

iButter: Neural Interactive Bullet Time Generator for Human Free-viewpoint Rendering

Liao Wang
ShanghaiTech University
Shanghai, China
wangla@shanghaitech.edu.cn

Ziyu Wang
ShanghaiTech University
Shanghai, China
wangzy6@shanghaitech.edu.cn

Pei Lin
ShanghaiTech University
Shanghai, China
linpei@shanghaitech.edu.cn

Yuheng Jiang
ShanghaiTech University
Shanghai, China
jiangyh2@shanghaitech.edu.cn

Xin Suo
ShanghaiTech University
Shanghai, China
suoixin@shanghaitech.edu.cn

Minye Wu
ShanghaiTech University
Shanghai, China
wumy@shanghaitech.edu.cn

Lan Xu
ShanghaiTech University
Shanghai, China
xulan1@shanghaitech.edu.cn

Jingyi Yu
Shanghai Engineering Research
Center of Intelligent Vision and
Imaging, School of Information
Science and Technology,
ShanghaiTech University
Shanghai, China
yujingyi@shanghaitech.edu.cn



Multi-view RGB Streams

Interactive Bullet-time Design

High-quality Bullet-time Effects

Figure 1: Our neural interactive bullet-time generator (iButter) enables convenient, flexible and interactive design for human bullet-time visual effects from dense RGB streams, and achieves high-quality and photo-realistic human performance rendering along the designed trajectory.

ABSTRACT

Generating “bullet-time” effects of human free-viewpoint videos is critical for immersive visual effects and VR/AR experience. Recent neural advances still lack the controllable and interactive bullet-time design ability for human free-viewpoint rendering, especially

under the real-time, dynamic and general setting for our trajectory-aware task. To fill this gap, in this paper we propose a neural interactive bullet-time generator (iButter) for photo-realistic human free-viewpoint rendering from dense RGB streams, which enables flexible and interactive design for human bullet-time visual effects. Our iButter approach consists of a real-time preview and design stage as well as a trajectory-aware refinement stage. During preview, we propose an interactive bullet-time design approach by extending the NeRF rendering to a real-time and dynamic setting and getting rid of the tedious per-scene training. To this end, our bullet-time design stage utilizes a hybrid training set, light-weight network design and an efficient silhouette-based sampling strategy. During refinement, we introduce an efficient trajectory-aware scheme within 20 minutes, which jointly encodes the spatial, temporal consistency and semantic cues along the designed trajectory,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '21, October 20–24, 2021, Virtual Event, China

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8651-7/21/10...\$15.00

<https://doi.org/10.1145/3474085.3475412>

achieving photo-realistic bullet-time viewing experience of human activities. Extensive experiments demonstrate the effectiveness of our approach for convenient interactive bullet-time design and photo-realistic human free-viewpoint video generation.

CCS CONCEPTS

• **Computing methodologies** → **Image-based rendering.**

KEYWORDS

free-viewpoint video; bullet-time; novel view synthesis; neural rendering; neural representation

ACM Reference Format:

Liao Wang, Ziyu Wang, Pei Lin, Yuheng Jiang, Xin Suo, Minye Wu, Lan Xu, and Jingyi Yu. 2021. *iButter: Neural Interactive Bullet Time Generator for Human Free-viewpoint Rendering*. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21)*, October 20–24, 2021, Virtual Event, China. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3474085.3475412>

1 INTRODUCTION

Novel view synthesis has been widely used in visual effects to provide unique viewing experiences. One of the most famous examples is the “bullet-time” effects presented in the feature film *The Matrix*, which creates the stopping-time illusion with smooth transitions of viewpoints surrounding the actor. Such human-centric free viewpoint video generation with bullet-time effects further evolves as a cutting-edge yet bottleneck technique with the rise of VR/AR over the last decade. How to enable flexible and convenient bullet-time design for human free-viewpoint rendering with fully natural and controllable viewing experiences remains unsolved and has recently attracted substantive attention of both the multi-media and computer graphics communities.

To produce such bullet-time effect for dynamic human activities, early solutions [5, 14, 58] require dense and precisely arranged cameras which are aligned on a track forming a complex curve through space and restricted to pre-designed camera trajectory without the ability of convenient and flexible trajectory design. The model-based solutions [11, 29] rely on multi-view dome-based setup for high-fidelity reconstruction and texture rendering in novel views to support flexible bullet-time design. However, the model reconstruction stages require two to four orders of magnitude more time than is available for daily interactive applications. The recent volumetric fusion methods [40, 49, 53] have enabled much faster human reconstruction by leveraging the RGBD sensors and modern GPUs. But they are still restricted by the limited mesh resolution and fail to provide photo-realistic bullet-time effects. The recent neural rendering techniques [24, 28, 34, 41, 42, 46] bring huge potential for neural human free-viewpoint rendering from multiple RGB input. However, these solutions rely on per-scene training or are hard to achieve real-time performance due to the heavy network. Only recently, some approaches [9, 45] enhance the neural radiance field (NeRF) [28] to break the per-scene training constraint, while other approaches [13, 30, 35, 36, 51] further enable real-time NeRF rendering for static scenes. However, researchers did not explore these solutions to strengthen the bullet-time design for human activities, especially for the flexible, interactive and real-time setting without specific per-scene training. Moreover, existing

approaches [9, 28, 45] refine the NeRF representation using all the input frames, leading to inefficient training overload for dynamic scenes without exploiting the temporal redundancy, especially for our trajectory-aware bullet-time design task.

In this paper, we attack these challenges and present *iButter* – a neural Interactive **B**ulle**T**-Time **G**ene**R**ator for human free-viewpoint rendering from multiple RGB streams. As shown in Fig. 1, our approach enables real-time immersive rendering of novel human activities based on neural radiance field, which is critical for artists to design arbitrarily bullet-time trajectories interactively, flexibly and conveniently. Once the trajectory designed, our approach further provide an efficient trajectory-aware refinement scheme within about 20 minutes to utilize the spatial and temporal redundancy for photo-realistic bullet-time rendering.

Generating such a human free-viewpoint video by combining bullet-time design with neural radiance field rendering in a real-time, general and dynamic manner is non-trivial. More specifically, we adopt doom-like dense multiple RGB streams as input and utilize the neural radiance field [28] as the underlying scene representation. During the real-time preview stage for flexible bullet-time trajectory design, inspired by the general NeRF approach [45], we adopt the network architecture with pixel-aligned feature extractor, implicit scene representation and radiance-based volumetric renderer so as to get rid of the tedious per-scene training, with the aid of a hybrid training set using both our doom system and the Twindom dataset. For further real-time inference in our bullet-time design task, we utilize the inherent global information from our multi-view setting via a efficient silhouette-based sampling strategy. We also adopt a light-weight network design for each architecture components, which serves as a good compromising settlement between fast run-time performance and realistic preview quality. With the interactively designed trajectory, we further introduce a novel efficient trajectory-aware refinement scheme to provide high-quality and photo-realistic bullet-time viewing experience along the trajectory. To this end, we encode the spatial consistency by re-rendering the target image into an adjacent view through our refined network in a self-supervised manner. We also encode the temporal consistency between two successive frames using the correspondences found by non-rigid deformation on the geometry proxies of the preview stage. Besides, we introduce a semantic feature aggregation scheme to enhance the rendering details for our refinement stage.

To summarize, our main contributions include:

- We present a neural bullet-time generation scheme for photo-realistic human free-viewpoint rendering from dense RGB streams, which enables flexible, convenient and interactive design for human bullet-time visual effects unseen before.
- We propose a bullet-time preview approach which extends the NeRF rendering to a real-time, general and dynamic setting, with the aid of hybrid training set, light-weight network design and an efficient silhouette-based sampling strategy.
- We introduce a trajectory-aware refinement scheme in 20 minutes which jointly encodes the spatial, temporal consistency and semantic cues for photo-realistic bullet-time viewing experience along the designed trajectory.

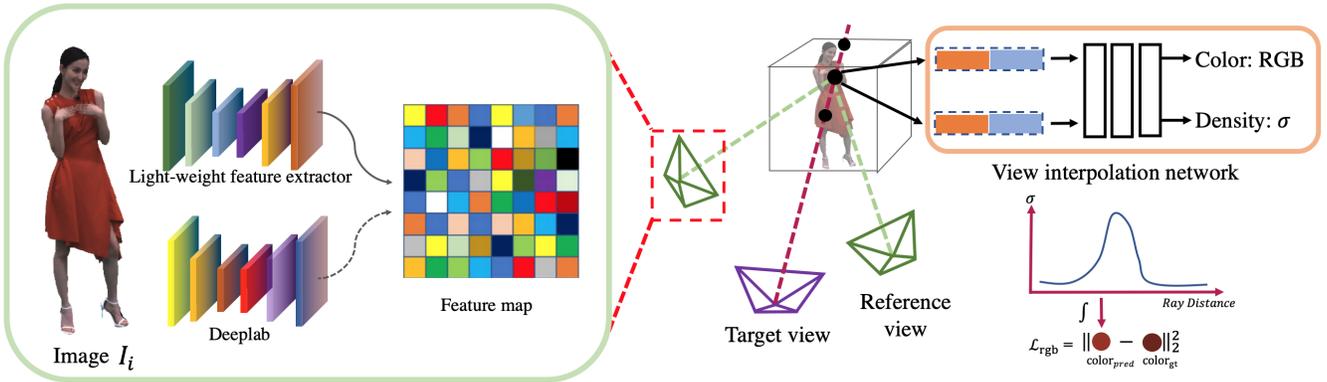


Figure 2: Illustration of our network architecture. Figure on the left demonstrates the feature extraction of our iButter. The Deeplab features (dot curve) only be used in high-quality bullet-time rendering stage. Features of sample points are fetched from reference views’ feature maps.

2 RELATED WORK

Human Modeling. Modeling a human is a critical step in generating high quality free-viewpoint rendering. Some works [11, 19, 29] have reconstructed high-quality dynamic geometric models and texture maps by building complex camera systems and controllable lighting systems to overcome the occlusion challenges. The time issue of model reconstruction is preventing these methods from daily interactive applications. With the utility of the depth sensor and the powerful computing power of GPU, some works [15, 31, 40, 47, 54] combine the volumetric fusion and non-rigid tracking to reconstruct free-form human body in real-time by using only a single RGBD sensor. However, the monocular RGBD methods are limited by the self-occlusion and the capture quality of the depth sensor and cannot reconstruct high-quality models and texture with details. Recently, some work[20, 21, 32] has been used to apply deep learning methods to recover the geometry and texture of the human body from a single image. Some work [12, 20, 21, 32, 37, 38, 56] formulate the human body reconstruction problem into the regression problem of the shape and pose parameters of the parametric models. And recent studies combine deep learning and different 3D representations including depth maps [12], voxel grids [56] and implicit functions[37, 38] to recover human model from single image. But these methods always fail to reconstruct high-quality geometry and fine-detailed texture in order to achieve photo-realistic rendering in novel views. Comparably, our approach enables photo-realistic human free-viewpoint video generation using only rgb inputs with mask information to generate SFS prior.

Neural Rendering. The recent progress of neural rendering [17] shows great capacity to achieve photorealistic novel views synthesis based on various data representation, such as pointclouds [46], voxels[23] and implicit function [25, 28, 37, 41]. For dynamic scenes neural rendering, NHR [46] using dynamic point clouds to encode sequence feature. Neural Volumes [24] encodes multi-view images into a latent code and decodes to a 3D volume representation followed by volume rendering. Recent work [33, 34] use radiance field to render dynamic scenes. They regress each time step into a canonical frame. However, these methods need per-scene training is hard to provide an instant preview for an unseen scene. Other methods, [6, 9, 45, 52] try to extend the ability of neural radiance field

to arbitrary new scenes. However, they cannot achieve real-time rendering to provide users with interactive bullet-time trajectory design and may have worse performance on the unseen scene than per-scene training models. While other methods [13, 18, 30, 36, 51] try to render static scene in real-time but require per-scene training. In contrast, our method can generalize to new, dynamic, real-world scenes for interactive preview without long time training and a high-quality refinement in a short time.

Free Viewpoint Rendering. Traditional image-based rendering generate novel view texture by using the input images blending [5, 8, 22, 26, 58]. They calculate blending weights by considering camera view angles and spatial distance. Other methods [3] use explicit geometry prior to guiding novel view rendering. However, these methods are limited by the accuracy of geometry prior. Another direction is using the multiplane images [2, 10, 27, 39] to represent the scene. While these methods cannot provide a wide range of free viewing rendering effects. Our approach can generate arbitrary free view rendering results from the user-designed bullet-time trajectory.

3 OVERVIEW

We propose a Interactive BULLEt-TIME GENEration, or iButter, which allows users to interactively making photo-realistic human bullet-time effects only from multi-view RGB image sequences without heavy 3D reconstruction. The iButter’s pipeline can be divided into two stages. The first stage is real-time dynamic 3D human pre-viewing, and the second one is high-quality bullet-time rendering, which are described as follows.

Real-time Dynamic 3D Human Previews. One key feature of our iButter is providing real-time previews for interactive virtual camera trajectory selection. And the selected trajectory will be applied to render high-quality bullet-time effects. Our iButter adopts recent neural radiance fields work [45] to our real-time photo-realistic rendering task and achieves generic view interpolation directly based on RGB images without per-scene training and time-consuming 3D reconstructions. In section 4.1, we first introduce the network architecture of iButter for generic view synthesis from multi-view video streams. Meanwhile, a silhouette-based sampling

strategy leverages inherent global information from human silhouettes and is utilized to significantly reduce computational complexity and accelerate neural training and rendering speed (section 4.2). Accordingly, users can design bullet-time trajectories interactively and get intuitive feedback from our renderer.

High-quality Bullet-time Rendering. Moreover, iButter can produce high-resolution photo-realistic rendering results with temporal and multi-view consistency on the given selected trajectory from users. To this end, we design a novel efficient trajectory-aware refinement scheme which is introduced in section 4.3. On the one hand, in the scheme, we enhance multi-view consistency by re-rendering a near reference camera with a rendered camera on the trajectory. On the other hand, we warp two adjacent rendered images on the trajectory according to model correspondences and force them to be the same for strengthening the temporal consistency of our iButter. Also, we introduce a semantic feature aggregation scheme to enhance rendering quality and network generalization of human-related scenes. After these self-supervised refinements, iButter is able to render high-quality images on the selected trajectory and preserves multi-view and temporal consistency.

4 IMAGE-BASED BULLET-TIME GENERATOR

Here, we introduce our neural image-based bullet-time generator (iButter). Firstly, we show the neural rendering network’s architecture which can achieve generic photo-realistic view interpolation. Based on this kind of network, we introduce a silhouette-based sampling strategy for fast training, rendering, and previewing. We also propose a semantic feature aggregation scheme to improve the rendering results of human-related scenes. Then user-selected trajectory will guide the trajectory-aware network refinement and help to enhance temporal and spatial consistency in a self-supervised manner.

4.1 Network Architecture

The IBRNet [45] synthesizes the novel target view by interpolating nearby source views instead of encoding an entire scene into a single network, which enables realistic rendering without per-scene training. We adopt this architecture to our bullet-time task with dynamic human and real-time free-view rendering. iButter evolves from this kind of architecture. The whole architecture can be divided into three parts: feature extraction, view interpolating, and volume rendering. Specifically, in the beginning, iButter selects K nearest reference cameras according to a given target camera’s parameters \mathbf{P}_o . In the first part, we extract features from input images. Let \mathbf{I}_i denotes the color image of i -th source view at one frame. We use a feature extractor $\Phi_f(\cdot)$ to extract feature maps from RGB images $\{\mathbf{I}_i\}$. So we have a set of cameras and feature maps $\mathcal{S} = \{(\mathbf{P}_i, \Phi_f(\mathbf{I}_i))\}_{i=1}^K$, where \mathbf{P}_i is the camera’s parameters of i -th source view. In the view interpolating part, iButter fetches features from \mathcal{S} for each sample points along camera rays and predicts their density σ and color \mathbf{c} via our view interpolation network in a hierarchical manner. Different from IBRNet, we replace the ray transformer with simple MLPs in the view interpolation network. Note that the output of the view interpolating part is colors and densities of sample points along rays. Finally, to render the color of a ray \mathbf{r}

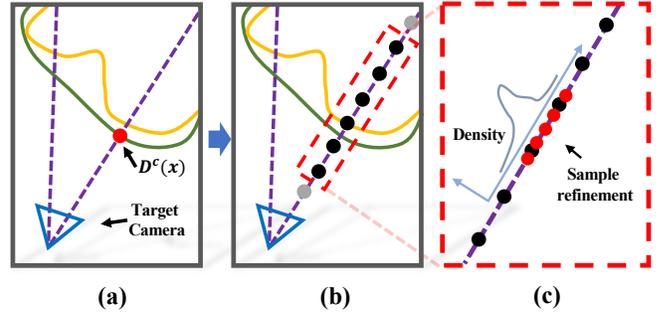


Figure 3: Demonstration of our silhouette-based sampling strategy. The orange curve represents the real geometry surface; The green curve is the surface generated by SfS. (a) iButter calculates depth values for camera rays on coarse geometry G; (b) Sampled points near the depth value in "coarse" stage; (c) Sampled points according to density distribution in "fine" stage.

through the scene, we accumulate colors along the ray modulated by densities in the volume rendering part. The volume rendering part is formulated as :

$$\tilde{C}(\mathbf{r}) = \sum_{k=1}^{n_s} T_k (1 - \exp(-\sigma_k \delta_k)) \mathbf{c}_k, \quad (1)$$

$$T_k = \exp\left(-\sum_{j=1}^{k-1} \sigma_j \delta_j\right),$$

where n_s is the number of sample points along the ray \mathbf{r} . σ_k , \mathbf{c}_k , δ_k denotes the density, color of the samples and the interval between adjacent samples respectively.

The architecture of our iButter is illustrated in Fig. 2. Please refer to [45] for more details.

4.2 Real-time Previews and Trajectory Selection

According to Equation 1, colors of sample points near human geometry surface contribute primarily to the ray color $\tilde{C}(\mathbf{r})$ in an ideal radiance field. Weights for sample points that are away from the surface will be close to zeros. So based on this observation, instead of sampling all points between near-far planes along with rays, we propose a sampling strategy based on multi-view silhouettes to sample around human geometry surface efficiently. This strategy can significantly accelerate both training and rendering speed by reducing useless sample points.

Specifically, for each frame with multi-view RGB image data, we generate a coarse 3D human mesh G . We deploy a fast Shape-from-Silhouette (SfS) method [50] to reconstruct geometry from multi-view human silhouettes in real-time. Having green screens in our multi-view capture system, we can fast extract human silhouettes from RGB images via chroma key segmentation and background subtraction.

Before point sampling along a ray \mathbf{r} , we first calculate the depth value from reconstructed G . Once the ray \mathbf{r} hits human geometry, we evenly sample points near the depth value within a small range in the coarse sampling stage and then follow the hierarchical sampling scheme in [28]. Fig. 3 illustrates our silhouette-based

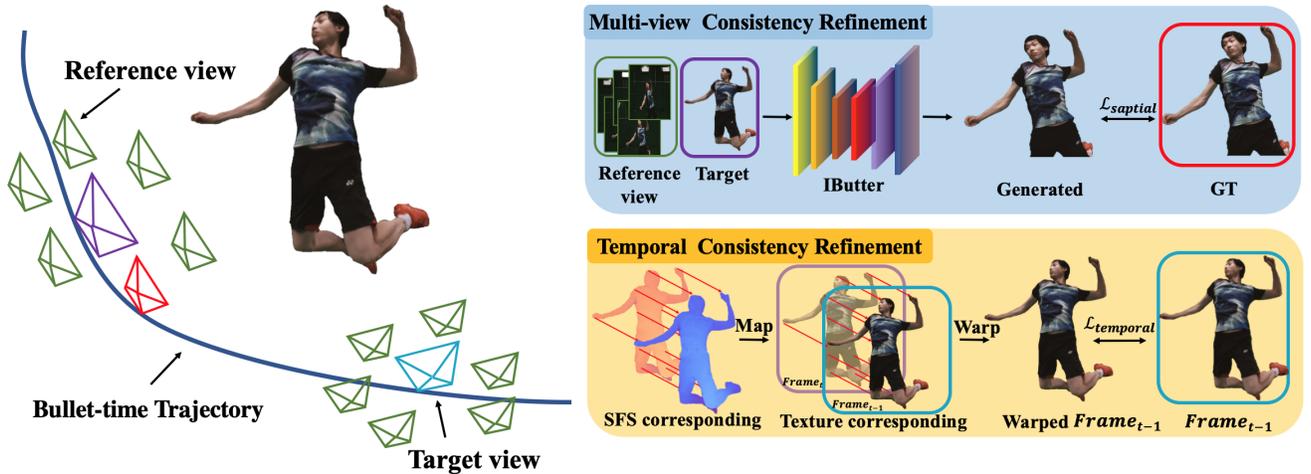


Figure 4: Illustration of our trajectory-aware refinement pipeline. Both spatial consistency loss and temporal consistency loss are applied to improve rendering quality.

sampling strategy. Besides, in the preview stage, we deploy a UNet-like light-weight network as the feature extractor Φ_f , which has a downsampling scale of 4. Please refer to the supplementary material for more network details. We formulate iButter $\Phi(\cdot)$ as:

$$\tilde{\mathbf{I}} = \Phi(\mathbf{S}, \mathbf{P}_v, \mathbf{G}), \quad (2)$$

where $\tilde{\mathbf{I}}$ is the rendered image in the target view.

All these efficient procedures enable a real-time rendering from multi-view live streams. With our iButter, users can select trajectory presets and adjust key frames through an user interface (UI) in an interactive way with instant feedbacks, which is artist friendly.

4.3 High-quality Bullet-time Rendering

Although the real-time version of iButter has achieved rather realistic rendering result, the synthesis quality is good but not enough for production. To this end, we propose feature aggregation and trajectory-aware refinement to boost rendering quality into a product level. Similarly, we also bring our silhouette-based sampling strategy introduced in § 4.2 into our high-quality bullet-time rendering so as to accelerate both training and rendering speed.

Semantic Feature Aggregation. As mentioned before, iButter produces a discrete radiance field according to fetched features of sample points from multi-view image feature maps. Ideally, more discriminative features will facilitate networks to approximate human geometry and appearance more correctly, produce a more accurate discrete radiance field, and better rendering results. Here, we aggregate both local features from our light-weight feature extractor and high-level semantic features from deeplab [7] segmentation pre-trained backbone to form a new discriminative feature extractor Φ_f , as illustrated in Fig. 2. Although increasing the computational complexity, the new feature extractor boosts the rendering quality and is one step closer to product-level results as discussed in section 6.2 .

Trajectory-aware Refinement. Given selected bullet-time trajectory, we set out to refine iButter for product-level rendering results

on it. Multi-view consistency and temporal consistency are two main challenges for such neural renderers like ours because iButter predicts radiance fields according to view image features instead of maintaining a constant representation. Without these consistencies, iButter will produce flickering and ghosting artifacts on result images. We attack these two challenges by introducing a training scheme to alleviate such artifacts based on given trajectory. Figure 4 demonstrates the pipeline of our trajectory-aware refinement.

The selected bullet-time trajectory defines K related reference cameras in each frame. So we can narrow down the training dataset by discarding other unrelated cameras. For enhancing multi-view consistency, we first render the target view image $\tilde{\mathbf{I}}_v^t = \Phi(\mathbf{S}^t, \mathbf{P}_v^t, \mathbf{G}^t)$ in the frame t . Next, we utilize $\tilde{\mathbf{I}}_v^t$ to re-render reference view with other $K - 1$ reference cameras, and supervise the network training with the ground truth. We formulate the multi-view consistency loss $\mathcal{L}_{spatial}$ as:

$$\mathcal{L}_{spatial} = \frac{1}{2} \sum_{t=1}^T \sum_{i=1}^K \|\mathbf{M}_v^t \cdot (\mathbf{I}_i^t - \Phi(\mathbf{S}^t \cup s_v^t \setminus s_i^t, \mathbf{P}_i^t, \mathbf{G}^t))\|_2^2, \quad (3)$$

where $s_v^t = (\mathbf{P}_v^t, \Phi_f(\tilde{\mathbf{I}}_v^t))$ is the generated camera-feature pair for target view camera on the trajectory, and T is the length of bullet-time trajectory; \mathbf{I}_i^t is the ground truth image of reference view i . \mathbf{M}_v^t is the human foreground mask generated by the coarse geometry at frame $t - 1$ for avoiding learning background pixels.

Since having coarse human geometry for each frame, we find correspondences between adjacent 3D models on timeline by a light-weight model tracking algorithm [48]. These model correspondences establish connections between these two adjacent frames and are used for image warping. Let $f_{t,t-1}(\cdot)$ represents the 2D flow which warps image from frame t to frame $t - 1$ and projected from model correspondences. We formulate the temporal consistency loss $\mathcal{L}_{temporal}$ as:

$$\mathcal{L}_{temporal} = \frac{1}{2} \sum_{t=1}^T \|\mathbf{M}_v^{t-1} \cdot (\tilde{\mathbf{I}}_v^{t-1} - f_{t,t-1}(\tilde{\mathbf{I}}_v^t))\|_2^2, \quad (4)$$



Figure 5: Several examples that demonstrate the quality and fidelity of the render results (right) from the trajectory user designed (left) from our system on the data we captured, including human portrait, human with objects and multi humans.

where \tilde{I}_v^{t-1} and \tilde{I}_v^t are target view images generated by iButter for adjacent frames.

5 IMPLEMENTATION DETAILS

Our multi-view data are captured from a dome system with up to 80 cameras arranged on a cylinder. All cameras are synchronized and capture at 25 frames per second. For training data, we collect 20 sets of datasets where the performers are in different clothing and perform different actions. Moreover, we use Twindom dataset [43] to generate multi-view synthetic datasets with single frame to extend training set in diversity. We first pretrain iButter both the feature extraction network and the view interpolation network end-to-end on training set using a single photometric loss:

$$\mathcal{L}_{rgb} = \frac{1}{2} \sum_{t=1}^T \sum_{i=1}^N \|M_v^t \cdot (I_i^t - \tilde{I}_i^t)\|_2^2, \quad (5)$$

where N is the number camera in the frame.

During trajectory-aware refinement, we combine all losses together and formulate total loss \mathcal{L} as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{rgb} + \lambda_2 \mathcal{L}_{spatial} + \lambda_3 \mathcal{L}_{temporal}, \quad (6)$$

where $\lambda_1 = 0.5$ and $\lambda_2 = \lambda_3 = 0.25$ in this paper. Note that we do not update the parameters of deeplab backbone during refinement.

We use Adam optimizer to train our networks with a base learning rates of 10^{-3} and 5×10^{-4} for feature extraction network and the view interpolation network respectively.

6 EXPERIMENTAL RESULTS

In this section, we evaluate the result of our iButter system on a variety of scenarios followed by the comparison with other methods, both qualitatively and quantitatively. We pre-train our model on a Nvidia GeForce RTX3090 GPU for 3 days with 400 rays per batch and it takes only 20 minutes for our trajectory-aware refinement using the same GPU format. We sample 10 points in the coarse

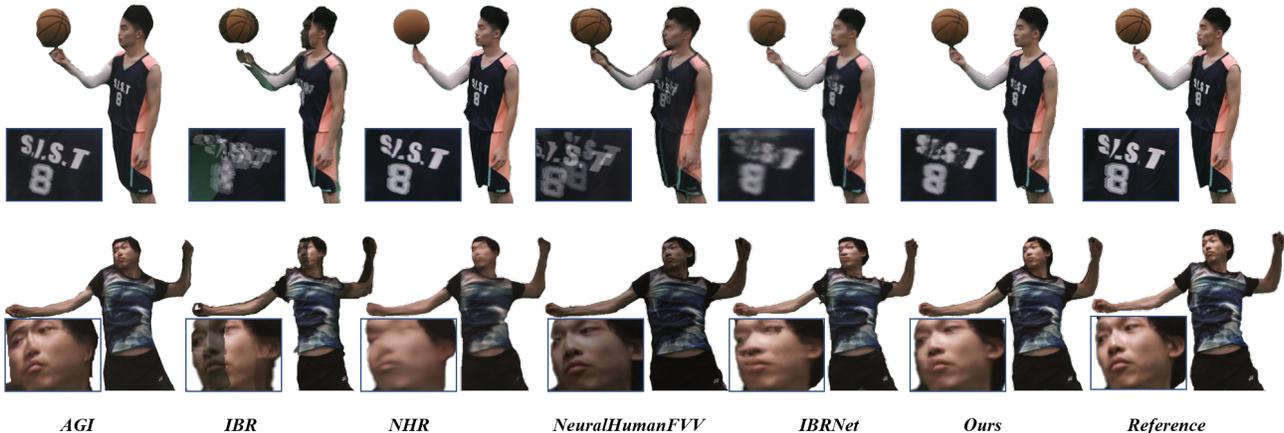


Figure 6: Comparison of our methods with AGI [1], IBR [4], NHR [46], NeuralHumanFVV [41], and IBRNet [45]. Our neural portrait achieves much more realistic rendering results.

Method	AGI	IBR	NHR	FVV	IBRNet	Ours
PSNR \uparrow	22.99	28.66	31.56	23.29	29.41	32.27
SSIM \uparrow	.9489	.9661	.9687	.9203	.9572	.9716
LPIPS \downarrow	.0716	.0749	.0644	.0863	.0846	.0659

Table 1: Quantitative comparison against various methods.

stage during the preview bullet-time design stage and sample 64 points in the coarse stage and additional 64 points in the fine stage during refinement. We render images with the output resolution of 512×384 in our interactive preview using about 0.15 seconds and 1024×768 in our final refinement result using about 30 seconds. Fig. 5 provides some representative results to demonstrate that our approach generates photo-realistic human free-viewpoint rendering from dense RGB streams, which enables flexible and interactive design for human bullet-time visual effects. Please refer to the supplementary video for more video results.

6.1 Comparison

For thorough comparison, we perform quantitative and qualitative comparisons against AGI [1] using the Agisoft software, the image-based method IBR [4], the point-based NHR [46], the NeRF-based IBRNet [45] and the hybrid texturing-based NeuralHumanFVV [41]. Note that NeuralHumanFVV and IBRNet are trained using the same training dataset as ours and IBR adopts the same SFS geometry proxy as ours for fair comparison. As shown in Fig. 6, the non-learning approaches AGI and IBR suffer from severe texturing artifacts due to inaccurate geometry reconstruction. Besides, NeuralHumanFVV and IBRNet suffer from mis-blending or blur artifacts, especially for the fast motion regions. The NHR achieves promising results in those slow-motion scenarios loses fidelity in fast motions due to the ambiguity of point feature learning. Moreover, it takes 7 days to train NHR on a specific sequence to achieve reasonable texturing results. In contrast, our approach achieves photo-realistic free-viewpoint rendering of human activities especially for those fast motion regions, which only needs less than 20

minutes for refinement. Our approach also enables human bullet-time visual effects design in a flexible and interactive manner, based on our real-time, dynamic NeRF-based rendering without per-scene training. For quantitative comparison, we adopt Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM) [57] and Learned Perceptual Image Patch Similarity (LPIPS) [55] as metrics on the whole testing dataset by comparing the rendering results with source view inputs. As shown in Tab. 4, our approach outperforms other methods in terms of PSNR and SSIM and achieves comparable performance against NHR in terms of LPIPS. These comparisons above illustrate the effectiveness of our approach for flexible bullet-time design and photo-realistic free-viewpoint rendering. More comparison details can be found in supplementary materials.

6.2 Evaluation

Trajectory-based Refinement. Here we compare our trajectory-based refinement scheme with the brute force scheme using all the input frames (denoted as **Brute force**). As illustrated in Fig. 7, our trajectory-aware scheme achieve more efficient refinement within 20 minutes by jointly utilizing the spatial and temporal redundancy for the bullet-time design task. Fig. 8 further provides the final PSNR with various refining time for two schemes. While brute force scheme suffers from inefficient refinement as ours even with longer refining time, our scheme achieve more efficient and temporal consistent rendering results.

Number of sample points and feature extraction network.

Here we evaluate the number of per-ray sample points and feature extraction network architecture in our interactive preview stage by rendering 200 novel views on a 24GB Nvidia GeForce RTX3090 at the resolution of 512×384 . As shown in Tab. 2 (see supplementary material for more qualitative and quantitative results), more sample points and deeper feature extractor lead to better rendering quality and more rendering time. Through these thorough evaluations, we choose to sample 10 points per-ray and 3 residual blocks in feature extraction in the interactive preview stage, which serves as a good compromising settlement between fast run-time performance and flexibly interactive ability for bullet-time design.

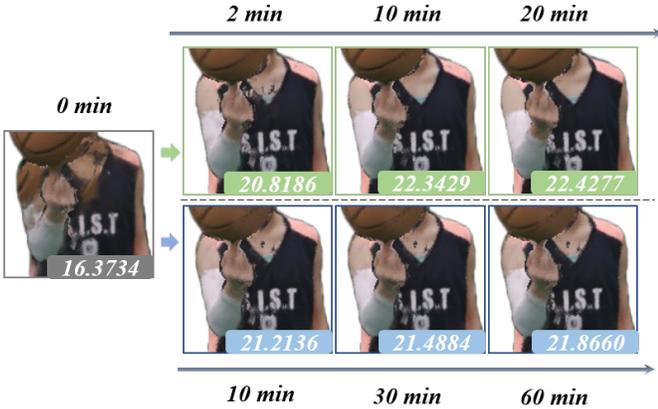


Figure 7: Evaluation of Trajectory-based Refinement. Our trajectory-based scheme (top) achieves more efficient and effective refinement than the brute force one (bottom). Note that our results with in 20 minutes are better than the results of brute force scheme in 60 minutes. PSNR of each cropped image is attached in the button of sub-figure.

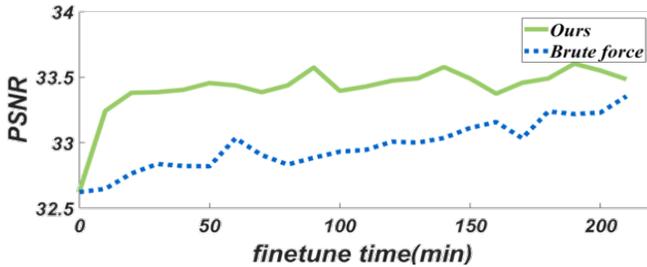


Figure 8: Quantitative evaluation of the refinement stage in terms of refining time.

Ablation study of refinement network. Here we evaluate the design of our refinement network. Let *w/o SFS*, *w/o SFS*, *w/o semantics*, *w/o multi-view*, *w/o temporal* denote our variations without using the SFS proxy, the semantics feature aggregation, the spatial multi-view supervision and the temporal supervision, respectively. Tab. 3 show that our full refinement scheme consistently outperforms other variations in terms of all the metrics (see supplementary material for qualitative comparison). These evaluations not only highlights the contribution of each algorithmic component but also illustrates that our refinement enables photo-realistic human free-viewpoint rendering for bullet-time design.

6.3 Limitation and Discussion

As a trial to provide an interactive bullet-time generator for unseen scenes, our iButter is subject to some limitations. First, our approach still relies on about 80 cameras to provide a 360-degree free-view rendering. It’s promising to reduce the camera number using human priors and even enabling in-the-wild capture with more data-driven scene modeling strategies. Secondly, there still exists some flicking artifacts near the boundary regions due to the challenging viewing directions and some complex self occlusion areas. This could be alleviated in the future by assigning weights of source views using the visibility information from SFS prior. Currently, our iButter

# Sample points	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	rendering time (s) \downarrow
5	34.09	.9800	.0328	.1293
10	34.61	.9822	.0263	.1529
20	34.91	.9833	.0241	.2429
40	34.90	.9828	.0248	.3258
60	35.01	.9832	.0239	.9423

Table 2: Quantitative evaluation on the number of the sample points in terms of various metrics.

	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
Ours	32.28	.9716	.0659
w/o SFS	26.62	.9259	.0934
w/o semantic	28.38	.9413	.0821
w/o multiview	27.42	.9404	.0898
w/o temporal	27.82	.9424	.0895

Table 3: Quantitative ablation study of refinement network.

achieves interactive dynamic 3D human preview in around 0.15 seconds. Speeding up our rendering for more fluent interaction is the direction we need to explore. Furthermore, our approach relies on a consistent lighting assumption and it’s promising to handle complex lighting conditions for view-dependent rendering. It’s also interesting to model the changes between the adjacent frames, so as to enable motion interpolation effects for our users.

7 CONCLUSION

We have presented iButter, a neural interactive bullet-time generator for photo-realistic human free-viewpoint rendering from dense RGB streams, which enables human bullet-time visual effects design in a flexible and interactive manner unseen before. Our real-time preview and design stage enables real-time and dynamic NeRF rendering of novel human activities without per-scene training, with the aid of a hybrid training set, light-weight network design and an efficient silhouette-based sampling strategy. Our efficient trajectory-aware refinement stage further enables photo-realistic bullet-time viewing experience of human activities by jointly utilizing the spatial, temporal consistency and semantic cues along the designed trajectory. Extensive experimental results demonstrate the effectiveness of our approach for convenient interactive bullet-time design and photo-realistic human free-viewpoint video generation, which compares favorably to the state-of-the-art. We believe that our approach renews the bullet-time design for human free-viewpoint videos generation with more flexible and interactive design ability. It serves as a critical step for interactive novel view synthesis, with many potential applications for fancy visual effects in VR/AR, gaming, filming or entertainment.

8 ACKNOWLEDGMENTS

This work was supported by NSFC programs (61976138, 61977047), the National Key Research and Development Program (2018YFB2100500), STCSM (2015F0203-000-06), SHMEC (2019-01-07-00-01-E00003) and Shanghai YangFan Program (21YF1429500).

REFERENCES

- [1] 2019. Agisoft photoscan professional. <https://www.agisoft.com/downloads/installer>.
- [2] Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew Duvall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. 2020. Immersive light field video with a layered mesh representation. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 86–1.
- [3] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. 2001. Unstructured lumigraph rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 425–432.
- [4] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. 2001. Unstructured Lumigraph Rendering. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. Association for Computing Machinery, New York, NY, USA, 425–432. <https://doi.org/10.1145/383259.383309>
- [5] Joel Carranza, Christian Theobalt, Marcus A Magnor, and Hans-Peter Seidel. 2003. Free-viewpoint video of human actors. *ACM transactions on graphics (TOG)* 22, 3 (2003), 569–577.
- [6] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. 2021. MVNeRF: Fast Generalizable Radiance Field Reconstruction from Multi-View Stereo. [arXiv:2103.15595 \[cs.CV\]](https://arxiv.org/abs/2103.15595)
- [7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence* 40, 4 (2017), 834–848.
- [8] Shenchang Eric Chen and Lance Williams. 1993. View interpolation for image synthesis. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. 279–288.
- [9] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. 2021. Stereo Radiance Fields (SRF): Learning View Synthesis from Sparse Views of Novel Scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.
- [10] Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz. 2019. Extreme view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7781–7790.
- [11] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. 2015. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 69.
- [12] Valentin Gabeur, Jean-Sébastien Franco, Xavier Martin, Cordelia Schmid, and Gregory Rogez. 2019. Moulding humans: Non-parametric 3d human shape estimation from single images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2232–2241.
- [13] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. 2021. FastNeRF: High-Fidelity Neural Rendering at 200FPS. [arXiv preprint arXiv:2103.10380 \(2021\)](https://arxiv.org/abs/2103.10380).
- [14] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. 1996. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 43–54.
- [15] Kaiwen Guo, Feng Xu, Tao Yu, Xiaoyang Liu, Qionghai Dai, and Yebin Liu. 2017. Real-time geometry, albedo, and motion reconstruction using a single rgb-d camera. *ACM Transactions on Graphics (ToG)* 36, 4 (2017), 1.
- [16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [17] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. 2018. Deep Blending for Free-Viewpoint Image-Based Rendering. *ACM Trans. Graph.* 37, 6, Article 257 (Dec. 2018), 15 pages. <https://doi.org/10.1145/3272127.3275084>
- [18] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. 2021. Baking Neural Radiance Fields for Real-Time View Synthesis. [arXiv \(2021\)](https://arxiv.org/abs/2021.08.01).
- [19] T. Kanade, P. Rander, and P. J. Narayanan. 1997. Virtualized reality: constructing virtual worlds from real scenes. *IEEE MultiMedia* 4, 1 (1997), 34–47. <https://doi.org/10.1109/93.580394>
- [20] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. 2018. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7122–7131.
- [21] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. 2019. Convolutional mesh regression for single-image human shape reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4501–4510.
- [22] Marc Levoy and Pat Hanrahan. 1996. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 31–42.
- [23] Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Hyeonwoo Kim, Florian Bernard, Marc Habermann, Wenping Wang, and Christian Theobalt. 2018. Neural Rendering and Reenactment of Human Actor Videos. [arXiv:arXiv:1809.03658](https://arxiv.org/abs/1809.03658)
- [24] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. 2019. Neural Volumes: Learning Dynamic Renderable Volumes from Images. *ACM Trans. Graph.* 38, 4, Article 65 (July 2019), 14 pages. <https://doi.org/10.1145/3306346.3323020>
- [25] Haimin Luo, Anpei Chen, Qixuan Zhang, Bai Pang, Minye Wu, Lan Xu, and Jingyi Yu. 2021. Convolutional Neural Opacity Radiance Fields. [arXiv:2104.01772 \[cs.CV\]](https://arxiv.org/abs/2104.01772)
- [26] Wojciech Matusik, Chris Buehler, Ramesh Raskar, Steven J Gortler, and Leonard McMillan. 2000. Image-based visual hulls. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 369–374.
- [27] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. 2019. Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines. *ACM Transactions on Graphics (TOG)* (2019).
- [28] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2020. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*. Springer, 405–421.
- [29] Armin Mustafa, Hansung Kim, Jean-Yves Guillemaut, and Adrian Hilton. 2016. Temporally Coherent 4D Reconstruction of Complex Dynamic Scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [30] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Chakravarty R. Alla Chaitanya, Anton Kaplanyan, and Markus Steinberger. 2021. DONeRF: Towards Real-Time Rendering of Neural Radiance Fields using Depth Oracle Networks. [arXiv:2103.03231 \[cs.CV\]](https://arxiv.org/abs/2103.03231)
- [31] Richard A. Newcombe, Dieter Fox, and Steven M. Seitz. 2015. DynamicFusion: Reconstruction and Tracking of Non-Rigid Scenes in Real-Time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- [32] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter Gehler, and Bernt Schiele. 2018. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *2018 international conference on 3D vision (3DV)*. IEEE, 484–494.
- [33] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin Brualla. 2020. Deformable Neural Radiance Fields. [arXiv preprint arXiv:2011.12948 \(2020\)](https://arxiv.org/abs/2011.12948).
- [34] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. 2020. D-NeRF: Neural Radiance Fields for Dynamic Scenes. [arXiv preprint arXiv:2011.13961 \(2020\)](https://arxiv.org/abs/2011.13961).
- [35] Daniel Rebain, Wei Jiang, Soroosh Yazdani, Ke Li, Kwang Moo Yi, and Andrea Tagliasacchi. 2020. DeRF: Decomposed Radiance Fields. [arXiv preprint arXiv:2011.12490 \(2020\)](https://arxiv.org/abs/2011.12490).
- [36] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. 2021. KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs. [arXiv preprint arXiv:2103.13744 \(2021\)](https://arxiv.org/abs/2103.13744).
- [37] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. 2019. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2304–2314.
- [38] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. 2020. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 84–93.
- [39] Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. 2019. Pushing the boundaries of view extrapolation with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 175–184.
- [40] Zhuo Su, Lan Xu, Zerong Zheng, Tao Yu, Yebin Liu, and Lu Fang. 2020. RobustFusion: Human Volumetric Capture with Data-Driven Visual Cues Using a RGBD Camera. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 246–264.
- [41] Xin Suo, Yuheng Jiang, Pei Lin, Yingliang Zhang, Minye Wu, Kaiwen Guo, and Lan Xu. 2021. NeuralHumanFVV: Real-Time Neural Volumetric Human Performance Rendering using RGB Cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.
- [42] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. 2020. Non-Rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Dynamic Scene From Monocular Video. [arXiv:2012.12247 \[cs.CV\]](https://arxiv.org/abs/2012.12247)
- [43] twindom [n.d.]. Twindom Dataset. <https://web.twindom.com/>.
- [44] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2016. Instance normalization: The missing ingredient for fast stylization. [arXiv preprint arXiv:1607.08022 \(2016\)](https://arxiv.org/abs/1607.08022).
- [45] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. 2021. IBRNet: Learning Non-View Image-Based Rendering. In *CVPR*.
- [46] Minye Wu, Yuehao Wang, Qiang Hu, and Jingyi Yu. 2020. Multi-View Neural Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision*

and Pattern Recognition (CVPR).

- [47] Lan Xu, Wei Cheng, Kaiwen Guo, Lei Han, Yebin Liu, and Lu Fang. 2019. Flyfusion: Realtime dynamic scene reconstruction using a flying depth camera. *IEEE transactions on visualization and computer graphics* 27, 1 (2019), 68–82.
- [48] Lan Xu, Zhuo Su, Lei Han, Tao Yu, Yebin Liu, and Lu Fang. 2019. UnstructuredFusion: realtime 4D geometry and texture reconstruction using commercial RGBD cameras. *IEEE transactions on pattern analysis and machine intelligence* 42, 10 (2019), 2508–2522.
- [49] Lan Xu, Zhuo Su, Lei Han, Tao Yu, Yebin Liu, and Lu FANG. 2019. UnstructuredFusion: Realtime 4D Geometry and Texture Reconstruction using CommercialRGBD Cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019), 1–1.
- [50] Sofiane Yous, Hamid Laga, Masatsugu Kidode, and Kunihiko Chihara. 2007. Gpu-based shape from silhouettes. In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*. 71–77.
- [51] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. 2021. PlenOctrees for Real-time Rendering of Neural Radiance Fields. In *arXiv*.
- [52] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. 2020. pixelNeRF: Neural Radiance Fields from One or Few Images. <https://arxiv.org/abs/2012.02190> (2020).
- [53] Tao Yu, Zerong Zheng, Kaiwen Guo, Pengpeng Liu, Qionghai Dai, and Yebin Liu. 2021. Function4D: Real-time Human Volumetric Capture from Very Sparse Consumer RGBD Sensors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR2021)*.
- [54] Tao Yu, Zerong Zheng, Kaiwen Guo, Jianhui Zhao, Qionghai Dai, Hao Li, Gerard Pons-Moll, and Yebin Liu. 2018. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7287–7296.
- [55] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 586–595. <https://doi.org/10.1109/CVPR.2018.00068>
- [56] Zerong Zheng, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu. 2019. Deephuman: 3d human reconstruction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7739–7749.
- [57] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612. <https://doi.org/10.1109/TIP.2003.819861>
- [58] C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. 2004. High-quality video view interpolation using a layered representation. *ACM transactions on graphics (TOG)* 23, 3 (2004), 600–608.

9 APPENDIX

A FEATURE EXTRACTION NETWORK ARCHITECTURE

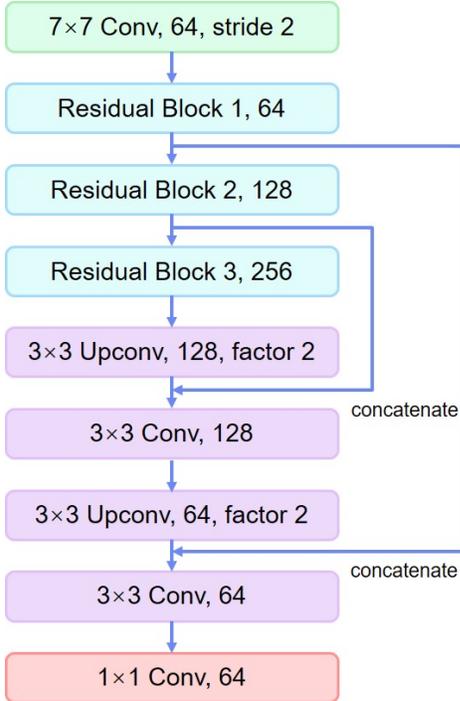


Figure 9: Light-weight Feature Extraction Network Architecture.

Fig. 9 shows an overview of our light-weight feature extractor network architecture, which is modified from [16] and [45]. Our network takes the source view as inputs and extracts their features using shared weights. “Conv” contains convolution, ReLU and Instance Normalization [44] layers. “Upconv” contains a bilinear upsampling with the factor and a “Conv” layer with stride 1. The detailed architecture of Residual Block is shown in Fig. 10. Compared to ResNet34 [16], we reduce convolution layers to speed up without losing performance in novel view synthesis.

B ADDITIONAL RESULTS

B.1 Comparison

For the adopted implementations, in AGI we adopt the commercial Agisoft Metashape Professional software while in IBR we reproduce the traditional paper called Unstructured Lumigraph Rendering using the same SFS geometry proxy. We utilize the released official implementations for both NHR and IBRNet, and provide our data to the authors of NeuralHumanFVV for both training and testing.

For the real-time performance of iButter, we render 200 512*384 images on an Nvidia GeForce RTX3090 GPU in average as shown in Tab. 4. Compared to the non-real-time methods (NeRF [28], IBRNet [45], NHR [46]), we can render at an interactive rate without per-scene training. Compared to those real-time methods (AGI [1] built textured mesh, IBR [4], NeuralHumanFVV [41]), we can provide

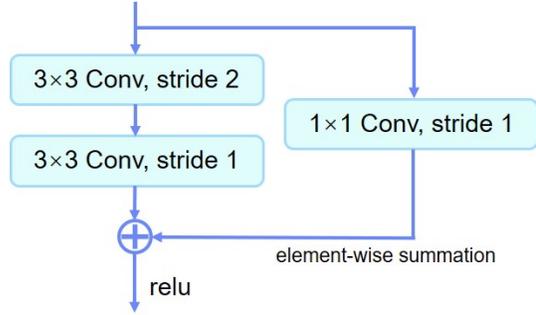


Figure 10: Architecture for the Residue Block.

Method	Ours	NeRF	IBRNet	NHR	AGI	IBR	FVV
time(s)	.1529s	26.78s	21.33s	.3217s	.0071s	.0253s	.0462s

Table 4: Quantitative rendering time comparison against various methods.

# Variations	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	rendering time (s) \downarrow
(a)	34.03	.9745	.0404	.1237
(b)	34.61	.9822	.0263	.1529
(c)	34.70	.9829	.0257	.1847
(d)	35.82	.9831	.0249	.2331

Table 5: Quantitative evaluation on the number of the sample points in terms of various metrics. (a) using two residual blocks and one ‘Upconv’ layer; (b) ours (three residual block); (c) using six residual blocks; (d) using nine residual blocks.

photo-realistic and dynamic FVV preview without tedious per-scene training or reconstruction.

B.2 Evaluation

Number of sample points and feature extraction network.

We provide qualitative evaluation of our models for using different number of sample points per ray in Fig. 11. We provide qualitative and quantitative evaluation of our feature extraction network in Fig. 12 and Tab. 5.

Ablation study of refinement network. We provide qualitative evaluation of our refinement network in Fig. 13.

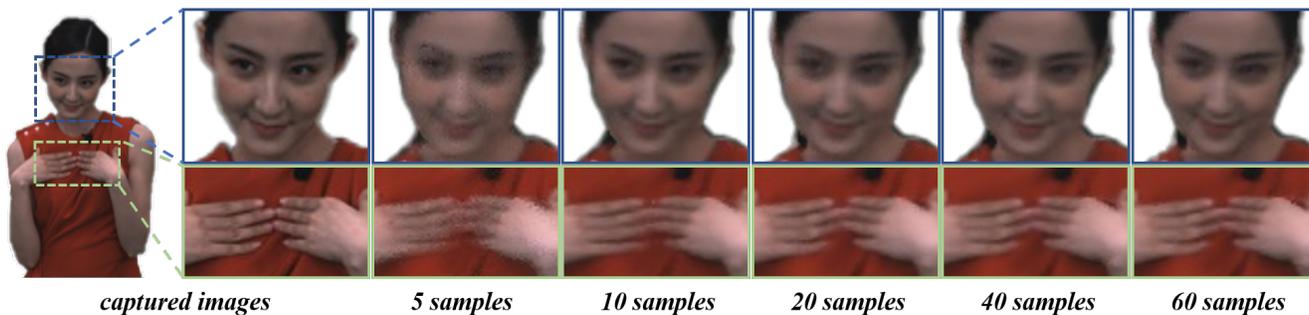


Figure 11: Evaluation of number of sample points in our interactive renderer. Blur and artifacts will appear when only 5 points are sampled. When the number of sampling points exceeds 10, the performance of image does not increase much.

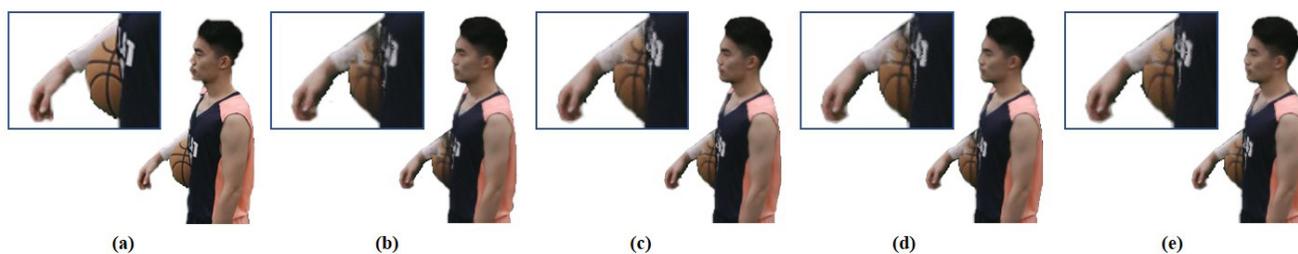


Figure 12: Qualitative ablation study of feature extraction network. (a) ground truth; (b) using two residual blocks and one 'Upconv' layer; (c) ours (three residual block); (d) using six residual blocks; (e) using nine residual blocks.

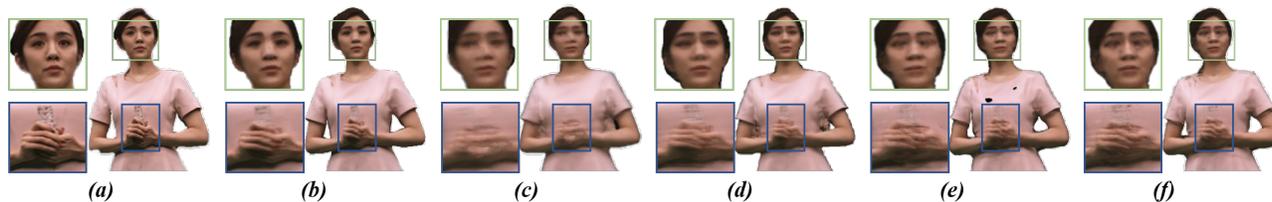


Figure 13: Qualitative ablation study of refinement network. (a) ground truth; (b) ours; (c) w/o SFS prior; (d) w/o semantic feature aggregation; (e) w/o multi-view consistency refinement; (f) w/o temporal consistency refinement.