sensors

MDPI

Communication

Generating High-Quality Panorama by View Synthesis Based on Optical Flow Estimation

Wenxin Zhang [1], Yumei Wang [1,2,*] and Yu Liu [1,2]

[1] School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China; zhangwx@bupt.edu.cn (W.Z.); liuy@bupt.edu.cn (Y.L.)
[2] Research Center of Networks and Communications, Peng Cheng Laboratory, Shenzhen 518066, China
* Correspondence: ymwang@bupt.edu.cn

Abstract: Generating high-quality panorama is a key element in promoting the development of VR content. The panoramas generated by the traditional image stitching algorithm have some limitations, such as artifacts and irregular shapes. We consider solving this problem from the perspective of view synthesis. We propose a view synthesis approach based on optical flow to generate a high-quality omnidirectional panorama. In the first stage, we present a novel optical flow estimation algorithm to establish a dense correspondence between the overlapping areas of the left and right views. The result obtained can be approximated as the parallax of the scene. In the second stage, the reconstructed version of the left and the right views is generated by warping the pixels under the guidance of optical flow, and the alpha blending algorithm is used to synthesize the final novel view. Experimental results demonstrate that the subjective experience obtained by our approach is better than the comparison algorithm without cracks or artifacts. Besides the commonly used image quality assessment PSNR and SSIM, we also calculate MP-PSNR, which can provide accurate high-quality predictions for synthesized views. Our approach can achieve an improvement of about 1 dB in MP-PSNR and PSNR and 25% in SSIM, respectively.

Keywords: panorama stitching; view synthesis; optical flow

Citation: Zhang, W.; Wang, Y.; Liu, Y. Generating High-Quality Panorama by View Synthesis Based on Optical Flow Estimation. *Sensors* **2022**, *22*, 470. https://doi.org/10.3390/s22020470

Academic Editor: Alessandro Leone

Received: 22 November 2021
Accepted: 5 January 2022
Published: 8 January 2022

1. Introduction

Panoramas have been around for more than one hundred years. Their ability to render a scene in all directions has made them popular in the field of scene visualization and photography. With the rise of the VR (Virtual Reality) industry and the popularity of VR headsets, one of the key enabling elements in creating immersive VR content is panorama, which can provide a compact representation of the scene and more abundant information. Creating a panorama involves a special image stitching task. Image stitching aims at stitching the overlapping region of multiple views collected by multiple cameras with limited angles of view into a wide-angle seamless image, and the panorama requires a viewing coverage of 360°.

Classical image stitching involves the following steps. Starting with image preprocessing, including geometric alignment and photometric correction, the projected transformation model (a homography matrix, generally) between a pair of images is established after feature extraction and feature matching. Then, the projected transformation model is utilized to align one image to another. Finally, the blending algorithm is used to provide the final results. Brown et al.'s AutoStitch [1] was the core algorithm used in commercial software. This used a global homography matric to project the image onto a cylindrical or spherical surface, then synthesized the images using a multi-resolution fusion algorithm. The global homography projection assumed that the overlapping area of the images lay in the same depth plane. Otherwise, the global homography matrix could not align points on different depth planes, which is also the cause of artifacts. Therefore, the APAP [2] of

Zaragoza et al. divided the image into dense grids. Each grid was mapped to the canvas of the final stitched image by using the local projection transformation., but this was only suitable for scenes with small parallax. Similar works, such as SPHP [3] by Chang et al., proposed a shape-preserving method from the perspective of shape correction and combined homography warp and spatial similarity to achieve perfect alignment in the overlapping area while maintaining the original viewing angle in non-overlapping areas. Lin et al.'s AANAP [4] also aimed at the shape correction problem of grid distortion, but discarded the constraint term and used global similarity transformation to correct the shape, improving the naturalness and viewing experience of the stitched image. Some methods attempted post-processing in order to eliminate structural distortions such as ghosting or truncation (structural discontinuity) in the overlap area caused by inaccurate registration parameters. The seam-guided [5] of Lin et al. tried to find an optimal seam to partially align the area near the seam to mitigate the distortion caused by unnatural fusion. However, seam-cutting has a huge impact on moving objects. When moving objects pass through the seam, their structures will be broken easily or ghosts will appear around them.

Completely different from traditional stitching methods, since VR capture devices are well-designed (with a regularly arranged camera array), regarding the stitching problem as a viewpoint synthesis problem is more suitable for the generation of high-quality panorama. In previous works on view synthesis, Thatte et al. [6] established a probability model of possible missing points to minimize the disocclusion holes in synthesizing novel views in order to solve the missing areas in the output image of the depth-based view synthesis method. In ref. [7], Zhang et al. addressed view synthesis from a single image. They reorganized the pixels of input view and learned the stereoscopic structure in the multi-scale feature map to synthesize the target view through the learning framework of structure awareness without information about the scene structure, such as depth. The authors of [8] proposed a view-dependent flow-based blending method to generate panoramas with motion parallax in real time.

Motivated by existing works, we propose an omnidirectional view synthesis approach based on optical flow to generate panoramas. In addition to necessary geometric correction and exposure adjustment, our method only relies on the pair of inputs to obtain a synthetic view without scene information or camera parameters. As for feature matching methods, camera intrinsics and extrinsics need to be accurate as they will affect the quality of the stitching. Theoretically, the flow-based method of stitching actually forms a unique projection model for each pixel located in the overlapping area. Therefore, even if the ratio of the overlapping area is insufficient, it will neither affect the novel view results nor destroy the structure of the non-overlapping area.

In this paper, we systematically review existing methods of image stitching and analyze their limitations with respect to panorama generation. According to this goal, we propose a method of panorama generation that is based on optical flow estimation and view synthesis. In detail: (1) we propose a novel optical flow estimation algorithm to obtain translation between views. The obtained flow is essential for subsequence reconstruction and blending. (2) We use the optical flow field to reconstruct the left and right views and combine distance weights and flows with the alpha blending algorithm to synthesize the novel view, which ensures high-quality panorama construction. (3) In the experiments, we show that the proposed approach outperforms previous methods in subjective experience and image quality assessment with the stitched panoramas generated by our method.

2. Related Work

2.1. Panorama Stitching

Panorama stitching is a technology that stitches images taken from different perspectives together to form a panorama. As for the stitching of an image pair, some homography-based methods [9,10] have been proposed. In these methods, a homography matrix is mainly designed to solve the problems of perspective distortion and shape distortion. In addition, the content-preserving warping is introduced to improve the poor correspon-

dences of low-texture regions in some studies [3,11]. In these studies, the images are first divided into a uniform dense grid mesh. Then, mesh-based warping and optimization are conducted by adding global similarity prior or local similarity transformation to obtain a more accurate alignment. Finally, the overlapping regions of the warped images are blended to obtain a smooth seamless stitched image.

Compared with traditional image pair stitching technologies, panorama stitching improves stereoscopic perception. Therefore, the challenging problem we need to address is that the stitched method can achieve artifact-free within reasonable parallax range to ensure the final panorama is more realistic. In ref. [12], a solution for generating stereo panoramas at a mega pixel resolution was presented and a flow-based up-sampling method was used to resolve the issue of stitching artifacts. Peleg et al. [13] proposed two optical omni-stereo panorama systems to capture images from different perspectives. The left and right panoramas could be spliced by obtaining multiple strips from the camera. While these works also focus on the perception of stitched panorama, flow estimation is at the root of solving parallax and artifacts.

2.2. Image-Based Rendering

Image-based rendering aims to synthesize a new viewpoint image of a scene from an input image sequence. Different methods can be used to obtain the information of the target view from the input in different ways. MPEG (Moving Picture Experts Group) divides view synthesis into two categories and gives the official solution VSRS [14]. One of these categories is image-based rendering (IBR), which refers to images from multiple viewpoints. Using 3D-Warping projection, view fusion, interpolation, and other technologies, images from virtual viewpoints can be directly generated [15–18]. Another method is model-based rendering (MBR), which needs to build an accurate 3D model of a real scene. If a model is built successfully, images can be obtained from any viewpoint [19–22].

The view synthesis method based on images [15] does not need scene geometry information, but the acquisition equipment of its inputs requires a regular and dense camera grid and the generation of the target view is usually a linear blend of inputs. Chaurasia et al. [16] estimated the depth of each view, mapped the color information and blending weight to the target view according to the depth, and compensated for the inaccuracy of the depth map by the super-pixel method. Zhou et al. [17] introduced a multi-plane image representation that was estimated by a convolutional network for stereo magnification. The image was represented over multiple RGB-α planes, where each plane was related to a certain depth. Novel views can be rendered using back-to-front composition based on this given representation. Flynn et al. [18] used a plane-sweep volume within a network architecture for image-based rendering. A color branch predicted the color values for each depth plane in the target view and another branch predicted the probability.

Some view synthesis methods need to rely on the proxy geometry structure of the scene. Kopf et al. [19] generated 3D proxy geometry by structure-from-motion (SfM) and multi-view stereo (MVS) and achieved a view of the target viewpoint by optimizing Markov Random Field. Aliev et al. [20] described Neural Point-Based Graphics, in which each 3D point was associated with a learned feature vector. These features were segmented into a target view and transformed through a rendering network to synthesize a novel view. However, the feature extractor needed to be re-trained when it was applied to a new scene. Thies et al. [21] used mesh instead of 3D points to embed feature vectors. Sitzmann et al. [22] avoided explicit proxy geometry and projected source images into a neural voxel grid, where each voxel was associated with a trainable feature vector.

In this paper, an optical flow-based view synthesis method is proposed for panorama stitching. In the proposed method, an optical flow estimation algorithm is designed by considering the consistency between pixels of the same world point, and the flow-adjusted version of the left and right views is reconstructed to reduce vertical disparity. Furthermore, we combine optical flow with an alpha-based blending algorithm to synthesize the target

view, which ensures a smooth seamless stitched image. Experimental results show that the proposed method performs better in terms of the visual experience.

3. The Proposed Method

3.1. System Overview

Our method starts with captured data. We use images collected by two products on the market—i.e., Insta360 pro and Facebook surround 360 [23]—to generate our experimental datasets. First of all, because the captured images are affected by the lens of cameras, distortion correction is a necessary pre-processing step in stitching. Then, we need to roughly estimate the overlapping area of the left and right images based on the camera structure. Secondly, we propose a novel optical flow algorithm to calculate the dense optical flow in the overlapping area to achieve pixel-level image matching and use this to reconstruct the left and right views used in the subsequent blending phase. Finally, we combine the flow-based weight with the alpha blending algorithm to synthesize the novel view. We illustrate the steps of our approach in Figure 1, some details will be given in Sections 3.2 and 3.3.

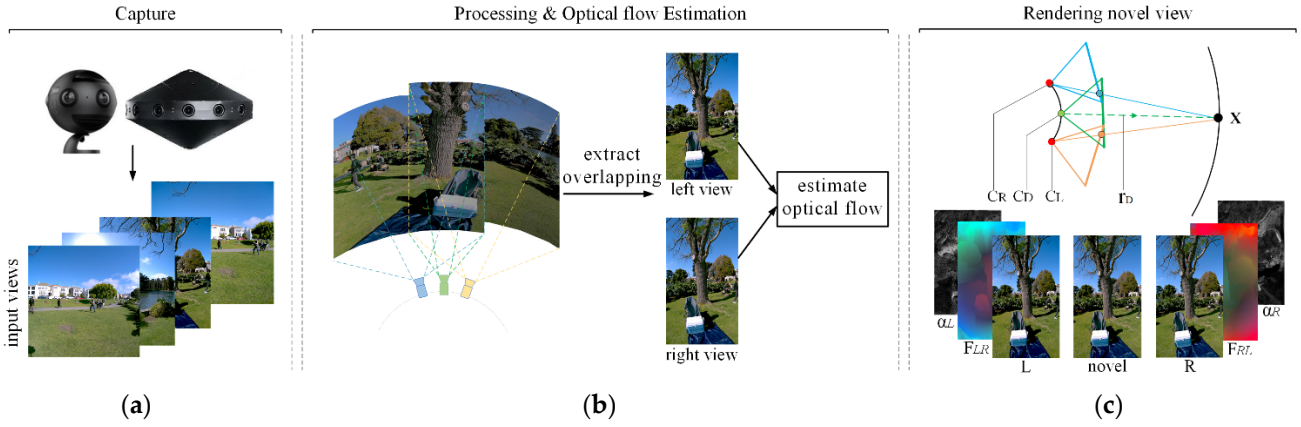

Figure 1. Overview of our approach. (**a**) Capture: experimental data are acquired by these two devices. (**b**) Processing and optical flow estimation: we divide the approximate projection area according to the angular space and calculate the optical flow of each pair of adjacent cameras to establish a dense correspondence (in Section 3.2). (**c**) Rendering: a blending algorithm combining an optical flow and alpha map is proposed to synthesize the view of novel viewpoints (in Section 3.3).

3.2. Optical Flow Estimation

Image-based rendering (IBR) aims to enable the synthesis of novel views of a scene directly from a set of input images. In this process, there are two factors that determine the quality of the synthetic view. First, a center pixel and a small block around it will be mapped to a new position along with the displacement vector of the center pixel, and the mapping error of pixels will cause cracks, holes, or noise. Secondly, due to the occlusion, the background that was originally occluded by the foreground becomes visible in the target view. The question of how to accurately calculate the displacement vector of the same scene point and map it to the corresponding position of the novel view and how to sample the left and right views to make the synthesized view perceptually indistinguishable from reality are two issues that need to be solved. In this paper, we utilize ab optical flow to synthesize the novel view.

Optical flow can detect the pixel motion of moving objects and can be widely used in the field of moving object detection and tracking. There are objects with different depth values in the scene described by a set of input images, which makes the relative position of objects in the same spatial position different in various views. We can use optical flow to describe this change. From the two input images I_L and I_R, we synthesize the image I_D of the desired view relying on optical flow. We use the optical flow field to approximate the displacement between two views, and the flow field can be well approximated as the

inverse depth (parallax) of the scene. Optical flow estimation can be regarded as point-by-point per-pixel matching. It returns a displacement vector for each pixel of the original image and maps it to a new position in the reference images. The pixel matching process is depicted in Figure 2.

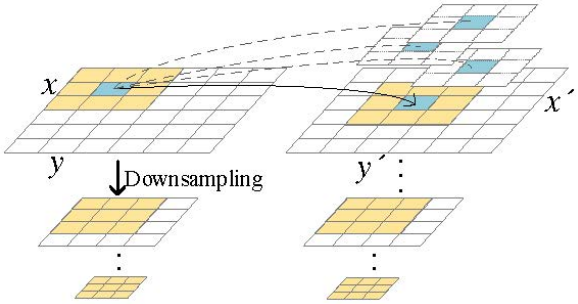

Figure 2. The patch search process in optical flow estimation. The best match is determined by minimizing the P_{error} of a small block around the pixel, and the pixel (x, y) in the current view is mapped to the (x', y') pixel position of the adjacent view.

First, we separate the gray components and alpha components of the input images (.png format images contain alpha channels; other formats need to be converted into .png). The alpha maps of two images are used to adjust the intensity of the image (gray components) to further alleviate the spatial gray difference between the two images after exposure adjustment. Additionally, this process is explained by Equation (1). Next, we downsample the images by scaling the image height and width by a factor of 2 as the inputs of the image pyramid structure to make this process faster and more stable. By gradually downscaling the image, the basic assumption of optical flow with small displacement is satisfied so as to deal with the large motion of objects in the scene. Finally, we perform patch searching for both gray and alpha pyramids at the same time from the top level of the pyramids with the lowest resolution. Here, we set the size of the patch to 2 N +1, setting N to 2. We calculate the intensity error of the pixel value in the patch recorded as P_{error}, as depicted in Equations (2)–(4):

$$I_R = I_R(\,x, y\,) \times \frac{\sum I_L(x,y) \times \alpha_L(x,y) \times \alpha_R(x,y)}{\sum I_R(x,y) \times \alpha_L(x,y) \times \alpha_R(x,y)} \tag{1}$$

$$I_{abs} = \sum_{v=-N}^{N} \sum_{u=-N}^{N} |I_L(x_L + u, y_L + v) - I_R(x_R + u, y_R + v)| \tag{2}$$

$$\alpha = \sum_{v=-N}^{N} \sum_{u=-N}^{N} \alpha_L(x_L + u, y_L + v) \times \alpha_R(x_R + u, y_R + v) \tag{3}$$

$$P_{error} = \frac{I_{abs}}{\alpha} \cdot [1 + 2 \times L_2(x_R - x_L, y_R - y_L)] \tag{4}$$

I_{abs} is calculated as the sum of absolute differences (SAD) of intensity in the search box; α is the dot product of alpha values in the patch; P_{error} is calculated as the second norm of displacement vector in the x direction and y direction plus the ratio of SAD to alpha, which matches the pixel characteristics in the patch more accurately.

Taking the occlusion caused by parallax and the different orientation of the image into account, only searching within the 5×5 tile does not enable us to obtain the correct correspondence for nearby objects. We expand the search box as a rectangle to update the initial match. The rectangle extends ortho to each side of the search direction (left or right), and the scope around $I_L(x, y)$ is limited to:

$$Rect\left(-k, -\left(\frac{k}{kRatio} + \frac{1}{2}\right), k + 1, 2\left(\frac{k}{kRatio} + 1\right)\right) \tag{5}$$

In Equation (5), the four items of *Rect* are the coordinates of the upper left point, width, and height of the searchbox, respectively. Here, k is a constant ranging from 0 to 1 that determines the degree of expansion and *kRatio* determines the aspect ratio of the search box. Rectangular search considers the depth varies with scene content, which is of benefit to the foreground content (as seen in Figure 3). We use the best match with the smallest error in the expanded box, which signifies the best correspondence. The displacement vector of the point (x, y) is described as (u, v), which is treated as flow.

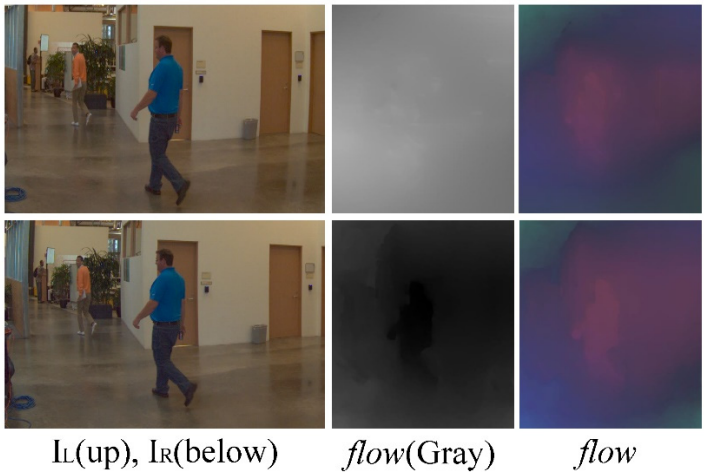

Figure 3. Rectangular search can better detect the foreground and keep its outline. The upper are the optical flows obtained by a 5×5 patch, and the lower are the results of the extended rectangular search box. A color wheel is used to color the gray version of the flow to give an obvious effect.

After patch searching, we obtain the primary matching of pixels depending on the similarity of the gray neighborhood. However, the gray variation is not obvious for the smooth areas (there may be some silhouettes and sloped surfaces) in the image. In order to maintain this edge information, more advanced features besides gray information are added in the process of calculating the matching error to improve the primary flow field. Gradient norm errors in the x and y directions of images are considered. We sweep the flow from two directions to obtain the final optical flow field F. The description of the details is given in Algorithm 1. In this step, we introduce the adjacent points at the four positions of up, down, left, and right to finally determine whether the current displacement vector indicates the optimal pixel correspondence. When obtaining the final optical flow field F, the transparency information of images is used to weight the flow and its gaussian blur version so as to improve the robustness of the optical flow estimation.

Algorithm 1. Calculate the optical flow field.

Input: I_L, I_R—the grayscale images of the left and the right views
α_L, α_R—the alpha maps of the left and the right views
dx_L, dy_L, dx_R, dy_R—gradients in two directions
$flow$, $blowflow$—the flow and its gaussian blur version
Output: F—the final optical flow field

function errorfunction$(x, y, flow.at(x, y))$
  $(x', y') \leftarrow (x, y) + flow.at(x, y)$
  $(i_L x, i_L y) \leftarrow (dx_L.at(x, y), dy_L.at(x, y))$
  $(i_R x, i_R y) \leftarrow (dx_R.at(x', y'), dy_R.at(x', y'))$
  $diff \leftarrow blurflow.at(x, y) - flow.at(x, y)$
  $error = \|(i_L x - i_R x, i_L y - i_R y)\|_2 + \sqrt{diff \odot diff}$

```
7.  end function
8.  for x from 0 to I_L.width do
9.      for y from 0 to I_L.height do
10.         currentError ← errorfunction(x, y, flow.at(x, y))
11.         (0,1), (0,−1), (−1,0), (1,0) successively assign to (i, j)
12.         error ← min[errorfunction(x, y, flow.at(x + i, y + j))]
13.         while error < currentError do
14.             flow.at(x, y) ← flow.at(x + i, y + j)
15.         end while
16.     end for
17. end for
18. k = 1 − α_L.at(x, y) × α_R.at(x, y)
19. F(x, y) ← (1 − k) × flow.at(x, y) + k × blurflow.at(x, y)
```

Algorithm 1 describes the calculation of the optical flow from the left view to the right view. Taking I_L as a reference, we need to find the correspondence in I_R and convert it accordingly to obtain the optical flow in the other direction. Figure 4 shows the result of our optical flow calculation. From the left to right are the left view, F_{LR}, F_{RL}, and the right view. We use colorwheel to color the displacement vector because the optical flow field describes the displacement vectors from two opposite directions and shows the difference in the color temperature.

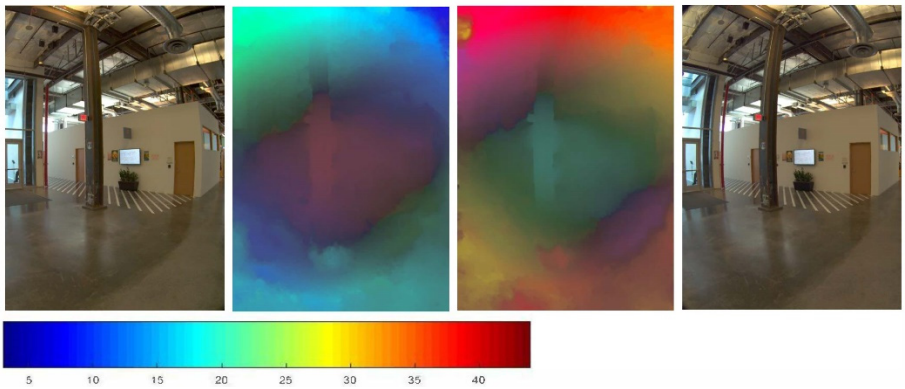

Figure 4. The optical flow approximates the inverse depth of the scene. A color bar of the optical flow field is shown below, and the value represents disparity in the x direction.

3.3. Reconstructed View-Based Blending Algorithm

Due to parallax and camera orientation, the overlapping areas of the left and the right images are not perfectly matched and blending directly based on distance weights will cause artifacts. By calculating the optical flow, we obtain the displacement between the overlapping area on the 2-dimensional plane. In order to synthesize the image of the target viewpoint, we consider obtaining an intermediate pattern depending on the optical flow to improve the corresponding accuracy of the overlapping area. Based on the intermediate pattern, we can synthesize the image of the target view I_D by using the spatial position relationship and image transparency; the geometric model is described in Figure 5.

The world point X projects to x_L and x_R in the left and the right views, respectively. We interpolate a novel viewpoint D and synthesize its imaged view I_D, where x_D is the scene point X projected in I_D. The dotted blue and yellow lines on the left in Figure 5 depict the plane-induced depth mismatch. We use optical flow fields F_{LR} and F_{RL} to reconstruct the left and the right views to compensate for the truncation and artifacts caused by this mismatch. We adjust the optical flow amplitude based on the position information to

determine the coordinates of sampling points x_L^* and x_R^*, which are needed for image reconstruction. The amplitude adjustment parameter δ is as follows:

$$\delta = \text{sigmod}\left(\frac{scale}{W} \times \left(pos - \frac{W}{2}\right)\right) \tag{6}$$

where *scale* is an adjustable parameter that controls the degree of change in the linear region of the sigmoid function curve. The larger the *scale* is, the more the displacement of the optical flow field plays a role in the sampling coordinates. *W* refers to the pixel width of the image, and *pos* refers to the column position of the pixel. For the left view, the left part is only slightly tuned by the F_{LR} according to the value range of the sigmoid function and is almost the same as the left side of the original image. Additionally, the right part is tuned a lot dependent on F_{LR}. The coordinates of the left and the right views warped by optical flow are:

$$x_L^* = x_L + \delta \times F_{LR}(x_L) \tag{7}$$

$$x_R^* = x_R + (1 - \delta) \times F_{RL}(x_R) \tag{8}$$

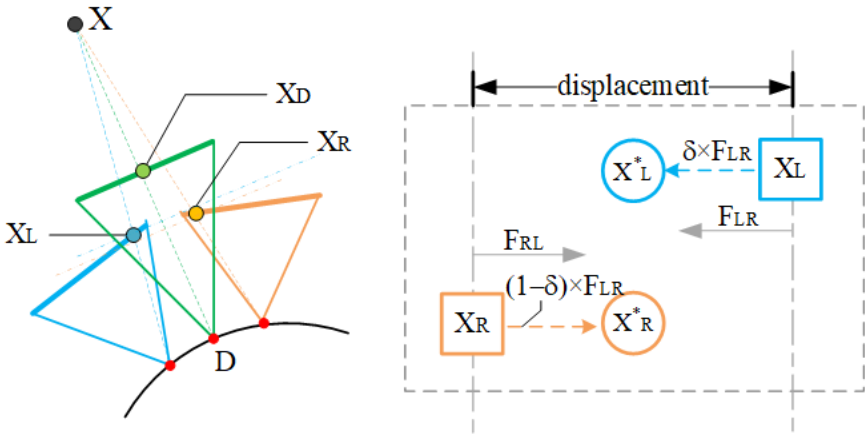

Figure 5. Get the reconstructed views by warping pixels depending on optical flow.

Figure 6 shows the left and the right views after optical flow adjustment. The top and the bottom of the first column are the original left and right views, and the second column is the value map of the adjustment function according to amplitude adjustment parameter. The brighter part indicates that the original view is adjusted a lot by the optical flow. The third column is the reconstructed left and right viewpoint images, depicting that the vertical distortion with red line is corrected after the optical flow adjustment. The last column is the synthesized view.

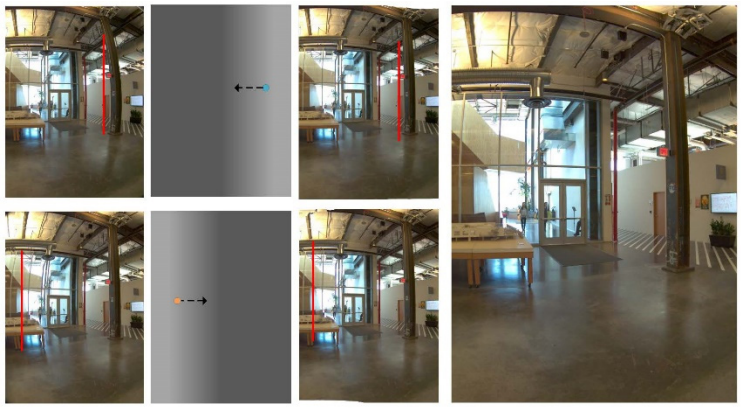

Figure 6. Optical flow adjustment and the synthesis of the novel viewpoint.

Vertical distortion is most noticeable for nearby scene objects and affects the viewing experience of users. Additionally, the synthesized image combines views from multiple perspectives, which leads to distorted scene objects in novel viewpoints. We use the alpha value, which indicates the transparency of the pixel, to blend the reconstructed views which can tolerate a certain degree of parallax. If the transparency is effectively controlled, the artifacts can be suppressed to a certain extent.

The alpha values of the left and the right images are represented by α_L and α_R, respectively. Here, we take $L_1 = \delta$ and $R_1 = 1 - \delta$ as the optical flow weights. In fact, L_1, R_1, and the alpha value of the pixel are employed to obtain the base K_L and K_R of the transparency weight value represented by L_2 and R_2, as shown in Equations (9) and (10):

$$K_L = e^{(k \times L_1 \times \alpha_L)} \tag{9}$$

$$K_R = e^{(k \times R_1 \times \alpha_R)} \tag{10}$$

where k is an adjustable coefficient. According to the alpha blending algorithm, the sum of the transparency weights of the two images is 1. Therefore, L_2 and R_2 are obtained according to Equations (11) and (12):

$$L_2 = \frac{K_L}{K_L + K_R + \varepsilon} \tag{11}$$

$$R_2 = \frac{K_R}{K_L + K_R + \varepsilon} \tag{12}$$

where ε is an extremely small positive number.

Finally, we use the difference in the absolute value of the pixel at the same position, which is referred as d, to evaluate the importance of these two weights. Additionally, we use *ratio* to express this importance:

$$ratio = \tanh(d) \tag{13}$$

where tanh ensures that when the difference of pixel values is large, the transparency weights of the image play a leading role in the process of blending, and we are able to obtain a compromise between the optical flow weight and alpha weight. Then, we obtain the final blending weights of the reconstructed left and right images according to Equations (14) and (15)

$$w_L = L_1(1 - ratio) + L_2 \times ratio \tag{14}$$

$$w_R = R_1(1 - ratio) + R_2 \times ratio \tag{15}$$

To synthesize the novel view I_D, we combine the pixels sampled at x_L^* and x_R^* from I_L and I_R, respectively. Additionally, we perform a convex combination:

$$I_D(x_D) = w_L \times I_L(x_L^*) + w_R \times I_R(x_R^*) \tag{16}$$

Note that $w_L + w_R = 1$ according to the fusion rule of alpha blending.

4. Experimental Results

4.1. Datasets

Since there is no benchmark dataset in the field of panorama stitching, we use real image datasets 'scene1' and 'scene2' captured by the Facebook surround360 device and 'fisheye' from the Insta360 device to build a panorama and use the public light field datasets HCI and 'Teddy' from Middlebury [24,25] to conduct the comparison between methods. In the ablation experiments, the stereo/flow 2012 dataset of KITTI [26] is used to evaluate each component of the algorithm. The camera array of Surround360 is composed of 14 cameras on the fixed camera rig with the same intrinsics at equal intervals. The optical centers of the cameras are in the same horizontal plane. The resolution of the images is 2048 × 2048.

The Insta360 camera rig is arranged by 6 fisheye cameras. We expand the fisheye images and extract the central area as the inputs according to the Equi-Rectangular Projection (ERP) projection theory, where the size of images is 1360 × 2042. Each scene of the HCI light field dataset is composed of 81 images by a 9 × 9 camera array with a resolution of 768 × 768. Each scene in the Middlebury dataset is composed of 6 views, and the resolution is larger than 600 × 500. The KITTI dataset contains stereo pairs and is from the real scene of autonomous driving, which is commonly used to evaluate vision algorithms.

Scene1 is a building interior scene that does not contain many complex textures but rather quantities of vertical or horizontal object edges. A good method should keep the original structure in the vertical direction. Scene 2 is a spacious outdoor scene with complex texture objects such as grass and branches. This kind of texture is prone to unsatisfactory stitching distortion artifacts, which poses a great challenge to the effectiveness of the methods. Moreover, Insta360 is an indoor scene with varying depth, and the images are in the ERP projection format, which means that the scene is deformed. Light field data [24] are acquired by a structured camera array and can be seen as a narrow baseline image pair. On the contrary, 'TEDDY' is an image pair with a wide baseline. KITTI data are collected by self-driving vehicles, usually including buildings, vehicles, roads, and trees. These real scenes have covered most of the complex textures to evaluate the performance of our method comprehensively.

4.2. Ablation Study

Flow Estimation (FE): The first step of our approach is optical flow estimation, which represents the correspondence between the left and right views. To verify the importance of this module, we compare the performance in the case of 'w/o FE' with 'Ours' in Table 1 and Figure 7. Concretely, we implement the optical flow part of surround360 [23] to calculate optical flow and replace our proposed optical flow estimation in the 'w/o FE'. From the comparison of the subjective results, it can be seen that the two are close. However, our method keeps the shape at the vertical and horizontal edges of the second scene. Therefore, 'Ours' can achieve a higher peak signal to noise ratio (PSNR) and structural similarity (SSIM), which means that our results have a lower noise error and higher structural similarity, as shown in Table 1.

Table 1. Ablation studies on flow estimation by Surround 360 [23], flow-based reconstruction, and flow-based blending.

Pipeline	MP-PSNR	PSNR	SSIM
w/o FE	26.7732	21.5039	0.6301
w/o FR	24.6985	18.4631	0.5833
w/o FB	25.9854	18.7952	0.5711
Ours	**28.0473**	**23.2075**	**0.7027**

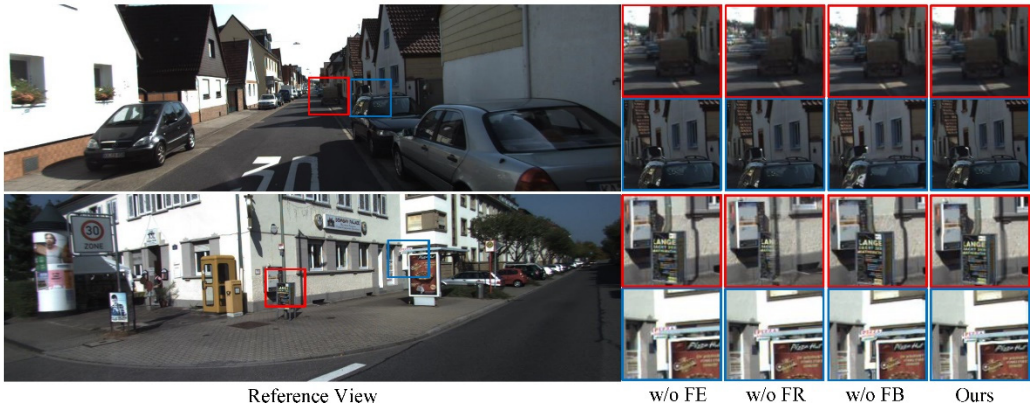

Figure 7. Taking the right view as the reference view for the ablation experiments to verify the effectiveness of flow estimation (FE), flow-based reconstruction (FR), and flow-based blending (FB).

Flow-based reconstruction (FR): Due to parallax, the baseline and rough estimation of the overlapping area, blending directly based on distance will lead to bad results. Note that we have calculated the displacement between pixels through optical flow estimation.

We first adjust the optical filed depending on the positional information and then fine-tune the two views using the adjusted flow field to obtain an intermediate pattern between the left and right views that can reduce artifacts in subsequent blending. To verify the necessity of this process, we eliminate the process of flow adjustment in the case of 'w/o FR'. As illustrated in Figure 7, 'w/o FR' shows a discontinuity of structure in the part circled in red.

Flow-based blending (FB): In this work, we utilize the alpha value to design our blending algorithm. The alpha map represents the visibility of each pixel of an image and is widely used in view synthesis. In this paper, a dual-weight blending algorithm based on optical flow is designed to determine the weights of the left and the right views in the final sampling. Comparing our method with the traditional alpha blending algorithm 'w/o FB', we find that the results of 'w/o FB' and 'Ours' are almost similar in some local parts. While there is a certain degree of deformation in the parts circled in blue in the first scene and red of the second scene in 'w/o FB', this can be controlled by our method, as shown in Figure 7.

4.3. Viewing and Analysis

For datasets that cannot form a panorama, subjective synthetic view results are displayed to ensure the rationality of the experiment in Figure 8. For the datasets that can form the panorama, we show the formed panoramas and calculate the image quality assessment metrics over the image sequence in addition, as shown in Table 2 and Figure 9. We compare our algorithm with other methods, such as APAP [2], AANAP [4], and the view synthesis method SM [17], from the left to the right in Figure 8. The first two rows are the contents of scene1, the next row is scene2, and the last scene is fisheye data. The next scene is HCI light-field data from [24], and the last scene is TEDDY from the Middlebury datasets [25]. The values under the figures are PSNR and SSIM, respectively.

In scene1, our method achieves better performance on the vertical edges of the building, while other methods show ghost and stitching distortions. In the fisheye scene, the contrast methods have artifacts and truncation effects on the green baffle and the black cable. On complex textures such as tree branches, our approach can also perform well without artifacts. While SM [17] obtains the best performance on TEDDY, our approach shows an irregular edge due to the large parallax that optical flow adjustment cannot overcome. Figure 1 shows the panoramas generated by our approach on three panoramic datasets.

We also calculate image quality assessment metrics on scene1, scene2, and fisheye data, as shown in Table 2. The difference between the values of a single image pair may be relatively large, so we evaluate the metrics on the whole dataset. The result of the single image pair is shown below the picture because it cannot form a complete panorama. We

calculate MP-PSNR [27], PSNR, and SSIM to evaluate these approaches. The SSIM value gain of our method is close to 0.1 in scene1, while the MP-PSNR and PSNR gains are close to 1dB in the fisheye scene. Experimental results show that our results are better than the comparison methods to a certain extent in terms of both subjective visual perception and objective assessment metrics.

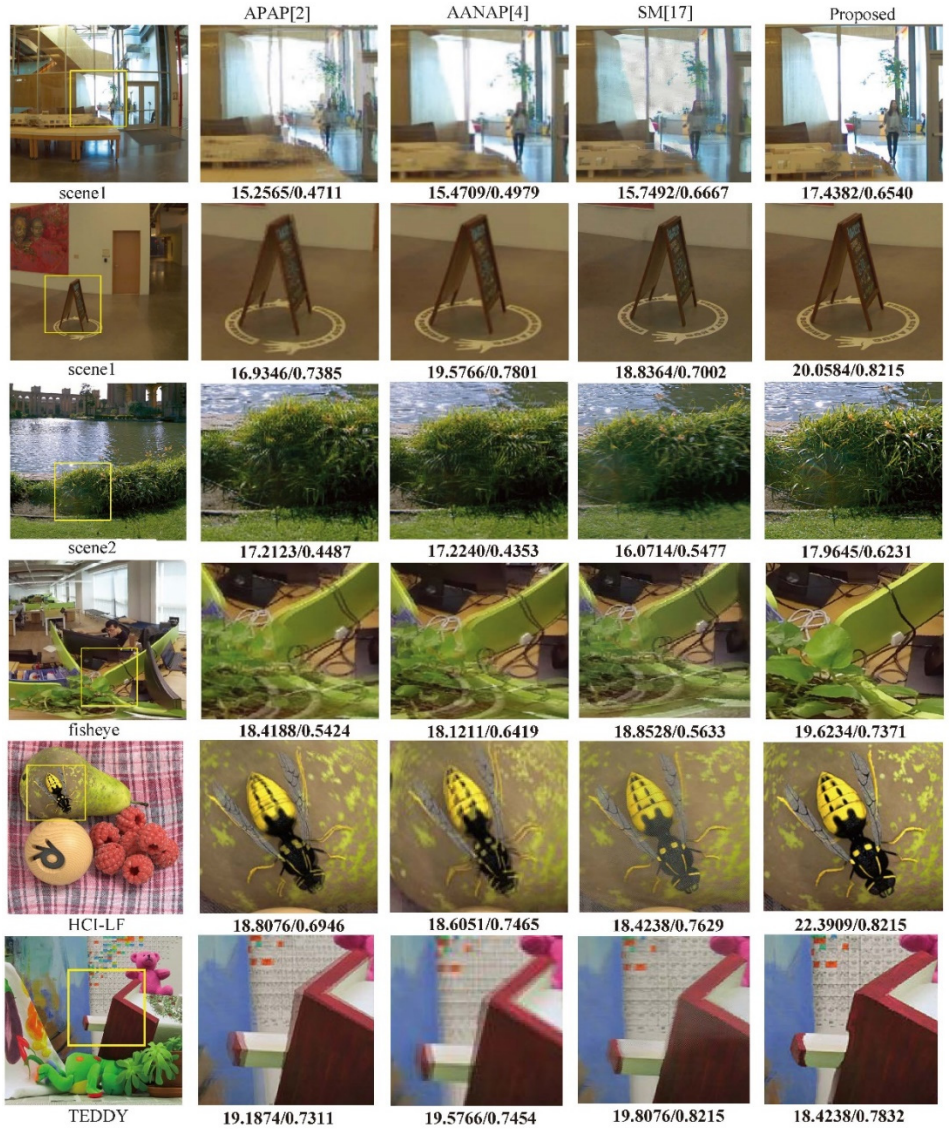

Figure 8. The comparison of the results obtained from other methods proposed for use in several scenes. The values below the images are PSNR and SSIM, respectively.

Table 2. MP-PSNR [27], PSNR, and SSIM comparison of the proposed method and [2,4,17]. The bold items indicate which method gets the better performance in each scene.

Datasets/ Models	APAP [2]			AANAP [4]			SM [17]			Proposed		
	*scene*1	*scene*2	*fisheye*	*scene*1	*scene*2	*fisheye*	*scene*1	*scene*2	*fisheye*	*scene*1	*scene*2	*fisheye*
MP-PSNR [27]	25.0831	24.1017	24.1743	25.9847	24.1833	24.5380	27.1593	25.0037	24.0649	27.1482	25.1794	24.5882
PSNR	17.3444	20.1031	20.1295	18.8771	18.1437	18.1437	**21.2529**	19.5766	20.7835	21.0021	20.1295	**21.2200**
SSIM	0.4015	0.6100	0.6218	0.4275	0.6463	0.7078	0.4979	0.7167	0.5732	**0.5518**	**0.7411**	**0.7914**

(**a**)

(**b**)

(**c**)

Figure 9. 'Ours' panoramic images. (**a**) scene1; (**b**) scene2; (**c**) fisheye.

5. Conclusions

In this paper, we proposed a view synthesis approach based on optical flow to solve the problem of panorama stitching. First, we estimated the optical flow field to match images that had a certain robustness with overlapping and non-overlapping areas. Secondly, the obtained optical flow field was used to warp the left and right views to obtain the reconstructed views for the subsequent blending stage. We showed that the reconstruction version of views could reduce the vertical distortion of the original image. In the rendering process, we considered the optical flow and the alpha value of pixels to interpolate each pixel in the novel view. Compared with existing methods, our flow-based view synthesized method was able to eliminate most artifacts and structural distortions.

Author Contributions: Conceptualization, W.Z.; methodology, Y.W. and Y.L.; software, W.Z.; validation, W.Z.; formal analysis, Y.W.; investigation, Y.L. and Y.W.; resources, Y.L. and Y.W.; data curation, W.Z.; writing—original draft preparation, W.Z.; writing—review and editing, Y.W. and Y.L., visualization, W.Z.; supervision, Y.W.; project administration, Y.W. and Y.L.; funding acquisition, Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Key R&D Program of China (No.2018YFB1800501 and No.2019YFB1803103).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Brown, M.; Lowe, D.G. Automatic Panoramic Image Stitching using Invariant Features. *Int. J. Comput. Vis.* **2007**, *74*, 59–73. [CrossRef]
2. Zaragoza, J.; Chin, T.-J.; Brown, M.S.; Suter, D. As-projective-as-possible image stitching with moving dlt. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2339–2346.
3. Chang, C.-H.; Sato, Y.; Chuang, Y.-Y. Shape preserving half-projective warps for image stitching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 3254–3261.
4. Lin, C.-C.; Pankanti, S.U.; Ramamurthy, K.N.; Aravkin, A.Y. Adaptive as-natural-as-possible image stitching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1155–1163.
5. Lin, K.; Jiang, N.; Cheong, L.-F.; Do, M.; Lu, J. Seagull: Seam-guided local alignment for parallax-tolerant image stitching. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 370–385.
6. Thatte, J.; Girod, B. A statistical model for disocclusions in depth-based novel view synthesis. In Proceedings of the 2019 IEEE Visual Communications and Image Processing (VCIP), Sydney, Australia, 1–4 December 2019; pp. 1–4.
7. Zhang, Y.; Zou, D.; Ren, J.S.; Jiang, Z.; Chen, X. Structure-preserving stereoscopic view synthesis with multi-scale adversarial correlation matching. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5853–5862.
8. Bertel, T.; Campbell, N.D.F.; Richardt, C. MegaParallax: Casual 360° Panoramas with Motion Parallax. *IEEE Trans. Vis. Comput. Graph.* **2019**, *25*, 1828–1835. [CrossRef] [PubMed]
9. Xu, B.; Jia, Y. Wide-angle image stitching using multihomography warping. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 1467–1471.
10. Chai, Q.; Liu, S. Shape-optimizing hybrid warping for image stitching. In Proceedings of the 2016 IEEE International Conference on Multimedia and Expo (ICME), Seattle, WA, USA, 11–15 June 2016; pp. 1–6.
11. Fan, X.; Lei, J.; Fang, Y.; Huang, Q.; Ling, N.; Hou, C. Stereoscopic Image Stitching via Disparity-Constrained Warping and Blending. *IEEE Trans. Multimed.* **2019**, *22*, 655–665. [CrossRef]
12. Richardt, C.; Pritch, Y.; Zimmer, H.; Sorkine-Hornung, A. Megastereo: Constructing high resolution stereo panoramas. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 1256–1263.
13. Peleg, S.; Ben-Ezra, M.; Pritch, Y. Omnistereo: Panoramic stereo imaging. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 279–290. [CrossRef]
14. Stankiewicz, O.; Wegner, K.; Tanimoto, M.; Doma'nski, M. Enhanced View Synthesis Reference Software (VSRS) for Free-VIEWPOINT Television. 2013. Available online: https://svn.multimedia.edu.pl/vsrs (accessed on 7 June 2021).
15. Kalantari, N.K.; Wang, T.-C.; Ramamoorthi, R. Learning-based view synthesis for light field cameras. *ACM Trans. Graph.* **2016**, *35*, 1–10. [CrossRef]
16. Chaurasia, G.; Duchene, S.; Sorkine-Hornung, O.; Drettakis, G. Depth synthesis and local warps for plausible image-based navigation. *ACM Trans. Graph.* **2013**, *32*, 1–12. [CrossRef]
17. Zhou, T.; Tucker, R.; Flynn, J.; Fyffe, G.; Snavely, N. Stereo magnification: Learning view synthesis using multiplane images. *ACM Trans. Graph. (TOG)* **2018**, *37*, 1–12. [CrossRef]
18. Flynn, J.; Neulander, I.; Philbin, J.; Snavely, N. Deepstereo: Learning to predict new views from the world's imagery. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5515–5524.
19. Kopf, J.; Cohen, M.F.; Szeliski, R. First-person hyper-lapse videos. *ACM Trans. Graph.* **2014**, *33*, 1–10. [CrossRef]
20. Aliev, K.-A.; Sevastopolsky, A.; Kolos, M.; Ulyanov, D.; Lempitsky, V. Neural point-based graphics. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 696–712.
21. Thies, J.; Zollhöfer, M.; Nießner, M. Deferred neural rendering: Image synthesis using neural textures. *ACM Trans. Graph. (TOG)* **2019**, *38*, 1–12. [CrossRef]
22. Sitzmann, V.; Thies, J.; Heide, F.; NieBner, M.; Wetzstein, G.; Zollhofer, M. Deepvoxels: Learning persistent 3d feature embeddings. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2437–2446.
23. Facebook. Surround360 System, Surround360 Website. 2019. Available online: https://github.com/facebook/Surround360 (accessed on 4 January 2022).
24. Honauer, K.; Johannsen, O.; Kondermann, D.; Goldluecke, B. A dataset and evaluation methodology for depth estimation on 4d light fields. In *Asian Conference on Computer Vision, Taipei, Taiwan, 20–24 November 2016*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 19–34.

25. Scharstein, D.; Szeliski, R. High-accuracy stereo depth maps using structured light. In Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, WI, USA, 18–20 June 2003; Volume 1, pp. 195–202.
26. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The kitti vision benchmark suite. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012.
27. Sandic-Stankovic, D.; Kukolj, D.; Le Callet, P. Dibr synthesized image quality assessment based on morphological pyramids. In Proceedings of the 2015 3DTV-Conference: The True Vision—Capture, Transmission and Display of 3D Video (3DTVCON), Lisbon, Portugal, 8–10 July 2015; pp. 1–4.