

Adaptive Multiplane Image Generation from a Single Internet Picture

Diogo C. Luvizon¹ Gustavo Sutter P. Carvalho^{1,2} Andreza A. dos Santos³ Jhonatas S. Conceição³
 Jose L. Flores-Campana³ Luis G. L. Decker³ Marcos R. Souza³ Helio Pedrini³
 Antonio Joia¹ Otávio A. B. Penatti¹

¹AI R&D Lab, Samsung R&D Institute Brazil, Campinas, SP, 13097-160, Brazil

²ICMC, University of São Paulo (USP), São Carlos, SP, 13566-590, Brazil

³Institute of Computing, University of Campinas (UNICAMP), Campinas, SP, Brazil, 13083-852

Abstract

In the last few years, several works have tackled the problem of novel view synthesis from stereo images or even from a single picture. However, previous methods are computationally expensive, specially for high-resolution images. In this paper, we address the problem of generating a multiplane image (MPI) from a single high-resolution picture. We present the adaptive-MPI representation, which allows rendering novel views with low computational requirements. To this end, we propose an adaptive slicing algorithm that produces an MPI with a variable number of image planes. We present a new lightweight CNN for depth estimation, which is learned by knowledge distillation from a larger network. Occluded regions in the adaptive-MPI are inpainted also by a lightweight CNN. We show that our method is capable of producing high-quality predictions with one order of magnitude less parameters compared to previous approaches. The robustness of our method is evidenced on challenging pictures from the Internet.

1. Introduction

Novel view synthesis from a single image is a very challenging problem that has gained attention from the computer vision community in the last few years. Despite the challenges involved in this task, for instance, estimating depth and generating color information on occluded regions, novel view synthesis unlocks a broad range of applications related to 3D visual effects from a single image, augmented reality systems, among many others.

Traditionally, methods for new view synthesis rely on stereo vision for obtaining both depth information and color inpainting [23, 43]. Among these methods, multiplane images (MPI) [31] has frequently been used to encode the

Figure 1: Our method predicts an adaptive multiplane image representation from a single picture, which can be easily rendered to produce new views of the scene.

scene. Contrarily to other representations, such as point clouds or 3D meshes, multiplane images are easily stored with traditional image compression algorithms and can be rendered efficiently on embedded systems, such as in smartphones or Smart TVs, even for high-resolution images.

Learning an MPI representation from a single image is a much more challenging problem, since depth and occluded colors have to be inferred from a flat image. Very recently, Tucker and Snavely [35] proposed a framework for learning an MPI from a single-view image. However, their method is still dependent on multiview (stereo) data during training. This fact hinders the robustness and applicability of the method, since such data is often collected from a moving camera (e.g., images from RealEstate10K dataset), therefore, being limited to static scenes. Contrarily, in our method we learn an MPI representation from single-view and unconstrained pictures from the Internet. As a result,

This work was funded by Samsung Eletrônica da Amazônia Ltda., through the project “Parallax Effect”, within the scope of the Informatics Law No. 8248/91.

our method benefits from a strong generalization capability. Another drawback of previous methods based on MPI is their high computational cost, since a high number of image planes are often used to achieve satisfactory results [30].

Considering the limitations of previous methods, we propose a new approach to generate an *adaptive multiplane image* (adaptive-MPI) from a single image. As shown in Figure 1, differently from traditional MPI, the adaptive version automatically selects the number of layers and the distance between them in order to better represent the content of the scene. Consequently, our method is able to produce results comparable to the state of the art while requiring very few image planes and running in a few seconds even for high-resolution images. The main contributions of our work are listed as follows. *First*, we propose a new method to learn an adaptive-MPI from single images from the Internet. *Second*, we propose a lightweight CNN for depth estimation and a robust distillation method for transfer learning from a larger network in high-resolution. *Third*, we propose an inpainting strategy that handles MPI with a variable number of layers. Our approach is evaluated for depth estimation with respect to our *teacher* network and on view synthesis on challenging pictures extracted from the Internet.

2. Related work

In this section, we review some of the most relevant works related to our method.

2.1. Single-image depth estimation

Estimating depth from a single image is an ill-posed problem due to scale ambiguity. With the advent of deep learning, predicting depth maps has become possible for well conditioned environment [7], such as indoor scenes. However, for reducing the epistemic uncertainty related to the task [17], an enough amount of data is required for more general cases. To handle this issue, methods based on stereo matching have emerged in the last few years [11, 34]. Such approaches rely only on well calibrated pairs of images for training, which is easier to capture than precise depth maps. However, learning from stereo pairs is not as effective as learning directly from ground-truth depth [9].

More recently, impressive results have been achieved by exploring stereo data from the Web [40], structure from motion (SfM) using Internet pictures [18], and from YouTube videos (mannequin challenge) [19]. The main advantage of these methods is that they benefit from a high variety of data. However, when multiple datasets with different scales and data distributions are mixed together, learning a common representation becomes a challenge. Ranftl et al. [25] proposed to solve this problem with a scale- and shift-invariant loss, that allows the network learning from multiple datasets simultaneously.

Considering the discussed previous work, we can notice that a key factor for robust depth estimation is training data. Nevertheless, some datasets may not be readily available or may require customization that makes their use difficult. Therefore, our main contribution in depth estimation is a robust distillation process for learning a lightweight model capable of generating high-quality and high-resolution depth maps, as demonstrated through our experiments.

2.2. View synthesis from a single view

Realistic view synthesis has gained great relevance in the areas of computer and robotic vision. Recent works have used deep learning-based approaches, which generally predict representations such as 3D meshes [20] or point clouds [39], while for more complex images, representations such as MPI [31] and LDI [26] have been used. Until recently, methods in the literature have only dealt with calibrated stereo images [8, 30, 43] and used fully-convolutional deep networks to predict an MPI representation, which is then rendered to extrapolate novel views.

Currently, there are still few studies that predict new views from a single image, such as in Tucker and Snavely [35]. In this case, the authors generated the training data for depth supervision using Simultaneous Localization and Mapping (SLAM) and SfM algorithms from videos with static content, while the network is optimized for both depth estimation and new view synthesis. Image segmentation has also been explored to simulate motion parallax effect [24]. Other methods propose to learn an LDI relying on intermediate tasks, such as depth and segmentation maps [4, 5], or even from pairs of stereo images [36]. Very recently, Shih et al. [27] proposed a framework for new view synthesis based on monocular depth estimation and on a cascade of depth and color inpainting. Despite the high-quality results, this method has an iterative process that becomes prohibitive for high-resolution images on computational restricted devices.

Similar to the work proposed by Tucker and Snavely [35], we predict an MPI from a single image. However, the planes of our MPI are defined in an adaptive way, depending on the input image, which is more suitable for view synthesis from a small number of image planes. In addition, our method does not rely on stereo data for training and performs efficiently due to our lightweight models.

2.3. Image inpainting

Several inpainting techniques have been proposed to fill missing or occluded parts of an image. These methods can be divided into traditional diffusion-based [1, 2] and patch-based [6] approaches, and more recent GAN-based methods [14, 22, 42]. Differently from classic image inpainting, multiplane images require a more sophisticated process, since only the background or specific borders in the image layers need inpainting. Since we are not considering

the case of stereo images for training, we propose an approach to learn an inpainting CNN for MPI layers based on single images and their respective depth maps.

3. Proposed method

3.1. Dataset preparation

The goal of our method is to produce an adaptive-MPI representation from unconstrained images. Publicly datasets for novel view synthesis are frequently based on static scenes, i.e., collected from videos without people or animals, such as RealEstate10K [43], limited to macro pictures captured by plenoptic cameras [16, 29], or restricted to a specific domain [10]. Available depth estimation datasets are also limited to specific domains (MegaDepth [18]) and low resolution images (NYU-Depth V2 [28]).

Nevertheless, the Internet is a massive source of high-resolution image data, with scenes of different modalities, including landscapes, portraits, groups of people, animals, food and vehicles, frequently on variable conditions of lightning. This abundance of image data could be used to train a robust inpainting algorithm, as well as to distill the knowledge from depth estimation methods. With this in mind, we collected a dataset of high-resolution Internet pictures, which is used for teaching our lightweight CNN models, both for depth estimation and inpainting tasks. Some examples of images from our dataset are shown in Figure 2.

Figure 2: Samples from our dataset and the pseudo ground truth generated by our ensemble approach.

3.1.1 Data collection

We collected 100K images from Flickr by performing queries based on 16 different keywords: *america, animals, asia, brazil, city, civilization, europe, food, indoor, landscape, night, objects, people, show, sports, and vehicle*. The queries were made for high-resolution images with no restrictions in the rights of use. All the collected files were filtered to remove duplicated pictures and to correct rotations. We split the remaining files as 95K training and 3K evaluation images.

3.1.2 Distilling depth knowledge

Recent depth estimation methods [18, 25] are capable of predicting high quality depth maps. However, these models are computationally expensive, specially for high-resolution images. Since our approach focuses on efficiency, we propose to distill the knowledge from MiDaS [25] by using our collected pictures from the Internet.

In knowledge distillation, an effective approach is to train the *student network* based on an ensemble of predictions from the *master network* [12]. In our method, we performed a $10\times$ ensemble by considering predictions in five different resolutions (squares with rows and columns with 512, 768, 1024, 1600, and 1920 pixels) and horizontal flipping. All the intermediate predictions were then normalized and resized with bilinear interpolation to the highest resolution, which are then combined by the median value at each pixel position, resulting in our *pseudo ground truth* depth maps. Figure 2 shows some examples of depth maps generated by this approach. For simplicity and efficiency during training, our method does not rely on intermediate feature supervision. Similarly to previous works, we use the term *depth map* interchangeably for *disparity map*.

3.2. Efficient depth estimation

Considering our objective to produce high-resolution effects efficiently, we propose a lightweight CNN for depth estimation. Previous methods for depth estimation are frequently based on U-Net [19] architectures. The drawback of traditional U-Net models are their high number of parameters and high memory usage, specially for high-resolution images. Instead, we propose to adapt EfficientNet [32] to perform depth estimation.

Specifically, we integrated the lightweight MobileNet V3 block [13] with Discrete Wavelet Transforms (DWT), as shown in Figure 3. We use DWT and Inverse DWT in a similar way to Luo et al. [21], i.e., for downscaling and upscaling feature maps, respectively. The DWT has the effect of decomposing an input tensor of shape $H \times W \times C$ into a set of approximation coefficients (AC) and detailed coefficients (DC), which represent the time-frequency decomposition of the input signal. By concatenating the AC and DC components, we recover a tensor with shape $H/2 \times W/2 \times 4C$. This allows our model to benefit from skip connections with lower resolutions compared to traditional U-Nets and to preserve high frequency information throughout the full network. In addition, we perform multi-scale depth supervisions to enforce learning from multi-resolution signals. Our design decisions are evaluated in the ablation studies.

3.3. Adaptive multiplane image slicing

The goal of the adaptive multiplane image slicing algorithm is to compute a small set of image planes that represents a 3D scene, while reducing the possible artifacts on

Figure 3: Depth estimation network (*lite*) based on MobileNet V3 blocks. Input and feature maps are downscaled by Discrete Wavelet Transforms (DWT) and upscaled by Inverse DWT. Supervision at intermediate resolutions are applied to enforce multi-resolution learning.

novel views. To this end, we propose a slicing algorithm based on depth information. This algorithm selects thresholds between the image planes from regions with high discontinuity in the depth map, assuming that the scene has some depth salience, such as an object in the foreground.

Given a depth map of a scene, we first pre-process it in order to enhance depth discontinuities. In this pre-processing stage, depth values are normalized to the interval $[0..255] \subset \mathbb{Z}^+$. Then, we apply a bilateral filter [33] followed by a Canny edge detector. The borders in the depth map are expanded by a morphological erosion. The effect of this process is shown in Figure 4. From the resulting depth map, we compute a *transition index*, defined by:

$$t = \frac{2h}{\max(1, h)}, \quad (1)$$

where h is the normalized histogram of the resulting depth map and 1 is set to 0.001 to avoid division by zero. Figure 5 shows the normalized histogram and the transition index for a given depth map.

Filtered depth Canny edges Eroded borders

Figure 4: Pre-processing of depth maps: proposed steps allow better delineation of object edges.

A high value in t corresponds to a valley or a discontinuity in the depth map, since it depends on the second derivative of h . Therefore, the local maximum values from t are used to select the threshold (transition) between two image planes. Once a transition t_i is selected, the corresponding value of i is stored in a vector T and its neighbors

Figure 5: Local and spaced maximum values in t define the transitions T_j between image planes. The first and last index of t are always selected as boundaries. Note that no transition was selected between 46 and 141 due to the low density in this interval.

$\{i - 1, \dots, i + 1\}$ in t are set to zero. This prevents from selecting transitions too close in the depth. In our implementation, we set $\delta = 8$. This process of transition selection repeats until no more values in t are above a threshold (set to 0.1 in our method), or a maximum number of selected transitions is reached. At the end of this process, we have $N + 1$ selected transitions, where N is the number of image planes. Finally, each image plane l_j is formed by the pixels lying in the depth interval $\{h_{T_j}, \dots, h_{T_{j+1}}\}$, where $j = \{0, \dots, N\}$ is the index of the selected transitions. The position of l_j on the Z axis is computed as the average of the depth values lying in the same interval. Thus, the number of image planes and the depth of each layer is adapted accordingly to the depth information of the scene.

3.4. Multiplane image inpainting

Once the image planes are computed by the adaptive slicing algorithm, some regions may require image inpainting to fill the holes on the generated novel views. A naive approach would be to inpainting after rendering. However,

this would be prohibitive to most of the real-time applications, since inpainting is computationally expensive and cannot be performed on current embedded devices in real-time for high-resolution images. Instead, we propose to inpaint the adaptive-MPI representation before rendering, which means that no additional post-processing will be required during novel views generation.

Differently from classic inpainting tasks based on regular regions [3] or on small free-form regions [42], in our case, we have to inpaint image planes that frequently contain very sparse visual information. In addition, the parts of the image that require inpainting are generally thin margins that follow the contour of the available color information. For this reason, we trained a lightweight inpainting CNN¹ for generating pixels in a masked region that is computed based on morphological operations.

Since our method is trained on single pictures from the Internet, therefore, no ground truth is available for occluded regions, we propose to train the inpainting model as follows: given an image layer computed by the adaptive slicing algorithm, we perform a morphological erosion in its mask in order to remove a border of pixels following the image contour. Then, the resulting image is concatenated with a mask corresponding to the removed pixels and fed to the network. The model is supervised with the original image layer (target). This process is illustrated in Figure 6 (top). During inference (Figure 6, bottom), the input mask is computed based on the dilated border from the current image layer, and both are fed to the network. The prediction is then used for inpainting. Transparent regions in the MPI are not considered in this process, such that pixels are generated on occluded regions only.

Figure 6: Illustration of the proposed inpainting process during training and inference. The goal of the inpainting is to generate pixels to extend the background image (input) according to a margin defined by morphological operations (input mask).

¹The CNN architecture for inpainting was inspired in [41] but with a smaller number of filters and convolutions.

4. Experiments

In this section, we present the implementation details of our method, as well as an experimental evaluation considering *depth estimation* and *view synthesis*, for both quantitative and qualitative results.

4.1. Implementation details

To train the depth estimation network, we used a composed loss function based on two terms, data and gradient losses:

$$\mathcal{L}_{data} = \frac{1}{N} \sum_{i=0}^N |\hat{y} - y| \quad (2)$$

$$\mathcal{L}_{grad} = \frac{1}{N} \sum_{i=0}^N \sum_{s=0}^3 |x(\hat{y}_s - y_s)| + |y(\hat{y}_s - y_s)|, \quad (3)$$

where \hat{y}_s and y_s are the predicted and the target values, considering a downscaling factor of 2^s. Similarly to [25], we normalize the target disparity maps between 0 and 10. The final depth loss is defined by

$$\mathcal{L}_{depth} = \mathcal{L}_{data} + \lambda \mathcal{L}_{grad}, \quad (4)$$

where λ is set to 0.5 in our experiments. This loss is applied individually to each output of our model (see Figure 3). At inference time, we use only the last prediction. The network was optimized with stochastic gradient descent and the Adam optimizer. Learning rate was initially set to 0.001 and reduced by a factor of 0.2 when validation plateaus. We trained the network with the images of our dataset rescaled to 1024 × 768 pixels, for about 1.2 million iterations using 6 images per batch.

For training the inpainting network, we randomly selected image layers computed by our adaptive slicing algorithm and included on it the respective layers farther in the depth range, resulting in one single image containing the pixels farther than a given depth threshold. The resulting image was then pre-processed as illustrated in Figure 6 and used for training. In the morphological dilation, we used a 3 × 3 filter with 40 iterations to produce a margin of about 40 pixels. We used pixel losses, similarly as in Equation 4, but assuming \hat{y}_s and y_s as RGB images. To increase the quality and sharpness of predicted inpainting, we also used a classic discriminator loss and a perceptual loss [15], respectively designated by \mathcal{L}_{disc} and \mathcal{L}_{perc} . The resulting loss for training the inpainting network is:

$$\mathcal{L}_{inpaint} = \mathcal{L}_{data} + \lambda \mathcal{L}_{grad} + \mu \mathcal{L}_{disc} + \nu \mathcal{L}_{perc}, \quad (5)$$

where we set $\lambda = 0.1$, $\mu = 0.01$, and $\nu = 0.01$. The inpainting network was also trained on our dataset with Adam and learning rate set to 0.0002 and 0.0001, respectively for the generator and discriminator models, for about 1 million iterations using 6 images per batch.

4.2. Depth estimation

We performed several experiments to evaluate the effectiveness of our depth estimation network, as well as to evaluate the proposed distillation process from MiDaS. In order to compare our *lightweight* network with the *teacher* network, i.e., MiDaS [25], we evaluated our approach on two public depth datasets, which are described next.

The NYU-Depth V2 dataset [28] is composed of 1449 pairs of images and depth maps densely annotated, from which 654 are used for evaluation. Despite the small number of samples with dense depth maps and the limited indoor images, this dataset is useful to perform ablation studies with the proposed CNN architecture, since the training process takes only one day on a single GPU.

The DIODE [37] is a depth dataset composed of diverse high-resolution images of both indoors and outdoors scenes, paired with accurate, dense and far-ranged depth measurements. This dataset contrast itself with other depth datasets due to its multiple scene types and employment of different sensors, aiding the generalization across different domains. This dataset is useful to evaluate our proposal on diverse and precisely annotated data, specially to compare the proposed network with the *teacher* model, considering the proposed knowledge distillation process.

4.2.1 Evaluation metrics

We evaluated our method considering classic depth estimation metrics, similarly to [7]. Namely, we used the *threshold* metric, considering 1.25 , 1.25^2 , and 1.25^3 , as well as RMSE (linear) and Absolute Relative difference (REL). However, since we learn our depth estimator from MiDaS, which has scale- and shift-invariant predictions, these metrics cannot be applied directly between the ground-truth and the prediction of our model. To allow a fair comparison, we followed the approach from [25] and aligned the depth predictions to the ground-truth based on the median and standard deviation. This evaluation is very useful in our approach, since we only require normalized depth to compute our adaptive-MPI representation. Furthermore, our dataset has no absolute ground truth depth nor camera parameters information, which prevents from evaluating depth predictions in absolute coordinates.

For new view synthesis, we use the SSIM [38] and PSNR metrics to compare predicted views with target images. Since our dataset has no paired calibrated image pairs, we compare the views produced by our adaptive slicing method with a densely sampled MPI projected to the target view point. Differently from [35], which uses a set of points with absolute depth information to align the MPI to the ground-truth depth, we are not able to provide a fair comparison on RealEstate10K since our method is relying on relative depths only.

4.2.2 Network architecture

In this part, we present a study with the depth estimation network that motivated our choices in the proposed architecture. Table 1 shows the results of our method when trained on NYU-Depth V2 only. As a baseline, we consider the same architecture as illustrated in Figure 3, but using supervision only in the last prediction, and convolutions with striding 2 and bilinear upsampling, respectively for downscaling and upscaling feature maps.

Table 1: Results for depth estimation on NYU-Depth V2 considering variations of our depth estimation network and training strategy. +MS: multi-scale predictions, +DWT: using direct and inverse DWT, +D: pre-training on our dataset. The model *lite* corresponds to the architecture from Fig. 3, and the *heavy* variation uses more filters (details in the appendix). Predictions were shifted and scaled with ground-truth based on median and standard deviation.

Method	1.25	RMSE	REL	Model Size
Baseline model (<i>lite</i>)	0.779	0.531	0.156	10.8 MB
+MS	0.793	0.514	0.149	10.8 MB
+MS +DWT	0.794	0.520	0.150	10.6 MB
+MS +DWT +D	0.835	0.452	0.129	10.6 MB
+MS +DWT +D (<i>heavy</i>)	0.869	0.416	0.113	25.2 MB

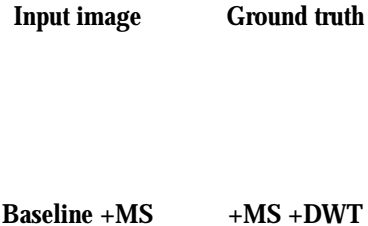


Figure 7: Predictions on NYU-Depth V2 considering our baseline model with multiple supervisions (+MS) and the proposed architecture with DWT (*lite* model version).

We improved the baseline by 1.4% in the 1.25 score (+MS in Table 1) just by including supervisions at multiple resolutions. This enforces the network to learn depth at different scales of the feature maps. Note that the intermediate supervisions are used only during training and, at inference time, only the last prediction is used. Replacing strided convolutions by DWT and bilinear upsampling by inverse DWT (row +MS +DWT) improved by an additional

0.1% and reduced the model size from 10.8 to 10.6 MB, since DWT has no trainable parameters. Despite this improvement could be considered marginal, we noticed that DWT results in sharper depth maps, as can be observed in Figure 7. This is an important factor for novel view synthesis, since most of the artifacts are produced in the borders of objects. We also show that a pre-training of the network in our dataset produces an improvement of 4.1%. Finally, we also show the results for a *heavy* version of the network, which has 25.2 MB and increases $\tau_{1.25}$ by 3.5%.

4.2.3 Comparison with MiDaS

In Table 2, we compare our depth estimation models with MiDaS. Predictions from MiDaS and our models were scaled and shifted to the ground truth, as suggested in [25]. Surprisingly, our *heavy* model, which has only 6.2% of the size of MiDaS (403 MB) and is $5\times$ faster, was able to surpass the *teacher* network on the DIODE dataset, while our *lite* model (38 times smaller than MiDaS and $14\times$ faster) reached comparable results. We believe that this fact results from our depth distillation procedure from MiDaS, which is based on an ensemble of $10\times$ predictions (as details in Section 3.1.2) and improves the robustness of our model. Moreover, MiDaS was trained with crops of size 384×384 , while our model was trained on 1024×768 , which is the native resolution of DIODE. We believe that this fact also explains our lower scores on NYU, which has lower resolution images, on which our method was not trained in. We also show in Figure 8 some qualitative results of our method compared to [25].

Table 2: Comparison of the results from our depth estimation models on NYU-Depth V2 and DIODE datasets. On DIODE, we consider MiDaS predictions in 384×384 (native resolution) followed by a rescaling, and direct prediction in 1024×768 . On NYU, all predictions are performed in 512×384 . Latency was computed on an Intel Core i5 CPU at 2.60 GHz.

Method	<i>Higher is better</i>			<i>Lower is better</i>		
	$\tau_{1.25}$	$\tau_{1.25}^2$	$\tau_{1.25}^3$	RMSE	REL	Latency
DIODE						
MiDaS $_{384 \times 384}$	0.649	0.825	0.899	7.976	0.465	2.620 s
MiDaS $_{1024 \times 768}$	0.600	0.802	0.888	9.575	0.467	19.112 s
Ours (<i>lite</i>)	0.614	0.812	0.894	6.420	0.440	1.308 s
Ours (<i>heavy</i>)	0.684	0.839	0.902	5.286	0.385	3.674 s
NYU-Depth V2						
MiDaS $_{512 \times 384}$	0.824	0.951	0.984	0.550	0.146	4.675 s
Ours (<i>lite</i>)	0.713	0.925	0.979	0.640	0.185	0.339 s
Ours (<i>heavy</i>)	0.734	0.914	0.962	0.622	0.180	0.939 s

Input image MiDaS Ours (*lite*)

Figure 8: Comparison of depth predictions by our *lite* model and MiDaS on test samples from our dataset.

4.3. View synthesis

As previously mentioned, our adaptive multiplane image slicing aims to use a small set of image planes to represent a 3D scene, while reducing the number of induced artifacts. To demonstrate the efficiency of the adaptive slicing algorithm, we used 1500 video sequences from RealEstate10K extracting a source and target image from each sequence. Varying the number of maximum planes, we used both uniform slicing and our adaptive algorithm to render the target view from the source frame using the camera transformation provided by the dataset. The results, presented in Table 3, show that for each number of planes the adaptive strategy outperforms the uniform slicing on both SSIM and PSNR metrics. It is important to notice that the difference between the two strategies is more significant as the number of planes gets smaller, reflecting the desired property of the adaptive slicing. Visual results of our method are shown in Figure 9. While 3D-Photography takes more than 400 seconds to produce an LDI running on CPU, our approach takes less than 10 seconds for 1024×768 images.

4.3.1 Subjective user study

In addition to the objective evaluation carried out, we also conducted a subjective study with users through 25 images chosen randomly from the test samples of our dataset to evaluate and compare our method with 3D-Photography [27] according to human perception. We generated videos with equivalent motion for both methods, placing them side by side in a single video. Three types of movement were used based on those provided by the 3D-Photography code. Each video was generated with 180 frames of size 1024×768 and 30 frames per second.

We asked users to evaluate two aspects of the videos: (a)

Input image 3D-Photography Ours

Figure 9: Qualitative results of our method compared to 3D-Photography on images from our dataset.

Table 3: Comparison between uniform and adaptive slicing using RealEstate10K. The number of maximum planes in the adaptive algorithm was varied as specified and the uniform slicing was performed with the number of planes resulting from our method.

Number of planes	Method	SSIM	PSNR
4	Uniform	0.6710	18.6921
	Adaptive (ours)	0.6818	18.9200
8	Uniform	0.6778	18.8438
	Adaptive (ours)	0.6836	18.9725
16	Uniform	0.6804	18.9034
	Adaptive (ours)	0.6841	18.9874

motion quality, in order to check whether the motion is natural and (b) amount of noise or artifacts present in the video. For both cases, we use the Absolute Category Rating, with 1 indicating poor and 5 indicating excellent. In addition, we allow the user to optionally comment on each assessment. We obtained, for each of the five video samples, the respective number of responses from users: 22, 18, 17, 15, and 15. For each of the videos, we calculated their average for the values of the motion and frame quality. Then, we applied the min-max normalization to change the range of averages from [1,5] to [0,100]. We also count, for each video, the number of times that someone assigned an equal or better rating to our method. Table 4 shows the Mean Opinion Score (MOS) and the percentage of times that our videos scored equal or higher for both aspects evaluated.

From these results, we can notice that 3D-Photography had some superiority in MOS. However, the difference is

Table 4: Comparison between our method and 3D-Photography according to users’ perception.

MOS	Motion Quality	Frame Quality
3D-Photography	70.24 ± 7.36	72.71 ± 8.61
Ours	65.07 ± 8.82	64.49 ± 14.23
Ours (% of)	63.85 ± 12.94	60.37 ± 23.17

not substantial, being comparable by the standard deviation. In addition, our method scored higher than or equal to more than 60% of the responses in both criteria, which is even more relevant considering that our method requires significantly less memory and computational processing.

5. Conclusions

In this paper, we described a novel method to estimate an adaptive multiplane image representation from a single image that allows novel view synthesis with low computational requirements. The main contribution presented in our work is a new algorithm that produces a variable set of image planes depending on the depth of the scene. In addition, we proposed a distillation technique that allows our lightweight CNN architecture to achieve depth estimation results comparable to a state-of-the-art model while requiring only 2.7% of the original model’s number of parameters (a reduction of 38 times in the model size). Therefore, our method is capable of producing an adaptive-MPI for high-resolution images in less than 10 seconds (about 44 times faster than 3D-Photography), resulting in an efficient method for new high-quality view generation.

References

- [1] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera. Filling-in by Joint Interpolation of Vector Fields and Gray Levels. *IEEE Transactions on Image Processing*, 10(8):1200–1211, Aug. 2001.
- [2] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image Inpainting. In *27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 417–424, USA, 2000. ACM Press/Addison-Wesley Publishing Co. ISBN 1581132085.
- [3] W. Cai and Z. Wei. PiiGAN: Generative Adversarial Networks for Pluralistic Image Inpainting. *IEEE Access*, 8: 48451–48463, 2020.
- [4] H. Dhamo, N. Navab, and F. Tombari. Object-Driven Multi-Layer Scene Decomposition From a Single Image. In *IEEE International Conference on Computer Vision*, pages 5369–5378, 2019.
- [5] H. Dhamo, K. Tateno, I. Laina, N. Navab, and F. Tombari. Peeking Behind Objects: Layered Depth Prediction from a Single Image. *Pattern Recognition Letters*, 125:333–340, 2019.
- [6] I. Drori, D. Cohen-Or, and H. Yeshurun. Fragment-Based Image Completion. *ACM Transactions on Graphics*, 22(3): 303–312, July 2003.
- [7] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, pages 2366–2374, 2014.
- [8] J. Flynn, M. Broxton, P. Debevec, M. DuVall, G. Fyffe, R. Overbeck, N. Snavely, and R. Tucker. Deepview: View synthesis with learned gradient descent. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2367–2376, 2019.
- [9] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep Ordinal Regression Network for Monocular Depth Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.
- [11] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [12] G. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*, 2015.
- [13] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam. Searching for MobileNetV3. In *IEEE/CVF International Conference on Computer Vision*, October 2019.
- [14] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and Locally Consistent Image Completion. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 36(4):1–14, 2017.
- [15] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual Losses for Real-time Style Transfer and Super-Resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.
- [16] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi. Learning-Based View Synthesis for Light Field Cameras. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 35(6), 2016.
- [17] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5574–5584. Curran Associates, Inc., 2017.
- [18] Z. Li and N. Snavely. Megadepth: Learning Single-view Depth Prediction from Internet Photos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018.
- [19] Z. Li, T. Dekel, F. Cole, R. Tucker, N. Snavely, C. Liu, and W. T. Freeman. Learning the Depths of Moving People by Watching Frozen People. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4521–4530, 2019.
- [20] S. Liu, T. Li, W. Chen, and H. Li. Soft Rasterizer: A Differentiable Renderer for Image-based 3D Reasoning. In *IEEE International Conference on Computer Vision*, pages 7708–7717, 2019.
- [21] C. Luo, Y. Li, K. Lin, G. Chen, S.-J. Lee, J. Choi, Y. F. Yoo, and M. O. Polley. Wavelet Synthesis Net for Disparity Estimation to Synthesize DSLR Calibre Bokeh Effect on Smartphones. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020.
- [22] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context Encoders: Feature Learning by Inpainting, 2016.
- [23] E. Penner and L. Zhang. Soft 3D Reconstruction for View Synthesis. *ACM Transactions on Graphics*, 36(6), Nov. 2017.
- [24] A. Pinto, M. A. Córdova, L. G. Decker, J. L. Flores-Campana, M. R. Souza, A. A. dos Santos, J. S. Conceição, H. F. Gagliardi, D. C. Luvizon, R. d. S. Torres, et al. Parallax motion effect generation through instance segmentation and depth estimation. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1621–1625. IEEE, 2020.

- [25] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun. Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- [26] J. Shade, S. Gortler, L.-w. He, and R. Szeliski. Layered depth images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 231–242, 1998.
- [27] M.-L. Shih, S.-Y. Su, J. Kopf, and J.-B. Huang. 3D Photography using Context-aware Layered Depth Inpainting. *arXiv preprint arXiv:2004.04727*, 2020.
- [28] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor Segmentation and Support Inference from RGBD Images. In *European Conference on Computer Vision (ECCV)*, pages 746–760, 2012.
- [29] P. P. Srinivasan, T. Wang, A. Sreelal, R. Ramamoorthi, and R. Ng. Learning to Synthesize a 4D RGBD Light Field from a Single Image. In *IEEE International Conference on Computer Vision*, pages 2243–2251, 2017.
- [30] P. P. Srinivasan, R. Tucker, J. T. Barron, R. Ramamoorthi, R. Ng, and N. Snavely. Pushing the Boundaries of View Extrapolation with Multiplane Images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 175–184, 2019.
- [31] R. Szeliski and P. Golland. Stereo Matching with Transparency and Matting. *International Journal of Computer Vision*, 32(1):45–61, 1999.
- [32] M. Tan and Q. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Machine Learning Research*, volume 97, pages 6105–6114, Long Beach, California, USA, 09–15 Jun 2019. PMLR.
- [33] C. Tomasi and R. Manduchi. Bilateral Filtering for Gray and Color Images. In *Sixth International Conference on Computer Vision*, pages 839–846. IEEE, 1998.
- [34] F. Tosi, F. Aleotti, M. Poggi, and S. Mattoccia. Learning Monocular Depth Estimation Infusing Traditional Stereo Knowledge. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2019.
- [35] R. Tucker and N. Snavely. Single-View View Synthesis With Multiplane Images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020.
- [36] S. Tulsiani, R. Tucker, and N. Snavely. Layer-Structured 3D Scene Inference via View Synthesis. In *European Conference on Computer Vision*, pages 302–317, 2018.
- [37] I. Vasiljevic, N. Kolkin, S. Zhang, R. Luo, H. Wang, F. Z. Dai, A. F. Daniele, M. Mostajabi, S. Basart, M. R. Walter, and G. Shakhnarovich. DIODE: A Dense Indoor and Outdoor DEpth Dataset. *CoRR*, abs/1908.00463, 2019.
- [38] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.
- [39] O. Wiles, G. Gkioxari, R. Szeliski, and J. Johnson. Synsin: End-to-end View Synthesis from a Single Image. In *Conference on Computer Vision and Pattern Recognition*, pages 7467–7477, 2020.
- [40] K. Xian, C. Shen, Z. Cao, H. Lu, Y. Xiao, R. Li, and Z. Luo. Monocular Relative Depth Perception with Web Stereo Data Supervision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 311–320, 2018.
- [41] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative Image Inpainting with Contextual Attention. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5505–5514, 2018.
- [42] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Free-form Image Inpainting with Gated Convolution. In *IEEE International Conference on Computer Vision*, pages 4471–4480, 2019.
- [43] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely. Stereo Magnification: Learning View Synthesis using Multiplane Images. In *Computer Graphics - ACM SIGGRAPH*, 2018.