Generation of Stereo Images Based on a View Synthesis Network

Yuan-Mau Lo 1, Chin-Chen Chang 2,*, Der-Lor Way 3 and Zen-Chung Shih 1

1 Institute of Multimedia Engineering, National Chiao Tung University, Hsinchu 300, Taiwan; mdes4214@gmail.com (Y.-M.L.); zcshih@cs.nctu.edu.tw (Z.-C.S.)
2 Department of Computer Science and Information Engineering, National United University, Miaoli 360, Taiwan
3 Department of NewMedia Art, Taipei National University of Arts, Taipei 112, Taiwan; adlerway@gmail.com
* Correspondence: ccchang@nuu.edu.tw

Received: 8 March 2020; Accepted: 25 April 2020; Published: 29 April 2020

Abstract: The conventional warping method only considers translations of pixels to generate stereo images. In this paper, we propose a model that can generate stereo images from a single image, considering both translation as well as rotation of objects in the image. We modified the appearance flow network to make it more general and suitable for our model. We also used a reference image to improve the inpainting method. The quality of images resulting from our model is better than that of images generated using conventional warping. Our model also better retained the structure of objects in the input image. In addition, our model does not limit the size of the input image. Most importantly, because our model considers the rotation of objects, the resulting images appear more stereoscopic when viewed with a device.

Keywords: stereo images; view synthesis; neural network; semantic segmentation; depth estimation

1. Introduction

In recent years, because of the commercialization of wearable systems and the vigorous development of related technologies, research on virtual reality has become increasingly popular. Among many related topics, the concept of stereo images is basic and essential. In the real world, the human eye recognizes three-dimensional objects, which is termed a stereoscopic sense [1]. Many studies have been conducted on stereo images, on such issues as the design of special cameras or devices to capture stereoscopic panoramas [2] and stitching stereoscopic panoramas from stereo images captured with a stereo camera [3]. Research on stereo images also benefits from the recent dramatic development of deep learning. After several representative networks were proposed, such as convolutional neural networks (CNNs) [4,5] and generative adversarial networks (GANs) [6], correlated studies on stereo images witnessed breakthrough developments, such as stereo matching and disparity estimation from a pair of stereo images [7,8], single view depth estimation [9–13], predicting new views or constructing a complete three-dimensional scene from an image sequence [14,15], and view synthesis for an object from only a single view [16–20].

However, these studies lack a method for generating a pair of stereo images from a single image. For example, if there is only one photo in an album, it cannot be used to obtain stereo images to present stereoscopic vision using the aforementioned methods. Xie et al. [21] proposed a CNN-based method to automatically convert videos from two to three dimensions, where a single image is input as the left-eye view, and the disparity of each pixel is predicted using the network to generate the right-eye view. The input (left-eye view) and output (right-eye view) can be combined into a red-cyan anaglyph.
A drawback of this method is that it is based on the conventional warping method, and considers only the disparity of each pixel and not the rotation of the entire object.

The conventional warping method warps an image using the disparity between each pixel in the left- and right-eye views, based on the different depths of pixels in the image. It only considers translations of pixels. However, when both eyes gaze at the same object in a scene, the object between the different views of the eyes appears to not only be translated, but also rotated. The rotation is an essential factor that affects the stereoscopic sense. Although the traditional warping method may display similar effects, because of different translation values of each pixel in the same object, the resulting image is usually too distorted or blurred and the rotation is insignificant, resulting in insufficient stereoscopic sense. Furthermore, in the conventional warping method, the pixels that are not in the original image but can be observed in the resulting image (i.e., the part that cannot be seen in the original image because of the field of view) cannot be effectively processed. Therefore, in this paper, we propose a new model that generates stereo images from a single image, considering both translations as well as the rotation of objects in the scene.

Methods that consider both translation and rotation can be found in many studies of view synthesis [14,20,22]. Flynn et al. [14] proposed a method that is similar to complementing frames in a video. They used an image sequence (e.g., a video) captured in motion as an input, and employed a network to predict a new view between two frames. Although their method can obtain high-quality results, it cannot be applied to a single image. Methods proposed by Park et al. [17] and Zhou et al. [20] have produced results on a single object, and their networks could effectively predict the portion of the object that was occluded in the original view. However, when these methods are applied to scene prediction, even if the approximate structure and texture can be generated, a considerable amount of distortion and blur may occur.

In this paper, we attempt to convert the view synthesis problem of the entire scene in a single image into a view synthesis problem of several objects to achieve our aim. We propose a new algorithm that allows the user to input a single image as the left-eye view. During the operation, the translation and rotation of each object in the scene are considered. This generates a right-eye view that can be combined with the input image to form a pair of stereo images, and the user can feel a stronger stereoscopic sense when viewing it.

2. Related Works

Luo et al. [8] treated the problem as a multiclass classification, where the classes are all possible disparities. They proposed a new matching network for stereo matching, and it produced accurate results in less than 1 s of GPU (graphics processing unit) computation. Kendall et al. [7] proposed a novel deep learning architecture for regressing disparity using a pair of stereo images. They used three-dimensional convolutions with disparity cost volume, representing geometric features, to predict a disparity map. Eigen et al. [10] proposed a multiscale deep network to solve the problem, and it consists of two parts. One makes a coarse global prediction, and the other refines this prediction locally. Zhou et al. [13] proposed two CNNs that can predict a depth map and camera pose, including position and direction information, from an image. During training, they used a video sequence as an input and the network selected one frame as the target image. Garg et al. [11] used the depth predicted from the left-eye view to compute a disparity map. They warped the right-eye view as a new left-eye view, and the error calculated from the two left-eye views was used to train the network. Godard et al. [12] furthered this method to propose a more careful approach. Instead of only warping the left-eye view as a new right-eye view or vice versa, they did both simultaneously. They first computed the predicted disparity from the left-eye view, and then warped the left- and right-eye views to obtain new right- and left-eye views, respectively.

Kulkarni et al. [16] proposed a network that can receive geometric parameters, such as poses, lights and shapes. They used a view of an object as an input and then added geometric parameters, such as pose and light, during the encoding process. After decoding was conducted, a new view
with information about these parameters could be produced. Tatarchenko et al. [18] added rotation information during the encoding process, and their network could directly generate pixels of the new rotated view. By contrast, Zhou et al. [20] proposed an appearance flow network (AFN), that does not generate pixels directly, but produces appearance flow instead. They sampled the original view with the flow to synthesize a new high-quality view. However, the shortcoming of this method is also obvious. Because the result is sampled from the original view, it is influenced by the correlation between these two views. When the new view and the original view share only a small number of pixels (i.e., the rotation angle is too large to make the new view completely different from the original view), the quality of the new view will be low.

After GANs [6] were proposed, some studies attempted to use them to solve the aforementioned problem. Park et al. [17] proposed a transformation-grounded view synthesis network based on AFNs [20] and GANs. They multiplied the result of the AFN with a visibility map, and converted the problem from that of a new view synthesis to that of image completion. To a certain extent, this facilitated the overcoming of the shortcoming of AFNs. Zhao et al. [19] used variational inference to generate global appearance details, such as shape and color, of the new view, and then used GANs to complete the details. However, these methods are designed only for view synthesis of objects, and produce low-quality results when the entire scene is the input. The networks are also limited, in that the trained model is applicable to only one type of object.

Flynn et al. [14] proposed a method with concepts similar to complementing frames. They used a sequence of multiple and consecutive—but not sufficient—photos of different views as inputs, and then used the network to predict new views that do not appear in the sequence; that is, they cascaded the inputs and results into a relatively smooth video. Hedman et al. [15] reconstructed a complete three-dimensional scene, containing color, depth, and normal information, from photo sequences. Images resulting from these methods are high-quality, and these methods can be applied to real scenes, provided the defined input is an image sequence. Xie et al. [21] proposed a CNN-based method to automatically convert videos from two dimensions to three dimensions. Their method is different from other methods in that their network directly predicts several disparity channels. To train the network end-to-end, they converted the conventional warping equation to a differentiable form. Although the method can be applied to a single image, it is still based on the conventional warping method, considering only the disparity of each pixel.

3. The Proposed Approach

The aim of our model is to allow a user to input a single image as the left-eye view and generate a right-eye view. In contrast to conventional warping-based methods, our method considers both the translation and rotation of objects. The output image can be combined with the input image to form a pair of stereo images. Figure 1 presents the flowchart of our model, which consists of five main parts.

![Flowchart of our model](image-url)
3.1. Depth Estimation

First, we estimated a depth map from an input image. We used the network and pretrained model proposed by Eigen et al. [9] because of its simple and efficient architecture. It consists of three parts that represent different scales. In the first scale, Eigen et al. trained two sizes of models. One model is based on an ImageNet-trained AlexNet [4] and the other is initialized by the VGG network [5]. We choose the VGG-initialized model for our model because of its higher performance.

The network uses three scales to complete the prediction task. First, it predicts a coarse but spatially varying set of features in scale 1. Then, it predicts a more detailed view at a midlevel resolution in scale 2. In scale 3, it refines the predictions at a higher resolution. Note that although the network can predict depth maps, surface normals and semantic labels from one image, we used it only for depth estimation because the semantic labels predicted by this network were not sufficiently accurate to fit our model. By contrast, the pyramid scene parsing network (PSPNet) [23] can deliver a higher performance.

Because the network is trained for the NYUDepth [24] indoor data set, the output size is only 147 × 109, which is smaller than the input image (558 × 501). Therefore, we could not use the network to compute disparities for all pixels directly. Also, the depth map predicted by the network does not cover the entire input image. The width of missing regions along the four borders was 5 pixels. Thus, to fit the input image, we extended the borders of the depth map and then upscaled it. To extend the depth map, we added five pixels to each border by copying the colors from the nearest pixels. This simple strategy was used because in human vision, information along the edges of a scene is often ignored and considered less important. To upscale the depth map, we used bicubic interpolation [25], which generates a smoother result than bilinear interpolation and nearest-neighbor interpolation. The result is presented in Figure 2. We use grayscales to facilitate visualizing the upscaled depth map.

![Figure 2. Upscaling the depth map: (a) the input image (558 × 501); (b) the extended depth map (157 × 119); and (c) the upscaled depth map (558 × 501).](image)

3.2. Semantic Segmentation

Our model predicts semantic labels from the input image in addition to estimating the depth map. In semantic segmentation [26], the PSPNet [19,23] can achieve high performance using various data sets and can also outperform the network proposed by Eigen et al. [9]. We used the PSPNet to predict semantic labels because of its high accuracy.

The PSPNet first uses ResNet [27] to generate a feature map from the input image, and then produces different subregional representations using the pyramid pooling module. After upsampling, concatenation and convolution layers, semantic labels that consider the global and local context information are predicted. Zhao et al. [23] trained several models using different data sets to measure the performance of the PSPNet. In our model, because the network and pretrained model that estimate the depth map are trained using the NYUDepth indoor dataset, we chose the model trained using the ADE20K [28] dataset for the PSPNet. It is suitable for our model and can be applied to both outdoor...
and indoor scenes. The ADE20K dataset contains 150 object categories. Although correctly identifying the category of each object is not essential to our model, the ability of PSPNet to accurately color most objects in a scene is helpful in subsequent processes. Also, the predicted labels and the input image are of the same size. Thus, we can directly use the labels to segment the image.

To compute the translation and rotation of each object, we must segment the input image into several objects. After segmentation was conducted, we separated regions that were unconnected or belonged to different categories. Our algorithm identifies the cluster of each region using the semantic labels. After clustering, we generated a cluster list that is then filtered with four conditions: (1) the labels of the region are the background categories, such as wall, sky, floor, ceiling, road and sidewalk; (2) the size of the region is too small; (3) most pixels of the region are at the borders of the image; and (4) the average depth of the region is too deep, which means that the object is too far away to result in a significant rotation in human vision.

If the region of a cluster meets any aforementioned conditions it will be removed from the cluster list, because in our model, we only focus on those regions or objects that are important, sufficiently large, and noticed easily by the human eye. After regions were separated, we segmented the input image into several objects, as presented in Figure 3. However, we eroded the background image because of the inaccuracy of the semantic labels along the borders of objects.

![Figure 3. Segmenting the input image into objects: (a) the input image; (b) after clustering; (c) the background after erosion; (d) the pillows; and (e) the bed.](image)

### 3.3. Translation and Rotation Computation

When computing translations, the sign of disparities from the left-eye view to the right-eye view is crucial. We used a form of the conventional warping method [21] to compute the disparity of each object. The equation is defined as follows:

\[ d = \frac{B(D - f)}{D} \]

where \( d \) is the disparity, \( D \) is the depth, \( B \) is the baseline, and \( f \) is the focal length. Thus, when \( D < f \), \( d \) carries a negative sign, and a positive sign otherwise.
We use this equation to compute disparities for both objects and background. For objects, $D$ is the average depth of the covered region, and we use the same disparity to translate all pixels in the region. For the background, $D$ is the depth of each pixel that translates with its own disparity. The left-eye view is translated to the right-eye view using a simple warping equation, defined as follows:

$$I_t(x, y) = I_s(x - d, y)$$

where $d$ is the disparity, $I_t(x, y)$ is the value of pixel $(x, y)$ in the result, and $I_s(x - d, y)$ is the value of pixel $(x - d, y)$ in the input image.

The rotation of an object is related to its depth. Between the left- and right-eye views, the rotation is larger if the object is closer. In addition, because we generated the right-eye view from the left-eye view, the rotation of the object was always clockwise. After obtaining the rotation limit, the rotation of an object is computed using smoothing. The equation for computing the rotation of an object is as follows:

$$r_i = \begin{cases} 3, & \text{for } D_{\text{avg}} \leq D_{\text{min}} \\ \frac{(3-1)(D_{\text{max}}-D_{\text{avg}})}{D_{\text{max}}-D_{\text{min}}} + 1, & \text{for } D_{\text{min}} \leq D_{\text{avg}} \leq D_{\text{max}} \\ 0, & \text{otherwise} \end{cases}$$

where $r_i$ is the index of the rotation, which is one of the inputs to the view synthesis network; $r_i \in (0 \cup [1,3])$. The corresponding angles of rotation are 0° and [5°, 15°], and $D_{\text{avg}}$ is the average depth of an object. The values of $D_{\text{max}}$ and $D_{\text{min}}$ depend on the range of the predicted depth map. In general, $D_{\text{max}} = 2.5$ and $D_{\text{min}} = 0.8$, and both are restrictions for ignoring objects that are either too far or too close.

### 3.4. View Synthesis of Objects

We only retained the disocclusion-aware AFN (DOAFN); we discarded the visibility map after training, which makes the network synonymous with the AFN, and modified some parts to fit our model. In addition to the visibility map, the differences introduced in using the DOAFN were that we changed the size of the rotation input from 17 × 1 × 1 to 3 × 1 × 1 and used bicubic sampling on the appearance flow.

One of the limitations of previous view synthesis networks [17,18,20] is that the trained model of the network is only for one object category. For example, if we train the network using the training data of cars, we cannot apply the trained model to the testing data of chairs. This is due to the complexity of the task of view synthesis, which usually requires 360° views to be generated using the input view. However, in our model, we rotate objects from the left-eye view to the right-eye view, which means that the object views change by only a few degrees and always rotate clockwise.

In the DOAFN, the size of the rotation input is 17 × 1 × 1, which is a one-hot array representing angles from 20° to 340° with an interval of 20°. For example, [0, 1, 0, 0, ... 0] represents a 40° rotation. However, in our model, we use a 3 × 1 × 1 array, representing angles from 5° to 15°. Because of the relatively small difference between the left-eye view and the right-eye view, not only did we reduce the range of rotation, but we also lessened the interval from 20° to 5°, which produces subtler changes. In addition, we were not required to consider negative degrees, such as −15° or 345°, which is a benefit of the property of clockwise rotation. These changes reduce the complexity of view synthesis and allow the training of various categories of objects at the same time. Our model is trained with 38 categories of indoor objects from the ShapeNet [29], including beds, benches, bookshelves, chairs, sofas and tables. The total number of objects in the training data set is 8233.

When we applied the trained model with our modified AFN to the objects segmented from the input image, the rotation index $r_i$ was sent into the one-hot array using linear interpolation, as follows:

$$x_{[r_i]} = 1 - (r_i - [r_i]), \quad x_{[r_i]+1} = 1 - x_{[r_i]}$$
where \( x_{tr_1} \) and \( x_{tr_1+1} \) are elements in the one-hot array. For example, if \( r_1 = 1.8 \), the one-hot array will be \([0.2, 0.8, 0]\), which represents \( 9^\circ \) \(((5 \times 0.2) + (10 \times 0.8) = 9\)\). Thus, the degrees of rotation are consecutive.

In the DOAFN, a differentiable sampling layer was observed following the appearance flow output. These layers are the kernels of the network. The appearance flow is a two-channel map that represents the correlation between pixels in the target view and in the input view, as presented in Figure 4. Figure 4a shows the input view of the objects, Figure 4b shows the RGB (red-green-blue) image of the appearance flow for visualization, and Figure 4c shows the target view of the objects generated using the input view and appearance flow. Following the appearance flow, the network can use bilinear sampling to sample pixels from the input view and then generate a new view. The bilinear sampling [17,20] equation is as follows:

\[
I_{t}^{i,j} = \sum_{(h,w) \in N} I_{s}^{h,w} \max(0, 1 - |F_{x,y}^{i,j} - h|) \max(0, 1 - |F_{x,y}^{i,j} - w|)
\]

where \( I_{t}^{i,j} \) is the target view, \( I_{s}^{i,j} \) is the input view, \((i, j)\) are the coordinates of the pixel in the target view, and \((h, w) \in N\) represents its four neighbors. \( F_{x,y}^{i,j} \) and \( F_{x,y}^{i,j} \) indicate the \( x \) and \( y \) coordinates of a target location \((i, j)\) in the appearance flow \( F \). When the DOAFN is trained, this operation is converted to the differentiable sampling layer that efficiently computes time. More importantly, it is differentiable, which benefits the gradient descent in learning the neural networks.

A problem with the DOAFN is that it is designed for \(256 \times 256\) images, which is usually too small a size for objects in a photo. In contrast to the depth map, objects segmented from the image have many high-frequency regions. The resulting image is blurred if we directly upscale the output of the network. Thus, we maintained the differentiable sampling layer because of its advantages during training. However, when we applied the trained model to the object, we extracted the appearance flow from the network and then up scaled it to the size of the input object. Subsequently, we used bicubic sampling to generate a new view.

The bicubic sampling [25] equation is as follows:

\[
I_{s}(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} I_{s}(x', y') W(x-x') W(y-y')
\]

\[(x', y') = F_{x,y}\]

where \( F_{x,y} \) is the value of coordinates \((x, y)\) in the appearance flow, which is a tuple of the correlated location in source image \( I_{s} \); source image \( I_{s} \) is the input object of the original size; \( I_{t} \) is the new view (i.e., the rotated object); coordinates \((x', y')\) represent \(4 \times 4\) neighbors of \((x', y')\) for \(i, j = 0, 1, 2\) and

Figure 4. (a) The input view of the objects, (b) the RGB image of the appearance flow for visualization, and (c) the target view of the objects.
3; and $W$ is an interpolation kernel function. The resulting image using bicubic sampling is clearer and sharper than that using direct upscaling. The results of rotating objects segmented from the input image are presented in Figure 5.

Figure 5. Results of rotating objects: (a) The pillows; (b) bed; (c) rotated pillows; and (d) rotated bed.

3.5. Right-Eye View Generation and Inpainting

After rotating the objects, we combined them with their own translations and the background image to generate a right-eye view. We eroded the objects because of noise produced by the view synthesis network along the edges of the rotated objects. This also made the missing regions larger. We used an inpainting technique to solve this problem.

We chose an inpainting method, proposed by Kawai et al. [30,31], which is based on patch matching and considers brightness changes, the spatial locality of textures, and symmetric patterns. A preliminary result is presented in Figure 6a. Blurring was still evident along the edges of the objects. To improve the effect of image inpainting, we modified the method proposed by Kawai et al. in two parts. In their method, before iteratively searching patches in the image to minimize the energy function, Kawai et al. randomly set the initial values of missing regions. Instead of this, in the first part, we used the result inpainted using the method of Telea [32] as the initial image. This reduced the computation time of iterations and improved the inpainting quality. The second part of our improvement involved using a reference image. Kawai et al. set the range of patch-searching as the data in the image excluding missing regions. This is an intuitive and efficient strategy in general image inpainting. In our model, most of the information of the generated right-eye view can be extracted from the left-eye view. Thus, we could limit the searching region to the left-eye view; we treated the left-eye view as a reference image to increase the probability of finding similar patches. The result using our improved inpainting method, which fixes most blurred regions and artifacts, is presented in Figure 6b.
We selected five imperfect and five perfect pairs under the default setting from these 23 data sets as the input images for the quantitative evaluation. The baseline method of Kawai et al. [30,31] for inpainting.

We used an Nvidia GeForce GTX 1080 GPU and an Intel® Core™ i7-7700 CPU with a 3.60 GHz Octa-core. Our environments required 2 to 3 min to process a 500 × 500 image with two objects. The most time-consuming part was the inpainting step because this step used the iteration technique and cannot be parallelized through the GPU. In addition, the computation cost was proportional to the number of objects and painted regions in the image. Although the input image size is not limited in our model, it is correlated with the computation time because of the number of iterations of inpainting. The number and the size of objects in the image also affected the computation time of bicubic sampling.

4. Results

We conducted several experiments on images from the Middlebury 2014 data sets [33]. We compared the results obtained from our model with those obtained using the conventional warping method, and discussed the advantages and limitations of our model. To clearly understand our model, we experimented with different components. The conventional warping method warps each pixel in the input image and then inpaints the result using the method proposed by Telea [32]. Due to the inaccuracy of the PSPNet labels on objects for the Middlebury 2014 datasets, in our model, we used manual labels and the method of Kawai et al. [30,31] for inpainting.

We used an Nvidia GeForce GTX 1080 GPU and an Intel® Core™ i7-7700 CPU with a 3.60 GHz Octa-core. Our environments required 2 to 3 min to process a 500 × 500 image with two objects. The most time-consuming part was the inpainting step because this step used the iteration technique and cannot be parallelized through the GPU. In addition, the computation cost was proportional to the number of objects and painted regions in the image. Although the input image size is not limited in our model, it is correlated with the computation time because of the number of iterations of inpainting. The number and the size of objects in the image also affected the computation time of bicubic sampling.

4.1. Quantitative Evaluation

The Middlebury 2014 data sets consist of 33 data sets of various scenes. Each data set contains imperfect and perfect pairs of stereo images under the default setting, and under different illuminations and exposures. The imperfect and perfect pairs are the stereo images with and without calibration errors, respectively. In addition, 23 of these datasets have calibration information, such as for baseline. We selected five imperfect and five perfect pairs under the default setting from these 23 data sets as the input images for the quantitative evaluation. The baseline is from the calibration information, and we selected the focal length that produces the result that is most similar to the ground truth.

The structural similarity (SSIM) index [34,35] is sensitive to nonstructural geometric distortions such as translation, rotation and scaling [36]. In our model, we emphasized the rotation of objects to enhance the stereoscopic sense. We used the complex wavelet SSIM (CW-SSIM) index [36] in this experiment because it is not affected by enhanced rotation and focuses on the SSIM between two images. The CW-SSIM is an extension of the SSIM to the CW domain. A higher CW-SSIM index indicates higher similarity in two images. Let $c_x = \{c_{xi}\}_{i=1}^N$ and $c_y = \{c_{yi}\}_{i=1}^N$ be two
sets of coefficients of the two images in the complex wavelet transform domain, respectively. The CW-SSIM index is given by

$$\tilde{S}(c_x, c_y) = \frac{2 \left| \sum_{i=1}^{N} c_{x,i} c_{y,i}^* \right| + K}{\sum_{i=1}^{N} |c_{x,i}|^2 + \sum_{i=1}^{N} |c_{y,i}|^2 + K}$$

where $c^*$ is the complex conjugate of $c$ and $K$ is a constant.

The CW-SSIM indices are presented in Table 1. Our model obtains the highest average CW-SSIM index, which implies that it better retains the structure when enhancing the effects of rotation. Moreover, the average value of perfect-pair images is lower because the images are relatively complicated; the average value of imperfect-pair images is higher because the images are relatively simple. The images of this experiment are presented in Figures 7 and 8.

| CW-SSIM     | Warping | Our Approach |
|-------------|---------|--------------|
| Imperfect Pair | 0.9341  | 0.9564       |
| Perfect Pair  | 0.8477  | 0.8492       |
| Imperfect Pair & Perfect Pair | 0.8909  | 0.9028       |

Table 1. Average values of imperfect- and perfect-pair images.

Figure 7. Imperfect-pair images using warping and our model. From top to bottom, these are recycle, motorcycle, adirondack, shelves and bicycle1.
4.2. Qualitative Evaluation

Herein, we present the results of various images. We used our model for the images from the Middlebury 2014 data sets. In these images, we used blue and red boxes to indicate our advantages and limitations, respectively.

Figure 9 presents the results of piano. We eroded the objects to eliminate the noises along the edges caused by modified AFN and bicubic sampling. Therefore, during inpainting, artifacts may sometimes be produced in the thin parts or along the edges, as indicated in the red boxes. These structural distortions are the reason our model obtains a lower CW-SSIM index than conventional warping does. Because of the consistency of the translation of pixels in an object, the results of our model are better than other approaches, as indicated in blue boxes.
Figure 9. Comparison of the results obtained using (a) warping and (b) our model.

Because all methods use the predicted depth map, the results may be poor if the depths are inaccurate, especially when the baseline is large and the objects are close to the eyes, as presented in Figure 10 (jadeplant). Thus, all these methods obtain low CW-SSIM indices. Also, human eyes cannot focus on objects in images when viewing stereo images of this type. The results of our model maintain the structure in the blue box.

Figure 10. Comparison of the results obtained using (a) warping and (b) our model.

Figure 11 shows that in the result using conventional warping, the tires of the motorcycle are obviously distorted. By contrast, our model produced a better result with a more complete structure. However, in the red box, an artifact occurred after inpainting because of failed patch matching.

Figure 11. Comparison of the results obtained using (a) warping and (b) our model.

As shown in Figure 12, in the red boxes, because of the limitation introduced by erosion, the handle of the basket is thinner in our results. In addition, the rightmost area in this image, which
includes parts of the basket, cannot be observed in the input image (i.e., the left-eye view). Thus, we can conclude that both conventional warping and our model cannot generate this area well. Nevertheless, our model produces better results than other approaches in representing the structure of other objects.

![Figure 12. Comparison of the results obtained using (a) warping and (b) our model.](image)

5. Conclusions

We proposed a model that considers both the translation and rotation of objects in a scene to generate stereo images from a single image. Our results are more stereoscopic than those using traditional warping. They also maintain the structure of objects and obtain a higher CW-SSIM index because we converted the scope of this task from that of a scene to that of objects. The results of our model also outperform those of methods that use conventional warping. In addition, our model does not limit the size of the input image.

In our results, artifacts could have occurred in the thin structure or along the edges of objects because of the inaccuracy of the depth map and labels in some situations; in the future, this can be resolved by using better networks. It is better to use an instance segmentation network, especially for object segmentation in an image, even though it is a difficult and complex network to use and current studies on them are insufficient.

Author Contributions: Methodology, Y.-M.L. and C.-C.C.; Supervision, D.-L.W. and Z.-C.S.; Writing—original draft, Y.-M.L.; Writing—review & editing, C.-C.C. and Z.-C.S. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank the Ministry of Science and Technology of Taiwan for financially supporting this research under Contract No. MOST 108-2221-E-239-016-.

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

References

1. Zhang, L.; Dugas-Phocion, G.; Samson, J.S.; Seitz, S.M. Single-view modelling of free-form scenes. *Comput. Anim. Virtual Worlds* 2002, 13, 225–235. [CrossRef]
2. Aggarwal, R.; Vohra, A.; Namboodiri, A.M. Panoramic stereo videos with a single camera. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, CO, USA, 23–25 June 1999; pp. 3755–3763.
3. Zhang, F.; Liu, F. Casual stereoscopic panorama stitching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2002–2010.
4. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; New York University: New York, NY, USA, 2012; pp. 1097–1105.
5. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv 2014*, arXiv:1409.1556.
6. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Advances in Neural Information Processing Systems; New York University: New York, NY, USA, 2014; pp. 2672–2680.

7. Kendall, A.; Martirosyan, H.; Dasgupta, S.; Henry, P.; Kennedy, R.; Bachrach, A.; Bry, A. End-to-end learning of geometry and context for deep stereo regression. arXiv 2017, arXiv:1703.04309.

8. Luo, W.; Schwing, A.G.; Urtasun, R. Efficient deep learning for stereo matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; University of Toronto: Toronto, ON, Canada, 2016; pp. 5695–5703.

9. Eigen, D.; Fergus, R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2650–2658.

10. Eigen, D.; Puhrsch, C.; Fergus, R. Depth map prediction from a single image using a multi-scale deep network. In Advances in Neural Information Processing Systems; New York University: New York, NY, USA, 2014; pp. 2366–2374.

11. Garg, R.; Carneiro, G.; Reid, I. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2016; pp. 740–756.

12. Godard, C.; Mac Aodha, O.; Brostow, G.J. Unsupervised monocular depth estimation with left-right consistency. arXiv 2016, arXiv:1609.03677.

13. Zhou, T.; Brown, M.; Snavely, N.; Lowe, D.G. Unsupervised learning of depth and ego-motion from video. arXiv 2017, arXiv:1704.07813.

14. 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.

15. Hedman, P.; AlsIsan, S.; Szeliski, R.; Kopf, J. Casual 3d photography. ACM Trans. Graphics (TOG) 2017, 36, 234. [CrossRef]

16. Kulkarni, T.D.; Whitney, W.F.; Kohli, P.; Tenenbaum, J. Deep convolutional inverse graphics network. In Advances in Neural Information Processing Systems; New York University: New York, NY, USA, 2015; pp. 2539–2547.

17. Park, E.; Yang, J.; Yumer, E.; Ceylan, D.; Berg, A.C. Transformation-grounded image generation network for novel 3d view synthesis. arXiv 2017, arXiv:1703.02921.

18. Tatarchenko, M.; Dosovitskiy, A.; Brox, T. Multi-view 3d models from single images with a convolutional network. In European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2016; pp. 322–337.

19. Zhao, B.; Wu, X.; Cheng, Z.Q.; Liu, H.; Feng, J. Multi-view image generation from a single-view. arXiv 2017, arXiv:1704.04886.

20. Zhou, T.; Tulsiani, S.; Sun, W.; Malik, J.; Efros, A.A. View synthesis by appearance flow. In European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2016; pp. 286–301.

21. Xie, J.; Girshick, R.; Farhadi, A. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2016; pp. 842–857.

22. Lo, Y.M.; Chang, C.C.; Way, D.L.; Shih, Z.C. A stereo images generating system considering both translation and rotation of objects. In Proceedings of the 2019 Joint International Workshop on Advanced Image Technology (IWAIT) and International Forum on Medical Imaging in Asia (IFMIA), Singapore, 6–9 January 2019.

23. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.

24. Silberman, N.; Hoiem, D.; Kohli, P.; Fergus, R. Indoor segmentation and support inference from rgbd images. Comput. Vis.–ECCV 2012, 2012, 746–760.

25. Keys, R. Cubic convolution interpolation for digital image processing. IEEE Trans. Acoust. Speech Signal Process. 1981, 29, 1153–1160. [CrossRef]

26. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected rfs. IEEE Trans. Pattern Anal. Mach. Intell. 2018, 40, 834–848. [CrossRef] [PubMed]
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
28. Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; Torralba, A. Semantic understanding of scenes through the ade20k dataset. *arXiv* 2016, arXiv:1608.05442. [CrossRef]
29. Chang, A.X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. Shapenet: An information-rich 3d model repository. *arXiv* 2015, arXiv:1512.03012.
30. Kawai, N.; Yokoya, N. Image inpainting considering symmetric patterns. In Proceedings of the Pattern Recognition (ICPR), 2012 21st International Conference, Tsukuba, Japan, 11–15 November 2012; pp. 2744–2747.
31. Kawai, N.; Sato, T.; Yokoya, N. Image inpainting considering brightness change and spatial locality of textures and its evaluation. *Adv. Image Video Technol.* 2009, 5414, 271–282.
32. Telea, A. An image inpainting technique based on the fast marching method. *J. Graphics Tools* 2004, 9, 23–34. [CrossRef]
33. Scharstein, D.; Hirschmüller, H.; Kitajima, Y.; Krathwohl, G.; Nešić, N.; Wang, X.; Westling, P. High-resolution stereo datasets with subpixel-accurate ground truth. In *German Conference on Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 31–42.
34. Wang, Z.; Bovik, A.C. A universal image quality index. *IEEE Signal Process. Lett.* 2002, 9, 81–84. [CrossRef]
35. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* 2004, 13, 600–612. [CrossRef] [PubMed]
36. Sampat, M.P.; Wang, Z.; Gupta, S.; Bovik, A.C.; Markey, M.K. Complex wavelet structural similarity: A new image similarity index. *IEEE Trans. Image Process.* 2009, 18, 2385–2401. [CrossRef] [PubMed]