360° Stereo Image Composition with Depth Adaption

Kun Huang, Fang-Lue Zhang, Junhong Zhao, Yiheng Li, and Neil Dodgson

Abstract—360° images and videos have become an economic and popular way to provide VR experiences using real-world content. However, the manipulation of the stereo panoramic content remains less explored. In this paper, we focus on the 360° image composition problem, and develop a solution that can take an object from a stereo image pair and insert it at a given 3D position in a target stereo panorama, with well-preserved geometry information. Our method uses recovered 3D point clouds to guide the composited image generation. More specifically, we observe that using only a one-off operation to insert objects into equirectangular images will never produce satisfactory depth perception and generate ghost artifacts when users are watching the result from different view directions. Therefore, we propose a novel per-view projection method that segments the object in 3D spherical space with the stereo camera pair facing in that direction. A deep depth densification network is proposed to generate depth guidance for the stereo image generation of each view segment according to the desired position and pose of the inserted object. We finally combine the synthesized view segments and blend the objects into the target stereo 360° scene. A user study demonstrates that our method can provide good depth perception and removes ghost artifacts. The per-view solution is a potential paradigm for other content manipulation methods for 360° images and videos.

Index Terms—Stereoscopic Panoramic Image, Image Composition, Image Synthesis, Virtual Reality

1 Introduction

Advances in virtual reality (VR) and digital media technology have allowed people to virtually teleport to a virtual environment. This immersive experience provides tremendous opportunities in entertainment, education, and enriched experiences not directly accessible owing to safety or cost [22]. An economical way to construct such a virtual scene is to capture omnidirectional stereo images or videos from the real world. Therefore, there have been emerging research interests in 360° image and video processing for better immersive experiences in VR applications. But the question of how to manipulate the content of 360° stereo images remains less investigated. As a fundamental task in content manipulation, seamless image composition and cloning have been well-studied in the computer graphics and vision communities, especially for 2D images and videos [11], [15], [29], [48]. However, as demonstrated in the most recent 360° image/video processing work, such as stabilization [43], depth estimation [49], optical flow estimation [25], and edit propagation [59], [60], the methods designed for normal 2D images cannot be easily extended to work for 360° images, because of their incorrect spatial relationship measurement in the spherical domain.

Besides the typical problems that any image composition method has to cope with, such as gradient mismatch and complex object boundaries, there is an additional challenge with 360° stereo images: the consistency of the depth perception when the user is focusing on any part of the composited result. That issue can be neglected in planar stereo image composition [44] where a pair of camera positions are defined to look at the scene center, since the field-of-view (FoV) is limited in 2D images. However, 360° stereo images allow users to rotate their view directions to focus on an arbitrary region of the scene. The 360° images/videos are pre-loaded for the left and right eyes and played by directly projecting the left/right panorama to the left/right viewport for efficiency. If the stereo composition is only conducted as a one-off operation for a predefined user position, i.e., directly pasting the source regions from the left-view and right-view to the stereo equirectangular images for the final result, the perceived depths will not be correct unless the user's view direction is the same as the predefined cameras. Fig. 1 demonstrates the issue: When a user rotates their head with a VR headset, the depths of scene points vary, so the fixed disparity of a stereo pixel pair generated by the

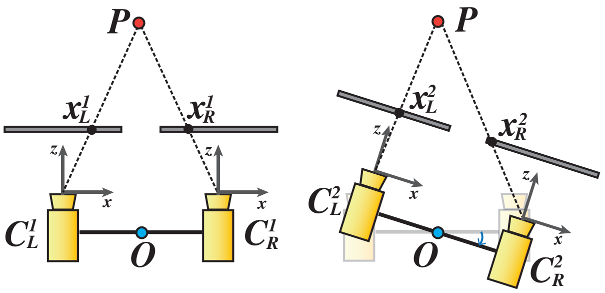


Fig. 1: The camera positions when watching stereo panoramas in a VR headset. P is a scene point, O is the virtual position of the user. When the user rotates their head (right), the two cameras are actually rotated about O, not their own centers, making the disparity $(x_L^1 - x_R^1)$ different from $(x_L^2 - x_R^2)$. This means that an image captured with this camera model has depth errors and ghosting when viewed from any rotated position.

- *K. Huang, F.-L. Zhang, Y. Li and N. Dodgson are from the School of Engineering and Computer Science, Victoria University of Wellington, New Zealand.*
E-mail: {kun.huang, fanglue.zhang, yiheng.li, neil.dodgson}@vuw.ac.nz
- *Junhong Zhao is with CMIC, Victoria University of Wellington*
- *Fang-Lue Zhang is the corresponding author.*

Manuscript received 2022; revised 2023.

one-off composition can never satisfy all possible view directions.

In this paper, we propose a novel method to insert stereo objects into a target 360° stereo image with a convincing appearance and well-preserved depth information when viewers change their viewing orientation. Our method uses the estimated depth information of both source and target stereo images to guide the generation of the stereo pair of inserted objects, ensuring the correct geometry when watching the object inserted at an arbitrary 3D position. For addressing the aforementioned depth inconsistency issue, we propose a solution where the generated disparities of stereo pixel pairs can fit different view directions of the virtual view pair. Instead of using a single pair of camera positions when generating the left and right panoramas, our camera model uses multiple pairs of camera positions facing in different directions. For different parts of the object, we separately generate the stereo content using the pairs of cameras looking in each direction and then combine the content of all the parts in the final left/right panoramas. More particularly, we build a deep neural network to learn to generate dense depth maps and object masks to produce high-quality stereo content when the object pose changes in the composited results, outperforming all the previous stereo image composition methods. In our user study, we find that our method offers the best depth perception, especially when the inserted object covers a large FoV in the final result.

Our contributions are as follows:

- An omnidirectional stereo image composition algorithm, which can composite a stereo object into a 360° panoramic background for VR applications. Our method has good fidelity, ensuring the fundamental 3D geometries of the inserted objects by guiding the content manipulation in 3D space.
- A novel solution to address the depth perception issue in stereo panorama content generation. We use a camera model that is more suitable to the geometry of stereo panoramas than a model that assumes projection onto a single plane.
- A deep model that is able to synthesize dense and accurate depth maps and object masks to facilitate stereo image generation for different object poses.

2 Related Work

Our work involves efficient compositing of 3D objects into stereoscopic 360° panoramas. We briefly cover the key related work in image manipulation, 360° image processing, stereoscopic editing, and 3D object manipulation.

2.1 Image Matting, Composition, and Segmentation

Image composition is a basic operation for content manipulation, used initially for film and video production [18]. Early methods focused on providing intelligent scissors for object segmentation to composite [36], [37]. In recent decades, alpha matting [48] and gradient-domain methods [38] have become mainstream approaches for composition. Matting allows us to extract accurate boundaries with transparency values of foreground pixels for realistic object insertion [8], [39]. Gradient-domain methods, such as Poisson Blending [38], help find a smooth transition between the background and the inserted foreground. Previous work also focused on various aspects of image blending, such as the environment lighting effects [46], [51], [62]. More recently, deep learning-based approaches have been proposed to increase the accuracy of the extracted soft masks of alpha matting [11], [13], [55] or improve the visual consistency between composited foreground and the target background [9], [24], [52]. The above methods handle 2D planar images very well. But they are not able to generate satisfactory results with stereo 360° images since an appropriate depth perception cannot be guaranteed.

2.2 360° Image Analysis and Processing

A great deal of recent work has attempted to understand and process 360° images and videos for better immersive experiences in VR applications. To provide better 3D information for mixed reality applications based on 360° videos, Feng et al. [16], [17] and Wang et al. [49] proposed deep depth estimation networks working on the spherical domain and built large panorama datasets for training their models. Deep learning techniques have also been used effectively for the semantic understanding of 360° images, including saliency detection [30], object recognition [41], and indoor holistic scene understanding [42]. Li et al. [23] developed a method of lighting and geometry estimation from 360 panoramic stereos. In the work of Li et al. [25], the dense correspondence estimation for 360° videos is improved by fusing the information of different sphere-to-plane projections. Although these techniques are capable of processing spherical 360° images properly, they are not able to be directly applied in stereo 360° image generation. Some researchers focus on omnidirectional view-synthesis from 360° image sequences to provide 6-DOF immersive experiences by explicitly [6] or implicitly [2] reconstructing 3D geometry. Zhao et al. [61] and Xu et al. [54] proposed to use convolutional neural networks to predict 360° HDR images for a better illumination effect when inserting virtual objects into a target scene. But they are not designed for manipulating stereo image content. To improve the interactive experiences, researchers presented methods for allowing a better user simulation [32] and adding social features to the VR video player [26], [33]. We focus on providing richer experiences by allowing the user to modify scene content.

2.3 Stereoscopic Image Editing

Stereoscopic image editing has attracted much research in the past decade, initially prompted by the needs of stereoscopic 3D film production [34]. Wang et al. [50] investigated a novel workflow called *StereoBrush* for users to convert a 2D stereoscopic image to 3D instantly by drawing strokes on the 2D image. Other research focuses on stereoscopic editing for stereo visual comfort by applying the mesh-based image warping manipulation methods to adjust the image structure. Tong et al. [44] proposed a novel system named *StereoPasting*, inspired by *StereoBrush*. It solves the stereoscopic composition task using an energy minimization warping formula. Users get instant feedback while painting strokes on the 2D foregrounds. Luo et al. [29] developed an algorithm for seamless stereoscopic image cloning, which manipulates on both color appearance and perceived depth. It estimates the disparity in the gradient domain to make the disparities of the cloning region continuous at the boundary, and also adjusts the shape and size of the cloning area by applying a perspective-aware warping with the constructed mesh based on the estimated disparity. Du et al. [14] introduced a 2D warping method for adjusting stereoscopic imagery, enabling users to perceive stereopsis in a new view. They use feature correspondences and straight-line constraints to guide the warping process, treating it as a quadratic energy minimization problem. However, these previous methods (*StereoPasting*, Luo et

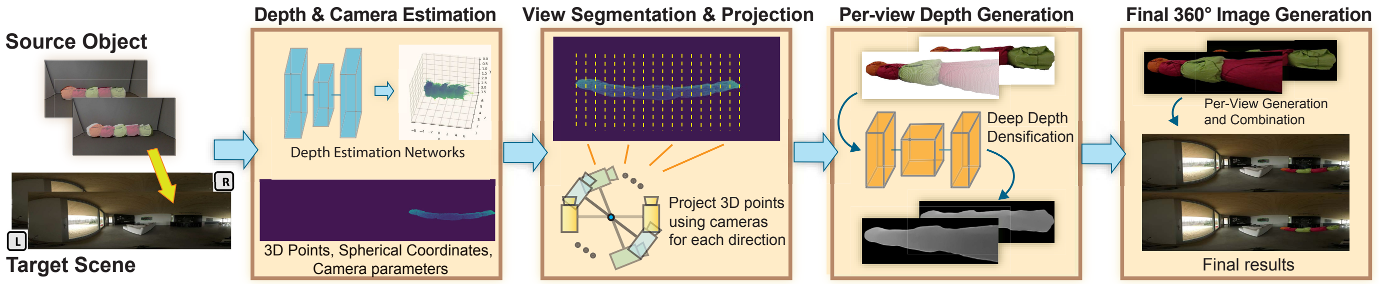

Source Object
Target Scene
R
L
Depth & Camera Estimation
Depth Estimation Networks
3D Points, Spherical Coordinates, Camera parameters
View Segmentation & Projection
Project 3D points using cameras for each direction
Per-view Depth Generation
Deep Depth Densification
Final 360° Image Generation
Per-View Generation and Combination
Final results

Fig. 2: The pipeline of our method. Given a stereo object, our approach manipulates the image content with guidance from 3D space to avoid distance metric issues when compositing the object into the target omnidirectional stereo background images.

al., Du et al.) are not plausible to deal with large perspective differences or occlusion between foreground and background because the generated disparities between corresponding pixel pairs can be significantly distorted. Furthermore, methods relying on mesh-based warping are unable to generate new image data that is required to fit the new perspective, if it was not present in the original input. They are also incapable of hiding information that is occluded under the new perspective because the region that should be occluded can only be narrowed down in the results.

Other research in stereoscopic image processing focuses on how to estimate accurate disparity/depth maps [3]. Recent advances in stereo depth estimation consist of deploying deep networks embedding all steps of traditional pipelines and combining effective learning modules [27], [45]. The estimated stereoscopic correspondences are used to conduct view-consistent image enhancement operations via deep networks, such as neural style transfer [5]. Due to the special distortions of 360° stereo images, deep networks that are delicately designed for estimating depth maps for stereo panoramic images were developed by Wang et al. [49], which assume vertical parallax between two views. Here, we use the depth information of the target scene and the geometric structure of the foreground object to allow the elements to be composited naturally while keeping a correct sense of occlusion and perspective.

2.4 Object Manipulation in 3D Space

Our work is also related to object modelling and editing methods in 3D for 2D images. Previous work that focused on editing a target object in 3D space needs either to reconstruct its basic geometry structure [7] or to use point clouds [31]. Van der Heuvel [47] and Criminisi et al. [10] introduced techniques of 3D reconstruction from single images, particularly for artificial objects that usually contain substantial prior geometric knowledge. Images of humans, which lack this geometry, do however contain prior structural knowledge that can be used to reconstruct free-form and texture-mapped models [58]. The reconstruction and manipulation of human models have been significantly advanced by neural network-based technologies [1], [40], where the geometry information is implicitly predicted and interpreted. To improve the fidelity of object insertion in a VR environment, Morioka et al. [35] proposed a method to let the inserted 3D object reflect the real-time lighting changes of the scene.

We choose to use point clouds to model the 3D foreground object because we can obtain depth information from stereo images of the object. The point clouds in 3D space enable flexibility when users edit the orientation or scale of the inserted objects without altering the underlying geometry structure. Furthermore, these 3D points are used to guide the warping and interpolation to generate the composited regions in the target equirectangular stereo pair. However, none of the above methods consider the depth consistency issue when the scene is presented as a 360° stereo image. In this paper, we propose a paradigm that can produce correct depth perception from an arbitrary view direction in a manipulated 360° stereo image.

3 Overview

Fig. 2 shows the pipeline of our method for compositing a stereo object into a target omnidirectional stereo background image. There are two key points in our method. First, we transform the common 2D image to 3D space, using an estimated depth map. After manipulating the image content with guidance, we further project the 3D coordinates to their spherical positions on the omnidirectional stereo (ODS) images to avoid the inconsistency distance metric issue in different image domains, as noted by Zhang et al. [59]. Second and more importantly, we apply per-view projection from 3D space to ensure appropriate disparities for different parts of the object. The 3D point clouds of the input stereo object and the target scene are reconstructed from the estimated disparities. According to the desired position and size of the inserted object, we transform the 3D points to a spherical coordinate system (θ, ϕ, ρ) with the user's virtual position as its origin, and segment the point clouds into multiple regions based on the horizontal angle θ. To generate proper depth perceptions for an arbitrary view direction, we build separate virtual camera pairs focused on each region, and apply per-view projections to obtain the initial sparse depth maps on the planar image domain. In our experiment, we find that denser segmentation always leads to higher visual quality. Therefore, we normally choose the smallest interval we can achieve to segment the 3D point cloud, which is the viewing angle covered by one column of the target equirectangular image. We then employ a deep depth densification model to estimate the dense depth maps and their alpha maps for all the view directions. The left and right color images are then generated with the guidance of the depth maps. For each view segment, we find the stereo equirectangular pixels within the segment's FoV and overwrite them with the corresponding pixels in the generated planar image pair for that view. Our proposed system also offers more possibilities for addressing the occlusion problem that arises when inserting a source object into a background scene, an area that has been less explored in previous work on stereo planar image composition with depth information. Properly handling occlusion is crucial for achieving natural depth perception in the final composite, as it eliminates any depth conflicts between the inserted object

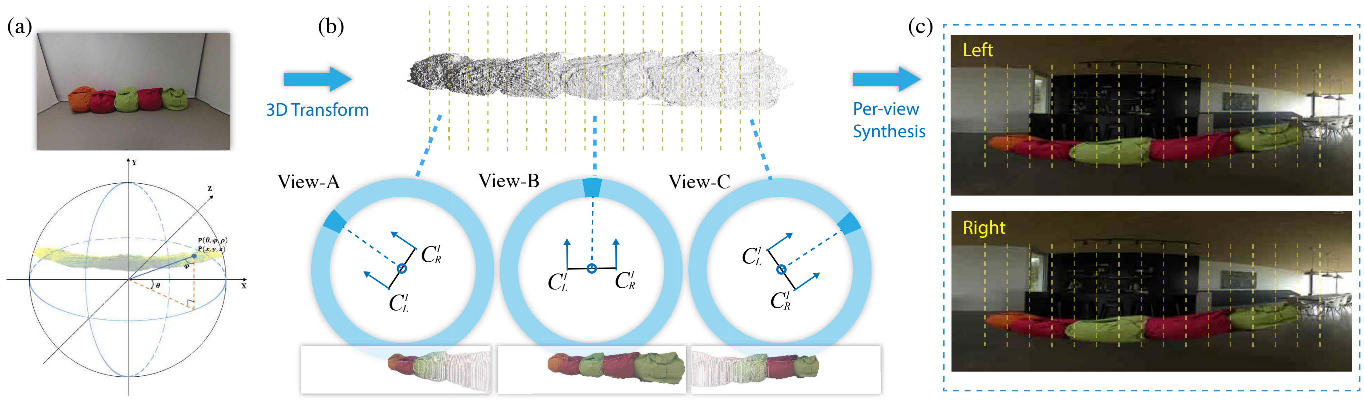


Fig. 3: Per-view image generation. (a) Spherical coordinate representation. (b) Per-view projection. (c) The stereo pair is generated with guidance from the densified depth maps for each view segment.

and the original panorama. However, creating a 3D scene from a panoramic image remains a challenging task due to the fact that depth information can only be obtained from the front view. Furthermore, current methods for 360° monocular depth estimation are inadequate in producing accurate real-world predictions because of the significant noise present in the input data.

4 Algorithm

The inputs of the method are a pair of stereo images of the object to be inserted, with a masked region-of-interest (RoI), and a stereo panoramic image pair as the target scene. The output is the composited image pair with correct depth perception of the inserted object for arbitrary view directions. The critical challenge is to find a proper binocular camera model to project inserted objects to target panoramic images while preserving correct depth perception. Another challenge lies in the generation of a complete and smooth depth map especially when the desired inserted 3D position is different from the original source position.

4.1 Sparse 3D Reconstruction

We first estimate the depth map of the foreground object and the region around the desired position of the target scene. For inserting the object into the stereo panoramic target scene represented by equirectangular projection, the user is required to specify the position and the size of the object. We project the RoI of the target scene to a planar image with a default FoV of 60° If the source object is in a stereo panorama as well, we use an FoV of no less than 60° to cover the horizontal angle extent of the object when projecting the object into a planar stereo crop. We apply Li et al's sequence-to-sequence correspondence perspective deep model, named the STereo TRansformer (STTR), to estimate the disparity map from the stereo pairs [27].

Using the predicted horizontal disparities between the input rectilinear stereo image pair, we can generate a depth map of the foreground objects based on camera parameters. Assuming the focal length is f and the baseline between the left and right camera is B, the depth values z for the pixels can be calculated from its estimated disparity d: $z = (f \times B)/d$.

Given a pixel (p_x, p_y) on the left-view image with a size of (W, H), we assumed a standard camera model located at the origin looking down the z-axis. The 3D coordinates of this point in the world coordinate system are obtained by:

$$(x,y,z) = \left(\frac{(p_x - W/2)\times z}{f}, \frac{(p_y - H/2)\times z}{f}, z\right), \qquad (1)$$

In the recovered sparse point cloud from the stereo pixels, we select the center point of the cloud as the object's reference point to place in the target scene and about which to make any transformations, such as scaling and rotation. It also helps the user to correctly position the object in the target scene. Fig. 3 shows an example of a recovered 3D point cloud.

4.2 View Segmentation and Projection

Having determined the transformations needed to meet the user's desires, including the size, orientation and position of the object, we obtain the 3D point cloud for the object that is to be inserted in the target scene. In order to tackle the challenge of varying depth perceptions, we partition the point cloud into distinct segments based on the respective viewing directions relative to the user's position. As shown in Fig. 3(a), we first transform world coordinates to a spherical coordinate representation by:

$$\theta = \arctan\left(\frac{x}{z}\right) \quad \phi = \arctan\left(\frac{y}{\sqrt[2]{x^2+z^2}}\right), \quad \rho = \sqrt[2]{x^2+y^2+z^2} \qquad (2)$$

Then we segment the point cloud according to the points' θ-values: we split the range of θ into N intervals, i.e., each interval covers a range of $\frac{2\pi}{N}$, and segment the points of the object based on the horizontal angle intervals in which they fall. In our experiments, we found that some segments might only contain too few points when the user-specified pose is largely different with the original pose, because the point cloud was recovered from the view of the original stereo image. Therefore, we produce a dense depth map for the desired pose and scale of the target object using the *deep depth densification* network proposed later in Sec. 4.3, and then perform view segmentation on the point cloud generated using the dense depth map.

Per-view Projection In this step, we treat each segment of the point cloud with a virtual camera pair that focuses on the individual segment's center. We generate a series of per-view projected point clouds, each in their specific camera space (Fig. 3(b)). The position and orientation of the user's binocular views are defined as follows. The virtual viewing camera pairs are located on the xz-plane of

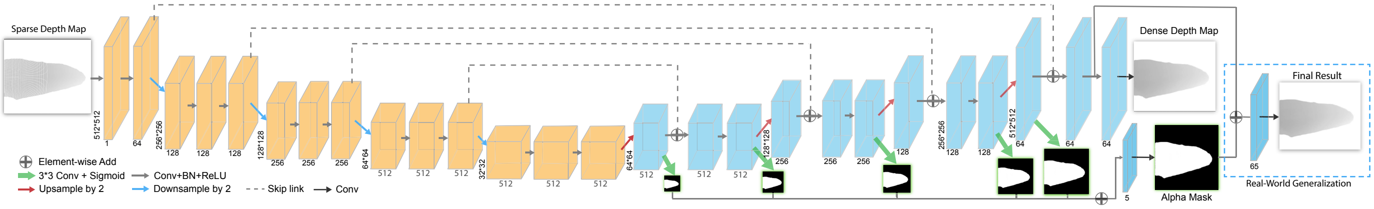


Fig. 4: The architecture of the deep depth densification network. We predict a dense depth map for the desired perspective and a soft mask that indicates the pixel's transparency and whether the pixel is solid.

the world coordinate system, with up-vector along the positive y-axis. The initial position of the two cameras, C_L^0 and C_R^0 are located on opposite sides of the origin on the x-axis with a distance B between the two cameras, which are at $C_L^0 = [-B/2, 0, 0]$ and $C_R^0 = [B/2, 0, 0]$, respectively. This initial pair's viewing directions are parallel to the positive z-axis. Given N intervals, the system calculates the viewing direction θ_i for the ith interval as:

$$\theta_i = 2\pi i/N - \pi \tag{3}$$

Then, the rotation matrix R_i for a such interval is computed based on the viewing direction, which is a rotation about y-axis by an angle θ_i. We can define the desired position of C_L^i and C_R^i with associated viewing direction as:

$$C_L^i = R_i C_L^0, \quad C_R^i = R_i C_R^0 \tag{4}$$

The projected 3D point cloud in the world coordinate system p_w will be further transferred to each segment space for both views, Q_L^i and Q_R^i with the specific camera matrix are expressed as:

$$Q_L^i = [R_i | C_L^i] p_w = P_L^i p_w, \quad Q_R^i = [R_i | C_R^i] p_w = P_R^i p_w \tag{5}$$

Using the known intrinsic matrix of both cameras for a specific segment, we then project the point clouds to 2D depth maps. The projected depth maps usually contain holes and gaps when the relative pose of the object to the camera changes. The following steps (see below) thus estimate a dense depth map to guide the generation of stereo RGB images for that view direction. Here, instead of just generating a depth map for the corresponding segment, we also generate sparse depths for the neighbouring regions in a certain field-of-view to facilitate the following dense depth map generation for each view.

4.3 Per-view Depth Generation

To ensure the perceived depths are correct when users are looking at different parts of the composited object, we propose to generate the dense depth maps and corresponding stereo RGB images based on the projected point clouds for each view direction separately. The stereo images for each view direction will be combined together to generate the final left and right panoramas.

Given that the desired pose of an object can vary considerably away from the captured pose, the projected depth maps can be very sparse for some view directions of the stereo camera pairs. We initially attempted to directly fill the missing 2D pixels on the depth map by morphological interpolation-based methods as described in [21]. However, an interpolation-based method works only when the pose changes are subtle so that the missing points can easily find valid depth values in their neighbourhoods. If the object has a relatively large scaling and rotation, the valid points are too sparse to provide sufficient reference values for the missing pixels to interpolate. Moreover, since we need to generate dense depth maps for each view, the long running time of morphological interpolation has a large negative impact on efficiency. Thus, we propose a *deep depth densification* network (DDDN) to solve the above issues when generating new depth maps. We show that our network generates dense depth maps efficiently with higher visual quality, especially for objects with sharp geometry features. To avoid the distortion in the equirectangular representation affecting the learning process, our deep model works in the 2D rectilinear image domain.

Architecture We build a convolutional generative network that takes a projected sparse depth map as input and predicts dense valid depths and transparencies for the target object. As shown in Fig. 4, a 2D sparse depth map goes through a U-Net-like architecture to produce a densified depth map. More specifically, we employ the following two learning schemes to improve the quality of generated depth maps: First, we explicitly predict a mask that indicates whether a pixel belongs to the target object in the final image, helping the network to learn whether a pixel should have a valid depth value. Second, we let the decoder learn from multi-scale masks to have a better capability of reconstructing the object's geometry structure. For each scale level, we use a separate convolutional layer, upsampling operation, and sigmoid function to predict the masks at different scales, which are then fused by a concatenation operation at the end with some higher-scale contextual information. Finally, the output mask and depth map are fused with a concatenation operation, followed by a 1 x 1 convolutional layer and a sigmoid function to generate the final predicted result.

In our experiment, the above schemes are shown to be effective, especially for improving the depth map quality for those regions with large sparsity.

Dataset We train and evaluate the above network using both synthetic and real-world datasets. We build a synthetic dataset with ground truth depth maps of rendered objects. We use Unity3D to render RGB images and their associated dense depth maps with different positions, scales, and orientations. We select 30 classes of 3D shapes in the ShapeNet dataset [4], covering a variety of categories of furniture, vehicle, housewares, and buildings. In each class, we randomly select 50 objects. For each object, we render 20 different poses, producing 20 dense depth maps, $\{D_k\}$, and their corresponding masks, $\{M_k\}$. The original pose for each object is chosen as its reference pose and a point cloud P_0 is reconstructed from its depth map D_0. The transformations, between the other poses and the reference pose are then applied on P_0 to generate the sparse depth maps, $\{\hat{D}_k\}$. We use the sparse depth map, $\hat{D}_k$, to simulate the input data for the real-world use case, where the sparse

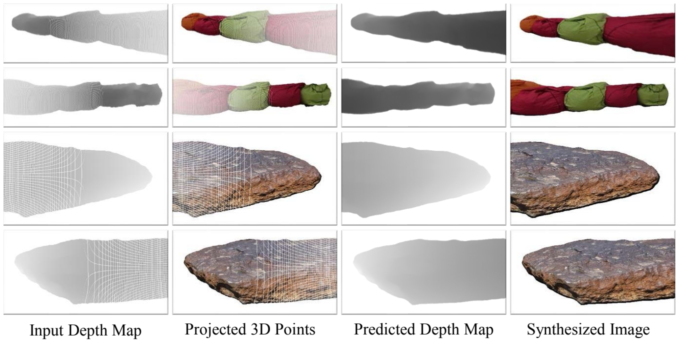


Fig. 5: The input sparse depth map (first column). The predicted dense depth maps (third column) by our network and synthesized dense RGB images (fourth column) with the guidance of the depth maps. The projected 3D points of the transformed source objects are shown in the second column.

depth map can be generated by transforming the reconstructed point clouds from the original stereo object to the desired 3D orientation and position. The corresponding dense map, D_k, and mask, M_k, can be used to supervise the learning process.

Our real-world dataset is drawn from the HRWSI dataset [53]. Our subset covers 15 different labels including living and non-living objects, such as humans, birds, buildings, and chairs, among others. To ensure the quality when generalizing to real-world data, we select 67 main objects that are completely captured within the images. For each of these objects, we construct the point cloud P_0 first from its original depth map D_0. We then produce 60 sparse depth maps $\{\hat{D}_k\}$ with corresponding camera pose (rotation R and translation T) $\{R_k^T\}$ and original point cloud P_0 in 30 different stereo camera-viewed poses. As we lack the ground truth dense depth maps for the transformed sparse depth maps, we adopt a self-supervised approach for generalizing to real-world images. We leverage the corresponding original view of each object as its reference view D_0, along with the predicted dense depth maps $F_d(\hat{D}_k)$ and the inverse camera pose $(R_k^T)^{-1}$ to enable us to carry out the self-supervised process effectively.

Training Procedure The neural networks are trained to fill the gaps and missing points while preserving the important geometry structures for a sparse depth map. We use the following losses in the training procedure:

(1) The reconstruction loss, L_r, an L2 loss applied on the predicted depth map with the ground truth mask, defined as:

$$L_r = \sum_{k=1}^{K} |D_k - F_d(\hat{D}_k) \cdot M_k|_2 \tag{6}$$

where $F_d(\hat{D}_k)$ represents the predicted dense depth map from the sparse map $\hat{D}_k$ and M_k is the ground truth mask

(2) The mask loss, L_m. The network generates masks of different scales, which help it to better learn the global structure of the shape. L_m is the loss of these smaller, subsidiary output masks, which is defined as:

$$L_m = \sum_{k=1}^{K} \sum_{l=0}^{L} |M_k^l - F_m^l(\hat{D}_k)| \tag{7}$$

where M_k^l is the ground truth mask for the scale l and $F_m^l(\hat{D}_k)$ is the predicted mask for scale l.

(3) The perceptual loss, L_p, applied on the depth map to produce better details. This makes the overall loss:

$$L = \lambda_r L_r + \lambda_m L_m + \lambda_p L_p \tag{8}$$

By default, we set $\lambda_r = 0.4$, $\lambda_m = 0.6$, and $\lambda_p = 1.0$ as the weights for different terms. We split our synthetic datasets into a training set of 19950 images and a test set of 9000 images. We train our network by 100 epochs or make an early stop when the losses on the validation data stop declining.

For the self-supervised real-world generalization procedure, we first transfer the predicted masked depth map along with the corresponding inverse camera pose back to their depth maps in the original pose. Then, we apply the following losses on the transferred depth map $F_d(\hat{D}_k)'$ and its corresponding transferred mask $F_m(\hat{D}_k)'$.

(4) The BerHu loss, L_b, applied on $F_d(\hat{D}_k)'$ for optimizing depth predictions from the original depth map D_0 with the corresponding transferred mask:

$$L_b = \begin{cases} |F_d(\hat{D}_k)' - D_0 \cdot F_m(\hat{D}_k)'| & |F_d(\hat{D}_k)' - D_0 \cdot F_m(\hat{D}_k)'| \leq C, \\ \frac{(F_d(\hat{D}_k)' - D_0 \cdot F_m(\hat{D}_k)')^2 + C^2}{2C} & |F_d(\hat{D}_k)' - D_0 \cdot F_m(\hat{D}_k)'| > C, \end{cases} \tag{9a}$$

$$C = 0.2max(|F_d(\hat{D}_k)' - D_0 \cdot F_m(\hat{D}_k)'|) \tag{9b}$$

(5) The BCEWithLogits loss, L_{bce}, applied on the predicted transformed mask with its masked original's mask.This makes the overall loss:

$$L = L_b + L_{bce} \tag{10}$$

We split our real-world datasets into a training set of 4020 images and a test set of 1206 images. We do the real-world generalization on our network and apply an early stop when the losses on the validation data stop declining.

We show some results of our method in Fig. 5 and compare our method with some alternatives in Fig. 6. These demonstrate that it is hard for interpolation-based methods to properly fill the missing pixels in the boundary parts since the transformed points can be very sparse in the projected depth map and cannot form any continuous line structures to wrap up the object. The use of the mask losses avoids the depth value “leaking” to pixels that should be outside of the object's contours and also makes the network more confident when estimating the depth values for pixels with strong geometry features. In Fig. 7 and Tab. 1, we compare the performance of our approach with and without the real-world generalization scheme, which has been incorporated to enable better generalization between synthesized and real-world data. This integration has led to a significant reduction in the amount of noise present in the generated depth maps.

Using such a network, we are able to rapidly produce all the dense depth maps where the camera pairs are focused on the centers of different 3D point segments.

4.4 Final Stereo Panorama Generation

Given the dense depth map for each view direction, we re-project each pixel to the original input stereo image to obtain its color value. In the original image, the system applies alpha matting [19] to generate soft edges for composition. To ensure a seamless composition, we also create an associated alpha mask by copying the alpha value of each referred original pixel. For the N segments of the point cloud, we thus obtain N stereo image pairs $\{I_n^r, I_n^l\}(n =$

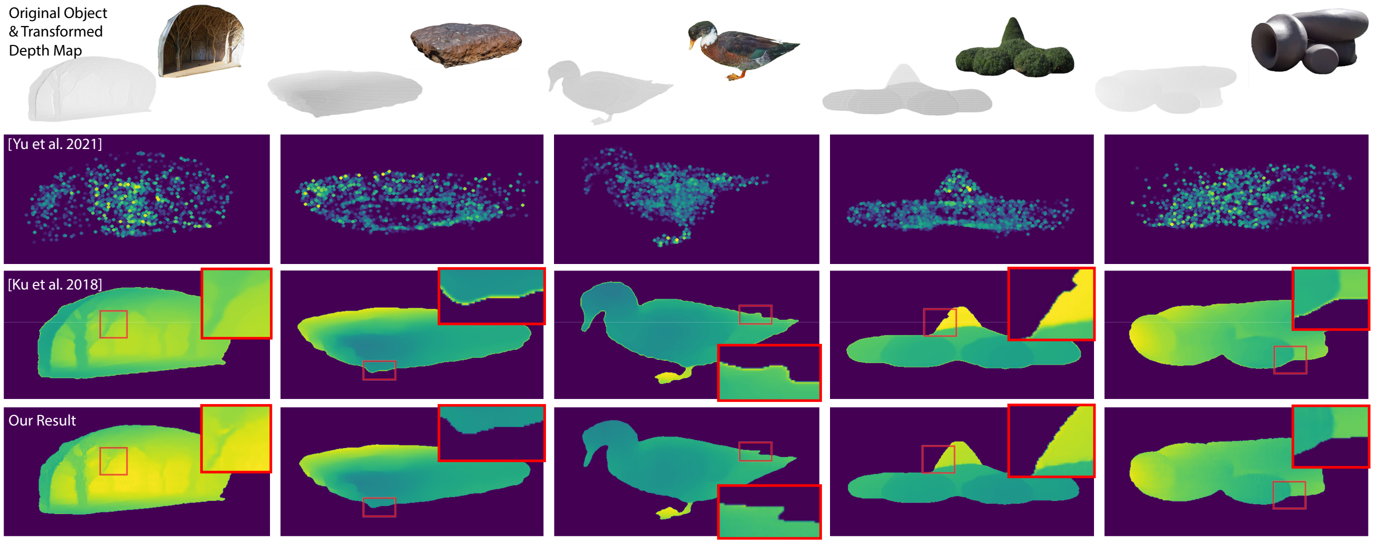


Fig. 6: Comparison on depth densification. We show the source objects and their sparse depth maps after 3D transformation in the top row and the generated depth maps by PointTr [57] (Row 2), depth map completion method of [21] (Row 3), and our depth densification network (Row 4).

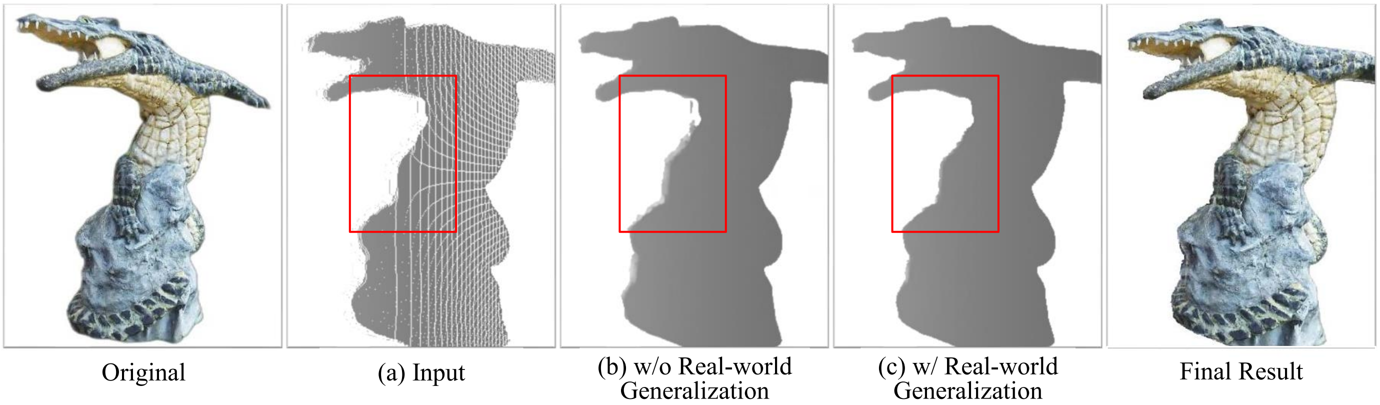

Original
(a) Input
(b) w/o Real-world Generalization
(c) w/ Real-world Generalization
Final Result

Fig. 7: The input sparse depth map, and the comparison of the predicted results if the real-world generalization is applied. The red rectangle region shows where real-world generalization makes a significant difference.

Method	MAE	RMSE	SSIM	PSNR
w/o Gen.	0.9317	5.3009	0.9150	35.6401
w/ Gen.	**0.7837**	**4.0211**	**0.9174**	**39.4472**

TABLE 1: Quantitative results on real-world datasets with two methods: without and with real-world generalization.

$1,2,\ldots,N$) with their alpha masks. To accelerate the process, we generate the depth map and stereo RGB images only in the involved part of the object and its neighbouring region, since the other parts, out of the specified view range, will not be used in the final result generation. We did experiment with feeding the sparse depth map of the *entire* object into the neural net, but found that this does not produce better quality, but rather reduces the resolution of the focused segment of the object in the resultant image.

We compose the synthesized left and right views of each segment to the left and right views of the target panorama respectively. For stereo panoramas captured by 360° cameras, we obtain their depths using the 360° depth estimation method proposed in [17] to align the 3D scenes of the target and the source images. Then with the camera model introduced in Eq. 4, for each pair of cameras (C_L^i, C_R^i), we use their view direction and the interval θ_i to obtain the affected pixels in the left and right views of the target panorama. Since the columns of an equirectangular image are naturally the pixels of different horizontal viewing angles, we just need to identify the affected columns and project those spherical pixels to the 2D image plane of the synthesized image I_i^r or I_i^l to find the pixels to overwrite the original colors. The pixels' alpha values will be used to ensure that only valid pixels will be used and seamlessly blended. Finally, depending on whether the inserted region contains background pixels surrounding the object boundary, we optionally perform Poisson Blending [38] on the synthesized regions in the left and interpolate the right view based on the geometric information.

We find that a larger number of segments always means better visual quality. Therefore, in our default settings, the viewing angle intervals are decided by the horizontal resolution of our target panoramic images and are normally set as the viewing angle represented by a single column. For example, for a panorama with a resolution of 3840×1920, we use $N = 3840$ and our interval is thus $360^\circ/3840 = 0.09^\circ$. In our experiments, we found that the per-column synthesis will not introduce artifacts regarding the spatial continuity of the object's appearance, because the changes between neighbouring columns are subtle and neglectable. Finally, we can compare the depth of the inserted object with that of the panorama to determine the occlusion relationship for better composition.

5 Experiments and Results

In this section, we evaluate the key parts of our system, demonstrate our 360° stereo image compositing results, and compare them with other stereo composition methods. We also conduct a user study to evaluate the depth perception quality of the generated results.

5.1 Implementation

The source object can be from any kind of stereo images, such as images captured by 2D/360° stereo cameras or rendered stereo images, and single RGB-D images. Our system enables users to interactively mark their objects of interest and obtain alpha masks for them, and provides a "one-off" preview of the composition results on the target scene, allowing users to view the changes in real-time. Furthermore, the DDDN can be used to give users greater flexibility in editing the desired pose of the inserted object, such as orientation and scale. Once a dense depth map is obtained, it can be projected into 3D space and aligned with the target scene for per-view segmentation. The partitioned 3D points can then be projected onto the 2D image domain using perspective projection and densified using our proposed DDDN for each view. Finally, the equirectangular projection is applied to project the specific column on the predicted depth map and mask onto the target column of the 360 images, which we refer to as "per-column composition". Furthermore, for cases where there is a significant difference in pixel colors between the source object and the target region, Poisson Blending, as outlined in [38], may be optionally employed. While the left view can be generated using the aforementioned procedure, the right view can be synthesized by harnessing geometric information to obtain and interpolate corresponding pixel colors, thereby ensuring consistency in color and the underlying 3D geometry between both views.

For consumer-grade stereo cameras for which the intrinsic parameters are available, we can directly use these to offer a better sense of the real-world size of the operated objects. For example, we used $f = 700$ and $B = 0.12$ for images captured by a ZED camera. Otherwise, we set some constant values for f and B for convenience and let the user adjust the object's size for the stereo data [53] collected from the internet. When refining the final appearance, gradient-domain methods such as Poisson Editing are provided as an optional operation for users.

When applying the per-column strategy on a single CPU core of an Intel Xeon W-2133 with an RTX3090 GPU, the average execution time of our Python implemented method is 45 seconds for generating a composition result with a resolution of 3840×1920 target 360° image ($N = 3840$, 0.09° as the view interval and the object covers an FoV of 90°), including object segmentation, depth estimation, point cloud processing, per-view projection and the final combine steps. The number of depth maps and masks that are produced is contingent on the number of columns (FoV) of the inserted object that is compositing on the equirectangular image. If a source object covers 90° of FoV in a 3840×1920 360° image, there will be a total of 960 depth maps generated for different camera views, in accordance with the original depth maps for each view.

To minimize the computation time and ensure interactive performance, we implement an alternative "key-column" strategy that generates a smaller number of depth maps, each covering multiple columns of pixels. We found that one depth map per 11 columns of pixels works well without creating obvious discontinuities between one set of columns and the next. In the example above, instead of generating 960 dense depth maps using the "per-column" strategy, the key-column strategy requires only 87 key columns for a 90-degree FoV coverage, each key column's depth map being used by the 5 columns of pixels to its left and the 5 to its right. The key-column strategy reduces the execution time to 9 seconds, effectively reducing redundancy and computational costs, thereby making our approach interactive.

5.2 Component Evaluation

5.2.1 Real-world Generalization

We use self-generated synthetic data to train our DDDN model for dense depth map generation, as there are no existing datasets that fully satisfied our requirements to the best of our knowledge. For example, the 4D Light Field Dataset proposed by Honauer et. al [20], while closely related to our problem, provides only a single depth map for the default view location, despite having multiple views of the same object.

It should be noted that, while synthetic data can be a useful starting point for the depth completion task, the trained model may not accurately capture the complexity and variability present in real-world data, as illustrated in Fig. 7 (b). We incorporate additional loss functions (L_b and L_{bce}) to optimize the predicted depth map, accounting for the complexities and robustness required for real-world data. These losses ensure that the predicted depth map (masked) matches the default depth map after a corresponding reverse transformation, facilitating consistent completion across views while linking to the original input. By training with real-world data, we have observed significant improvement, as shown in Fig. 7 (c).

We assess the effectiveness of our approach using standard metrics for depth map densification, including MAE, RMSE, SSIM, and PSNR, on a real-world dataset, as described in Section 4.3. Tab. 1 presents the quantitative results. The results demonstrate that our DDDN model, with the inclusion of real-world generalization, outperforms our original model in terms of these metrics. This validates the robustness and effectiveness of our proposed approach in real-world scenarios.

5.2.2 Comparing Rendering Methods

We compare three different rendering schemes for 360° stereo image composition: per-column, key-column and one-off. The one-off scheme is the baseline against which we compare our new schemes (per-column and key-column). The one-off scheme is a 2D stereo composition method that takes the stereo contents as input and applies one-off object insertion operations on the left and right views of the target image. This produces incorrect stereo disparities over large parts of the 360° image. In contrast, our new schemes are able to generate correct depth perception. A basic requirement for a stereo image pair is that there should be only horizontal disparity for each corresponding pixel pair when the viewer is looking at that point, to fit the layout of human eyes. However, the stereo 360° images created using the one-off method provide roughly

Method	Per-column	Key-column	One-off
Disparity Difference	**0.6544**	**0.7733**	**3.0327**

TABLE 2: Quantitative results on corner points of a chess board with different rendering methods: per-column, key-column (11 columns), and one-off.

accurate disparity results only for the area directly in front of the capture cameras. Behind the cameras, the perspectives generated are reversed, and to the left and right of the cameras, there is vertical disparity. This is because the two cameras are placed at fixed positions, spaced apart by the average distance between a person's eyes. Looking directly ahead, their separation is correct, left–right. But at 90° to that direction, the relative positioning of the two cameras is not left–right but instead front–back, resulting in a scale change rather than a stereo disparity: this is incorrect, does not provide correct stereo perception, and can result in eye strain, such as Fig. 12. Indeed, the one-off method produces displacements in all directions. The direction and magnitude of displacement depends on the object's distance from and angle to the pair of cameras. To ascertain the scope of this problem and whether our schemes resolve it, we conducted a quantitative analysis of the disparities generated by the three schemes, comparing them against the disparities present in the ground truth (GT) stereo 360° image pair generated by Unity3D.

We use Unity3D to generate a chessboard that covers 119.25° FoV in the final stereoscopic results with the correct depth adaption across all the covered regions. Using such synthetic data, we are able to avoid possible 3D reconstruction errors when working on real-world data, so that the comparison can focus only on the generated disparities. For the one-off scheme, we use only the initial left and right camera positions described in Sec. 4.2 and render two 360° images as the stereo pair. For our per-column and key-column schemes, we segment the 3D scene as in Sec. 4.2 according to the view angle range covered by a number of columns of the panorama and render the pixels within the view range using the pair of cameras looking at that direction. Then the per-column or key-column results are combined together to form the final stereo panorama with depth-adapted left and right content.

Tab. 2 presents a quantitative analysis of the disparities generated by the per-column, key-column (11 columns), and one-off schemes. We compare the average Euclidean distance between disparity vectors against the GT stereo 360° image pair produced by Unity3D at the centered different positions ($\theta = 0^\circ$, $\phi = -70^\circ, -35^\circ, 0^\circ, 35^\circ, 70^\circ$). Our schemes achieves significantly lower Euclidean distance differences than the one-off scheme on the disparity vector between the left and right views of the covered region, which shows the generalization ability of our method when compositing to different regions on a sphere. It is worth noting that the key-column scheme generates comparable evaluation results to the per-column approach, while substantially decreasing redundancy and computation time in comparison.

5.2.3 Deep Depth Densification

Our deep model for depth densification is built to generate a dense depth map from a sparse projected point cloud. To demonstrate the necessity of the dedicated deep network, we compare it with two possible alternatives: depth map completion methods and point cloud completion methods. Most depth map completion methods are designed to improve the depth map quality of a given RGB-D image. They need a complete RGB image to guide the depth completion, which is not available in our task because the dense RGB image for the desired pose is also missing. Thus, we choose to compare with the method of [21] because it can perform depth completion based on only an input sparse depth map.

Some results of the depth completion method proposed in [21] are shown in the third row of Fig. 6. Due to the limitations of their morphological operations when filling the missing pixels, their method fails to maintain geometric details, e.g., the trees' edges in the first example in Fig. 6. Their method can also easily propagate incorrect depth values to its neighbours, causing undesirable depth effects like in the boundary regions of the second and fifth examples. Finally, their method may generate incorrect object shapes as in the third and fourth examples, because their method cannot predict which positions have valid pixels of the foreground object. Our deep depth densification method overcomes the above issues and produces higher-quality depth maps. It should be also noted that our method is 6 times faster than the method of [21].

In the comparison between point cloud completion and our deep depth densification, we feed the transformed sparse point cloud to one of the state-of-the-art deep point cloud completion methods, PointTr [57] and project it to generate the depth map for the target view. We found that the point cloud completion method focuses more on the global 3D structure and the integrity of the model and fails to generate sufficient depth details for a specific view. The mandatory sampling step of their method is also a reason for the failure of the dense depth map generation. It cannot guarantee that all the geometric details of the depth map are preserved after the sampling step. Some typical examples are shown in Fig. 6, where our method generates depth maps with significantly better visual quality than PointTr.

We also consider NeRF-based approaches to directly learn to generate novel views. Unlike most NeRF-based approaches that need a number of input views to reconstruct a neural radiance field, PixelNeRF [56] and Depth-Supervised NeRf [12] only need one or few input images to synthesize new images of novel perspectives. However, PixelNeRF and DSNeRF algorithms face challenges in generating high-quality textures from stereo pairs, as the short stereo baseline typically leads to limited depth constraints estimated through structure-from-motion. This limitation results in unsatisfactory visual quality for composition tasks. Moreover, the algorithms often recover incomplete point clouds, typically only capturing the surface facing the camera, which further limits the ability to generate clear and sharp textures.

5.3 Results

Some composition results of our approach are presented in Fig. 9. For each example, the stereo panoramic scenes are shown at the bottom using the equirectangular projection of its left view. The inset windows demonstrate the segmented source objects and the zoom-in windows visualize the composited stereo objects using anaglyph images. We also include all the stereo 360° results in our supplementary materials, which can be viewed with VR headsets to achieve a better depth perception. From the results where we change the objects' orientation, size and position, we can see that our method is suitable for processing panoramic images as it is able to recover and manipulate the 3D geometry information to guide the pixel generation. We naturally avoid the computation for adapting to equirectangular distortions when pasting the object into an arbitrary position. We also generate correct panoramic disparities using this 3D-guided approach. Fig. 8 shows a result where we achieve natural

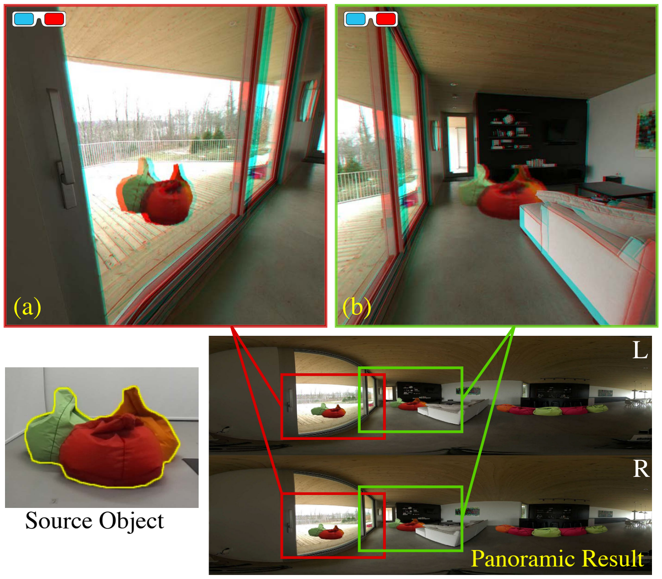


Fig. 8: Composition results by the proposed system. (a) Foreground objects are rotated about the y-axis, and the red beanbag blocks the corresponding region of the orange one. (b) Composited foreground objects are occluded by the existing objects based on their depths.

composition results by considering the occlusion relationship with the backgrounds. The occlusions between the composited objects and the original object show the depth consistency achieved by our method. In Fig. 9 (a) and (f), we apply Poisson Blending to one view to achieve natural color matching with the target scene, and then leverage geometric information to transfer and interpolate corresponding pixel colors from one view to the other through the image warping (point cloud-based) technique , thus ensuring color and underlying 3D geometry consistency between both views. In addition, our method has no restrictions on the size of the source object or the FoV covered in the target scene, which is particularly suitable for 360° image editing tasks. One reason is our proposed work manipulates the point cloud in the 3D space and then projects it to the target panoramic images. The other reason is that the target panoramic (equirectangular) image already covers $360^\circ \times 180^\circ$ of a scene, so the target 360° images will always be able to incorporate any size of composited object. Fig. 10 shows two such examples, where our approach achieves correct disparities in the left and right end of the composited objects.

5.4 Comparisons

Directly treating equirectangular images as normal 2D images when applying the image cloning operation cannot generate correct equirectangular distortion, which is important for maintaining the inserted object's shape when viewing it in a headset or a 360° image player. Considering the incorrect equirectangular distortion will also lead to problematic disparities when viewing the region of interest of the result, we therefore do not make further comparisons with the 2D planar image composition methods proposed for monocular 2D images.

For objects that only cover a narrow FoV, an alternative method to insert stereo objects to the target panoramic scene is to project the relevant part of the panorama to a 2D image plane and to then insert the object into the planar stereo image before projecting back. The composition methods of Luo et al. [29] and Tong et al. [44] are proposed for image cloning and composition in stereo 2D images. The latter method needs a substantial amount of user interaction. Therefore, we choose to compare with the method of Luo et al. that relies on mesh-based deformation to demonstrate the effectiveness of our stereo content manipulation and generation method. In Fig. 11, we show their composited stereo images and the 2D result generated using our deep depth densification and view-dependent content generation. Due to the limited capability of the mesh-based deformation method on the perspective changes of the object, our method can produce more realistic results when the desired orientation and relative position of the target object is notably different from the original capture.

Fig. 12 shows examples of both the one-off method and our method. Viewing the anaglyph stereo images, it is clear that our method produces only desirable horizontal disparities, while the one-off method produces incorrect results that include undesirable vertical disparities. We evaluate the perceived visual quality of the two methods in our user study.

5.5 Alternatives and additions to the algorithm

Pixel color inpainting is a useful technique in composition. In our case, it might lead to higher quality results, but it raises additional research questions. We have not applied any pixel inpainting technique onto our incomplete inserted objects because our focus is primarily on maintaining the composited object's disparity consistency across all view directions in the 360° images. If one did wish to apply inpainting, then a substantial unsolved issue is how to handle the inconsistency in the occluded regions for the left and right eyes. This inconsistency makes it crucial to prioritize creating a consistent color pattern in each view during any inpainting process. Any pattern mismatch between views could negatively impact the correctness of the disparities. Rather than use inpainting, we addressed the missing region color issue through our point cloud-based image warping technique that uses the predicted depth map to retrieve pixel color from corresponding positions.

We recognise that using the low-quality estimated depth map and alpha mask as input can result in imperfect final composited results. For example, if the depth estimation technique fails to accurately estimate the depth of the object, the foreground object might appear distorted or disconnected from its surroundings especially when its orientation is changed. Similarly, if the alpha matting process is not accurate, the edges of the foreground object might appear jagged or rough, resulting in an unnatural-looking composite image. To overcome these issues, we adopted one of the state-of-the-art depth estimation techniques [27] which delivers highly precise depth estimates for objects. Where this proves to be insufficiently accurate, we can encourage users to use some incorporated user interaction masking techniques to further improve the accuracy of the input mask.

In addition, we noted that low-quality depth maps can lead to floating points around the new shape after the object's orientation is changed. Our depth densification network denoises these floating points, which provides more tolerance for low-quality depth maps. We also include preprocessing techniques on the depth map that reduce noise points in the input. These additions to the algorithm enhance input accuracy so that our proposed system produces more realistic and satisfactory outcomes.

Finally, an alternative approach would be to use 3D point clouds rather than depth maps. However, we see several challenges with this approach. For instance, if we segment views based

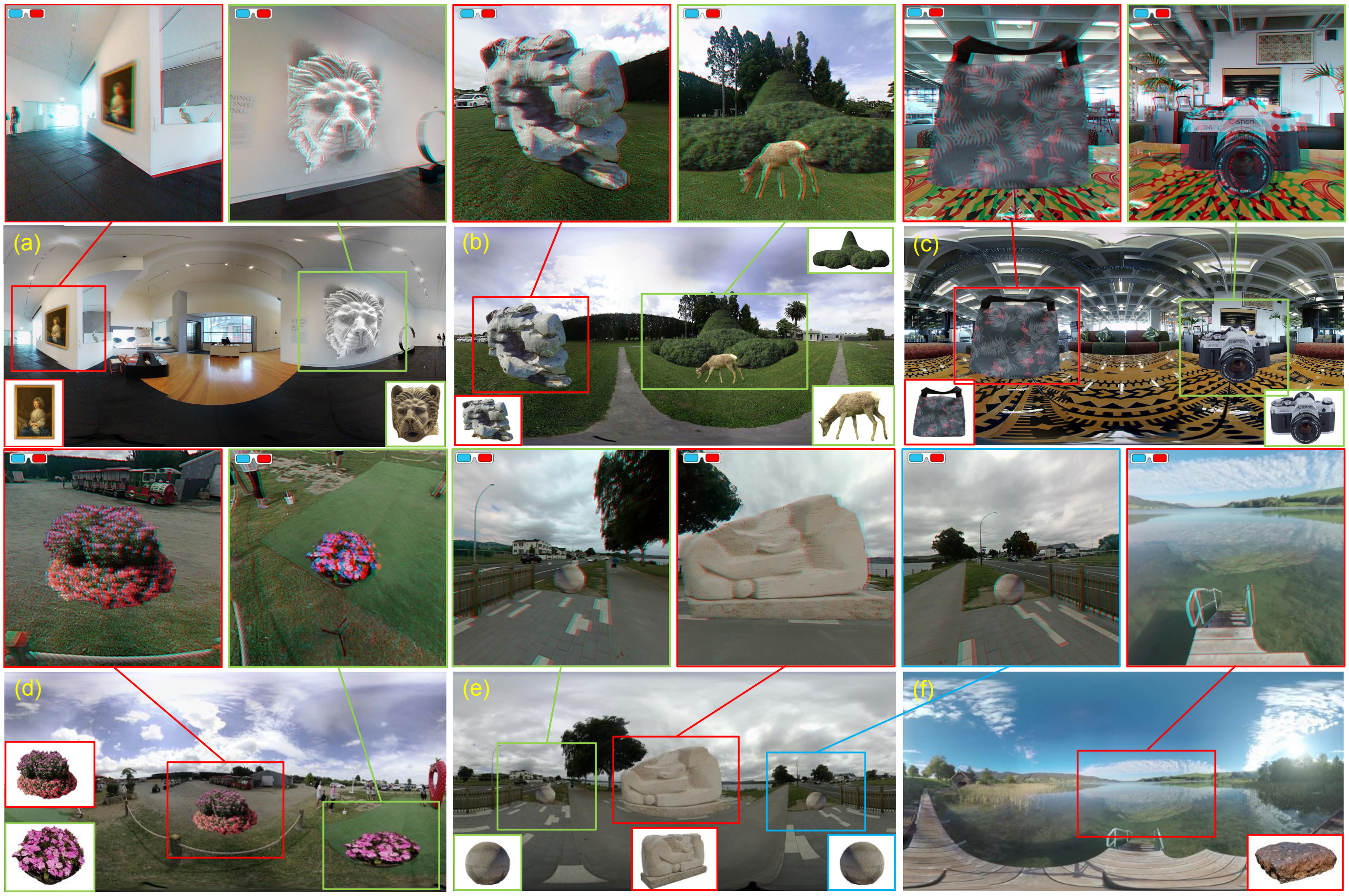


Fig. 9: Results of the proposed method. We show the anaglyph images of the zoom-in windows of the composited objects and only include the left view of the stereo panoramic composition results.

on the density of the point cloud, it could disrupt the disparity coherence of the inserted object in the final ODS image because the appropriate disparity for any given part of the inserted object should be determined by the user's viewing direction, rather than by the density of the point cloud itself.

The task of completing the depth map itself could be addressed by a point cloud upsampling technique. However, such upsampling algorithms primarily aim to enhance the global 3D structure and integrity of the inserted object. As an example, consider the method of Liu et al. [28]. It takes a point cloud of 256, 1024, or 4096 points, and densifies this by a factor of four, so a maximum of 16384 points. On the other hand, our depth densification network has a 512×512 depth map as both input and output. This is a considerably higher number of points than in the point cloud upsampling method. As a result, a complete single view of the inserted object is sufficient for us to reconstruct, and our approach can provide much finer details at the local level.

5.6 User Study

We conduct a user study to validate our method subjectively. We used ten panoramic images of different foreground objects. We generated two results for each scene: one result where we directly project all 3D points to the left and right equirectangular image to guide the pixel generation with a fixed pair of cameras (one-off) and the other result using our per-view projection, where we choose the FoV of each column of a 3840×2160 panoramic image as the interval of our view-segmentation, i.e., 0.09° There were 14 participants (six males, and eight females, aged from 25 to 52). After a short training session using two stereo scenes, the participants were asked to watch the ten groups of two stereo panoramas in Oculus Quest 2, and assess the two panoramas based on the visual quality by giving a score from 1-bad to 5-good. They were also asked to mark three positions where they found the visual qualities were most different between the two shown panoramas. Finally, they were asked to choose one of the two compared panoramas that had better visual comfort. We presented the foreground objects in front of a black background to avoid any confounding factors from a textured or image-based background.

Fig. 13 and Tab. 3 report the user study results. The per-view result achieves a much higher mean score than the one-off results which may be attributable to no ghost artefacts being perceived by participants, especially at the boundary or the regions with sharp and clear structures. This is consistent with our intuition that using a large number of segments with small intercepts can benefit the capture of foregrounds by rendering all parts of the foreground from appropriate eye positions. We performed a paired-sample t-test between the scores of the results of the one-off method and our approach. As shown in Tab. 3, the result indicates that the visual quality of our method is significantly better than the one-off method at a significant level $\alpha = 0.05$. Most of the participants reported that the regions far away from the composited object's center have noticeable quality differences. Depending on the textures and

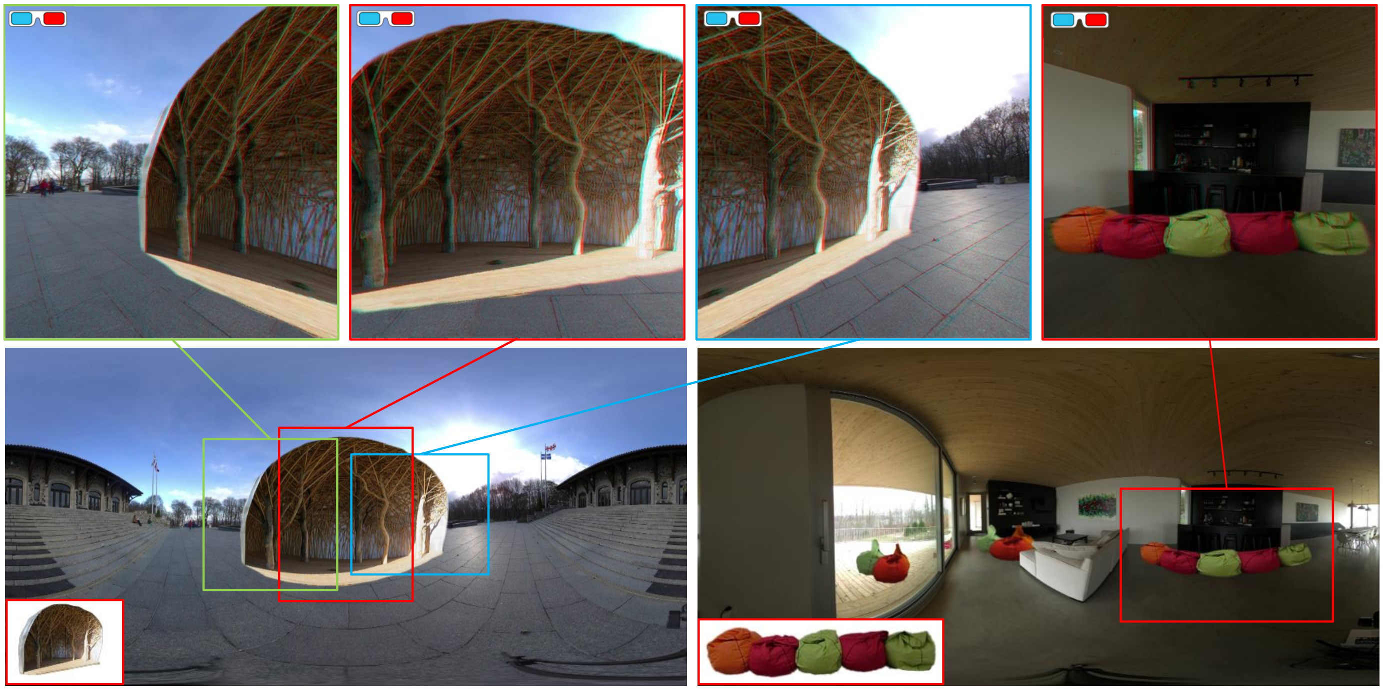


Fig. 10: Composition results with long objects inserted. Our method can allow arbitrary sizes of insertions covered in the target scene.

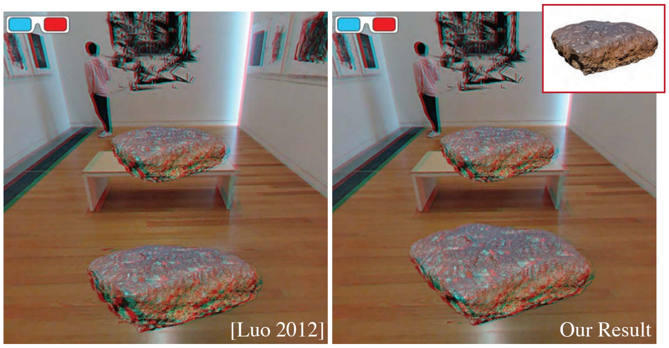
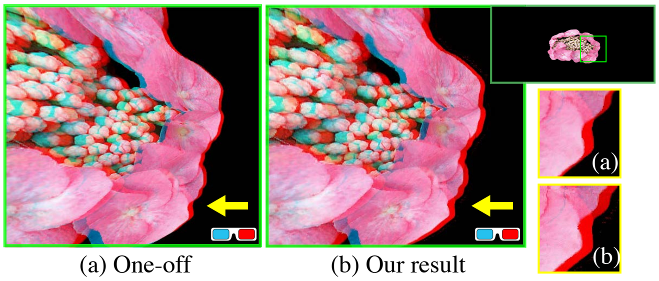


Fig. 11: Comparing our method with the 2D stereoscopic image composition method described in [29]. We inserted two rocks into the target scene at different depths and heights. Note that our method can adaptively adjust the 3D pose of the inserted object for a more natural result.

Score	Mean	Std	P-Value
One-off	2.807	0.847	5.607e-20
Per-column	3.714	0.742	

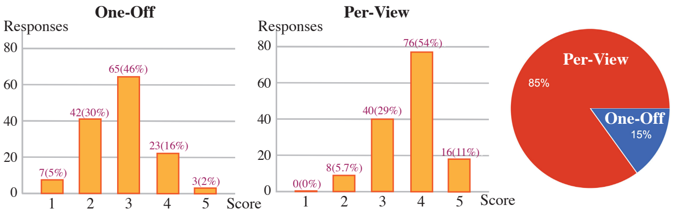
TABLE 3: Statistics of the user study results. The paired-sample t-tests are performed between the scores of the two methods.

colors, the participants might have a different level of sensibility to such a difference, which causes a minor variance in the reported positions. In terms of visual comfort preference, the preferred method is our per-view approach in 85% of the responses. For more information on the details and results of our user study please refer to our supplementary document. The above user experiments have been approved by the Human Ethics Committee of Victoria University of Wellington (ID: 0000025362).



(a) One-off (b) Our result

Fig. 12: Rectilinear views of synthesized stereo objects. It can be seen in the anaglyph images that the one-off method can not guarantee the horizontal disparities required for correct depth perception.



(a) Visual Quality Score Distribution (b) Visual Comfort

Fig. 13: The user study results. (a) The visual quality score distributions of the two methods (on scale of 1-5). (b) The visual comfort preference.

Limitations and Future Work The proposed approach has three limitations. First, if the depth estimation method fails to predict an accurate depth map of the source stereo object, our image generation method based on depth maps might not be able to produce satisfactory results when the user wants to change the object's 3D pose due to the incorrect 3D-to-2D projection. Second, our approach does not estimate the illumination of the target scene and the composited object. In future work, we will reconstruct the lighting information and the 3D geometry of the target scene to illuminate the object for better consistency. Finally, our current system relies on the user's input to decide the object's 3D pose and position. More advanced pose adjustment methods can be potentially employed to create more realistic results.

6 Conclusion

The goal of this paper is to address stereo 360° image composition with desired poses and positions of the inserted object, especially when the user composites an object with desired pose and scale that covers a large FoV into an ODS image. The goal has been achieved by developing a novel composition algorithm that keeps the basic 3D geometry of the composited object, while also achieving a high-quality depth perception for an arbitrary view in the panoramic scene. Particularly, a per-view projection method can make the composited content adapt to different view directions. The results show that the composited foregrounds can keep geometry information when the perspective, position or size of the object change. The user study demonstrates our method achieves the highest quality of depth perception when we make the per-view projection method with a fine segmentation.

Acknowledgments

This work was supported by Marsden Fund Council managed by Royal Society of New Zealand (No. MFP-20-VUW-180). Thanks to the Te Pataka Toi Adam Art Gallery for sharing their space with us for taking photos for Fig. 9(a) and Fig. 11.

References

[1] S. Athar, Z. Shu, and D. Samaras. Flame-in-nerf: Neural control of radiance fields for free view face animation. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pp. 1–8. IEEE, 2023.

[2] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of CVPR*, pp. 5470–5479, 2022.

[3] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):993–1008, 2003.

[4] 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.

[5] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua. Stereoscopic neural style transfer. In *Proceedings of CVPR*, pp. 6654–6663, 2018.

[6] R. Chen, F.-L. Zhang, S. Finnie, A. Chalmers, and T. Rhee. Casual 6-dof: free-viewpoint panorama using a handheld 360° camera. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2022. doi: 10.1109/TVCG.2022.3176832

[7] T. Chen, Z. Zhu, A. Shamir, S.-M. Hu, and D. Cohen-Or. 3-sweep: Extracting editable objects from a single photo. *ACM TOG*, 32(6):1–10, 2013.

[8] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski. A bayesian approach to digital matting. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. II–II. IEEE, 2001.

[9] W. Cong, J. Zhang, L. Niu, L. Liu, Z. Ling, W. Li, and L. Zhang. Dovenet: Deep image harmonization via domain verification. In *Proceedings of CVPR*, pp. 8394–8403, 2020.

[10] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *International Journal of Computer Vision*, 40(2):123–148, 2000.

[11] Y. Dai, H. Lu, and C. Shen. Learning affinity-aware upsampling for deep image matting. In *Proceedings of CVPR*, pp. 6841–6850, 2021.

[12] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of CVPR*, pp. 12882–12891, 2022.

[13] H. Ding, H. Zhang, C. Liu, and X. Jiang. Deep interactive image matting with feature propagation. *IEEE Transactions on Image Processing*, 31:2421–2432, 2022.

[14] S.-P. Du, S.-M. Hu, and R. R. Martin. Changing perspective in stereoscopic images. *IEEE Transactions on Visualization and Computer Graphics*, 19(8):1288–1297, 2013.

[15] Z. Farbman, G. Hoffer, Y. Lipman, D. Cohen-Or, and D. Lischinski. Coordinates for instant image cloning. *ACM TOG*, 28(3):1–9, 2009.

[16] Q. Feng, H. Shum, R. Shimamura, and S. Morishima. Foreground-aware dense depth estimation for 360 images. *Journal of WSCG*, 28(1-2):79–88, 2020.

[17] Q. Feng, H. P. Shum, and S. Morishima. 360 depth estimation in the wild-the depth360 dataset and the segfuse network. In *Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 664–673. IEEE, 2022.

[18] R. Fielding. *The Technique of Special Effects-Cinematography.[With Illustrations.]*. London & New York, 1965.

[19] T. Germer, T. Uelwer, S. Conrad, and S. Harmeling. Pymatting: A python library for alpha matting. *Journal of Open Source Software*, 5(54):2481, 2020. doi: 10.21105/joss.02481

[20] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke. A dataset and evaluation methodology for depth estimation on 4d light fields. In *Proceedings of the Asian Conference on Computer Vision*, pp. 19–34. Springer, 2017.

[21] J. Ku, A. Harakeh, and S. L. Waslander. In defense of classical image processing: Fast depth completion on the cpu. In *Proceedings of the Conference on Computer and Robot Vision*, pp. 16–22. IEEE, 2018.

[22] H. Laga, L. V. Jospin, F. Boussaid, and M. Bennamoun. A survey on deep learning techniques for stereo-based depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[23] J. Li, H. Li, and Y. Matsushita. Lighting, reflectance and geometry estimation from 360 panoramic stereo. In *Proceedings of CVPR*, pp. 10586–10595. IEEE, 2021.

[24] J. Li, J. Zhang, S. J. Maybank, and D. Tao. Bridging composite and real: towards end-to-end deep image matting. *International Journal of Computer Vision*, 130(2):246–266, 2022.

[25] Y. Li, C. Barnes, K. Huang, and F.-L. Zhang. Deep 360° optical flow estimation based on multi-projection fusion. In *Proceedings of the ECCV*, pp. 336–352. Springer, 2022.

[26] Y.-J. Li, J. Shi, F.-L. Zhang, and M. Wang. Bullet comments for 360° video. In *Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces*, pp. 1–10. IEEE, 2022.

[27] Z. Li, X. Liu, N. Drenkow, A. Ding, F. X. Creighton, R. H. Taylor, and M. Unberath. Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6197–6206, 2021.

[28] H. Liu, H. Yuan, J. Hou, R. Hamzaoui, and W. Gao. Pufa-gan: A frequency-aware generative adversarial network for 3d point cloud upsampling. *IEEE Transactions on Image Processing*, 31:7389–7402, 2022.

[29] S.-J. Luo, I.-C. Shen, B.-Y. Chen, W.-H. Cheng, and Y.-Y. Chuang. Perspective-aware warping for seamless stereoscopic image cloning. *ACM TOG*, 31(6):1–8, 2012.

[30] G. Ma, S. Li, C. Chen, A. Hao, and H. Qin. Stage-wise salient object detection in 360° omnidirectional image via object-level semantical saliency ranking. *IEEE Transactions on Visualization and Computer Graphics*, 26(12):3535–3545, 2020.

[31] P. Mandikal, K. Navaneet, M. Agarwal, and R. V. Babu. 3d-lmnet: Latent embedding matching for accurate and diverse 3d point cloud reconstruction from a single image. *arXiv preprint arXiv:1807.07796*, 2018.

[32] D. Martin, A. Serrano, A. W. Bergman, G. Wetzstein, and B. Masia. Scangan360: A generative model of realistic scanpaths for 360° images. *IEEE Transactions on Visualization and Computer Graphics*, 28(5):2003–2013, 2022.

[33] D. Medeiros, R. d. Anjos, N. Pantidi, K. Huang, M. Sousa, C. Anslow, and J. Jorge. Promoting reality awareness in virtual reality through proxemics. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pp. 21–30, 2021. doi: 10.1109/VR50410.2021.00022

[34] B. Mendiburu. *3D movie making: stereoscopic digital cinema from script to screen*. Routledge, 2012.

[35] H. Morioka, H. Okubo, and H. Mitsumine. A handy system for natural composition of cg and real scene with real-time reflection of lighting changes. In *Proceedings of the IEEE Virtual Reality*, pp. 231–232. IEEE, 2016.

[36] E. N. Mortensen and W. A. Barrett. Intelligent scissors for image composition. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, pp. 191–198, 1995.

[37] E. N. Mortensen and W. A. Barrett. Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60(5):349–384, 1998.

[38] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*, pp. 313–318. ACM New York, NY, USA, 2003.

[39] A. R. Smith and J. F. Blinn. Blue screen matting. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, pp. 259–268, 1996.

[40] S.-Y. Su, F. Yu, M. Zollhöfer, and H. Rhodin. A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose. *Advances in Neural Information Processing Systems*, 34, 2021.

[41] Y.-C. Su and K. Grauman. Learning spherical convolution for 360° recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8371–8386, 2022.

[42] C. Sun, M. Sun, and H.-T. Chen. Hohonet: 360 indoor holistic understanding with latent horizontal features. In *Proceedings of CVPR*, pp. 2573–2582, 2021.

[43] C. Tang, O. Wang, F. Liu, and P. Tan. Joint stabilization and direction of 360° videos. *ACM TOG*, 38(2):1–13, 2019.

[44] R.-F. Tong, Y. Zhang, and K.-L. Cheng. Stereopasting: interactive composition in stereoscopic images. *IEEE Transactions on Visualization and Computer Graphics*, 19(8):1375–1385, 2012.

[45] F. Tosi, M. Poggi, A. Benincasa, and S. Mattoccia. Beyond local reasoning for stereo confidence estimation with deep learning. In *Proceedings of the ECCV*, 2018.

[46] J. Unger, A. Wenger, T. Hawkins, A. Gardner, and P. Debevec. Capturing and rendering with incident light fields. Technical report, University of Southern California, 2003.

[47] F. A. Van den Heuvel. 3d reconstruction from a single image using geometric constraints. *ISPRS Journal of Photogrammetry and Remote Sensing*, 53(6):354–368, 1998.

[48] J. Wang, M. Agrawala, and M. F. Cohen. Soft scissors: An interactive tool for realtime high quality matting. *ACM TOG*, 26(3):9–es, 2007.

[49] N.-H. Wang, B. Solarte, Y.-H. Tsai, W.-C. Chiu, and M. Sun. 360SD-Net: 360° stereo depth estimation with learnable cost volume. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 582–588. IEEE, 2020.

[50] O. Wang, M. Lang, M. Frei, A. Hornung, A. Smolic, and M. Gross. Stereobrush: interactive 2d to 3d conversion using discontinuous warps. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*, pp. 47–54, 2011.

[51] A. Wenger, A. Gardner, C. Tchou, J. Unger, T. Hawkins, and P. Debevec. Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM TOG*, 24(3):756–764, 2005.

[52] X. Wu, X.-N. Fang, T. Chen, and F.-L. Zhang. Jmnet: A joint matting network for automatic human matting. *Computational Visual Media*, 6(2):215–224, 2020.

[53] K. Xian, J. Zhang, O. Wang, L. Mai, Z. Lin, and Z. Cao. Structure-guided ranking loss for single image depth prediction. In *CVPR*, June 2020.

[54] J.-P. Xu, C. Zuo, F.-L. Zhang, and M. Wang. Rendering-aware hdr environment map prediction from a single image. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 2857–2865, 2022.

[55] N. Xu, B. Price, S. Cohen, and T. Huang. Deep image matting. In *Proceedings of CVPR*, pp. 2970–2979, 2017.

[56] A. Yu, V. Ye, M. Tancik, and A. Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of CVPR*, pp. 4578–4587, 2021.

[57] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12498–12507, 2021.

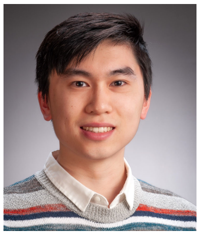
[58] L. Zhang, G. Dugas-Phocion, J.-S. Samson, and S. M. Seitz. Single-view modelling of free-form scenes. *The Journal of Visualization and Computer Animation*, 13(4):225–235, 2002.

[59] Y. Zhang, F.-L. Zhang, Y.-K. Lai, and Z. Zhu. Efficient propagation of sparse edits on 360° panoramas. *Computers & Graphics*, 96:61–70, 2021.

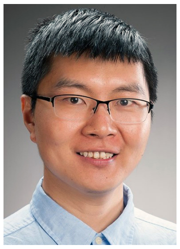
[60] Y. Zhang, F.-L. Zhang, Z. Zhu, L. Wang, and Y. Jin. Fast edit propagation for 360 degree panoramas using function interpolation. *IEEE Access*, 10:43882–43894, 2022.

[61] J. Zhao, A. Chalmers, and T. Rhee. Adaptive light estimation using dynamic filtering for diverse lighting conditions. *IEEE Transactions on Visualization and Computer Graphics*, 27(11):4097–4106, 2021.

[62] D. E. Zongker, D. M. Werner, B. Curless, and D. H. Salesin. Environment matting and compositing. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, pp. 205–214, 1999.



Kun Huang is currently a PhD candidate with Victoria University of Wellington, New Zealand. He received a Bachelor's and M.S. degree from Victoria University of Wellington in 2017 and 2021, respectively. His research interests include 360 image and video editing, virtual reality and mixed reality. He is a member of IEEE. He is a student branch chair of the IEEE New Zealand Central Section.



Fang-Lue Zhang is currently a senior lecturer with Victoria University of Wellington, New Zealand. He received a Bachelor's degree from Zhejiang University, Hangzhou, China, in 2009, and a Doctoral degree from Tsinghua University, Beijing, China, in 2015. His research interests include image and video editing, mixed reality, and image-based graphics. He is a member of IEEE and ACM. He received Victoria Early Career Research Excellence Award in 2019. He is on the editorial board of Computer & Graphics. He is a committee member of the IEEE New Zealand Central Section.

Junhong Zhao is currently a Research Fellow with the School of Engineering and Computer Science of Victoria University Of Wellington. She completed her doctoral degree in 2015 at the Institute of Electronics of the Chinese Academy of Sciences. She worked at the Institute of Information Engineering of the Chinese Academy of Sciences as an Assistant Researcher from 2015 to 2017. From 2018 to 2022, she was working with the Computational Media Innovation Centre (CMIC) at Victoria University of Wellington as a postdoctoral research fellow. Her research interests include machine learning, image processing and computer vision.

Yiheng Li received the M.S. degree in Computer Graphics in 2022 from Victoria University of Wellington, New Zealand. He is currently working at Sony China Software Center. His research interests include synthetic dataset generation, high-performance computing and panoramic image processing.

Neil Dodgson is Professor of Computer Graphics and Dean of Graduate Research at Victoria University of Wellington. His PhD is in image processing, from the University of Cambridge. He spent 25 years at Cambridge, becoming full professor in 2010. He moved to Wellington in 2016 to lead the computer graphics group there. His research is in 3D TV, subdivision surfaces, imaging, and aesthetics. He is a Chartered Engineer and a Fellow of Engineering New Zealand and of the Institution of Engineering and Technology (IET) and the Institute for Mathematics and its Applications (IMA) in the UK.