https://doi.org/10.2352/EI.2025.37.8.IMAGE-274
© 2025, Society for Imaging Science and Technology

RGBD Routed Blending: A 3D Reconstruction Pipeline For Video Conferencing

Fan Bu[1]***, Qian Lin***[2]***, and Jan Allebach***[1]
[1]***School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN***
[2]***HP Labs, HP Inc., Palo Alto, CA***

Abstract

With the widespread use of video conferencing technology for remote communication in the workforce, there is an increasing demand for face-to-face communication between the two parties. To solve the problem of difficulty in acquiring frontal face images, multiple RGB-D cameras have been used to capture and render the frontal faces of target objects. However, the noise of depth cameras can lead to geometry and color errors in the reconstructed 3D surfaces. In this paper, we proposed RGBD Routed Blending, a novel two-stage pipeline for video conferencing that fuses multiple noisy RGB-D images in 3D space and renders virtual color and depth output images from a new camera viewpoint. The first stage is the geometry fusion stage consisting of an RGBD Routing Network followed by a Depth Integrating Network to fuse the RGB-D input images to a 3D volumetric geometry. As an intermediate product, this fused geometry is sent to the second stage, the color blending stage, along with the input color images to render a new color image from the target viewpoint. We quantitatively evaluate our method on two datasets, a synthetic dataset (DeformingThings4D) and a newly collected real dataset, and show that our proposed method outperforms the state-of-the-art baseline methods in both geometry accuracy and color quality.

Introduction

Since the Covid-19 pandemic, video conferencing has seen a rise in usage for remote communication in the workforce. Typically, participants use electronic devices with a single RGB camera, which, ideally, captures a frontal view of the user's face when positioned directly in front. However, due to disparities in camera and screen positions, the visual quality is often compromised. Instances where users look away from the screen or their body postures face away from the camera hinder face-to-face interaction between participants [1].

Some image processing techniques have been developed to adjust the viewing angle based on RGB camera inputs [2, 3, 4, 5, 6]. This approach requires only commercial RGB cameras, but it either incurs high computational costs or lacks consistent frontal views across video frames.

Other researchers use RGB-D cameras to track and fuse the 3D surface of target objects dynamically [7, 8, 9, 10, 11, 12, 11, 13, 14]. These algorithms update a canonical model to retain temporal information, which is a time-consuming matching process for dynamic objects and thus hinder their application in high-quality real-time video conferencing.

Another solution is to use multiple RGB-D cameras and fuse their inputs in 3D space [15, 16, 17, 18, 19, 20, 21, 22, 23, 24]. This type of approach treats the object being rendered as a static object and captures the RGB-D data from multiple camera perspectives at the same time. As an output, a color image is rendered from the target camera viewpoint. Such a scheme requires fewer computational resources but is usually affected by noise from the depth inputs, leading to 3D geometry errors.

To help reduce such errors, we propose RGBD Routed Blending, a novel two-stage pipeline that fuses multiple noisy RGB-D images in 3D space and renders virtual color and depth output images from a new camera viewpoint. We envision our work leading toward a real-time solution to 3D fusion and rendering tasks where the depth sensor contains various types of noise, and the camera locations are not free to move. The primary contributions of our work are summarized as follows:

- We propose a novel deep-learning-based end-to-end two-stage pipeline that reads RGB-D images from multiple cameras as input and renders an RGB-D image pair from a target viewpoint as output.
- We propose RGBD Routing Network, an RGBD-image-based depth map denoising approach inspired by [22], to predict a denoised depth image and its associated confidence map for each RGB-D input. This improves the robustness of the denoising performance under various types of depth sensor noise.
- We quantitatively evaluate our method on a synthetic dataset [25] with 3D humanoid animations and a self-collected real dataset using four RGB-D cameras for video conferencing image quality evaluation.
- We utilize blending weights in the color rendering process to fill in the vacant pixels that the cameras cannot cover, improving the overall visual quality of the output color image.

Related Works

Facial Synthesis

Some methods work from the original camera viewpoint and synthesize facial images directly from color inputs and unable to convert side face images to frontal face images [4, 26, 3]. Others are subject-dependent and require pre-training for specific subjects [5, 27]. While some algorithms don't require prior training of target subjects [28, 29, 30, 31, 32, 33, 34, 35, 36, 37], they face challenges in balancing rendering quality and flexible camera viewpoints.

In contrast, 3D model-based approaches capture target faces with RGB or RGB-D cameras to generate 3D models [38, 39, 40, 41, 42]. These models allow rendering faces from any viewpoint, even in real-time applications. However, recent solutions aim to avoid pre-building 3D models. They correspond 2D fea-

ture points to 3D space and locate these points within the pipeline [6, 31, 43, 44, 45]. Nevertheless, these approaches often produce distorted images of clothing, glasses, and hair. While a few solutions achieve better image quality, they lack consistency across sequential video frames or fail to meet real-time requirements for video conferencing.

Single-camera Dynamic Reconstruction

A series of solutions employ RGB-D cameras as input devices. Since a single RGB-D frame still lacks 3D information from multiple views. DynamicFusion [12] proposed a pipeline that tracks the target object using the temporal information from the RGB-D camera and inpainting the invisible parts when rendering the image output. Many studies [7, 8, 9, 10, 11, 12, 11, 13, 14] have leveraged similar schemes in which RGB-D cameras continuously collect data to maintain a 3D canonical model in the back-end related to the geometry of the dynamic object.

This solution offers the advantage of requiring less hardware. However, it comes with a drawback of high computational complexity due to maintaining a canonical model with 3D grids or feature points in the back-end. As a result, current dynamic reconstruction methods are not yet suitable for real-time video conferencing applications.

Multi-camera Static Reconstruction

As mentioned in previous sections, acquiring information from a single camera is limited, and manipulating temporal information increases computational complexity. Researchers have addressed these challenges by obtaining 3D surface information from multiple camera angles [46, 47]. Various approaches have been used, such as customized hardware setups with mirrors and projectors [48], or using multiple depth cameras for simultaneous RGB-D data collection [49, 18, 24]. These methods leverage synchronized cameras to measure the video conferencing participant as a static object, eliminating the need for processing temporal signals.

Before rendering the output color image, depth information of the target object is fused using multi-view 3D reconstruction algorithms. Traditional techniques merge point clouds or transfer depth maps into 3D volumes for fusion [15, 50, 51, 19], with Truncated Signed-Distance Function (TSDF) being a commonly used method [15].

However, these algorithms are prone to noise and outliers, leading to 3D errors in the fused geometry. To improve accuracy, online optimization [52, 53, 54] and deep neural networks [55, 56, 23, 22] have been employed. For example, 3DMV [23] uses TSDF fusion and a 3D neural network for segmentation. Octnet [57] and OctnetFusion [58] employ octree structures and deep 3D convolutional networks. RayNet [59] incorporates perspective projection and occlusion physics. SurfaceNet [60] converts stereo images to 3D voxels and uses a computationally expensive deep neural network for fusion. RoutedFusion [22] accelerates processing with two sub-networks, Depth Routing Network and Depth Fusion Network. It takes the depth maps as input and leverages the Depth Routing Network for denoising, followed by the Depth Fusion Network to fuse a TSDF volume without color information.

Another series of 3D reconstruction algorithms is represented by NeRF [61], which uses a deep Multi-Layer Perceptron (MLP) to represent the scene's geometry and appearance by fitting a neural radiance field to RGB images. After the training, it can synthesize novel views of a 3D scene with high quality. The primary drawbacks of NeRF [61] are its long training and rendering times. Rendering must happen in real-time for interactive applications, making its slow rendering speed a significant limitation. Many subsequent studies have attempted to accelerate the training and rendering processes. KiloNeRF [62] divides the scene into thousands of smaller MLPs and significantly speeds up the rendering process and can render an image in approximately 22 milliseconds. InstantNGP [63] introduces multiresolution hash table that allows the use of a smaller network without sacrificing quality, and achieves training of high-quality neural graphics primitives in a matter of seconds. However these algorithms are designed for static objects, making them ill-suited for dynamic environments like video conferences where each moment introduces a new scene. Even if each scene only requires a few seconds for training, it's still challenging to achieve real-time 3D reconstruction and rendering. By the time of submission, some studies, such as Tensor4D [64], HumanRF [65], and HexPlane [66] attempt to decompose voxels with temporal information from 4D to 2D to enhance the algorithm's computational speed in dynamic scenes. Although achieving real-time performance and high resolution simultaneously remains a challenge with this approach, it still represents a promising research direction for the future.

Method

As described in Section , the present state-of-the-art 3D fusion algorithms take into account one or two of the following three technical requirements: real-time performance, anti-noise robustness, and color quality. To achieve all three credentials described above in video conferencing applications, we propose an end-to-end two-stage 3D surface fusion and coloring algorithm called RGBD Routed Blending. Fig. 1 shows the overall flowchart of our proposed method.

The input of our method is the RGB-D images of a video conferencing participant from multiple pre-calibrated and pre-synchronized cameras. Therefore, the target participant can be treated as a static object during the processing. In the first stage, the geometry fusion stage, we propose an RGBD Routing Network followed by a Depth Integrating Network to fuse the RGB-D input images to a 3D volumetric geometry. As an intermediate product, this fused geometry is sent to the second stage, the color blending stage, along with the input color images to render a new color image from the target viewpoint. Sections and describe the two stages in detail. In a real-world videoconferencing environment, a depth image is not necessary, but generating a depth image can help us better measure the geometric accuracy of the algorithm. Consequently, we choose to render an RGB-D image pair with a frontal view of the video conferencing participant as the final output of our algorithm.

Geometry Fusion

The goal of geometry fusion is to construct the 3D information of an object by fusing the RGB-D camera inputs. Inspired by RoutedFusion [22], our geometry fusion stage consists of two connected deep neural networks to better deal with the noise carried in the depth images. We name our geometry fusion stage as RGBDRoutedFusion. The first neural network is the RGBD Rout-

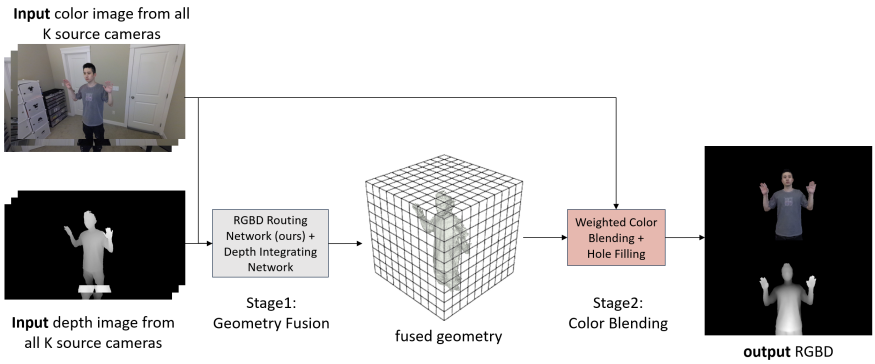


Figure 1. *The overall flowchart of our proposed RGBD Routed Blending algorithm based on RGB-D camera images.*

ing Network, followed by the second neural network the Depth Integrating Network as shown in Fig. 2.

For the RoutedFusion [22] algorithm, it focuses solely on depth images, omitting color information. Employing the original Depth Routing Network from the RoutedFusion [22] approach would segregate depth and color processing, resulting in discrepancies between color and object contours in the final renderings. Such mismatches often manifest in noisy regions of depth images and along object boundaries. In Section , we expanded the input format to RGB-D image pairs, thereby integrating both depth and RGB data. By leveraging the inherent clarity of color images, we aim to mitigate the noise present in depth images.

RGBD Routing Network

In Fig. 2, the RGBD Routing Network reduces input image depth noise by processing color and depth information. It generates a noise-reduced depth map and confidence map using a U-Net [67] encoder-decoder model. The network includes convolutional layers, activation functions, max-pooling layers, and upsampling layers. We removed normalization layers to mitigate depth-dependent bias, following RoutedFusion [22].

For all K source cameras, $K = 3$ in our experiment, each RGB-D image from the k^{th} camera is processed by the RGBD Routing Network separately. The input tenor $D^{RGBD}_{k=1,2,...,K} \in \mathbb{R}^{\text{width}\times\text{height}\times 4}$ is a 4-channel RGB-D image, and the two output tensors are the denoised depth map $\hat{D}_k$ and the corresponding confidence map C_k. Both output tensors have a channel size of 1 and will be used in the later Depth Integrating Network for geometry fusion.

During training, the RGBD Routing Network and the Depth Integrating Network are trained separately. The loss function of the RGBD Routing Network consists of depth prediction loss and confidence loss [22]. The depth prediction loss can be obtained by adding the L1 loss of the predicted depth value and the ground-truth depth value, and the L1 loss of the predicted depth map gradient and the ground-truth depth map gradient [68]. The confidence value plays a weighting role in the loss function so that it can be trained in an unsupervised manner [69]. We expect that the higher the confidence, the lower the overall loss. Therefore, the final loss function L_{Routing} for the RGBD Routing Network is:

$$L_{\text{Routing}} = \sum_i \left\{ c_i L_1(y_i, \hat{y}_i) + c_i L_1(\bigtriangledown y_i, \bigtriangledown \hat{y}_i) + \lambda \log c_i \right\} \quad (1)$$

where i is the index of pixels, y is the predicted denoised depth map, $\hat{y}$ is the ground-truth no-noise depth map, c is the confidence map, $\bigtriangledown$ is the gradient, and λ is an empirical hyperparameter, which we set to 0.015, following [22].

Depth Integrating Network

In Fig. 2, the Depth Integrating Network fuses denoised depth images to create a 3D TSDF volume with preserved TSDF values V and weights W. We leverage the same architecture as RoutedFusion [22]. For detailed information on the network structure and hyper-parameters, please refer to Appendix **??**. We confirmed that joint training doesn't enhance overall performance, as claimed in RoutedFusion [22]. Hence, the Depth Integrating Network is independently trained without the RGBD Routing Network. A distinction between our method and RoutedFusion lies in the optimal setting for the hyper-parameter S. While RoutedFusion suggests that S is best set to 5, our findings indicate that its optimal value fluctuates depending on the dataset. As detailed in Section , we adopt $S = 9$ for DeformingThings4D [25] dataset and $S = 5$ for our real dataset.

Color Blending

Weighted Color Blending

As described in Section , our proposed algorithm has obtained a TSDF volume based on the RGB-D image inputs $D^{\text{RGBD}}_{k=1,2,...,K}$ from all the K source cameras. The fused surface can be calculated by searching for the zero level set [70] among the final TSDF values V^*_K.

To take advantage of the high-resolution color images from each camera, we, inspired by Buehler [71], executed a Weighted Color Blending scheme that projects color pixels onto the fused 3D surface and then combines them according to their blending weights $w_{s,k}$. More details are introduced in Appendix **??**. When pixels from multiple cameras fall at the same point p, the final color Ω^*_p is the weighted average of these pixel colors $\Omega_{p,k}$:

$$\Omega^*_p = \frac{\sum_k \Omega_{p,k} \cdot w_{p,k}}{\sum_k w_{p,k}} \quad (2)$$

Fig. 3 shows an example of the contribution of the three different source cameras to the final rendered image result. Fig. 3 (a, c, e) are the results of projecting the color pixels from each camera onto the 3D surface. Fig. 3 (b, d, f) shows the weights corresponding to each pixel of the images in Fig. 3 (a, c, e), with brighter colors representing higher weights and vice versa. Fig.

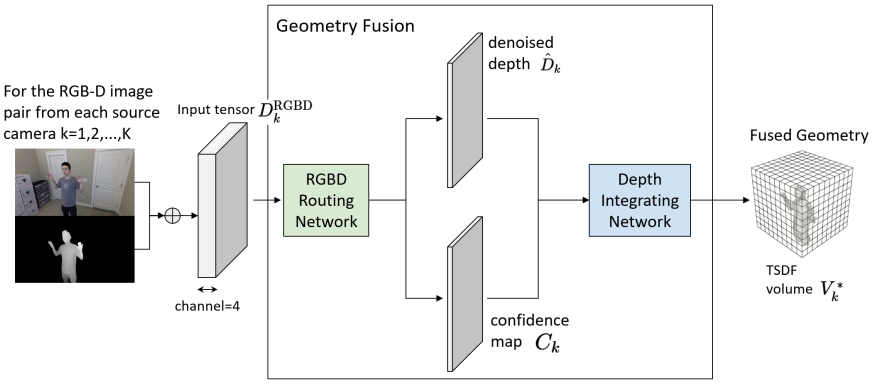


Figure 2. *The flowchart of the RGBDRoutedFusion network that we propose in the geometry fusion stage.*

3 (g) is the weighted average RGB image Ω^* produced by the example color pixels and their blending weights. There will still be some tiny spots on the 3D surface that are not covered by any source cameras. From the perspective of the target camera, these tiny spots look like blank holes, as shown in Fig. 3 (g).

Hole Filling

Denote Ω^* as the weighted average RGB image produced by the Weighted Color Blending module, as shown in Fig. 3 (g). If a pixel with index h in the image Ω^* is not covered by the color pixels from any of the source cameras, then the pixel Ω_h^* will appear as a black cavity.

Define M^* as the valid mask of image Ω^*, where colors are assigned by the Weighted Color Blending module. Define M_{not}^* as the negative of the mask M^*. Then, M_{not}^* contains all the background pixels and the black cavities. To localize these blank pixels more accurately, we dilate the valid mask M^* of image Ω^* with a step size of 1, and then erode it with a step size of 1. After the erosion, we calculate the intersection of M_{not}^* and the eroded mask, i.e.:

$$M_H = \text{Erode}(\text{Dilate}(M^*)) \bigcap M_{not}^* \tag{3}$$

In this way, we localize M_H, the mask of the black holes, without destroying the contour of M^* so that they can be filled with colors in the next step.

Given the mask of the holes M_H, we fill each pixel h of these cavities H by the weighted average of the colors $\Omega_{\delta h}^*$ and blending weights $W_{\delta h}^*$ of their neighboring pixels δh, i.e.:

$$\Omega_h^* = \frac{\sum_{j \in \delta h} \Omega_{\delta h}^* \cdot w_{\delta h}^*}{\sum_{j \in \delta h} w_{\delta h}^*} \tag{4}$$

where h is the index of each blank pixel within the cavities mask M_H, $\delta h \in M^*$ is the indices of the valid neighboring pixels of h, $W_{\delta h}^*$ is the blending weights of pixels δh provided by the Weighted Color Blending module, and the Ω_h^* is the output color of the holes filled by the Hole Filling module. An example of the final RGB image produced by the Color Blending module is shown in Fig. 3 (h).

Datasets and Setup

Training Datasets

Datasets for the RGBD Routing Network

For the RGBD Routing Network, its training input is required to contain paired RGB-D images, and it is also necessary to include noise-free depth images as ground-truth when calculating the routing loss in Eq. 1. A possible training approach is to adopt a synthetic dataset, such as the SceneNet [72], which provides RGB-D images of various virtual indoor scenes. We simulate noisy depth images input by overlaying depth noise on the ground-truth depth images. In this paper, three different types of noises are created to simulate the possible depth noise produced by various depth sensors. The three types of noise are (1) Depth Value noise; (2) Salt-and-pepper noise that simulates outliers; and (3) Gaussian blur that mimics the sticky pixels caused by miscalibration between camera color and depth sensors.

Datasets for the Depth Integrating Network

The Depth Integrating Network is trained using a dataset containing complete 3D models of various objects. This network is capable of generating TSDF volumes and corresponding depth images from any camera viewpoint. For training purposes, we utilize ShapeNet [73] following the methodology of RoutedFusion [22]. While the Depth Integrating Network does not require color images, the depth images used as input may still contain noise. Therefore, similar to the RGBD Routing Network, we randomly apply one of three simulated noise types to the noise-free ground-truth depth images.

Test Datasets

The test set is crucial for evaluating the rendering quality of the videoconferencing algorithm. It should include RGB-D images of humanoid models from various viewpoints relevant to the videoconferencing environment. Ground-truth images are necessary for accurate numerical computations. To achieve this, we utilize both a synthetic dataset (add our simulated noises), DeformingThings4D [25], and our self-collected real dataset for quantitative evaluation. Figure 4 provides an overview of the camera locations used in these datasets, with additional details provided in Appendix **??**. We believe that the experimental results presented in this paper should also be verifiable in similar datasets, such as the commercial RenderPeople dataset [74].

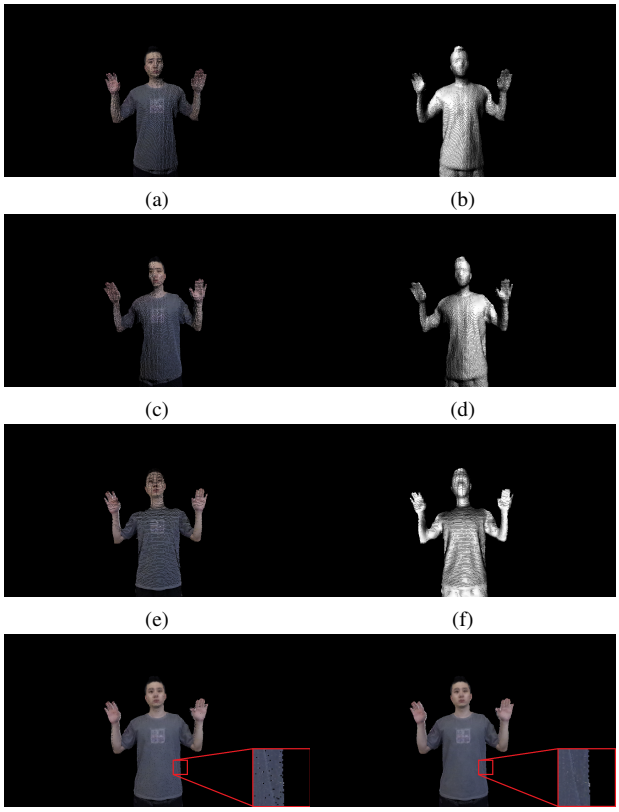


Figure 3. *Example of our proposed Color Blending module. (a, c, e): The projected color pixels from different source cameras; (b, d, f): The blending weights of each pixel with respect to (a), (c), and (e); (g) The weighted average result; (h) The final output color image after hole filling.*

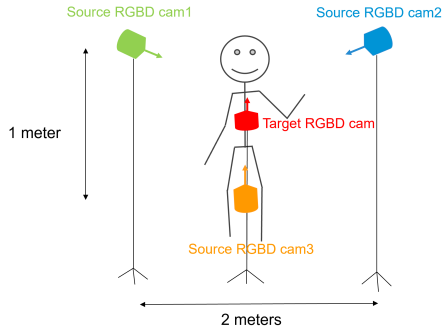


Figure 4. *Sketch of the camera locations when generating synthetic data and collecting real data.*

Experiments

Baseline Methods

Based on their generality and novelty, we have selected these three representative algorithms, Truncated Signed-Distance Function (TSDF) [15], Starline [24], and RoutedFusion [22], as benchmark algorithms. For a fair comparison in the Experiments Section , we choose the voxel size = $\frac{1}{128}$ meter when implementing these algorithms.

For the TSDF and the Starline algorithms, the truncated distance = $\frac{1}{32}$ meter. When comparing RoutedFusion with other methods, since RoutedFusion only utilizes depth images and not color images, we treat it as a substitute for the geometry fusion stage. Unlike TSDF, RoutedFusion has no concept of truncated distance, but a similar term S is used to determine the range of the influence of each pixel on the depth images, and we followed such an idea in our algorithm (Section). The network framework and the pre-trained weights of RoutedFusion are provided by Weder et al. [22], with the voxel size = $\frac{1}{128}$ meter and $S = 9$.

For ablation studies, the Geometry Fusion stage has four options: TSDF, Starline-TSDF (modified TSDF), RoutedFusion, and RGBDRoutedFusion (ours). The Color Blending stage can be divided into two sub-modules: Coloring and Hole Filling. We combine different choices for each submodule to permute nine candidate ablation pipelines.

Method 1 is the original TSDF algorithm [15], which performs geometry fusion without color information. Method 2 adds colors to Method 1 by averaging the colors in each 3D voxel. Method 3 replicates the Starline algorithm [24], incorporating the Starline-modified TSDF module [24] and the Weighted Color Blending module. Method 4 improves Method 3 by adding the Hole Filling module to compare its contribution to the pipeline. Method 5 is RoutedFusion [22], utilizing their pre-trained weights for experimental results, which is also a geometry fusion algorithm based solely on depth images without color information. Method 6 adds the Weighted Color Blending module to Method 5 for coloring the fused surface. Method 7 further incorporates the Hole Filling module to Method 6 to assess its influence on the final results. Method 8 is our experimental method, comprising a geometry fusion stage (RGBDRoutedFusion) and a coloring stage (Weighted Color Blending module), without the Hole Filling module. Method 9 is our proposed final algorithm, which adds the Hole Filling module to Method 8.

Evaluation Metrics

Both the geometry quality and the coloring quality affect the overall quality of the test algorithms. In practical applications, we tend to filter out the background to save computational resources. Therefore, we quantitatively measure the geometry and the coloring quality of pixels related to the foreground portrait. More details are attached in Appendix **??** and **??**.

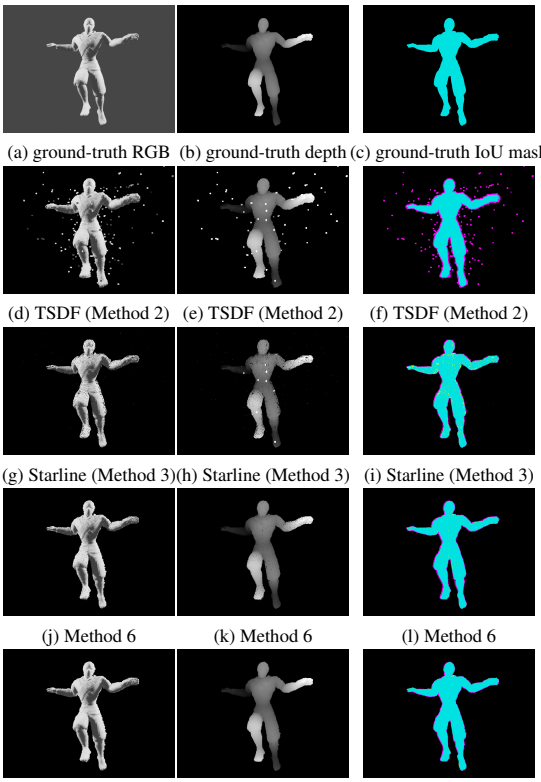
Results on the Synthetic Dataset

We compare our proposed algorithm with baseline methods on the simulated noisy test dataset using the introduced measures. The input depth images are generated according to Section , while the RGB images remain unchanged.

We select the TSDF [15] (Method 2), the Starline [24] (Method 3), the RoutedFusion [22] + WeightedColorBlending (Method 6), and our proposed pipeline (Method 9) as the four representative methods and show two representative visual sample results in Figs. 5 and Appendix **??**.

When the input depth image has significant noise or outliers, as shown in Tables 1, our algorithm outperforms the other benchmark algorithms in geometry accuracy.

We then analyze each algorithm from a holistic perspective regarding the color quality of their output RGB images. Regardless of the noise scale, the results presented in all Tables are con-



(a) ground-truth RGB (b) ground-truth depth (c) ground-truth IoU mask

(d) TSDF (Method 2) (e) TSDF (Method 2) (f) TSDF (Method 2)

(g) Starline (Method 3) (h) Starline (Method 3) (i) Starline (Method 3)

(j) Method 6 (k) Method 6 (l) Method 6

(m) Ours (Method 9) (n) Ours (Method 9) (o) Ours (Method 9)

Figure 5. *Results of our method and the baseline methods on the synthetic dataset where the input depth images contain* ***depth and outliers noise****. The left column is the output RGB image of each method. The middle column is the output depth image of each method. The right column is the IoU of each method, where the intersection areas, the missing areas, and the false prediction areas are colored in cyan, yellow, and magenta, respectively.*

sistent, i.e., Starline [24] (Method 3) performs the best within the intersection mask, and our solution performs best within the union mask. Considering a video conference scenario, despite the fact that one can remove the background, the final RGB image presented to the user by each pipeline is directly related to the union mask. Therefore, the measurement result within the union mask is more noteworthy than the measurement within the intersection mask, which indicates that our proposed algorithm outperforms the other algorithms in terms of the rendering result of the output color image.

The conclusions are evident when comparing algorithm results in Fig 5. The figures have three columns: output RGB images, depth images, and predicted masks versus ground-truths. Intersection masks (in cyan) show correct predictions. Pixels in the intersection mask belong to both the ground-truth mask and the predicted mask. Missing pixels (in yellow) are those in the ground-truth mask but missed by the prediction, while false predictions (in magenta) are pixels in the predicted mask but not in the ground-truth mask. From the result images, we observe that depth image noise indeed leads to geometry errors, and deep-learning-based methods generally exhibit better resilience to depth noise compared to traditional computer vision algorithms. By comparing the performance of our algorithm (Method 9) horizontally across all visual results generated by different types

Method Index	Pipeline Modules			Geometry Error			
	Stage1	Stage2		Intersection over Union	Intersection Mask	Union Mask	IoU punished Depth RMSE (mm) ↓
	Geometry Fusion	Coloring	Hole Filling	Segmentation IoU ↑	Depth RMSE (mm) ↓	Depth RMSE (mm) ↓	
1	TSDF [15]	×	×	0.7655	99.3956	1259.5125	129.8512
2	TSDF [15]	Color Averaging	×	0.7655	99.3956	1259.5125	129.8512
3	Starline-TSDF [24]	Weighted Color Blending	×	0.8771	101.2050	867.2225	115.3856
4	Starline-TSDF [24]	Weighted Color Blending	√	0.9015	101.4389	777.1757	112.5272
5	RoutedFusion [22]	×	×	0.8654	32.6371	910.0795	37.7122
6	RoutedFusion [22]	Weighted Color Blending	×	0.8939	32.5036	804.2666	36.3623
7	RoutedFusion [22]	Weighted Color Blending	√	0.8974	32.5272	792.6656	36.2452
8	RGBDRoutedFusion (ours)	Weighted Color Blending	×	0.9249	**27.8211**	682.0188	30.0805
9	**RGBDRoutedFusion (ours)**	**Weighted Color Blending**	√	**0.9331**	27.8302	**644.6320**	**29.8270**

Geometry quality evaluation on the synthetic dataset where the input depth images contain depth and outliers noise following Eq. ??.

Method Index	Color Quality (Intersection Mask)				Color Quality (Union Mask)			
	Y channel of YCrCb		All RGB channels		Y channel of YCrCb		All RGB channels	
	PSNR ↑	SSIM ↑	LPIPS Alex Loss ↓	LPIPS VGG Loss ↓	PSNR ↑	SSIM ↑	LPIPS Alex Loss ↓	LPIPS VGG Loss ↓
1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
2	19.7613	0.9661	0.0310	0.0465	13.0263	0.9141	0.1275	0.0893
3	23.1388	**0.9902**	**0.0073**	**0.0146**	13.9261	0.9346	0.1408	0.0842
4	22.9929	0.9838	0.0134	0.0321	16.5174	0.9613	0.0988	0.0723
5	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
6	23.2471	0.9867	0.0103	0.0207	16.0785	0.9561	0.0654	0.0575
7	23.2190	0.9847	0.0125	0.0266	16.9943	0.9653	0.0495	0.0497
8	**23.3847**	0.9871	0.0092	0.0196	16.6647	0.9611	0.0552	0.0549
9	23.3432	0.9852	0.0112	0.0252	**18.5115**	**0.9737**	**0.0345**	**0.0410**

Color quality evaluation on the synthetic dataset where the input depth images contain depth and outliers noise. Method Index is the same as Table 1.

of noise, we can observe consistent performance across various noise types.

For ablation studies, we investigate the contribution of each sub-module to the overall pipeline. In the geometric fusion stage, our proposed deep-learning-based algorithm RGBDRoutedFusion shows strong robustness among different types of noise. In the coloring stage, since Weighted Color Blending takes full advantage of the fact that the resolution of the color image is higher than the resolution of the 3D geometry, the Weighted Color Blending module outperforms the Color Averaging module in terms of both the geometric accuracy and the color quality. In the Color Blending stage, both Weighted Color Blending and Hole Filling enhance the geometry accuracy and the color quality of the overall pipeline. Detailed comparisons are listed in Appendix **??**.

Results on the Real Dataset

The real dataset was collected with four RGB-D cameras, as stated in Section . Unlike the synthetic dataset, the data collected by the real cameras are not perfect. There are mainly two types of errors in the real camera data. One is the synchronization errors from each camera's color and depth sensors (Appendix Fig. **??**), and the other is the calibration errors of the extrinsic and intrinsic matrices between all camera sensors. The lack of synchronization between the two camera sensors impacts the edge contours, subsequently influencing our IoU calculations. Therefore, the data collected from the target camera position is not strictly accurate and can only be interpreted as reference images, not as ground-truth images.

We present each algorithm's geometry accuracy and color quality on the real dataset in Tables 3 and 4. Similar to Fig. 5, we show the final rendering results in Fig 6.

By comparing Table 3 with Table 1, the geometric accuracy of each method on the real dataset slightly differs from their eval-

uation results on the synthetic dataset. The real dataset is neither noise-free nor heavy-noise but a light-noise dataset. In this case, Method 4 and Method 9 jointly obtained the best performance. Considering that the scores of these two schemes are close, it is challenging to strictly deduce which of these two methods has the absolute advantage in geometric accuracy. Each of them has its own merits. Note that the dataset reference images are subject to data-collecting errors. Therefore, the evaluation of the geometry quality should only serve as a reference rather than a strict groundtruth.

The color quality of each method is evaluated in Table 4. Starline [24] (Method 3) outperforms other methods within the intersection mask, and ours (Method 9) tops within the union mask, which agrees with the conclusion in Table 2 for the synthetic dataset. Given that when presenting final results to users, the color quality over the union mask is of greater importance than the color quality over the intersection mask, our algorithm performs relatively better in this regard.

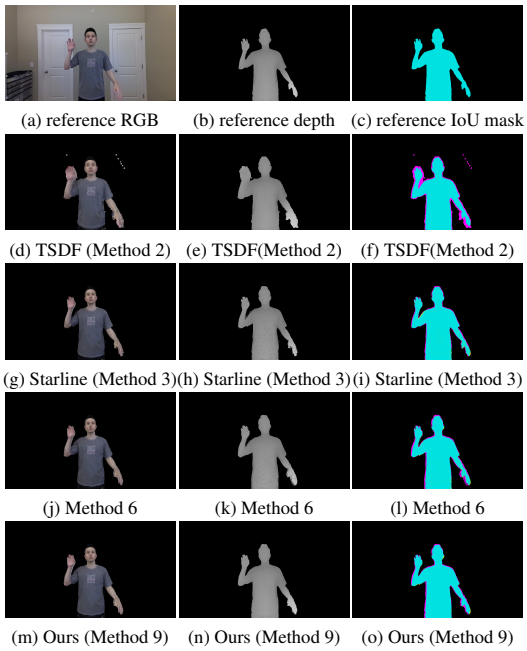


Figure 6. *Results of our method and the baseline methods on our collected real dataset. The left column is the output RGB image of each method. The middle column is the output depth image of each method. The right column is the IoU of each method, where the intersection areas, the missing areas, and the false prediction areas are colored in cyan, yellow, and magenta, respectively.*

Conclusion

We propose RGBD Routed Blending, a novel two-stage pipeline for video conferencing. It fuses multiple noisy RGB-D images in 3D space and renders virtual color and depth output images from a new camera viewpoint. We evaluate our method on both synthetic and real datasets, demonstrating its superior performance compared to state-of-the-art baseline methods in terms of geometry accuracy and color quality. By proposing the RGB-DRoutedFusion network (Section), a subnet in the geometry fusion stage of our pipeline, we demonstrate that our deep neural network with RGB-D input has better 3D reconstruction performance than the network leveraging depth images only. Our method also improves the color quality of the overall rendering results by filling the cavity pixels with their neighboring information.

Method Index	Pipeline Modules			Geometry Error			
	Stage1	Stage2		Intersection over Union	Intersection Mask	Union Mask	IoU punished Depth RMSE (mm) ↓
	Geometry Fusion	Coloring	Hole Filling	Segmentation IoU ↑	Depth RMSE (mm) ↓	Depth RMSE (mm) ↓	
1	TSDF [15]	×	×	0.8848	20.0414	402.5280	22.6507
2	TSDF [15]	Color Averaging	×	0.8848	20.0414	402.5280	22.6507
3	Starline-TSDF [24]	Weighted Color Blending	×	0.9163	20.0808	335.0653	21.9140
4	**Starline-TSDF [24]**	**Weighted Color Blending**	√	**0.9304**	20.1618	**304.8974**	21.6697
5	RoutedFusion [22]	×	×	0.8406	31.9583	457.5653	38.0165
6	RoutedFusion [22]	Weighted Color Blending	×	0.8936	31.8170	369.9614	35.6039
7	RoutedFusion [22]	Weighted Color Blending	√	0.9033	31.8051	352.1910	35.2106
8	RGBDRoutedFusion (ours)	Weighted Color Blending	×	0.9144	19.8470	338.3810	21.7048
9	**RGBDRoutedFusion (ours)**	**Weighted Color Blending**	√	0.9285	**19.8294**	309.8048	**21.3560**

Geometry quality evaluation on our collected real dataset.

Method Index	Color Quality (Intersection Mask)				Color Quality (Union Mask)			
	Y channel of YCrCb		All RGB channels		Y channel of YCrCb		All RGB channels	
	PSNR ↑	SSIM ↑	LPIPS Alex Loss ↓	LPIPS VGG Loss ↓	PSNR ↑	SSIM ↑	LPIPS Alex Loss ↓	LPIPS VGG Loss ↓
1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
2	23.2661	0.9732	0.0294	0.0511	21.7625	0.9629	0.0479	0.0632
3	**24.2709**	**0.9780**	**0.0179**	**0.0236**	21.9541	0.9482	0.0727	0.0632
4	24.2625	0.9722	0.0285	0.0405	**22.8712**	0.9616	0.0476	0.0576
5	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
6	23.9192	0.9757	0.0206	0.0270	21.1359	0.9472	0.0726	0.0634
7	23.9174	0.9707	0.0329	0.0427	21.9806	0.9612	0.0448	0.0521
8	24.0535	0.9758	0.0209	0.0275	21.4448	0.9489	0.0710	0.0624
9	24.0524	0.9711	0.0322	0.0426	22.4581	**0.9633**	**0.0435**	**0.0504**

Color quality evaluation on our collected real dataset. Method Index is the same as Table 3.

One drawback we observed in all benchmark algorithms, including ours, is that alignment errors of RGB-D sensors, caused by the high-speed motion, reduce the quality of the output image. For example, the RGBD camera we use to collect our real dataset is equipped with an RGB sensor and a depth sensor, and there is a $< 0.01s$ difference between their data collection times. Accordingly, we observe a mismatch between the edges of the color image and the depth image. Future work should improve sensor synchronization or investigate new methods to align the unmatched pixels.

References

[1] D. M. Grayson and A. F. Monk, "Are you looking at me? eye contact and desktop video conferencing," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 10, no. 3, pp. 221–243, 2003.

[2] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, "Few-shot adversarial learning of realistic neural talking head models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9459–9468.

[3] E. Zakharov, A. Ivakhnenko, A. Shysheya, and V. Lempitsky, "Fast bi-layer neural synthesis of one-shot realistic head avatars," in *European Conference on Computer Vision*. Springer, 2020, pp. 524–540.

[4] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "First order motion model for image animation," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[5] A. Bansal, S. Ma, D. Ramanan, and Y. Sheikh, "Recycle-GAN: Unsupervised video retargeting," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 119–135.

[6] T.-C. Wang, A. Mallya, and M.-Y. Liu, "One-shot free-view neural talking-head synthesis for video conferencing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recog-*

nition, 2021, pp. 10 039–10 049.

[7] A. Bozic, P. Palafox, M. Zollhöfer, A. Dai, J. Thies, and M. Nießner, "Neural non-rigid tracking," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 727–18 737, 2020.

[8] A. Bozic, M. Zollhofer, C. Theobalt, and M. Nießner, "Deepdeform: Learning non-rigid RGB-D reconstruction with semi-supervised data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7002–7012.

[9] K. Guo, F. Xu, T. Yu, X. Liu, Q. Dai, and Y. Liu, "Real-time geometry, albedo, and motion reconstruction using a single RGB-D camera," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, p. 1, 2017.

[10] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger, "Volumedeform: Real-time volumetric non-rigid reconstruction," in *European Conference on Computer Vision*. Springer, 2016, pp. 362–379.

[11] M. Slavcheva, M. Baust, and S. Ilic, "Sobolevfusion: 3d reconstruction of scenes undergoing free non-rigid motion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2646–2655.

[12] R. A. Newcombe, D. Fox, and S. M. Seitz, "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 343–352.

[13] W. Gao and R. Tedrake, "Surfelwarp: Efficient non-volumetric single view dynamic reconstruction," *arXiv preprint arXiv:1904.13073*, 2019.

[14] W. Lin, C. Zheng, J.-H. Yong, and F. Xu, "Occlusionfusion: Occlusion-aware motion estimation for real-time dynamic 3d reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1736–1745.

[15] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer Graphics and Interactive Techniques*, 1996, pp. 303–312.

[16] W. Dong, Q. Wang, X. Wang, and H. Zha, "PSDF fusion: Probabilistic signed distance function for on-the-fly 3d data fusion and scene reconstruction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 701–717.

[17] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. Ieee, 2011, pp. 127–136.

[18] C. Zhang, Q. Cai, P. A. Chou, Z. Zhang, and R. Martin-Brualla, "Viewport: A distributed, immersive teleconferencing system with infrared dot pattern," *IEEE MultiMedia*, vol. 20, no. 1, pp. 17–27, 2013.

[19] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3d reconstruction at scale using voxel hashing," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 6, pp. 1–11, 2013.

[20] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "Elasticfusion: Real-time dense slam and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2016.

[21] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, p. 1, 2017.

[22] S. Weder, J. Schonberger, M. Pollefeys, and M. R. Oswald, "Routedfusion: Learning real-time depth map fusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4887–4897.

[23] A. Dai and M. Nießner, "3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 452–468.

[24] J. Lawrence, D. B. Goldman, S. Achar, G. M. Blascovich, J. G. Desloge, T. Fortes, E. M. Gomez, S. Häberling, H. Hoppe, A. Huibers *et al.*, "Project starline: A high-fidelity telepresence system," 2021.

[25] Y. Li, H. Takehara, T. Taketomi, B. Zheng, and M. Nießner, "4dcomplete: Non-rigid motion estimation beyond the observable surface," pp. 12 706–12 716, 2021.

[26] T.-C. Wang, M.-Y. Liu, A. Tao, G. Liu, J. Kautz, and B. Catanzaro, "Few-shot video-to-video synthesis," *arXiv preprint arXiv:1910.12713*, 2019.

[27] W. Wu, Y. Zhang, C. Li, C. Qian, and C. C. Loy, "Reenactgan: Learning to reenact faces via boundary transfer," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 603–619.

[28] H. Averbuch-Elor, D. Cohen-Or, J. Kopf, and M. F. Cohen, "Bringing portraits to life," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 6, pp. 1–13, 2017.

[29] E. Burkov, I. Pasechnik, A. Grigorev, and V. Lempitsky, "Neural head reenactment with latent pose descriptors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 786–13 795.

[30] L. Chen, R. K. Maddox, Z. Duan, and C. Xu, "Hierarchical cross-modal talking face generation with dynamic pixel-wise loss," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7832–7841.

[31] J. Geng, T. Shao, Y. Zheng, Y. Weng, and K. Zhou, "Warp-guided GANs for single-photo facial animation," *ACM Transactions on Graphics (ToG)*, vol. 37, no. 6, pp. 1–12, 2018.

[32] S. Ha, M. Kersner, B. Kim, S. Seo, and D. Kim, "Marionette: Few-shot face reenactment preserving identity of unseen targets," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 10 893–10 900.

[33] Y. Nirkin, Y. Keller, and T. Hassner, "FSGAN: Subject agnostic face swapping and reenactment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7184–7193.

[34] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "Animating arbitrary objects via deep motion transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2377–2386.

[35] K. Vougioukas, S. Petridis, and M. Pantic, "Realistic speech-driven facial animation with gans," *International Journal of Computer Vision*, vol. 128, no. 5, pp. 1398–1413, 2020.

[36] A. Jamaludin, J. S. Chung, and A. Zisserman, "You said that?: Synthesising talking faces from audio," *International Journal of Computer Vision*, vol. 127, no. 11, pp. 1767–1779, 2019.

[37] H. Zhou, Y. Liu, Z. Liu, P. Luo, and X. Wang, "Talking face generation by adversarially disentangled audio-visual representation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 9299–9306.

[38] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing Obama: learning lip sync from audio," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017.

[39] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: Image synthesis using neural textures," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–12, 2019.

[40] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt, "Real-time expression transfer for facial reenactment." *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, pp. 183–1, 2015.

[41] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of RGB videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2387–2395.

[42] D. Vlasic, M. Brand, H. Pfister, and J. Popovic, "Face transfer with multilinear models," in *ACM SIGGRAPH 2006 Courses*, 2006, pp. 24–es.

[43] O. Fried, A. Tewari, M. Zollhöfer, A. Finkelstein, E. Shechtman, D. B. Goldman, K. Genova, Z. Jin, C. Theobalt, and M. Agrawala, "Text-based editing of talking-head video," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–14, 2019.

[44] K. Nagano, J. Seo, J. Xing, L. Wei, Z. Li, S. Saito, A. Agarwal, J. Fursund, H. Li, R. Roberts *et al.*, "pagan: real-time avatars using dynamic textures." *ACM Trans. Graph.*, vol. 37, no. 6, pp. 258–1, 2018.

[45] K. Olszewski, Z. Li, C. Yang, Y. Zhou, R. Yu, Z. Huang, S. Xiang, S. Saito, P. Kohli, and H. Li, "Realistic dynamic facial textures from a single image using gans," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5429–5438.

[46] S. J. Gibbs, C. Arapis, and C. J. Breiteneder, "Teleport–towards immersive copresence," *Multimedia Systems*, vol. 7, no. 3, pp. 214–221, 1999.

[47] L. Muhlbach, M. Bocker, and A. Prussog, "Telepresence in videocommunications: A study on stereoscopy and individual eye contact," *Human Factors*, vol. 37, no. 2, pp. 290–305, 1995.

[48] A. Jones, M. Lang, G. Fyffe, X. Yu, J. Busch, I. McDowall, M. Bolas, and P. Debevec, "Achieving eye contact in a one-to-many 3d video teleconferencing system," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, pp. 1–8, 2009.

[49] A. Maimone, J. Bidwell, K. Peng, and H. Fuchs, "Enhanced personal autostereoscopic telepresence system using commodity depth cameras," *Computers & Graphics*, vol. 36, no. 7, pp. 791–807, 2012.

[50] S. Izadi, R. A. Newcombe, D. Kim, O. Hilliges, D. Molyneaux, S. Hodges, P. Kohli, J. Shotton, A. J. Davison, and A. Fitzgibbon, "Kinectfusion: real-time dynamic 3d surface reconstruction and interaction," in *ACM SIGGRAPH 2011 Talks*, 2011, pp. 1–1.

[51] C. Zach, "Fast and high quality fusion of depth maps," in *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, vol. 1, no. 2. Citeseer, 2008.

[52] C. Zach, T. Pock, and H. Bischof, "A globally optimal algorithm for robust tv-l 1 range image integration," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[53] K. Kolev, M. Klodt, T. Brox, and D. Cremers, "Continuous global optimization in multiview 3d reconstruction," *International Journal of Computer Vision*, vol. 84, no. 1, pp. 80–96, 2009.

[54] N. Savinov, L. Ladicky, C. Hane, and M. Pollefeys, "Discrete optimization of ray potentials for semantic 3d reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5511–5518.

[55] N. Savinov, C. Hane, L. Ladicky, and M. Pollefeys, "Semantic 3d reconstruction with continuous regularization and ray potentials using a visibility consistency constraint," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5460–5469.

[56] A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, and M. Nießner, "Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4578–4587.

[57] G. Riegler, A. Osman Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3577–3586.

[58] G. Riegler, A. O. Ulusoy, H. Bischof, and A. Geiger, "Octnetfusion: Learning depth fusion from data," in *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 57–66.

[59] D. Paschalidou, O. Ulusoy, C. Schmitt, L. Van Gool, and A. Geiger, "Raynet: Learning volumetric 3d reconstruction with ray potentials," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3897–3906.

[60] M. Ji, J. Gall, H. Zheng, Y. Liu, and L. Fang, "Surfacenet: An end-to-end 3d neural network for multiview stereopsis," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2307–2315.

[61] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[62] C. Reiser, S. Peng, Y. Liao, and A. Geiger, "Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 335–14 345.

[63] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.

[64] R. Shao, Z. Zheng, H. Tu, B. Liu, H. Zhang, and Y. Liu, "Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 632–16 642.

[65] M. Işık, M. Rünz, M. Georgopoulos, T. Khakhulin, J. Starck, L. Agapito, and M. Nießner, "Humanrf: High-fidelity neural radiance fields for humans in motion," *arXiv preprint arXiv:2305.06356*, 2023.

[66] A. Cao and J. Johnson, "Hexplane: A fast representation for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 130–141.

[67] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer, 2015, pp. 234–241.

[68] D. Yong, P. Mingtao, and J. Yunde, "Probabilistic depth map fusion for real-time multi-view stereo," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE, 2012, pp. 368–371.

[69] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[70] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988.

[71] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, "Unstructured lumigraph rendering," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, 2001, pp. 425–432.

[72] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison, "Scenenet RGB-D: 5m photorealistic images of synthetic indoor trajectories with ground truth," *arXiv preprint arXiv:1612.05079*, 2016.

[73] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[74] Renderpeople, "Renderpeople dataset," https://humandataset.com.

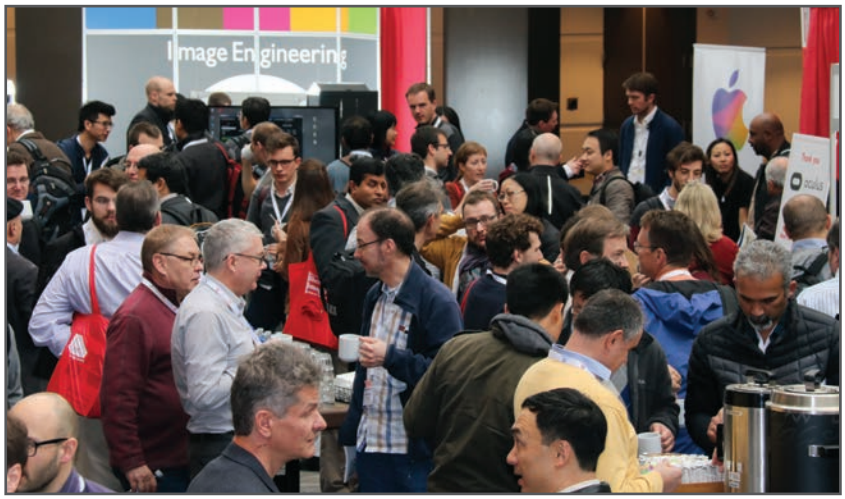
Author Biography

Fan Bu received a bachelor's degree in mechanical engineering from Huazhong University of Science and Technology (2015) and a master's degree in mechanical engineering from the University of Michigan (2017). At the time this article was completed, he was a Ph.D. student in electrical and computer engineering at Purdue University, working on research projects in communications, networking, signal and image processing.

JOIN US AT THE NEXT EI!

electronic IMAGING

Imaging across applications . . . Where industry and academia meet!



• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

ADVANCING IMAGING SCIENCE 75 1947 IS&T 2022 imaging.org