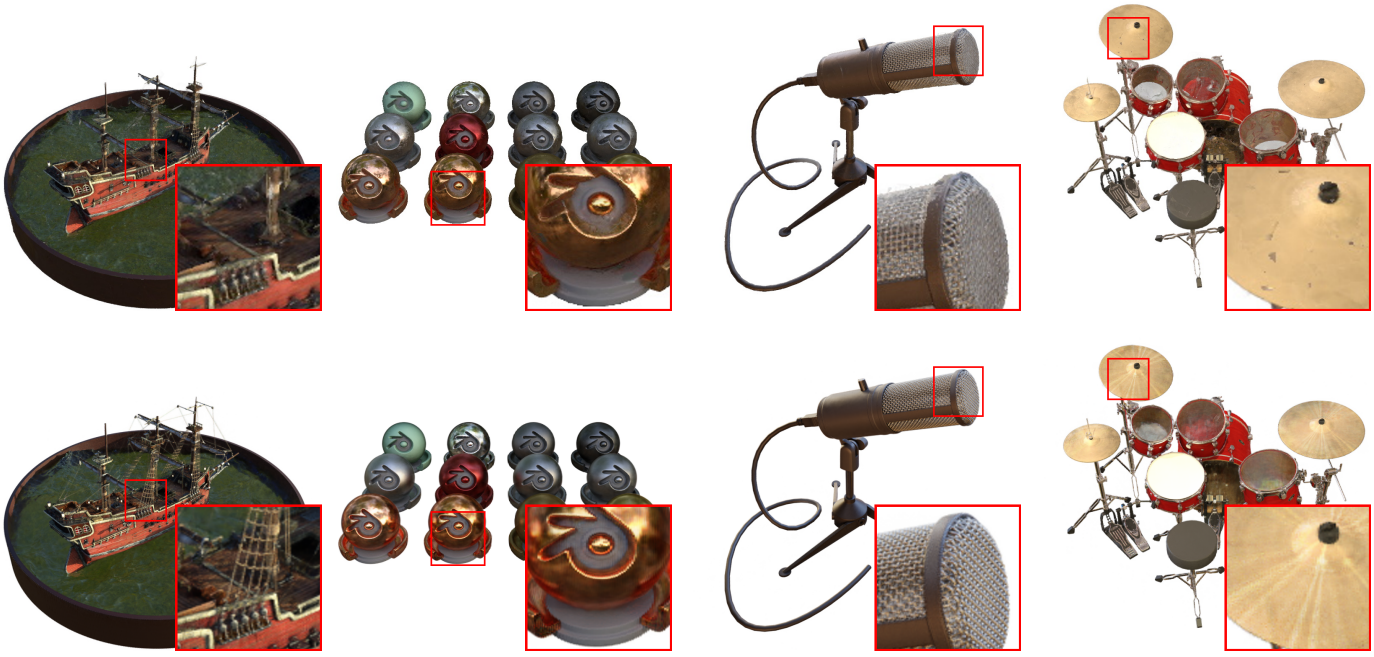


Figure 5.2: Qualitative comparison between nvdiffrec (results from [3]) and DeepBillboard (results from [24]).

visual quality.

5.2.2 Comparison with Mesh Optimization-based Method using NeRF

Table 5.1: Quantitative comparison of NVDiffRec and DeepBillboard. Average results for the eight scenes in the NeRF realistic synthetic dataset. Each scene consists of 100 training images, and 200 test images, with masks and known camera poses. Results from NeRF diffrec are taken from diffrec paper [1].

Method	PSNR↑	SSIM↑	LPIPS↓
NVDiffRec [3]	29.05	0.939	0.081
DeepBillboard (PlenOctree) [24]	**31.71**	**0.958**	**0.053**

Next, we compare DeepBillboard to NVDiffRec, a very recently published method for high quality mesh extraction from NeRF. Looking at Figure 5.2, the visual quality seems clearly better in DeepBillboard than in NVDiffRec. For example, the rope of the ship and the weave of the mic are not well reproduced in NVDiffRec. Also, looking at the drums, we can see that there are areas where the gloss is not well acquired in NVDiffRec. Table 5.1 is a quantitative comparison. From this table, we can see that DeepBillboard provides better visual observations than NVDiffRec, even quantitatively.

5.2.3 Quality as a 3D object

Finally, Figure 5.3 shows how NeRF Billboard was actually employed in the CG software. The model generated with a clean dataset (NeRF Synthetic Dataset) can be displayed

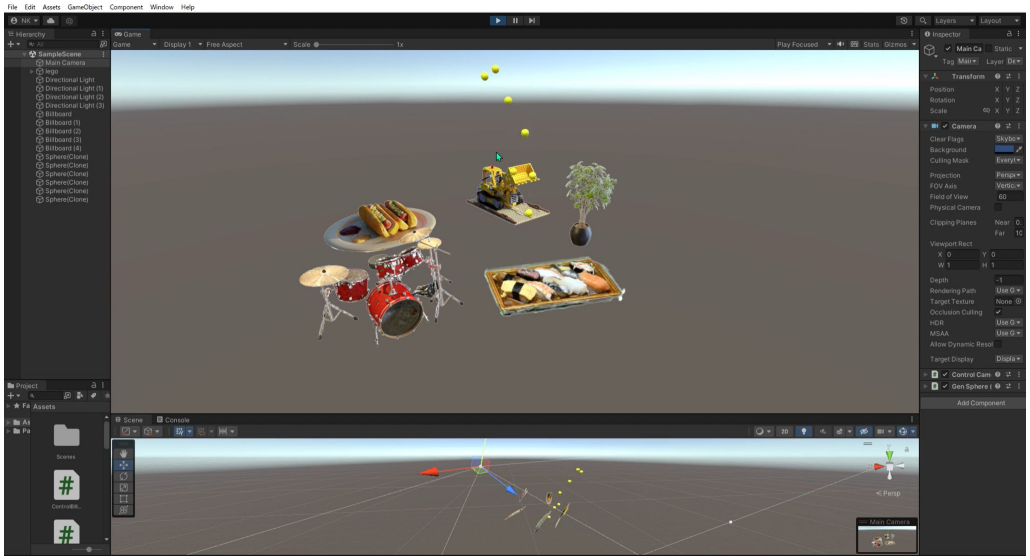


Figure 5.3: Interactive use of Deep Billboard. `https://www.youtube.com/watch?v=_-hvK3IUZn8`

at about 30 fps with sufficient resolution without any discomfort, but compared to other regular 3D objects, it felt a bit delayed. Also, the way the light hit the objects appeared to vary from object to object was a bit disconcerting. The sushi model generated with images taken by oneself seemed a bit flat, but this could be due to the fact that the training data had not been undistorted.

5.3 Discussion

Limitation of Mesh and Point Cloud The quality of meshes and point clouds tended to be low overall. This is because (1) mesh and point cloud cannot reproduce viewpoint-dependent changes in appearance, and thus cannot generate optimized observations for each viewpoin. (2) NeRF cannot accurately predict the density in details, so it tends to produce shape errors that expand the size of the object and the color of the empty space is assigned, resulting in an overall blurred shape and a grayish color.

Haze near the boundary of an object The reason for the haze is probably that NeRF does not acquire a field near the boundary of the object with a complete separation between the object and the background when training. For example, if the background is always black, it is no problem for NeRF to learn that the area near the object's surface is also black on certain viewpoints. To solve this problem, we can simply perform high fidelity background removal before training, or prepare several patterns of backgrounds in training data, for example.

5.4 Summary

NeRF Billboard can create pseudo-3D objects in virtual worlds using NeRF's real-time, high-quality novel view synthesis as it is. The NeRF Billboard was shown to be clearly better

than meshes and point clouds both qualitatively and quantitatively in terms of appearance. On the other hand, there are some problems, such as the fact that the accuracy of the model depends greatly on how the background is processed and the training data is captured, there is some delay, and the lighting is fixed differently from CG space, but these problems will be solved gradually in the future.

Chapter 6

World Billboards

6.1 Implementation

As mentioned earlier, World Billboard combines DeepBillboard and video prediction for data-driven modeling of dynamic objects, i.e., objects that change appearance and shape by themselves or by being manipulated by someone else. For video prediction, we use DSSM (Deep State Space Models), a deep learning-based basic method. This section describes how to use DSSM in conjunction with DeepBillboard and how it can be implemented. In order to simplify the problem set up, the experiment was conducted using only toy dolls that move on their own.

6.1.1 Input/Output Definitions

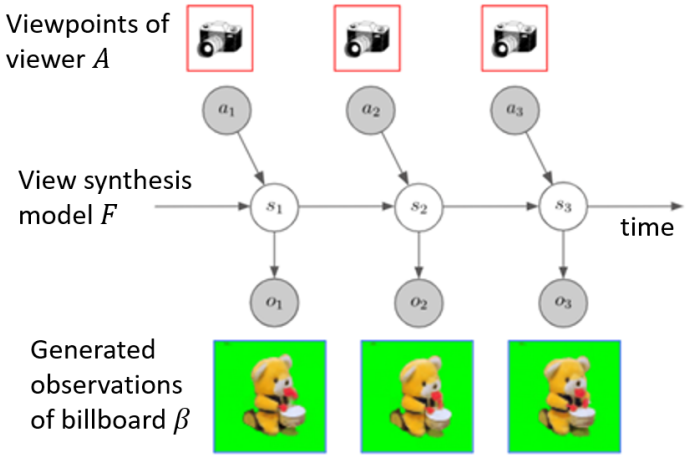


Figure 6.1: Graphical model of our DSSM and definitions of inputs and outputs.

Since DeepBillboard is an object-by-object pseudo-3D reproduction method, DSSM desires to be able to predict the video that will be pasted directly onto the billboard. Therefore, the DSSM input is the viewpoint, and the output is the video such that the target object is in the center of the frame.

Output In order to learn video predictions correctly and efficiently, the outputs are defined in detail as follows.

1. The object is at the center of the frame
2. The rendered size of the object is fixed
3. The object keeps horizontal orientation in the frame
4. Background is removed

As the distance between the viewpoint and the subject increases, the object should appear smaller, but since the billboard itself appears smaller as the distance increases, the size of the object should not be changed. Similarly, the object should appear to tilt as the viewpoint tilts, but the tilt of the object should not change with the tilt of the viewpoint because the billboard itself appears to tilt. Finally, in order for the backside of the object to be properly reflected in the virtual world, the areas other than the object must be able to be processed transparently.

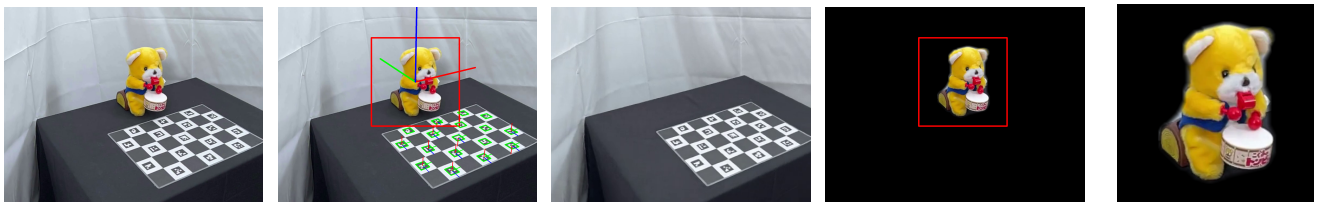
Input Based on the output definition, the only input required is the relative position of the viewpoint from the object center. (Strictly speaking, the horizontal direction cannot be defined directly above the object, so we assume that the object is not observed from the top.) Relative position can be decomposed into relative angle and relative distance, but since the change in appearance with distance is relatively small, we use only relative angle as input for DSSM in this case.

6.1.2 Photography Device

In this and the following sections, we will explain how to capture the object and how to preprocess the captured video in order to obtain the input and output data described above.

Figure 6.2: A look of the taking video data. We use a smartphone and an external physical gimbal.

As shown in Figure 6.2, we use iPhone13 Pro, with external physical stabilizer and Record3D (an iPhone app). A ChArUco board is also used to estimate the camera's position, orientation and target object position. Initially, we used an action camera, but we



(a) input frame (b) pose estimation (c) background prediction (d) background removal (e) output frame

Figure 6.3: Chromakey-Free Video Preprocessing

found that the camera's internal parameters were constantly changing due to digital camera stabilization, which is one of the strengths of action cameras, making it impossible to accurately estimate the camera's position and posture. Therefore, we selected these devices in order to provide image stabilization without such digital image stabilization. (Theoretically, it should be possible to learn from a dataset created from shaky video without image stabilization, but we believed that a simple video prediction model such as DSSM would also learn the behavior of shaking, so we placed importance on physically stabilizing the camera.) Record3D is an iPhone application that does not dare to apply digital image stabilization. This app can also capture depth, but we did not use it. We shot about 15 minutes of video at $1920 * 1440$ @10fps, about 12 minutes for training and the rest for evaluation. Objects were captured by randomly moving the camera from about $120\,^\circ$ left and right and $75\,^\circ$ up and down in front of the object.

6.1.3 Preprocess

Video data preprocessing consists of roughly three processes. We use OpenCV for the image processing described below.

Pose Estimation First, from the ChArUco board in the video, we estimate the camera's pose and the object's position. Figure 6.3b is a visualization of the results. First, the ChArUco board is used to determine the relative pose of the board with the camera as the origin. Next, the camera's pose is obtained with the board as the origin. Finally, since the relative positions of the board and the target object is fixed, the position of the target object in the image can be determined by projecting the three-dimensional position of the target object with the board as the origin onto the frame.

Background Removal Although it is easy to remove the background by using a green back screen or something, it is troublesome to prepare such an environment. We therefore devised a method to remove the background by generating a background image using NeRF (Figure 6.3c, Figure 6.3d). There are many off-the-shelf tools that can automatically remove the background, but they are inconvenient as they are not accurate enough when the object is captured small in the frame or when captured with complex backgrounds, and these tools do not support the background when the camera moves and the background changes. We trained NeRF to generate the background from 15 seconds of video taken in the room beforehand. This NeRF-based background processing is versatile, but it can still introduce

noise if the background is complex, so this time we shot the target with a monotonous background.

Target Cropping Finally, crop the image with the background removed. The center of the cropping is the center of the object in the image obtained earlier, and the size of the cropping is scaled inversely proportional to the distance between the object and the camera. The cropped image is resized to 64*64 and used for training.

6.1.4 Deep State Space Model

We use the DSSM for video forecasting, or more precisely, we use the RSSM (Recurrent State Space Model), which is an extension of the DSSM, following the implementation of [73]. This model adds not only a vector of stochastic transitions to the latent variables in the DSSM, but also a vector of deterministic transitions. This allows for better inference of macroscopic information such as the angle of the object being rendered. The input to the model is a 3D view with positional encoding. When training, the first frame is also given as input and the next 10 frames are predicted. The dimensions of both the stochastic and deterministic latent variables are 64.

6.2 Results

6.2.1 Prediction Quality

We evaluate the correctness of the video prediction, i.e., whether the appearance changes plausibly when the viewpoint changes, and whether the object deforms plausibly. We compare the predictions for the 10 frames in the test data (Figure 6.4). In this case, since only the viewpoints are input to the model, we prepared a video created by collecting frames corresponding to the viewpoints of the training data that are closest to each viewpoint of the test data as a comparative target of plausibility of the video prediction. (Named “near” in the figures) To check for changes between frames, we also visualize the areas where there have been time changes in HSV color space. (Named “dx” in the figures.) Note that the movement of the target object is not known from the first frame and the change of viewpoint, and is probabilistic, so the prediction does not have to be the same as the true frames displayed with “true” with respect to the object’s movement.

In Figure 6.4, comparing “true” and “pred”, we can see that the object’s orientation is correctly estimated and generated. Also, from ‘pred dx‘, there is no abrupt change in the object’s posture, and the change is as smooth as that of “true dx”. Next, comparing “near” and “pred” and their time diffs, we can see that “near” also seems to have the correct posture, but the object suddenly deforms like a warp, which means that it cannot generate a continuous moving images, whereas “true” shows smooth changes in head and hand movements. From these results, it can be said that the trained video prediction model is able to generate smooth motion in the correct posture and meets the requirements of DeepBillboard in these respects.

Next, Figure 6.5 shows the output of the long-time (150 step, 15 seconds) prediction. It is known that DSSM tends to fail in the prediction of longer steps if it is not trained well,

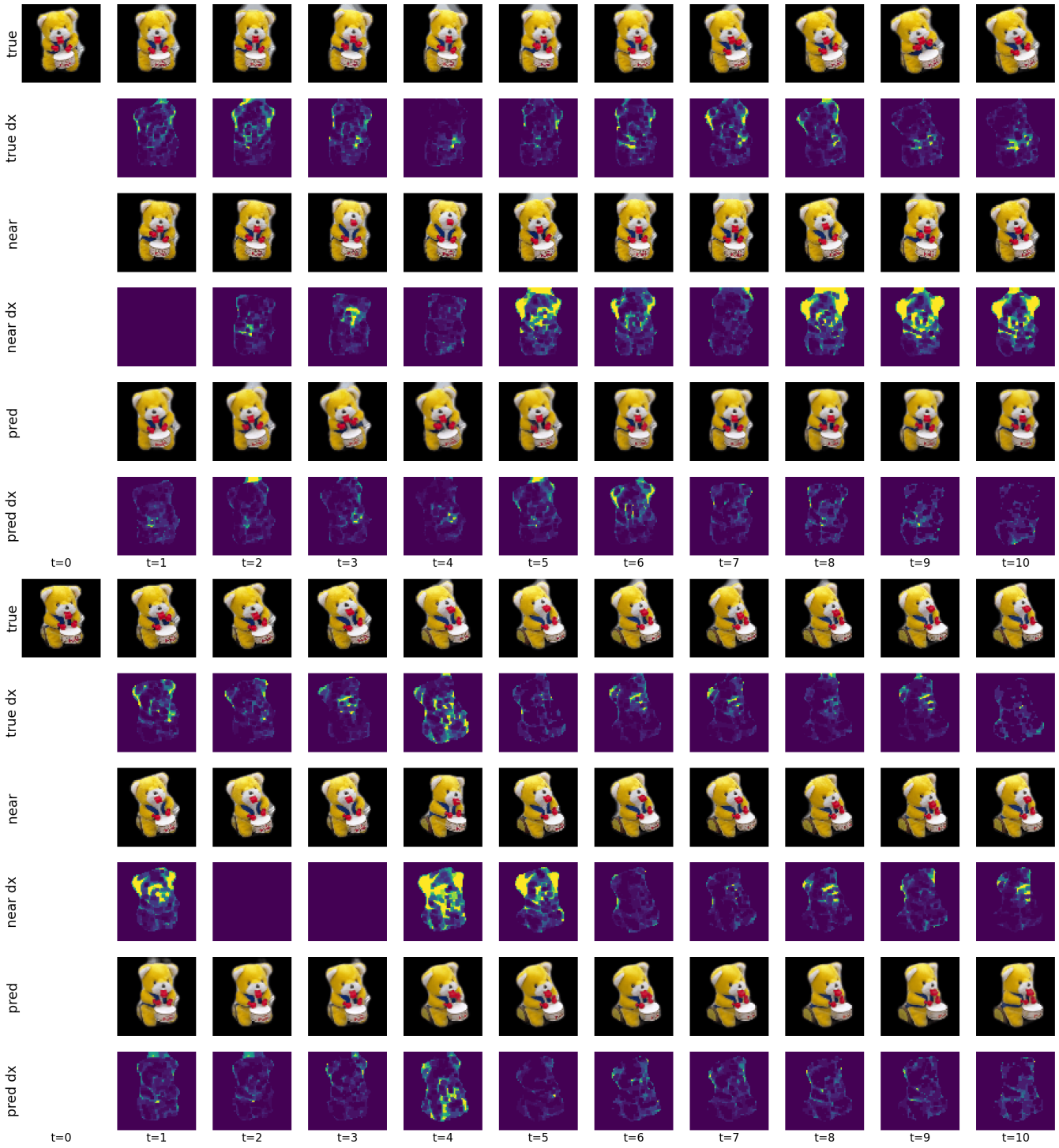


Figure 6.4: Comparison of the results of video prediction conditioned on viewpoint series. “true” is ground truth, “near” is the frame of training data closest to the viewpoint, and “pred” is prediction by DSSM. “dx” visualizes the areas in those videos that have changed over time.

but the figure shows that our DSSM is able to predict the correct direction of the generated image without violently failing in the long-term prediction. However, it cannot be said that the DSSM faithfully reproduces the motion of the object, for example, around $t = 25 \sim 35$, the object is frozen with no motion at all, and the overall number of hand movements is apparently less than the correct image.

6.2.2 Quality as a 3D object

Finally, Figure 6.8 shows how World Billboard was actually employed in the CG software. First of all, it seems that this model has an angle at which the aforementioned freezing phenomenon tends to occur. Also, when the viewpoint movement was stopped, the model tended to repeat the same movement. In addition, the resolution is too rough to be considered realistic, and the model does not blend well with the CG space, making it look as if it is floating. On the other hand, when viewed from a distance, it can be said to look like a 3D model that moves as it should.

6.3 Discussion

Unnatural Freezes or Excessive Repetition This is likely caused by the fact that DSSM is not able to separately learn changes in the observed images due to changes in viewpoint and changes in the observed images due to changes in time, so it is influenced by the viewpoint information and reproduces movements that are not originally related to that viewpoint but are more likely to be observed at that viewpoint. Another likely cause is that when the system is given an unknown viewpoint series, such as when the viewpoint change stops, it is unable to make use of what it has learned, and its predictions are about to fail. To solve this, it may be necessary to improve the system so that it can learn latent variables and their transitions in a way that removes viewpoint information, or so that it can generate video while predicting explicit rough 3D structures.

Decrease in frequency of local movements The posture of the object itself was reproduced correctly, but the frequency of local movements, such as the movement of the bear's head and hands, for example, tended to appear much less frequently. This is probably because DSSM has learned that when predicting the future, it is more plausible if the object does not move much, i.e., it is easier to keep the generation loss small. To solve this problem, one approach would be to accelerate the learning of motion by using a model that evaluates the likelihood of the generated video and the generated series of latent variables, or another approach would be to reduce the likelihood of not moving too much by conditioning the video prediction on past video with a longer number of steps to begin with.

Visual Quality Simply increasing the resolution and training the DSSM may seem to solve the problem, but the DSSM tends to produce blurred images because it learns with reconstruction errors. Instead, it is possible to train a model that forcibly converts the low-resolution DSSM results into a high-resolution model, but this is not easy because it is difficult to obtain the plausibility of temporal changes in local visual features.

6.4 Summary

The object orientation itself can be correctly predicted and generated, and the motion can be generated smoothly. On the other hand, the frequency of small movements tends to be low, the movements are sometimes incorrect depending on the viewpoint, and the visual reality is still insufficient, leaving room for improvement.

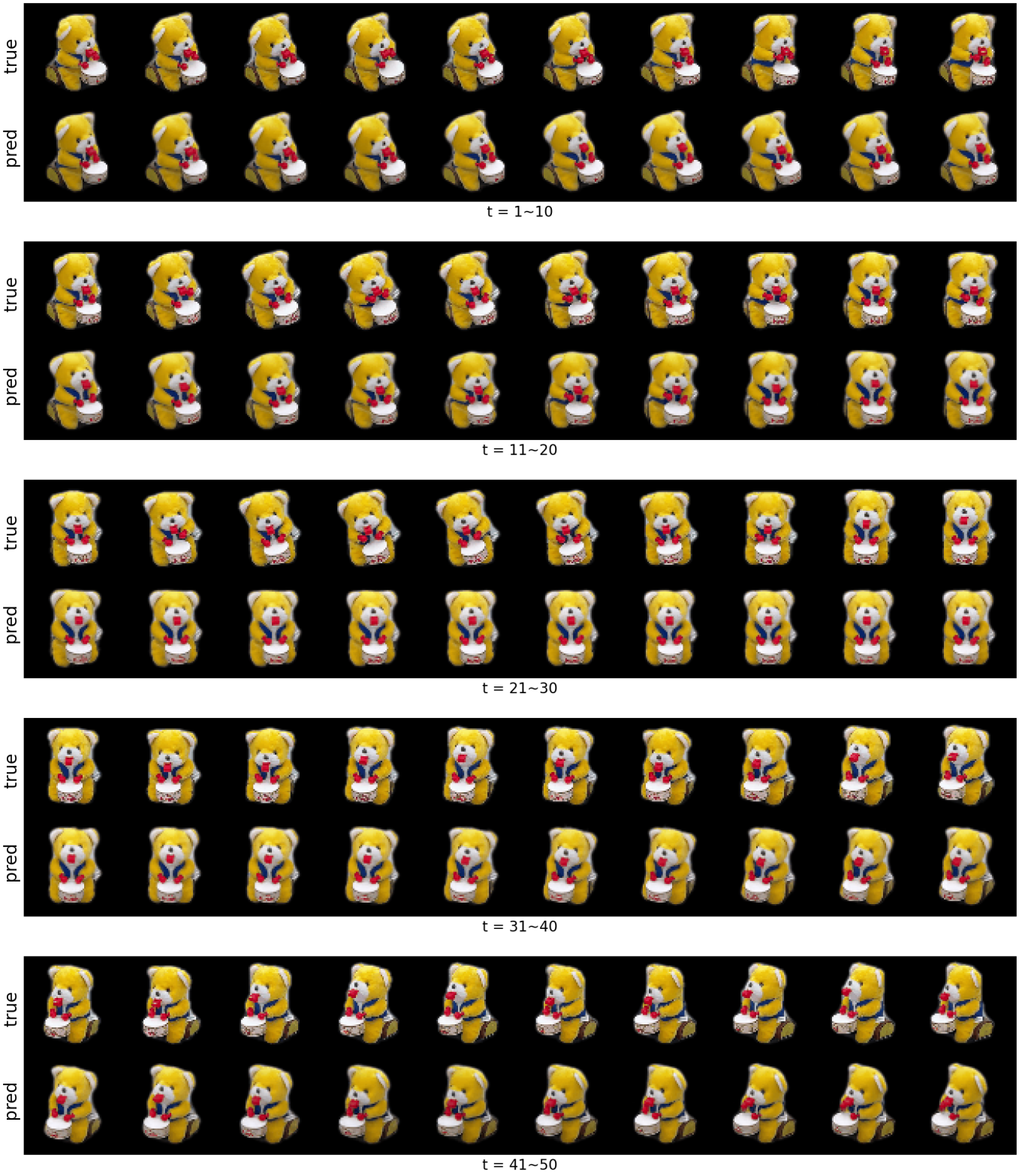


Figure 6.5: 150 frames predicted by DSSM. (1 of 3)

Figure 6.6: 150 frames predicted by DSSM. (2 of 3)

Figure 6.7: 150 frames predicted by DSSM. (3 of 3)

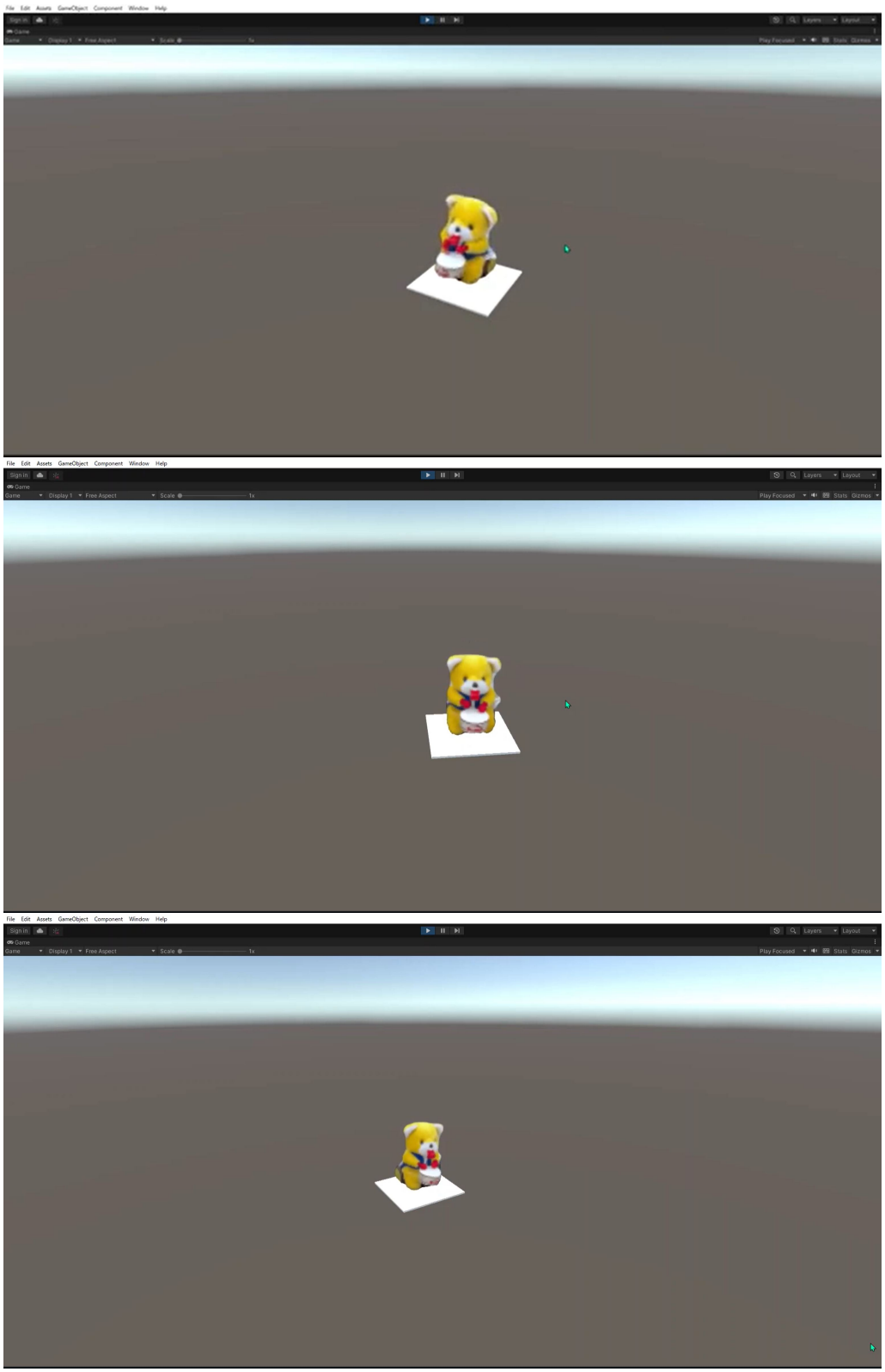


Figure 6.8: Interactive use of World Billboard. `https://www.youtube.com/watch?v=DtniMnkj0qQ`

Chapter 7

Disucussion

We have looked at the use of DeepBillboard with NeRF Billboard and World Billboard as examples. Here we will discuss DeepBillboard in general in several respects.

Visual quality of the Object Itself As for static objects, the NeRF Billboard example shows that they works well enough in the virtual world at sufficient resolution and at sufficient fps. As for dynamic scenes, however, sufficient fps was achieved, but not yet sufficient resolution. Since NeRF Billboards's very fast rendering is in virtue of PlenOctree's caching of the color density field (which can result in file sizes of several hundred MB to several GB), if we try to reproduce a dynamic 3D scene as precisely as NeRF, the caching size will simply be too large for the time resolution, which is not very realistic. A video prediction-based model such as World Billboard can generate images at high speed. However, for high-resolution training and generation that captures temporal changes in local visual features, it will be important to use an approach that is based on video prediction but can directly change observations in the observation space, such as Stable Diffusion [74], rather than an approach that represents everything in the latent space, such as DSSM, since there is too much information on the surface of the object and it is impossible to fully express it in latent variables. In the World Billboard experiment, there was also a problem that the object motion was too influenced by the viewpoint probably because the viewpoint and motion were not learned separately when training video predictions. This could be solved by generating images based on explicit prediction and understanding of rough object's 3D shape. Again, it will be important not to explicitly predict the high-resolution 3D structure, but rather that the 3D structure is only an aid and the observation is generated directly in the space of the observed image.

Quality as a 3D object In both NeRF Billboard and World Billboard, we felt that there are still barriers to making DeepBillboard objects feel completely identical to regular 3D models. The main reasons are (1) insufficient visual quality (World Billboard), (2) a little delay, (3) not being able to reflect ambient light, and (4) sketchy physical interactions. (1) was mentioned in the previous paragraph. For (2), there may be approaches such as a) improving Internet communication technology, b) generating observations in advance by predicting VR users' movements a few milliseconds in advance, c) generating some or all of the possible observations a few milliseconds in advance and sending them all to the user,

and then selecting the appropriate observations on the user side device, etc. For (3), we believe that this can be solved in the near future, since the successor method of NeRF in recent years has already solved this problem. As for (4), in particular, the reality is diminished when touching the Deep Billboard object in VR and the object penetrates the hand. Also, our World Billboard does not infer the shape of the object at all, so it can only give a fake 3D mesh, such as a sphere, and complex interactions are not possible. These problems can be solved by Deep Billboard inferring more accurate shapes, but in practice there are few situations where accurate shapes are required, so it may be more practical to define rule-based interactions with the hand, for example.

Dataset Making Deep Billboard also aims to make it easy for anyone to create 3D objects, but currently, especially creating a World Billboard data set is very laborious. First, the required length of time for the data is very long, at least 15 minutes, and it is necessary to take video from various angles, with as little bias as possible, taking care to fit the object in the camera's angle of view. Other requirements include, for example, markers, monotonous backgrounds, and appropriate lighting. We would also like to automate the process of taking data using a robot. In doing so, we will be able to pursue efficient data capturing methods. Also, meta-learning could be used to improve the learning process so that new targets can be learned more efficiently from a small amount of data if similar data is available.

7.1 Limitations

In theory, Deep Billboard can model everything in the physical world in 3D in a data-driven way, as long as you have video data of the object. However, there are several major limitations. (1) Since the accuracy of reproduction depends directly on the accuracy of video prediction, objects with complex appearance changes that cannot be learned well by video prediction models cannot be reproduced well. (2) While explicit shapes are not required to be calculated, complex physical interactions cannot be represented if the 3D shape is not known. (3) The shape and size of the Billboard canvas is fixed, so objects that grow in size or split up cannot be reproduced. Solving these limitations are our future work.

Chapter 8

Conclusion

We propose Deep Billboard, an extension of the existing Billboard concept. We show that a wide range of objects can be modeled by Deep Billboard in a data-driven way while avoiding the difficulty of 3D reconstruction, and can be used like regular objects by providing interactivity. We hope to see more deep learning research based on Deep Billboard developed in the future and we hope to see more realistic objects in the virtual world.

Acknowledgements

I would like to thank everyone at the Yoichi Ochiai Laboratory and the Yutaka Matsuo Laboratory at the University of Tokyo for their support. I am grateful to Mr. Ryo Shimizu for his assistance in the MITOU Program. This work was supported by the Strategic Information and Communications R&D Promotion Programme (SCOPE) of the Ministry of Internal Affairs and Communications of Japan and New Energy and Industrial Technology Development Organization (NEDO) of Ministry of Economy, Trade and Industry of Japan.

References

[1] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[2] Quei-An Chen. Nerf_pl: a pytorch-lightning implementation of nerf, 2020.

[3] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Mueller, and Sanja Fidler. Extracting Triangular 3D Models, Materials, and Lighting From Images. *arXiv:2111.12503*, 2021.

[4] Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. Banmo: Building animatable 3d neural models from many casual videos. In *CVPR*, 2022.

[5] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021.

[6] Aldo Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 16, No. 2, pp. 150–162, 1994.

[7] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *International journal of computer vision*, Vol. 38, No. 3, pp. 199–218, 2000.

[8] Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, Vol. 35, No. 2, pp. 151–173, 1999.

[9] Michael Waechter, Nils Moehrle, and Michael Goesele. Let there be color! Large-scale texturing of 3D reconstructions. *ECCV*, 2014.

[10] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. *SIGGRAPH*, 2001.

[11] Paul Debevec, C. J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. *SIGGRAPH*, 1992.

[12] Daniel Wood, Daniel Azuma, Wyvern Aldinger, Brian Curless, Tom Duchamp, David Salesin, and Werner Stuetzle. Surface light fields for 3D photography. *SIGGRAPH*, 2000.

[13] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, Vol. 32, No. 8, pp. 1362–1376, 2009.

[14] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. *arXiv preprint arXiv:1708.05375*, 2017.

[15] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2626–2634, 2017.

[16] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[17] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[18] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Trans. Graph.*, Vol. 38, No. 4, pp. 65:1–65:14, July 2019.

[19] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields, 2021.

[20] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*, 2021.

[21] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[22] Daniel Rebain, Wei Jiang, Soroosh Yazdani, Ke Li, Kwang Moo Yi, and Andrea Tagliasacchi. Derf: Decomposed radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14153–14161, 2021.

[23] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. *arXiv preprint arXiv:2103.10380*, 2021.

[24] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for real-time rendering of neural radiance fields. In *ICCV*, 2021.

[25] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. *ICCV*, 2021.

[26] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. *arXiv*, 2021.

[27] Petr Kellnhofer, Lars C Jebe, Andrew Jones, Ryan Spicer, Kari Pulli, and Gordon Wetzstein. Neural lumigraph rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4287–4297, 2021.

[28] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. *ICCV*, 2021.

[29] Kacper Kania, Kwang Moo Yi, Marek Kowalski, Tomasz Trzciński, and Andrea Tagliasacchi. Conerf: Controllable neural radiance fields, 2021.

[30] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B Tenenbaum, and Jiajun Wu. Neural radiance flow for 4d view synthesis and video processing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14324–14334, 2021.

[31] Yunzhu Li, Shuang Li, Vincent Sitzmann, Pulkit Agrawal, and Antonio Torralba. 3d neural scene representations for visuomotor control. In *Conference on Robot Learning*, pp. 112–123. PMLR, 2022.

[32] Rahul G Krishnan, Uri Shalit, and David Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

[33] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

[34] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[35] Abe Davis, Marc Levoy, and Fredo Durand. Unstructured light fields. In *Computer Graphics Forum*, Vol. 31, pp. 305–314. Wiley Online Library, 2012.

[36] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 31–42, 1996.

[37] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 43–54, 1996.

[38] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019.

[39] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Theobalt Christian, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. *arXiv preprint arXiv:2107.13421*, 2021.

[40] Alex Trevithick and Bo Yang. GRF: Learning a general radiance field for 3d scene representation and rendering. *arXiv*, 2021.

[41] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021.

[42] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021.

[43] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. *arXiv preprint arXiv:2103.15595*, 2021.

[44] Alexander W. Bergman, Petr Kellnhofer, and Gordon Wetzstein. Fast training of neural lumigraph representations using meta learning. In *NeurIPS*, 2021.

[45] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *NeurIPS*, 2021.

[46] Wang Yifan, Shihao Wu, Cengiz Oztireli, and Olga Sorkine-Hornung. Iso-points: Optimizing neural implicit surfaces with hybrid representations. In *CVPR*, 2020.

[47] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[48] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *International Conference on Computer Vision (ICCV)*, 2021.

[49] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, Vol. 27, , 2014.

[50] M. Gadelha, S. Maji, and R. Wang. 3d shape induction from 2d views of multiple objects. In *2017 International Conference on 3D Vision (3DV)*, pp. 402–411, Los Alamitos, CA, USA, oct 2017. IEEE Computer Society.

[51] Philipp Henzler, Niloy J. Mitra, and Tobias Ritschel. Escaping plato's cave: 3d shape from adversarial rendering. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[52] Thu Nguyen-Phuoc, Christian Richardt, Long Mai, Yong-Liang Yang, and Niloy Mitra. Blockgan: Learning 3d object-aware scene representations from unlabelled images. In *Advances in Neural Information Processing Systems 33*, Nov 2020.

[53] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, Vol. 29. Curran Associates, Inc., 2016.

[54] Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Josh Tenenbaum, and Bill Freeman. Visual object networks: Image generation with disentangled 3d representations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, Vol. 31. Curran Associates, Inc., 2018.

[55] Yiyi Liao, Katja Schwarz, Lars Mescheder, and Andreas Geiger. Towards unsupervised learning of generative models for 3d controllable image synthesis. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[56] Attila Szabó, Givi Meishvili, and Paolo Favaro. Unsupervised generative 3d shape learning from natural images. *CoRR*, Vol. abs/1910.00287, , 2019.

[57] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *The IEEE International Conference on Computer Vision (ICCV)*, Nov 2019.

[58] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *arXiv*, 2020.

[59] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[60] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. Gram: Generative radiance manifolds for 3d-aware image generation. In *arXiv*, 2021.

[61] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *arXiv*, 2021.

[62] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis, 2021.

[63] Alec Rivers, Takeo Igarashi, and Frédo Durand. 2.5d cartoon models. *ACM Trans. Graph.*, Vol. 29, No. 4, jul 2010.

[64] Maki Kitamura, Yoshihiro Kanamori, and Reiji Tsuruno. 2.5d modeling from illustrations of different views. *International Journal of Asia Digital Art and Design Association*, Vol. 18, No. 4, pp. 74–79, 2014.

[65] Tamás Umenhoffer, László Szirmay-Kalos, and Gábor Szijártó. Spherical billboards and their application to rendering explosions. In *Proceedings of Graphics Interface 2006*, GI '06, p. 57–63, CAN, 2006. Canadian Information Processing Society.

[66] Peter Lindgren. Volumetric smoke in real-time applications: by use of billboard-and shader-methods, 2008.

[67] Cristina Amati and Gabriel J. Brostow. Modeling 2.5d plants from ink paintings. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, SBIM '10, p. 41–48, Goslar, DEU, 2010. Eurographics Association.

[68] Marcel Germann, Alexander Hornung, Richard Keiser, Remo Ziegler, Stephan Würmlin, and Markus Gross. Articulated billboards for video-based rendering. In *Computer Graphics Forum*, Vol. 29, pp. 585–594. Wiley Online Library, 2010.

[69] Thomas Leimkühler and George Drettakis. Freestylegan: Free-view editable portrait rendering with the camera manifold. Vol. 40, No. 6, 2021.

[70] Dominik Borer, Lu Yuhang, Laura Wuelfroth, Jakob Buhmann, and Martin Guay. Rig-space neural rendering. *arXiv preprint arXiv:2003.09820*, 2020.

[71] Rahul Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31, 2017.

[72] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013.

[73] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pp. 2555–2565, 2019.

[74] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.