UKRAINIAN CATHOLIC UNIVERSITY

BACHELOR THESIS

Synthesizing novel views for Street View experience

Author:
Anastasiia LAZORENKO

Supervisor:
Philipp KOFMAN

A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science

in the

Department of Computer Sciences
Faculty of Applied Sciences

Lviv 2019

Declaration of Authorship

I, Anastasiia LAZORENKO, declare that this thesis titled, “Synthesizing novel views for Street View experience” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Bachelor of Science

Synthesizing novel views for Street View experience

by Anastasiia LAZORENKO

Abstract

Navigational applications often suffer from restricted and granular movement possibilities caused by a limited capture of real-world locations. Even the largest collections of street photos like Street-View, Mapillary [31], and SPED win more in geographical coverage than in qualitative capture of specific scenes. A possible solution to this problem could be post-processing of available image collections and generation of new photos that would restore the missing parts. This is the task of novel view synthesis - a known area in computer graphics and vision, that has shown impressive results over last several years [26], [27], [33], etc. However, the problem of real-world outdoor scene reconstruction is the most challenging, and is still a subject to active research. In this work we will explore different approaches to novel view synthesis and evaluate some of them on the sparse real-world imagery from Street-View dataset.

Acknowledgements

Genuine gratitude to Philipp Kofman for supervision of this work.

Contents

List of Abbreviations

NVS	**N**ovel **V**iew **S**ynthesis
SFM	**S**tructure **F**rom **M**otion
RBF	**R**adial **B**asis **F**unction
GAN	**G**enerative **A**dversarial **N**etwork
NeRF	**N**eural **R**adiance **F**ield
SIFT	**S**cale **I**nvariant **F**eature **T**ransform
ORB	**S**cale **I**nvariant **F**eature **T**ransform
SoTA	**S**tate **O**f **T**he **A**rt
MVS	**M**ulti **V**iew **S**tereo
CV	**C**omputer **V**ision
CPW	**C**ontent **P**reserving **W**arps
SSIM	**S**tructural **SIM**ilarity index
MS-SSIM	**M**ulti **S**cale **S**tructural **SIM**ilarity index
PSNR	**P**eak **S**ignal to **N**oise **R**atio
MSE	**M**ean **S**quare **E**rror
LPIPS	**L**earned **P**erceptual **I**mage **P**atch **S**imilarity

Chapter 1

Introduction

Virtual navigation for either leisure, commercial, or educational purposes has become an important part of our lives. Especially so at these times of restricted traveling possibilities. An important problem in this area is that of *novel view synthesis*, which refers to the generation of additional images of a known scene from different perspectives.

In this work we will review the main types of NVS approaches, and their historical development. Then we will evaluate several of them on outdoor urban imagery. The best known application with the highest coverage rate for virtual world navigation is Google StreetView, so improving its experience would make the most impact. We will describe the process of collecting data from the service, the challenges of NVS in such a varied environment, and finally compare the performance of several methods on this dataset. The two main flaws that we will try to solve with the latest approaches to novel view synthesis are:
1) lack of smoothness in transitions between photo-locations
2) restriction of movement, only allowing to follow a trajectory of the capturing device.

Chapter 2

Background

2.1 NVS definition

The goal of Novel View Synthesis (NVS) is to generate a realistic scene representation for an arbitrary camera position based on available views of that scene. Basically, figuring out what would something look like when viewed from a different angle, having previously seen it from just a few different camera positions.

This high-level problem formulation in practice includes several sub-tasks. First of all, understanding the spacial context of the scene. This can be achieved by either an explicit reconstruction of the underlying 3D shapes or by leveraging deep networks and learning relations between the neighbouring views and the target image or learning the patterns of pixel flow between adjacent views. Both strategies have several disadvantages in a general setup, but have successful applications under specific constrains. For example, explicit usage of precomputed 3D shapes on datasets of standalone synthetic objects that belong to a common domain like furniture models or car designs can simplify the task greatly, but would be useless for real-world imagery datasets that deal with significantly wider variety of object domains, cluttered scene compositions and occlusions. And vice versa, using approaches calibrated for complicated scenes on standalone objects would fail to recreate the object at the level of detail as high as it is usually required for applications offering a single-item overview.

Another common sub-task in NVS pipeline for real-world imagery is camera calibration, performed as part of data preprocessing. While there are some datasets collected with identical cameras (like Street-View and KITTI), they are either small or commercialised. So open-source image collections captured with variable hardware (e.g. Mapillary platform for street-level photography, Real-Estate dataset of indoor apartment photos, etc.) need to first be standardized. Also, when trying to use unstructured photo collections, pose estimation and pose correction are necessary.

As an optional but quite helpful step for unconstrained collections of outdoor photography some approaches [NeRF-W] also use semantic segmentation to filter out images that contain massive occludors. Occlusion can be caused by an unfortunate camera position or a captured passing-by object, and can complicate models' interpretation of the scene.

NVS applications

Computer graphics

Novel View Synthesis first appeared in computer graphics industry as an alternative to volume rendering. A traditional volume rendering process was operating a 3D model and used computationally expensive physics engines to perform tasks like texture mapping, shadow locating, determining local lighting conditions, glares, deformations, etc. Often if a shift of viewpoint is very slightly (e.g. in smooth 3D animation or visual effects), triggering the whole rendering process from scratch just wasn't efficient. So, a group of image-based rendering algorithms were proposed that synthesised intermediate views between already rendered frames with image interpolation. Consequently compensating with memory cost due to the need to store those key frames, for improvement of more crucial characteristics like: rendering latency and frame rate.

2.1.1 Visual effects in cinematography and animation

For some visual effects, NVS also found application cinematography. A descriptive example of using view interpolation mentioned above is the Bullet Time scene in 1999's Matrix. Making a high-resolution shot of such slow action would have required way more cameras than it was physically possible to place on a camera rig. So in order to up-sample the captured frames and produce a pleasantly looking slow-motion, novel view synthesis was used to generate frames as if taken from viewpoints on a rig between the real cameras. [CVPR 20] Interestingly, such collaborations also bring ideas from animation innovations to computer graphics works. For example, Disney's idea to split a scene into several layers according to depth of its objects inspired multi-plane data structure used for small interpolations on densely photographed environment. As well as later representations built on it (more details on that in Scene Representations section). The main idea behind it is to shift the planes with higher depth value less in order to simulate the natural slow motion of farther placed objects.

2.1.2 Virtual Reality

In its core, novel view synthesis is a way to reconstruct complete visual information about a world scene. And thus is essential for realistic observation experience in VR applications. one would need to obtain a photo or a rendering for every reasonable angle and position for the whole scene. Which would be quite costly (if even possible). This makes novel view synthesis essential for virtual reality experiences that recreate real-world places or require high rendering speed on lower-class hardware.

Apart from VR, novel view synthesis has applications in regular fields that use embedded interactive 3D presence modules. Some specific examples include: providing unrestricted view of items in online shopping, fixing the eye contact issue in video-conferencing applications [5], adding depth and inter-occlusion to custom backgrounds, improving presence experience in navigation software, enhancing photo editing by allowing 3D object manipulation [14] or perspective changes, making an interactive 3D scenery from users' sketches, etc.

Virtual Tourism

Virtual tourism (especially relevant at the time of writing) and the concomitant immersive routing experience is (arguably) the most interesting and challenging application, as it's concerned with the raw outdoor scenery. In one or another form virtual tourism existed for a long time, but restriction of movement and discrete transitions between available viewpoints make it feel less real. To quickly obtain intermediate frames and make movement smoother, novel view synthesis is required.

For a long time virtual tourism was only available in a form of prerecorded video sequence or quantized navigation interface between photographed locations [from first rel. works]. Recent spread of capturing technology (available drones, Street View car fleets, etc) and advances in view synthesis for natural scenery made possible experiences like Google Earth (where the popular sightseeing destinations are well-captured by aerial photography and available for for view from any position in high resolution, and the rest of the world is recreated in lower fidelity, more like for reference).

2.1.3 Training improvement for other computer vision tasks

Collecting an optimal amount of data is undeniably important for successful training and better generalisation of machine learning models. For many areas of computer vision, there's a limited choice of big datasets, and construction of custom ones is time-consuming. Dataset augmentation is a known technique for artificial upsampling of data that helps overcome this problem. However, it has high variety and size requirements for initial datasets as well (due to commonly used generative models which require more data). Several NVS methods are well optimised to work on limited amount of input views, and can produce additional data samples for smaller datasets. Some of computer vision tasks, like object detection, semantic segmentation, and classification tasks, have been proven to even gain performance and robustness when trained on NVS-augmented data. [24]

2.1.4 Robotics

Novel view synthesis is concerned with both static (mostly) and dynamic (e.g.[35]) scenes. In vision-dependent robotics systems, synthesis of future frames under different views helps in planning of collaborative activities [15], and making robot traversability predictions in unstable environments [12].

2.1.5 3D Video stabilization

When a camera capturing some video- or photo-sequence has a jittery trajectory (e.g. first-person camera for streaming extreme sports, hand-held camera capture an amateur video), producing rapidly changing views, we would like to stabilize the resulting video for a more comfortable watching. This can be achieved by selecting several key camera positions from existing chaotic trajectory, constructing a smoother one, and generating missing frames between those key positions. To generate an intermediate view, different approaches either use frame stitching with 3D proxies (Kopf, Cohen, and Szeliski 2014) or image warping (CPW) guided by detected feature points. The latter one can also be enhanced if textureless region (like ground or building) that lack feature points are handled as planar surfaces transformed by estimated homography, as described in [39].

2.2 Evolution of NVS approaches

2.2.1 First works. Interpolation between densely-spaced views

Contrary to common belief of *View Interpolation for Image Synthesis (1993)* by Shenchang Eric Chen and Lance Williams [3] being the seminal paper in the field [29], the idea of generating intermediate views using interpolation technique was first introduced in *A Novel Approach to graphics (1992)* paper by Tomaso Poggio and Roberto Brunelli [23]. The authors tried to decrease the computational complexity of graphic animation rendering, by synthesising intermediate views from a set of stored key frames instead of computing the frame from a 3D volume. Due to specifics of the intended application, the views of interest were assumed to be described by a set of input features/conditions (e.g. a 2D position, an action name, or a facial expression in case of character animation), and generated rather realistic than theoretically correct. For each animation setup a network of radial functions would be created based on initially given key data samples (frames along with meta features), where each radial function would be responsible for one known frame. Then, much like in neural networks, a novel frame could be obtained from superposition of results these functions would give for its input features. To overcome the challenge of direct generation of pixel values, such networks would actually produce an approximate sketch of final frame - a matrix of control points on which the new frame's texture would be later inpainted. [23].

View Interpolation for Image Synthesis (1993) however had a different approach for interpolation process - applying morphing algorithm to image pairs with precomputed correspondence maps. Image morphing is generally performed in two stages: first, the input images are warped into an approximated common shape to avoid mutual cross-fading effect after pixel re-combinations; after that, each pixel of the resultant image is computed as a weighted combination of input pixels from respective area. The last stage can be a simple nearest neighbour combination or can perform an adaptive interpolation, that takes into consideration not only a specific region of image, but also the underlying textures and presence of any edges. Both stages require a prior knowledge about pixel correspondences in a given image pair. Which used to require human supervision and posed a strong argument against usage of morphing algorithms for systems with high latency requirements. However, as the experiments in this [3] paper were performed on synthetic images, the image ranging values (basically the distance between an image point and the optical center of the virtual camera) could be used for finding such per-pixel correspondences. Furthermore, to avoid this still time-expensive search step, the correspondence maps (or "morph" maps) were precomputed and stored as arrays of 3D offset vectors, unique to morphing direction for each pair. This decision, although compromised method's memory cost, did accomplish the goal of minimising the rendering latency.

Both of described approaches assumed camera step (i.e. baseline) to be very small, and could only preform view interpolation on a well-captured scene. For more advanced tasks, like interpolation on sparse datasets and extrapolation in principle, a better understanding of the underlying 3D structure was required. From this point, NVS approaches range between two main categories:

- **geometry-based synthesis**, splitting the NVS task into explicit 3D shape reconstruction or acquisition, and then using it to re-render the scene for a novel 3D view point;

- **learning-based synthesis**, leveraging the power of deep neural networks to predict the output frame based on input images, often implicitly working with geometry obtained through convolutions or encoding.

2.2.2 Geometry-based approaches

3D reconstruction being a field of computer vision on its own, gave geometry-based approaches a good choice of reconstruction techniques for all different tasks.

Scene geometry can be described with volumetric primitives (points, polygonal meshes, voxels, etc.) or with mappings from known pixels into 3D space known as depth maps. Sometimes it makes sense to introduce a custom volumetric primitive designed for a specific case. Like trees of polyhedral primitives in [6] entitled to model basic architectural building blocks and simplify the task of building's model reconstruction).

In NVS applications designed for single-class standalone objects, it became common to use a predefined base 3D model with features shared by most of class samples, and adjust it according to input images of a specific object. While for more challenging cases of multi-class objects of non-uniform shapes, an explicit 3D model should be constructed under a supervision from user controls, or using approximation techniques. Shape approximation technique was chosen depending on the number of available views and preferred data structure for shape representation.

The shape representation that is most flexible in terms of input requirements is point cloud. Point clouds can be used for both individual objects and natural scenes, and are produced by a Structure-From-Motion (SFM) algorithm, which only requires multiple input views (technically, at least two). The idea behind SFM reconstruction is that if every point visible from a camera belongs to a ray passing though this camera's optical center, then given a big amount of points and several cameras, we can detect planar relation between such rays, and recover coordinates of both points and cameras relative to a basis placed on an arbitrary chosen camera. The most challenging part of SFM actually precedes this pose estimation process and lies in point matching across the available frames. Point correspondence is now performed by feature detection algorithms, that are designed to be invariant to some image transformations, thus able to find same unusual key points (e.g. pixels on edge intersections, contrasting textured regions, thin lines, or sharp contours) on images from different camera positions. Examples of widely known feature detectors nowadays are SIFT (Scale-invariant feature transform), SURF(Speeded Up Robust Features), and ORB (Oriented FAST and Rotated BRIEF). For example, point clouds were used in *Content-Preserving Warps (CPW) for 3D Video Stabilization (2009)* [17] that pioneered 3D video stabilization a structure-aware image warping algorithm was used that worked with scene's point clouds. The generation of a plausible frame sequence for a new, smoothed camera trajectory is performed in the following way. First, the initial trajectory is recovered, and the point cloud representation of a captured scene is reconstructed using SFM algorithm. Then the smoother trajectory is aligned around the initial one, and initial frames are warped into novel appearance simultaneously minimising the displacement of point cloud and the distortion the current 2D frame. Of course, the resulting stabilized video sequence is not quite geometrically coherent due to described compromise, but the error is not noticeable to human perception. [17] Later, this method was improved to also handle less textured areas, that lacked feature points necessary for successful image warping. In *Plane-Based Content Preserving Warps for Video Stabilization* [39] it was suggested to label such regions during frame segmentation, and treat them as planar surfaces that can be manipulated according to an estimated homography matrix. The texture-rich regions are left unlabeled, and are handled by the standard CPW, thus not degrading performance for previous cases.

On the other hand, depth values can be obtained from any amount of initial views. Since depth maps have a simpler structure than actual 3D shapes, they can be successfully predicted even from a single frame by deep learning methods. For cases with several perspectives available, as well as for imagery taken with a stereo camera, multi-view stereo (MVS) algorithms are widely used.

2.2.3 Learning-based approaches

The reasoning behind using imagery alone in learning-based NVS comes from the interpretation of NVS goal in context of real-world applications. Yes, in many cases realism and image quality is preferred over exact geometrical accuracy, as long as this geometry error is left unnoticed by observers. Getting a highly-detailed surface model of non-trivial scenes (nature photography, elaborately detailed objects, different object compositions, etc) is still a challenging task. So for the cases where details matter using methods purely based on image data is recommended. Intuitively, unless a NVS task was constrained to a specific data domain, applying machine learning methods should make more sense, as unseen data could be derived based on more high-level data patterns.

With recent breakthroughs in machine learning field, neural networks became capable of solving more abstract and complex tasks, and gained lots of attention from other fields. Soon the first works in NVS using the power of neural networks were introduced. They were obtaining an implicit geometry representation from input imagery using convolutional neural networks, and then conditionally generating pixels of the novel image from scratch. [30] [34] This formulation was too complicated for a network to learn on limited data. Therefore, early learning-based approaches were still limited to applications with a small number of object domains, and required lots of training data in order to generalise.

A better problem formulation was suggested by Zhou et al in *View Synthesis by Appearance Flow* [38]. Instead of pixel values, the network was predicting flow vectors, that held the coordinates of the source pixel from input image. This way the learning task was simplified and was more view transformation than naive image synthesis. Nevertheless, there it also had a downfall in its inability to generate regions that are not present on the source image. So, later followed an attempt [28] trying to combine pixel-wise generation and flow prediction with a learned confidence mechanism.

Another way to simplify the learning task - or rather to separate the part that actually needs to be learned from simpler parts that don’t worth spending time to teach the network to perform - is to optimize the scene representation:

Scene representation

In the simplest case, NVS is used on a posed photo collection, and produces novel views in a form of simple 2D images. However, a good choice of data structure for representing a scene can simplify the task significantly, and offer competitive results at fewer computational complexity. For example, for interpolation between densely-sampled photo set, multi-plane images are able to use the depth information alone to transform the input imagery into a qualitatively acceptable results. They transform an image into a stack of planes, each holding a unique part of image, according to its known or estimated depth. So that a 3D presence effect can be achieved by slight shifting of the sub-planes according to their depth values (when those with smaller depth move more drastically and recreate the contrast between easy foreground changes and almost static background called motion parallax).

A good example in context of learning-based approaches is DeepStereo work by Flynn et al [7]. Authors there represent scenes with multi-plane sweep volumes so that input views are reprojected onto a common viewpoint plane for several depth values and stacked together before being fed to the network. This representation spares the network a need to learn concepts of camera rotation, and limits the pixel areas used for determining values on synthesized view. In a later work [27], Yicun et al suggested an improved version of similar construct by replacing pre-set discrete depth values with self-learned depth ranges, and reprojected views with pixel displacements.

2.3 Main challenges

2.3.1 Disoclusions, Holes, Cracks, Artifacts

Both geometry- and learning-based approaches have issues that need to be addressed when choosing a method for application. When geometry-based method is used on sparse dataset and gets to reconstruct disocluded regions, it often suffer from 'holes' in rendered scenes in case a novel viewpoint is significantly different from original ones. To overcome this issue, a variety of hole-filling CV algorithms was developed. Most of them attempting to retrieve its texture from neighbouring patches or closest. Although they don't perform well on highly-detailed areas and areas of high curvature [16]. Another approach to this problem considers dividing scene objects into segments, and assuming all pixels that belong to the same segment have common texture. In this case holes fillings are derived from semantically reasonable neighbouring areas. Of course, a deep learning approaches already exist for hole filling. E.g. Ambient Point Clouds for View Interpolation by Goesele et al, 2010 [8].

Meanwhile, learning-based approaches that avoid 3D supervision are left with no context about underlying shapes can produce visual artifacts like ghosting of dislocated objects, sketchiness of thin shapes, and uncontrolled patterns on non-Lambertian surfaces. An alternative approach that allows to not deal with pixels directly and operate on gradients instead.

Chapter 3

Related Works

The ambitious goal of implementing an unconstrained interactive walk through a reconstructed real-world environment has been a subject to research for a long time.

3.1 Movie Maps: An application of the optical videodisc to computer graphics

One of the first attempts of a our world walk-through system was developed as early as in 1980-s. Back then, displaying a scene was costly, so a choice of route could not been performed interactively. The solution and the main innovation of the Movie Maps work was simultaneous usage of two optical videodiscs to serve the current tour sequence and to preload views for potential next location at real time speed.

3.2 The Virtual Museum Interactive 3D navigation of a Multimedia database

Developed in 1992, the Virtual Museum was aimed at bringing more immersive experience to virtual museum visits. For exhibition objects it had a pre-rendered animation sequences, triggered on item selection. Its innovation was however in a different aspect, - panning and transitioning moves allowed to simulate a discretized walk around the museum building. All the virtual navigation and animation was served using a real-time video decompression method.

3.3 Modeling and Rendering Architecture from Photographs: A hybrid geometry- and image-based approach

The first related work actually leveraging the power of an NVS approach was introduced by Debevec et al in 1996. Using photogrammetric modeling to obtain scene geometry approximation allowed to simplify stereopsis task and obtain more detailed depth maps. Which in turn improves the texture mapping process, and produces a much higher-quality views even from sparse frames. Its tight focus on building domain, however, poses significant limitation in applications. Theoretically it could be extrapolated to an architecture-oriented reconstruction of some city, but not to a general-purpose virtual touring, as it would need to give up its building-optimized scene representation structure in order to model other objects.

3.4 Google Maps' Street View Experience

The first outdoor walk-around experience with huge world coverage was introduced as part of Maps experience by Google. User movement there is still limited to the trajectory of capturing camera, but each photographed location is represented with a pan-able panorama, giving sense of local.

3.5 Google Earth

Presented in 2001, Google Earth is specifically made for virtual tourism, and also uses NVS for environment completion. The main tourist destinations are densely-captured by aerial cameras, while the rest of the world is reconstructed. We haven't found any official disclosure about used method or training data, but we suspect that they used satellite data, as there are many NVS approaches that performed qualitatively better when tested on Street View.

3.6 Building Rome in a Day

A work on 3D city reconstruction from unstructured datasets of amateur photography. This task is complicated by the need of pose estimation and camera calibration for the various data sources. Despite the impressive results, using such reconstruction for touring would be a turn back to the need volumetric rendering. Also, being reconstructed at as large scale it would lack the detailedness under a close-up walk of a virtual tourist. [1]

3.7 DeepStereo

A research work from 2016 on novel view synthesis for Street View dataset. Used a learning-based approach, that was still a novelty at that time. Results were comparable to SoTA and qualitatively satisfying, but the involved rendering process was causing huge latency even at inference time. This approach could be used to record a sequence for tour guide to voice, it wouldn't allow an interactive walk through.

3.8 Fast View Synthesis with DeepStereo

A much faster solution released in 2019 by Tewodros et al [9] built on base of the previous method. Reduced inference latency from minutes to seconds, and so far is one of the best candidates. yet not tested on different dataset. However, with more recent approaches in competition can be outperformed.

3.9 NeRF-W

[20] A recent approach, extending NeRF [19] working using a radiance function abstraction to represent all density and color of a scene perceived from a particular viewpoint under particular direction. Utilising large unstructured datasets of sightseeing spots, this approach is able to reconstruct a scene with impressive accuracy despite inconsistent visibility conditions and distinguish sample-dependent features from core scene structure. However, it is only applicable to recreation of popular

spots because of the amount of data needed to make multi-layer perceptron represent this a scene.

Chapter 4

Choosing NVS approaches for our problem

4.1 Specifics of our problem

4.1.1 Challenges of outdoor NVS

Synthesising outdoor views is quite a challenging task due to variety of our world. A photograph of some street or landscape would naturally contain more inter-composed objects from different domains, with elaborate shapes (e.g. thin streetlights, non-uniformly shaped flora, decorations, and even passing-by people) and different depth, contributing to higher occlusion rates. It would also include more complex textures, and non-Lambertian surfaces causing non-trivial illumination effects (like reflections in the water, soft reflection in a shop window, partial transparency of a glass building, and scattered light rays under tree crowns). On top of that, outdoor scenes undergo constant change of visibility conditions, - from day to night, from sunny weather to fog or rain, etc. Therefore, even in carefully constructed datasets a scene often has inconsistent appearance across many views.

To deal with both impaired visibility and lighting inconsistency we could pre-process data with some image restoration module to return the photographs to their basic form. A huge success in such task was achieved by restoration neural networks [2]. However, most of them are strictly specialized for canceling out some particular effect when trained (e.g. morphology-based de-rain [22], GAN-based de-rain [32], defog [18]), and would pose a bigger problem to be combined. A more elegant solution was proposed in NeRF-W [20], where transient (sample-dependent visuals, caused by variable conditions on scene) and static (structural information) features are extracted by two different networks, and only static component is used to produce a scene coherent for different viewing angles.

4.1.2 StreetView Dataset

For our experiments we collected Street View data for several parts of Paris city, totalling 450K photographs for 70K locations. All data was obtained through Google Static Street View API. For experiments on the approach with single-image input, there's also a publicly available "PitOrlManh" dataset containing similar data for three US cities. We didn't use this one on other approaches because of its greater camera rotation per location, which complicates the fetching of related views.

Street View imagery is available as photos of panorama, sliced by rotation angle and focal width that can be set in the request. Panoramas can be identified by hash names or by coordinates of the photo-locations. To get all panoramas available in some area, we first determine all photo-locations in this area using geopy library, then gather metadata like whether it is an indoor or outdoor photo, or the hash names that are later used as a more convenient identificator. Then, we complete direct requests to the Static Street View API to fetch photographs in six directions from each panorama.

Baseline

In a dataset of photographed route sequences, a baseline refers to a distance between the closest two locations, in other words a step of between two captured locations. The smaller baseline, the denser scene sampling would be, and the better setting for interpolation. In KITTI dataset [21], the smallest constant baseline value (which can be artificially increased) is 0.4m, while for Street View the it is on the order of 1-3m in Paris (for more the 50% of all samples) but is more variable.

4.1.3 Solution requirements

Since we are trying to recreate pretty real places, that can be used for navigational or educational purposes apart from the main entertainment intention, our primary concern should be the accuracy of synthesised scenes. In future work it could be interesting to also generate imaginary environments based on learned scene patterns across the world and conditioned with location, illumination and other potential features to see what does the network associate different world areas with. However, this would be task on intersection of NVS and image geolocalization domains, and would need much more research.

At the same time, for such visually-oriented activity as virtual tourism, the aspects of image quality and its aesthetic appeal are equally important as the accuracy of the scenery. In case of synthesis problem, image quality means not only high resolution and sharpness, but also preferably the lack or minimization of visual artifacts like ghosting of transient occludors, cracks in reconstructed geometry and local blurriness that are typical for generative solutions.

We've already seen in the Background section, there's no golden solution that could simultaneously provide high reconstruction accuracy in geometry and the high level of detail in texture. Therefore, we can assume a **perceptual accuracy** could be sufficient. This means we will still keep high requirements for structural accuracy, but can compromise it slightly if a combination with better detailedness is available.

Another requirement following from navigational nature of the recreated world is that text should be preferably valid. The reason behind this is that users might find information like street names or shop signboards important for their future routes and general world orientation. If followed, it would weight against GANs that have a common issue of not being able to generate coherent text imagery. This requirement is, however, secondary, because navigational information can be embeded into the interface, and despite local writings being a representation of the culture at the virtually visited destination, it is rather complementary, and should probably be addressed after the main goal is achieved.

So far we've discussed performance requirements. But for an interactive system we also have to comply with a reasonable maximum margin for **latency** of the inference. We've seen some well-performing works take minutes to generate a new frame, and therefore be useless for our application despite high benchmark scores.

To sum up, we need a high perceptual accuracy requirement for scenes synthesis, meaning highest possible accuracy with possible slight compromise in favor of better image quality; and an interactive inference rate for the system as whole, meaning minimal latency of a scene generation.

4.2 Approaches chosen for experiments

In order to meet our solution requirement for accuracy, we can't use "hallucinating" generative algorithms as they are optimized to produce realistic but genuinely incorrect results. It also limits us to interpolation, because any synthesis in directions beyond the known views would need predictions based on statistical priors instead of specific objects or even possible underlying geometry.

The following approaches were chosen based on requirements described in the previous subsection and their stated performance on related datasets (e.g. KITTI).

4.2.1 Base model - Fast View Synthesis with DeepStereo

[10] Based on Deep Stereo research, which has a documented performance on Street View dataset, this approach claims to keep a comparable accuracy with fixed latency issue.

4.2.2 Self-guided Novel View Synthesis via Elastic Displacement Network

[27] Proposes an optimized structure for scene representation called Layer Displacement Maps to model geometric transformation in a computationally-inexpensive way and simplify the learning task for the network.

4.2.3 Monocular Neural Image-based Rendering with Continuous View Control

[4] Specifically optimized for continuous interpolation to simulate smaller movement steps. Combines direct pixel prediction performed by deep network with reasoning based on the underlying 3D geometry.

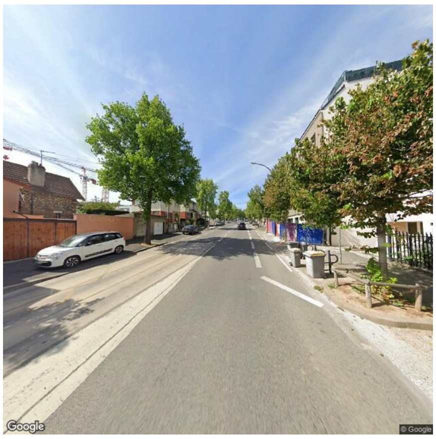
4.2.4 SynSin: End-to-end View Synthesis from a Single Image

[33] Knowing that the baseline value (step-size between photo-locations) in Street View dataset is inconsistent, we would like to also include for comparison one approach that works with single-image input. In case of close accuracy report, a less resourceful method would obviously be preffered. This particular work was originally trained and optimized for indoor task, but showed unexpectedly good generalisation scores when tested on other setups, including outdoor scenery.

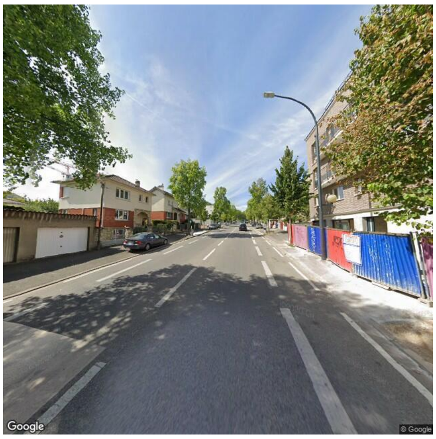
Chapter 5

Experiments

5.1 Data preparation - Multi-view grouping and pairing



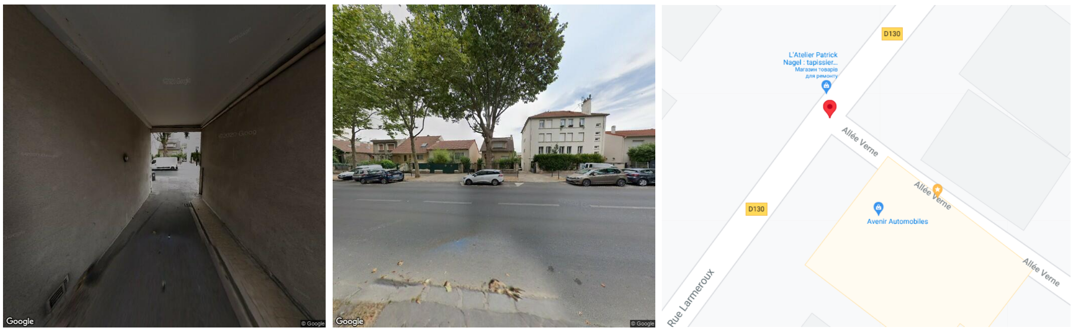
(A) source view



(B) target view

The problem from which we started this work is that photo locations in Street View are not dense enough to provide a seamless transition between them. As we can see on Figure X, it is also not enough to obtain multiple views of the same part of scene. This is why most of our experiments are based on single-view models. Still, approaches that work with single-image inputs require two views at the training time - a source view from which the camera movement and prediction is happening, and a target view, holding the destination. To obtain such photo pairs, we find the the closest location with regards to the great-circle distance between, computed by Haversine formula, and then choose an angle such that both views would be facing forwards (i.e. centered on the road and consecutive). Thus having maximum common context. The photographs per location are indexed with their original panorama identifier and their azimuth shifting by 60 degrees relative to the North. To determine which azimuth we need to get for two given locations x, y, we compute the arc tangent between $y_2 - y_1$ and $x_2 - x_1$, and reason from specific cases.

To reduce complexity of finding the closest location we split the map into an imaginary grid, so that every location would belong to an indexable square area (and would likely share it with the closest locations already, depending on the set precision). If the amount of target locations inside the area is not sufficient, we expand the attention area concentrically. However, there are cases when the closest location is actually on the neighbouring street our a nearby underground road. This is where the choice of Paris streets brings advantages, because the data on street junctions is available in Google's 2014 Hash Code dataset. So we can determine if a suggested target location belongs to the same street as the source one or at most to a junction in order to avoid misleading data samples. Some inconsistencies, however, could not be easily detected, so will still be present in the dataset. For example, when a reasonable closest point is occluded by signs, sidewalk borders, tunnels, greenery or by a privacy blur. See the following figures for example.



(A) source view (B) target view (C) not suspicious locations

FIGURE 5.2: E.g.: undetected occlusion when going through a tunnel

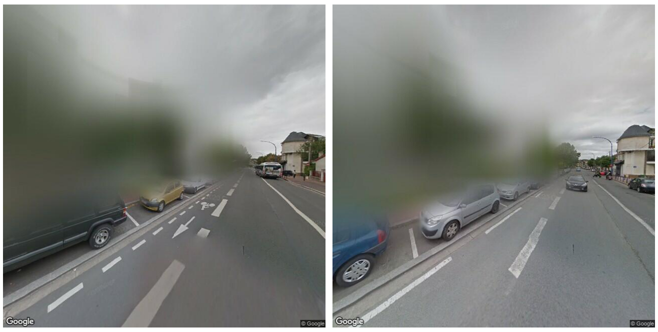


FIGURE 5.3: E.g.: Privacy blur

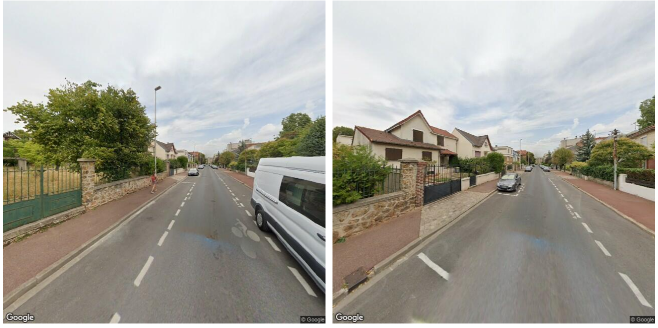


FIGURE 5.4: E.g.: Tree completely occluding a house on the lef

5.2 Metrics

Qualitative assessment of novel view synthesis is hard to perform without human supervision. Moreover, the actual requirement for a synthesised view is to be realistic, and indistinguishable by users from real views. The discriminator network sounds like a good match for this goal. To also analyse the coherence of synthesised images, we would compare Inception Score (IS) of initial dataset and produced images. Inception Score evaluation uses a pre-trained Inception network (e.g. on ImageNet dataset) to classify output imagery and then analyses the distribution of classification labels. For a well-performing generation network, this analysis should yield a higher classification confidence and a higher label variety. [25], [11]

For **Quantitative** evaluation we use combinations of the following metrics

- **L1** - posed as difference between the predicted and the ground-truth values calculated per pixel; it is less prone to over-penalizing big errors caused by outsider samples, than the more common L2.

- **MS-SSIM**. SSIM (Structural SIMilarity index) is a metric specifically introduced to evaluate perceptual quality like a human vision system would. This is achieved by looking at the neighbouring pixels' distribution parameters when computing a per-pixel value. So that the formula per pixel p comparison between images x and y is defined as:
$SSIM(p) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \cdot \frac{2\sigma_x y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$
Where μ, σ refer to mean and variance of a considered image patch around p, and C_1, C_2 are constants that ensure stability for cases when a denominator approaches 0. The two terms of a formula represent luminance and contrast comparisons. MS-SSIM is an improved version of basic SSIM, that considers perceptual accuracy at different scales, as it has been proved to be a factor in human perception. Due to its nature of using the surrounding patches of image, MS-SSIM performs worse on the edge regions of an image, and therefore needs to be coupled with another metric for objective evaluation. [37]

- **LPIPS** - evaluates the distance between feature embedings of two images, obtained with VGG classification network trained on ImageNet. [36] This allows to find the semantical proximity, and would be useful for encouraging scene coherence.

- **PSNR** (Peak signal-to-noise ratio) - uses a relation between the maximum possible signal and the present noise. For two images x, y o size w, l a PSNR value would be computed as:
$PSNR(x, y) = 10 \cdot log_{10}(255^2 / (\frac{1}{w \cdot l} \cdot \sum_i \sum_j (x_{ij} - y_{ij})^2))$ [13]
Despite being a less robust metric for structural comparison on its own (since its MSE term could produce the same evaluation for several different visual errors), PSNR provides a valuable measure noise presence, and would be useful to include for image clarity concerns.

5.3 Main experiment setup

Our main experiment is to test different approaches on Street View data and learn possible directions or problems that need to be addressed for its possible application.

All models are trained in using Google Colaboratory, on high-RAM GPU accelerator. If an approach doesn't have an implementation publicly available, we implement it using a PyTorch-based framework Catalyst for code re-usability.

To get the baseline sooner, we started with the **Monocular Neural Image-based Rendering with Continuous ViewControl** (will be referred to as MNCV) approach, which had a checkpoint of a model pretrained on KITTI dataset available. Since KITTI is also a collection of street imagery, it was a good fit. At the same time

StreetView's examples require larger viewpoint changes, so the task was still challenging. We first compare the how good does their model generalise to a slightly different domain of data, and get the initial evaluation purely out of their checkpoint. Then we fine-tune it on 200K of Street View data samples, and compare the performance with previous state.

	L1	SSIM
MNCV (pretrained on KITTI)	0.453373	0.5229340
MNCV (fine-tuned on StreetView)	0.344686	0.5370018

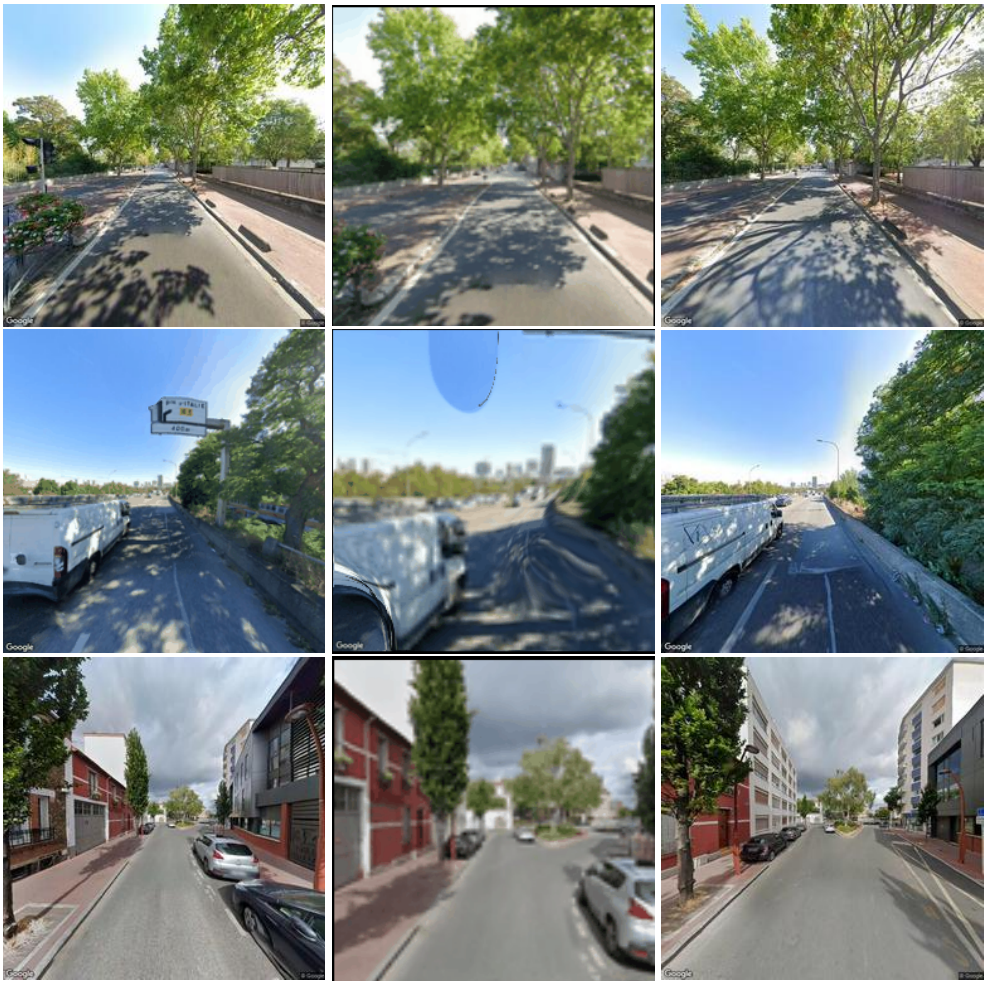


FIGURE 5.5: Performance of MNCV pretrained on KITTI.
(1) source view, (2) predicted target view, (3) ground truth target view

It appears model's performance have improved through fine-tuning, since it got more exposure to data with larger change between the views. The typical appearance of a StreetView sample might be also different from the dataset used for pretraining e.g. due to being focused on a different locale.

Due to the tendency of generative/semi-generative models training from scratch to take much longer, and due to a late start of working on the experimental part, the rest of the experiments will become available until the final presentation day.

5.4 Additional experiments

Street data contains a lot of transient occlusions, which impair the scene construction. After having initial results for the chosen approaches, an extra experiment would be to use preprocessed data with removed occludors and certain types of objects (e.g. cars, pedestrians).

As described in [37], the choice of metrics in the cost function can influence perceptual quality of restored imagery. As another experiment, we would like to test different metric combinations to achieve better image quality.

Bibliography

[1] Sameer Agarwal et al. "Building Rome in a Day". In: *Commun. ACM* 54.10 (Oct. 2011), 105–112. ISSN: 0001-0782. DOI: 10.1145/2001269.2001293. URL: https://doi.org/10.1145/2001269.2001293.

[2] Codruta Orniana Ancuti, Cosmin Ancuti, and Philippe Bekaert. "Single Image Restoration of Outdoor Scenes". In: *Computer Analysis of Images and Patterns*. Ed. by Pedro Real et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 245–252. ISBN: 978-3-642-23678-5.

[3] Shenchang Eric Chen and Lance Williams. "View Interpolation for Image Synthesis". In: *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '93. Anaheim, CA: Association for Computing Machinery, 1993, 279–288. ISBN: 0897916018. URL: https://doi.org/10.1145/166117.166153.

[4] Xu Chen, Jie Song, and Otmar Hilliges. "NVS Machines: Learning Novel View Synthesis with Fine-grained View Control". In: *CoRR* abs/1901.01880 (2019). arXiv: 1901.01880. URL: http://arxiv.org/abs/1901.01880.

[5] Antonio Criminisi et al. "Efficient Dense Stereo with Occlusions for New View-Synthesis by Four-State Dynamic Programming". In: *International Journal of Computer Vision* 71 (Jan. 2007), pp. 89–110. DOI: 10.1007/s11263-006-8525-1.

[6] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. "Modeling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-Based Approach". In: *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '96. New York, NY, USA: Association for Computing Machinery, 1996, 11–20. ISBN: 0897917464. DOI: 10.1145/237170.237191. URL: https://doi.org/10.1145/237170.237191.

[7] John Flynn et al. "Deep Stereo: Learning to Predict New Views from the World's Imagery". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 5515–5524.

[8] Michael Goesele et al. "Ambient Point Clouds for View Interpolation". In: *ACM SIGGRAPH 2010 Papers*. SIGGRAPH '10. Los Angeles, California: Association for Computing Machinery, 2010. ISBN: 9781450302104. DOI: 10.1145/1833349.1778832. URL: https://doi.org/10.1145/1833349.1778832.

[9] Tewodros Habtegebrial et al. "Fast View Synthesis with Deep Stereo Vision". In: *CoRR* abs/1804.09690 (2018). arXiv: 1804.09690. URL: http://arxiv.org/abs/1804.09690.

[10] Tewodros Habtegebrial et al. "Fast View Synthesis with Deep Stereo Vision". In: *CoRR* abs/1804.09690 (2018). arXiv: 1804.09690. URL: http://arxiv.org/abs/1804.09690.

[11] Martin Heusel et al. "GANs Trained by a Two Time-Scale Update Rule Converge to a Nash Equilibrium". In: *CoRR* abs/1706.08500 (2017). arXiv: 1706.08500. URL: http://arxiv.org/abs/1706.08500.

[12] Noriaki Hirose et al. "GONet++: Traversability Estimation via Dynamic Scene View Synthesis". In: *CoRR* abs/1806.08864 (2018). arXiv: 1806.08864. URL: http://arxiv.org/abs/1806.08864.

[13] Alain Horé and Djemel Ziou. "Image Quality Metrics: PSNR vs. SSIM". In: *2010 20th International Conference on Pattern Recognition*. 2010, pp. 2366–2369. DOI: 10.1109/ICPR.2010.579.

[14] Natasha Kholgade et al. "3D Object Manipulation in a Single Photograph using Stock 3D Models". In: *ACM Transactions on Computer Graphics* 33.4 (2014).

[15] Jangwon Lee and Michael Ryoo. "Learning Robot Activities from First-Person Human Videos Using Convolutional Future Regression". In: (Mar. 2017).

[16] Shuai Li, Ce Zhu, and Ming-Ting Sun. "Hole Filling With Multiple Reference Views in DIBR View Synthesis". In: *IEEE Transactions on Multimedia* 20.8 (2018), pp. 1948–1959. DOI: 10.1109/TMM.2018.2791810.

[17] Feng Liu et al. "Content-Preserving Warps for 3D Video Stabilization". In: *ACM Trans. Graph.* 28.3 (July 2009). URL: https://doi.org/10.1145/1531326.1531350.

[18] Wei Liu et al. *End-to-End Single Image Fog Removal using Enhanced Cycle Consistent Adversarial Networks*. 2019. arXiv: 1902.01374 [cs.CV].

[19] Ricardo Martin-Brualla et al. "NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections". In: *CoRR* abs/2008.02268 (2020). arXiv: 2008.02268. URL: https://arxiv.org/abs/2008.02268.

[20] Ricardo Martin-Brualla et al. "NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections". In: *CVPR*. 2021.

[21] Moritz Menze and Andreas Geiger. "Object Scene Flow for Autonomous Vehicles". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[22] Ranjan Mondal et al. "Morphological Networks for Image De-raining". In: *CoRR* abs/1901.02411 (2019). arXiv: 1901.02411. URL: http://arxiv.org/abs/1901.02411.

[23] Tomaso Poggio and Roberto Brunelli. "A Novel Approach to Graphics". In: (June 1993).

[24] Konstantinos Rematas et al. "Novel Views of Objects from a Single Image". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.8 (2017), pp. 1576–1590. DOI: 10.1109/TPAMI.2016.2601093.

[25] Tim Salimans et al. "Improved Techniques for Training GANs". In: *CoRR* abs/1606.03498 (2016). arXiv: 1606.03498. URL: http://arxiv.org/abs/1606.03498.

[26] Yujiao Shi, Hongdong Li, and Xin Yu. "Self-Supervised Visibility Learning for Novel View Synthesis". In: *CoRR* abs/2103.15407 (2021). arXiv: 2103.15407. URL: https://arxiv.org/abs/2103.15407.

[27] Yujiao Shi, Hongdong Li, and Xin Yu. "Self-Supervised Visibility Learning for Novel View Synthesis". In: *CoRR* abs/2103.15407 (2021). arXiv: 2103.15407. URL: https://arxiv.org/abs/2103.15407.

[28] Shao-Hua Sun et al. "Multi-view to Novel View: Synthesizing Novel Views with Self-Learned Confidence". In: *European Conference on Computer Vision*. 2018.

[29] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[30] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. *Multi-view 3D Models from Single Images with a Convolutional Network*. 2016. arXiv: 1511.06702 [cs.CV].

[31] Frederik Warburg et al. "Mapillary Street-Level Sequences: A Dataset for Lifelong Place Recognition". In: *Computer Vision and Pattern Recognition (CVPR)*. 2020.

[32] Yanyan Wei et al. *DerainCycleGAN: Rain Attentive CycleGAN for Single Image Deraining and Rainmaking*. 2021. arXiv: 1912.07015 [cs.CV].

[33] Olivia Wiles et al. "SynSin: End-to-end View Synthesis from a Single Image". In: *CoRR* abs/1912.08804 (2019). arXiv: 1912.08804. URL: http://arxiv.org/abs/1912.08804.

[34] Jimei Yang et al. *Weakly-supervised Disentangling with Recurrent Transformations for 3D View Synthesis*. 2016. arXiv: 1601.00706 [cs.LG].

[35] Jae Shin Yoon et al. "Novel View Synthesis of Dynamic Scenes With Globally Coherent Depths From a Monocular Camera". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[36] Richard Zhang et al. "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric". In: *CoRR* abs/1801.03924 (2018). arXiv: 1801.03924. URL: http://arxiv.org/abs/1801.03924.

[37] Hang Zhao et al. "Loss Functions for Image Restoration With Neural Networks". In: *IEEE Transactions on Computational Imaging* 3.1 (2017), pp. 47–57. DOI: 10.1109/TCI.2016.2644865.

[38] Tinghui Zhou et al. "View Synthesis by Appearance Flow". In: *CoRR* abs/1605.03557 (2016). arXiv: 1605.03557. URL: http://arxiv.org/abs/1605.03557.

[39] Zihan Zhou, Hailin Jin, and Yi Ma. "Plane-Based Content Preserving Warps for Video Stabilization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013.