Novel View Synthesis from a Single Image: Model Analysis Across Datasets

Elena Berman
eaberman@stanford.edu

Calyx Liu
calyx21@stanford.edu

Abstract

We present a scoped analysis of multi-plane image generation for the single-image novel view synthesis problem. We investigated several model approaches for single-image novel view synthesis, including Worldsheet, SynSin, and Single-View Multi-Plane Images (MPI). Ultimately, due to model constraints, we chose to focus on analyzing how image generation varies in the Single-View MPI model. Using a model pre-trained on a popular YouTube video clip dataset, RealEstate10K, we ran an experiment comparing initial layer generation on RealEstate10K data versus data from an Impressionist landscape fine arts dataset. We used a small sample size, but found no significant difference in PSNR and SSIM metrics for the model on the different datasets on the comparison between initial layers generated and input images. In the future, the model's viability in transfer learning with different data types could be analyzed through qualitative and quantitative comparisons.

1. Introduction

The problem of generating novel views from a single image has been gaining greater attention in 3D computer vision in recent years [1]. We investigated this problem because we were especially intrigued by the idea of only a single image being needed because of the greater flexibility it provides for applications and the unique technical challenges that come with generating something that has never been seen. Specifically, we investigated how transfer-learning may work in these models. Models used for this problem are largely trained on video-based data in order to compare "ground truth" values for new image views to generated new image views. The realm of fine art is well-suited as an application for this problem, since in art, we would only have one image that can be used as input. The problem of generating novel views of paintings has been demoed in models such as Hu et. al's Worldsheet, which includes a couple examples in their model page (https://worldsheet.github.io) [1]. However, no quantitative analysis on just fine arts datasets has been done on this problem to our knowledge. Our initial attempts were based on the work of Hu et al. on Worldsheet [1] because their model generated amazing images, but we soon pivoted to investigate other models as well due to limitations of our development environment. In our work, we worked on applying a single-view model to a dataset with Impressionist landscapes.

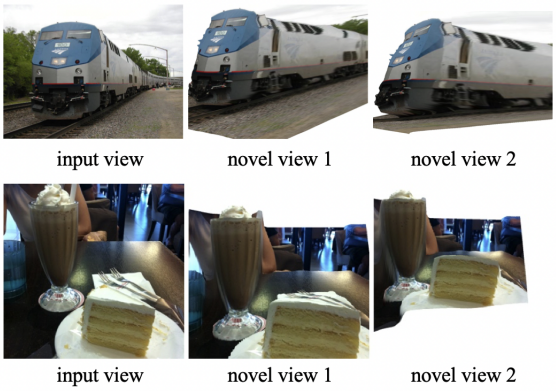

Figure 1. Two examples of Novel View Synthesis from a Single Image, from Figure 1 in Hu et. al's paper. Notice that the far-left image in both rows is the input image, and then the subsequent columns are novel views generated using the Worldsheet model.

2. Background and Related Work

2.1. Novel View Synthesis from One Image

Typical view synthesis models rely on a multiple image input. However, over the past few years, there has been a rise in the number of models attempting to generate novel views with only one image. In order to synthesize views from one image, existing models may predict depth from a single image, in addition to developing methods that examine and "imagine" what may be in the scene and visible from new perspectives but not visible from the original image's perspective [2]. An example of novel view synthesis from one image is shown in Figure 1, in which a model predicts two new views from the single input image.

2.2. Worldsheet

As mentioned in the introduction, we started with Worldsheet. They used a new technique where they construct a mesh of the scene by laying a lattice grid on top of the image and warping it based on grid offset and depth. They are then able to use this information to project the image's features according to the new view's angle. Finally, they use an inpainting network to fill in any missing sections and refine details on the already visible parts of the image.

2.3. SynSin

Another model that we tried to use and that Worldsheet compared their results to is SynSin [3]. SynSin's approach involves rendering a 3D point cloud based on features in the image but positioned to be from the new view angle. They then use a refinement module to clean up the details and inpaint any previously hidden sections.

2.4. Single-View Multi-Plane Images (MPI)

Tucker & Snavely [2] published the first method to synthesize novel views from a single-image using multiplane methods in 2020. Multi-plane images were first developed for use in stereo problems; however, Tucker & Snavely applied these to the single-image novel view problem (Figure 2). Multi-plane images are a set of layered-images, each parallel to the original image but corresponding to a different depth. Each plane/layer has RGBA values. These images are generated by an approach based on previous research the authors had done on a model called Stereo-Mag ([4]), which utilizes internet video data to train a deep network that predicts depth in images and generates multi-plane layers for a single image. It does this by warping each plane layer and applying it smoothly to the image. In addition to the deep network that generates multi-plane images, Tucker & Snavely's Single-View MPI model uses scale-invariant techniques for training by computing a scale factor from the data, and a loss function that combines L1 per-pixel loss, an "edge-aware smoothness" loss, and a sparse depth supervision loss. The development of a scale-invariant technique is significant because it enables training on internet video data in which the scale is not previously known. The Single-View MPI model is evaluated on the RealEstate10K dataset, along with a couple other datasets. Numbers are computed for PSNR and SSIM between source and target values in the generated image (a predicted view) versus the known image (actual view) derived from video sequences.

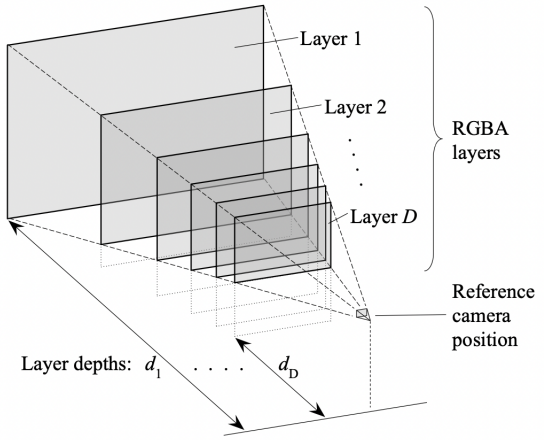

Figure 2. Tucker & Snavely's Multi-Plane Image approach: Figure 2 from their original paper

3. Approach

3.1. The Data

3.1.1 Real Estate 10K

The Real Estate 10K dataset contains images from video clips from 10,000 YouTube videos. Example images are shown in Figure 3. In total, there are 10 million frames in the dataset. The training dataset consists of 90% of the images in the dataset, with the test dataset comprising the remaining 10%. Each video clip has a text file containing time stamps, a camera intrinsic matrix, and camera pose matrices for each image frame. Additionally, because of the video clips that can serve as "ground truth" for novel view synthesis, various models have been trained on Real Estate 10K data/real estate videos from YouTube, including Worldsheet ([1]), StereoMag ([4]), Single-View MPI ([2]), and SynSin ([3]).

Because of this, the Real Estate 10K dataset was particularly significant for our purposes. However, the data format is notable. The dataset consists of .txt files corresponding to YouTube videos, and therefore requires pre-processing and additional code in order to obtain images. We discuss our approach to this in Section 3.2.1.

3.1.2 Impressionist Landscape Paintings

Additionally, we chose to use a fine arts dataset to analyze how novel view synthesis models performs on images that are extremely different. The dataset we used is the Impressionist Landscapes dataset on Kaggle. This is a dataset that includes 5,000 paintings in 1024x1024 RGB format. Impressionist landscapes present an interesting opportunity to evaluate how models perform in a "transfer learning" fashion. The fine arts may be additionally challenging in depth

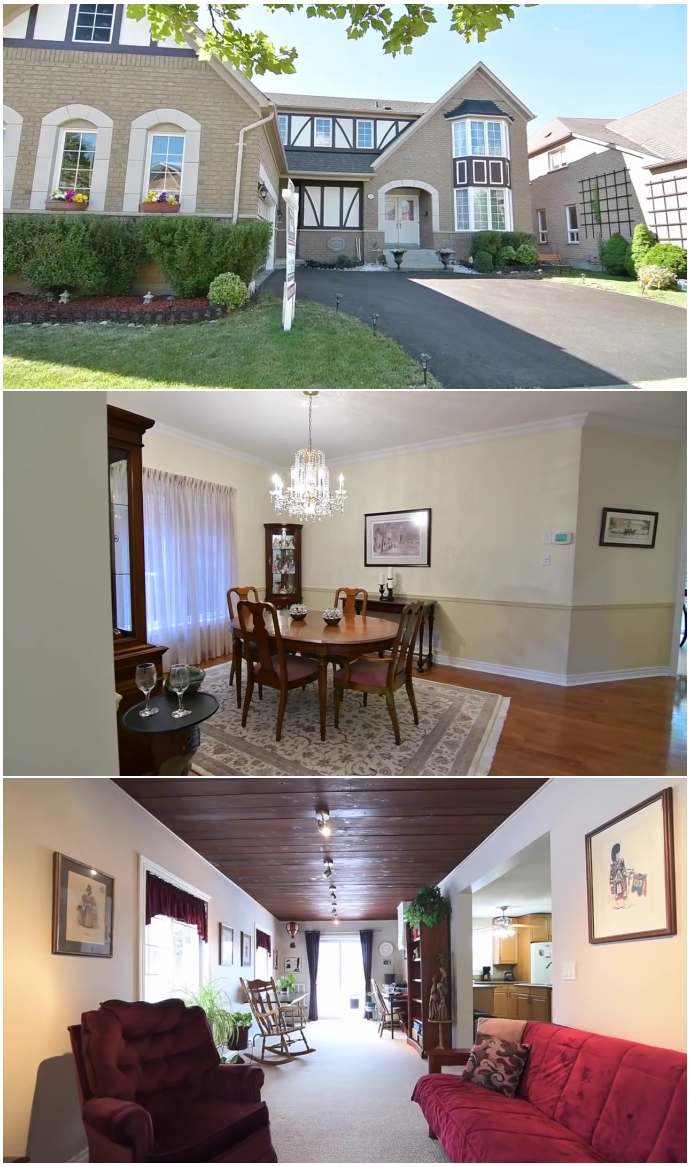

Figure 3. Three sample images from RealEstate10K dataset

prediction step if the original model has been trained on video data. Sample images from this dataset are shown in Figure 4.

3.1.3 Matterport3D

Originally, we planned to utilize the Matterport dataset, which is another dataset used in Worldsheet and other models for single-view novel synthesis. In order to access the dataset, we need the permission of the group who owns the dataset. We reached out to the group, and they initially responded to us letting us know that they would let us use it if we provided the names of our PIs on the project. We reached out to the CS 231A course staff and obtained per-

Figure 4. Three sample images from Impressionist Landscape dataset (works in this figure are by Alfred Sisley, Claude Monet, and William James respectively)

mission to use the CS231A instructors' names as our listed PIs. We emailed the Matterport owners back with the new names a couple weeks before this project was due, but they have not responded yet despite a follow-up email we sent a few days later, so we could not use that dataset in our project.

3.2. Initial Experiments

Originally, we only wanted to expand on Worldsheet's model and run it on the RealEstate10K and the Matterport3D datasets to establish a baseline before moving on to the Impressionist dataset and our modifications. In order to accomplish this, we started by trying to get Worldsheet's pretrained models running and extracting the files we need from the RealEsate10K dataset.

3.2.1 RealEstate10K Data Extraction

The RealEstate10K dataset contains frames from 10,000 Youtube videos of house tours. It is structured as a collection of text files where each file contains a URL to the corresponding Youtube video and several lines representing specific frames in the video. Each line contains a timestamp as well as the camera intrinsics and pose. In order to convert these text files into frames that could be used by the model, we wrote a Python script (with greatly appreciated help from Josh Wiedemeier!) that downloads the Youtube video to the VM before iterating through each frame and saving the ones that correspond to the timestamps listed in the dataset. Due to what Worldsheet needs in order to run, we also created a separate text file for each video containing the camera parameters for each frame that was successfully saved. After 2 days of running (with interruptions due to Youtube blocking us for sending too many requests), we were able to obtain frames for about 1500 videos.

3.2.2 Worldsheet Model

In order to get a working baseline model from Worldsheet's code, we needed a VM that had a GPU. So, we tried using Google Colab and a GCP virtual machine.

Unfortunately, Google Colab has strict timeouts that result in the VM powering down and the notebook to be reset. Therefore, it was not usable as the notebook would time out in the middle of uploading the dataset.

Due to its ability to store persistent data, we had a little more luck with the GCP VM. However, a significant setback in getting our model working was that when using Hu et al.'s GitHub repository, there were many dependencies that had to be installed, including other Facebook Research repositories. Because of this, we had to increase the memory of our virtual machine during the process. Additionally, when we tried to run the pre-trained model after installing all of the dependencies by following their instructions, we kept on running into error after error. One of us had permission errors while the other one had countless errors related to pytorch. After several days of trying to get the model running, we decided to look for another one.

3.2.3 SynSin Model

The next model we tried was SynSin due to it being featured in Worldsheet's paper. However, we continued to run into errors with installing dependencies. Eventually, they were installed. However, when the jupyter notebook containg a simple demo of SynSin was run, it failed due to not being able to find a CUDA module. After several more days, we turned to a third model.

3.3. Analyzing the Pre-trained MPI Model

Finally, we pivoted to analyzing the Single-View MPI model [2]. This model uses a very different framework than Worldsheet in order to approach the single-image novel view synthesis problem. In particular, multi-plane images (as shown in Figure 2) are generated from training on online videos (as shown in Figure 5). The overall loss function utilized in training is:

$$\mathcal{L} = \lambda_p \mathcal{L}^{pixel} + \lambda_s \mathcal{L}^{smooth} + \lambda_s \mathcal{L}^{depth}$$

The steps in which the edge-aware smoothness loss ($\mathcal{L}^{smooth}$) and view synthesis loss ($\mathcal{L}^{pixel}$) are applied is shown in Figure 5.

In our project, we specifically analyzed the multi-plane image layer-generation model pre-trained on RealEstate10K's model. This model utilized L1 pixel loss for its view synthesis loss, which compared the view generated from the model to the actual target image using RealEstate10K video clips:

$$\mathcal{L}^{pixel} = \sum_{channels} \frac{1}{N} \sum_{(x,y)} |\hat{I}_t - I_t|$$

The MPI layer-generation model is particularly interesting to analyze because the multi-plane images are then used to implicitly "inpaint" novel views in an image. In the next section, we describe our experiment to quantitatively analyze this model.

4. Experiment

4.1. Single-View Multi-Plane Image

In order to evaluate the Single-View MPI model, we first used the demo Colab notebook available through the GitHub repo (https://github.com/google-research/google-research/tree/master/single_view_mpi). This Colab notebook implements the model used that takes in a single input

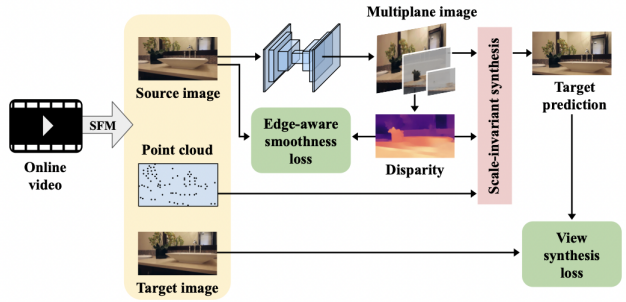

Figure 5. The Single-View MPI training and model approach, Figure 3 in Tucker & Snavely's original paper

image and returns layers. Then, it renders the layers in an interactive view.

We analyzed the model in two steps: quantitatively and qualitatively. First, we investigated it quantitatively. Given an input image, the model generates layers. We compared the input image to the first layer generated using two common metrics for analysis on this task: PSNR and SSIM. PSNR stands for "Peak Signal-to-Noise Ratio." It is used to calculate the "cleanliness" of a reconstructed image by measuring the amount of noise in an image vs its clarity with no noise. SSIM stands for "Structural Similarity Index." It measures the similarity between two given images. This is useful for our purposes because it allows the generated image to be generated to a "ground truth" image which is an actual picture of the view being generated. Note that this comparison is different than the comparison done in Tucker & Snavely's original paper since we are not comparing a generated image to a "ground-truth" image. Instead, there is no "ground-truth" image in our comparison. However, we thought this comparison would be interesting because it would give context for the similarity, using PSNR and SSIM, between depth planes generated by the MPI model and the original image. Additionally, we thought that this comparison had the potential to be interesting because we could compare metric numbers across two datasets: RealEstate10K, on which the model was originally trained, and Impressionist Landscape Paintings, which presents a completely new type of data in comparison.

In order to do this quantitative analysis, we added functions to calculate PSNR and SSIM in numpy from image arrays, as well as functions to sample images from the each dataset and run the model on these images to the demo Colab. Because the datasets are huge and we had a limited amount of time to finish this project, we chose to sample 100 images from each dataset and run the Single-View MPI model on these images, and calculate PSNR and SSIM values. Results are shown in Table 1. The PSNR and SSIM values are average values across the dataset. Additionally, in order for the model to run, we resized the RealEstate10K

Dataset	Sample Size	PSNR	SSIM
Impressionist Landscape Paintings	100	0.99	76.27
RealEstate10K	100	0.99	77.83

Table 1. Mean PSNR and SSIM values for original image and first layer generated by MPI model.

images to be square-shaped in the same size as the Impressionist images (1024 x 1024) by padding them vertically and cropping on the far-right side. Note again that these numbers are not directly comparable to PSNR and SSIM numbers from the original Single-View MPI paper since we compared original image to the first generated layer for each calculation instead of a "ground truth" image.

We see that the PSNR and SSIM values are not significantly different between the Impressionist Landscape Paintings and the RealEstate10K datasets. This could suggest that the layer-generation approach that the model uses is transferable to different types of image data. Although the model to generate multi-plane images was trained on RealEstate10K originally, the similarity between the original image and layers generated in the model is consistent across datasets. A limitation of this analysis is the limited scope: we only investigated the first layer given our time constraints, and it is possible that the first layer is more similar than subsequent layers. In the future, more analysis can be done on the multi-plane image generation steps across different image types.

Next, we investigated the model qualitatively. To do so, we used an image from the Impressionist dataset and an image from the RealEstate10K dataset and generated layers and novel views using the demo notebook. Demo videos showing the result are available in our GitHub (the "Video Demos" folder). The layers generated are shown in Figures 6 and 7. In our analysis of the interactive components (available in our Demo Videos) we noticed that in the RealEstate10K demo, the background seemed very sharp and there was limited blurriness or distortion. The Impressionist demo was also well-formed and showed depth in many places we expected; however, some portions such as the cloudy sky (in the Monet painting) seemed more distorted. Since we only used a sample size of one for each, it is difficult to draw conclusions about whether this variation is due to a difference in datasets or another factor; however, it could suggest that the medium or content represented in the Monet painting is more difficult to capture without distorting compared to the RealEstate10K image.

4.2. Project Code

The code utilized in this project is available on GitHub through the following link: https://github.com/CalyxLiu/CS231A-Project.

It includes scripts used to process the RealEstate10K

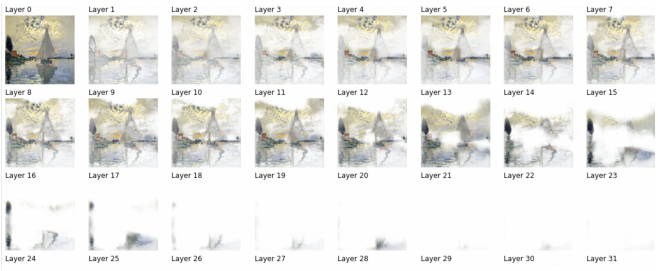

Figure 6. Visualization of MPI-generated layers for an Impressionist image

Figure 7. Visualization of MPI-generated layers for a RealEstate10K image

files as well as the two Colabs used to compute the results obtained in Table 1 (one for the Impressionist Landscape Paintings dataset and one for the RealEstate10K dataset). Additionally, it includes a demo video of the novel views generated from an Impressionist landscape painting using Tucker & Snavely's single-view MPI model.

5. Conclusion

We present a scoped analysis of single-view MPI on two datasets. On the quantitative side, we present metrics calculated from the initial layer of a generated multiplane image as compared to the actual image in samples from two different datasets, and do not observe a significant difference. On the qualitative side, we ran the model on a randomly-selected image from each dataset and observe similarities and differences in distortion and apparent accuracy of depth. Expanding on these analyses on the future by using different metrics to compare layers and using larger data samples for both quantitative and qualitative analyses could provide more information on how viable transfer-learning is in the single-image novel view synthesis problem.

Through the course of this project, we followed a nonlinear path in investigating multiple models, informing us on the different approaches to single-image novel view synthesis, from Worldsheet to MPI.

Additionally, we interacted with running scripts to download and extract images from videos from YouTube. In the future, we believe it would be advantageous in the field of computer vision if the image datasets were available as pixels instead of as YouTube URLs, since it may create a barrier or cause duplicate work in data processing to access data for future research.

References

[1] Ronghang Hu, Nikhila Ravi, Alexander C. Berg, and Deepak Pathak. Worldsheet: Wrapping the world in a 3d sheet for view synthesis from a single image. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021.

[2] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[3] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. SynSin: End-to-end view synthesis from a single image. In *CVPR*, 2020.

[4] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *CoRR*, abs/1805.09817, 2018.