Sat2Vid: Street-view Panoramic Video Synthesis from a Single Satellite Image

Zuoyue Li[1] Zhenqiang Li[2] Zhaopeng Cui[3*] Rongjun Qin[4] Marc Pollefeys[1,5] Martin R. Oswald[1*]
[1]ETH Zürich [2]The University of Tokyo [3]Zhejiang University [4]The Ohio State University [5]Microsoft

Abstract

We present a novel method for synthesizing both temporally and geometrically consistent street-view panoramic video from a single satellite image and camera trajectory. Existing cross-view synthesis approaches focus on images, while video synthesis in such a case has not yet received enough attention. For geometrical and temporal consistency, our approach explicitly creates a 3D point cloud representation of the scene and maintains dense 3D-2D correspondences across frames that reflect the geometric scene configuration inferred from the satellite view. As for synthesis in the 3D space, we implement a cascaded network architecture with two hourglass modules to generate point-wise coarse and fine features from semantics and per-class latent vectors, followed by projection to frames and an upsampling module to obtain the final realistic video. By leveraging computed correspondences, the produced street-view video frames adhere to the 3D geometric scene structure and maintain temporal consistency. Qualitative and quantitative experiments demonstrate superior results compared to other state-of-the-art synthesis approaches that either lack temporal consistency or realistic appearance. To the best of our knowledge, our work is the first one to synthesize cross-view images to videos..

1. Introduction

Street-view images have been proven to be helpful for exploring remote places or for strategic ground planning in emergency or intelligence operations. They are useful for a variety of applications in virtual or mixed reality, realistic simulations and gaming, viewpoint interpolation, or cross-view matching. Nevertheless, their acquisition is rather expensive, and regular updates to capture changes are required for some tasks. On the other hand, satellite images are regularly captured, easier to obtain, have significantly better earth coverage, and are generally much more widely available than street-view images. The generation of street views from given satellite or aerial images is thus an attractive and interesting alternative for the aforementioned applications.

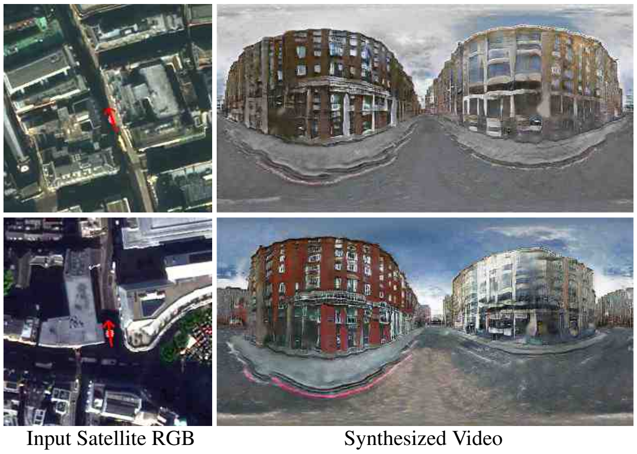

Figure 1: **Street-view panoramic video synthesis results of our method (*animations*).** For a single satellite image and a given trajectory (indicated by ↑ in the figure), we learn to synthesize a corresponding street-view panoramic video with both geometrical and temporal consistency.

While single street-view image generation from satellite images has recently been investigated [27, 20], these methods are not suitable to create continuous view-point changes around a given location since they built upon random generators and lack constraints on the correspondence between frame pixels. They are thus unable to synthesize temporally and geometrically consistent image sequences that are desired for a better visual experience.

In this paper, we approach the novel task to synthesize street-view panoramic video sequences as realistically as possible and as consistent as possible from a single satellite image and given viewing locations. To achieve this, instead of resorting to 2D generators like [27, 20] and generating images individually, we propose to generate the whole scene in a 3D representation of point cloud, and establish the correspondence between these visible points and the 2D frame pixels. In this way, the projected views from the entire generated scene instance will be naturally consistent by design. In order to generate image frames as good as a single image, we design a two-stage 3D generator in a coarse-to-fine manner that exploits the characteristics of different

[*] Corresponding author.

Please use **Adobe Reader / KDE Okular** to view ***animations***, which are also available with higher resolution in our **supplementary materials** and **arXiv** (`https://arxiv.org/abs/2012.06628`).

3D convolutional neural networks. Fig. 1 presents two examples of our synthesized results, which well demonstrate the temporal consistency of our generated video.

Our major **contributions** can be summarized as follows. **(1)** We present the first work for satellite-to-ground video synthesis from a single satellite image with a trajectory. **(2)** We propose a novel cross-view video synthesis method that ensures both spatial and temporal consistency by explicitly modeling a cross-frame correspondence using a 3D point cloud representation and building projective geometry constraints into our network architecture. **(3)** Our method outperforms multiple baseline methods both qualitatively and quantitatively on a newly-constructed dataset for cross-view video synthesis that is expanded from the London panorama dataset [20]. The source code and pre-trained models will be made publicly available upon publication.

2. Related Work

Cross-view synthesis focuses on synthesizing from a completely different view of the given image. Most existing works in this field are targeted at single image synthesis. A very typical application is to generate the street view from a given satellite image. Zhai *et al*. [46] proposed to learn to map the semantic segmentation from the aerial to the ground perspective, which can be further used to synthesize ground-level views based on GANs [8]. Regmi *et al*. [27, 28] proposed to use conditional GANs to learn the aerial or ground view images together with semantic segmentation. In order to keep the geometrical consistency, Lu *et al*. [20] proposes a differentiable geo-transformation layer that turns a semantically labeled satellite depth image to corresponding street-view depth and semantics for further street-view panorama generation. Turning to the field of cross-view video synthesis, there is no much work involving in yet as the problem becomes even harder. Although the video can be synthesized frame-by-frame by the image synthesis method, its temporal consistency is hard to be guaranteed, which is important for a video.

Video synthesis is a field that attracted more attentions in the community and have various forms according to the given input, which can be roughly divided into the following three categories. **(1)** Unconditional video synthesis [18, 31, 39, 40] generates video clips from given input random variables by extending the current GAN frameworks on (spatial) images further into the temporal dimension. **(2)** Future video prediction [7, 10, 16, 17, 19, 22, 25, 41, 42] aims at inferring the future frames of a video based on the current observations so far. **(3)** Video-to-video synthesis [2, 4, 21, 43, 44] is closer to our task, which maps a video from a source domain to a target domain (*e.g.*, generating RGB images from a sequence of semantic segmentation masks or depth images). Compared to the image-to-image translation task, it emphasizes the coherency of the generated video frames over time. Wang *et al*. [44] aimed to achieve this by leveraging a generative adversarial learning framework and spatio-temporal adversarial objective. Mallya *et al*. [21] proposed an enhanced method that achieves consistency over a longer time by a guidance image projected from an incrementally colored point cloud during the subsequent frame generation. Nevertheless, the cross-view video synthesis setting in our work is still different from all these categories, which should consider both the temporal consistency between video frames and the geometrical consistency between top and ground views.

Novel view synthesis and neural rendering technologies develop rapidly recently with the advancements in deep neural networks. Many state-of-the-art works focus on the synthesis from a single image. SynSin [45] proposed an end-to-end view synthesis pipeline via a learned point cloud and a differentiable soft z-buffer method, where a point is projected to a region in the image plane with some radius using α-compositing with other projected points (regions). Shih *et al*. [32] regarded the input depth image as a layered structure, and the learning-based inpainting model synthesizes color-and-depth content into the occluded region in a spatial context-aware manner. These works usually assume that the viewpoint changes are small, which makes it nearly impossible to directly employ them. On the other hand, synthesis and rendering with arbitrary viewpoint changes often achieved by multiple images input [36, 38, 35, 23, 24]. Traditional methods usually adopt the image-based rendering technique [33] to generate novel views. Riegler *et al*. [29] employed differentiable reprojection of image features. Sitzmann *et al*. [35] learned a 3D-structured scene representation from only 2D supervision that encodes the view-dependent appearance of a 3D scene. Sitzmann *et al*. [36] further proposed a implicit 3D scene representation which could be also learned from 2D images via a differentiable ray-marching algorithm. Mildenhall *et al*. [24] propose to represent scenes as 5D neural radiance fields which could render photorealistic novel views of complex scenes. Meshryet *et al*. [23] uses additional depth and semantic information of the point cloud, together with an encoded latent vector to achieve realistic rendering with different styles. Recent surveys on neural rendering can be found in [37, 14]. All these methods require a set of images or the built point cloud as input in order to learn detailed 3D scene representation with the deep network. Since our input is only a single satellite image, it is even more difficult for the network to learn meaningful representation.

3. Method

We introduce a novel framework for synthesizing street-view panoramic video from a single satellite image and provide an overview of our proposed pipeline in Fig. 2. As shown in the figure, we use a cascaded network architecture

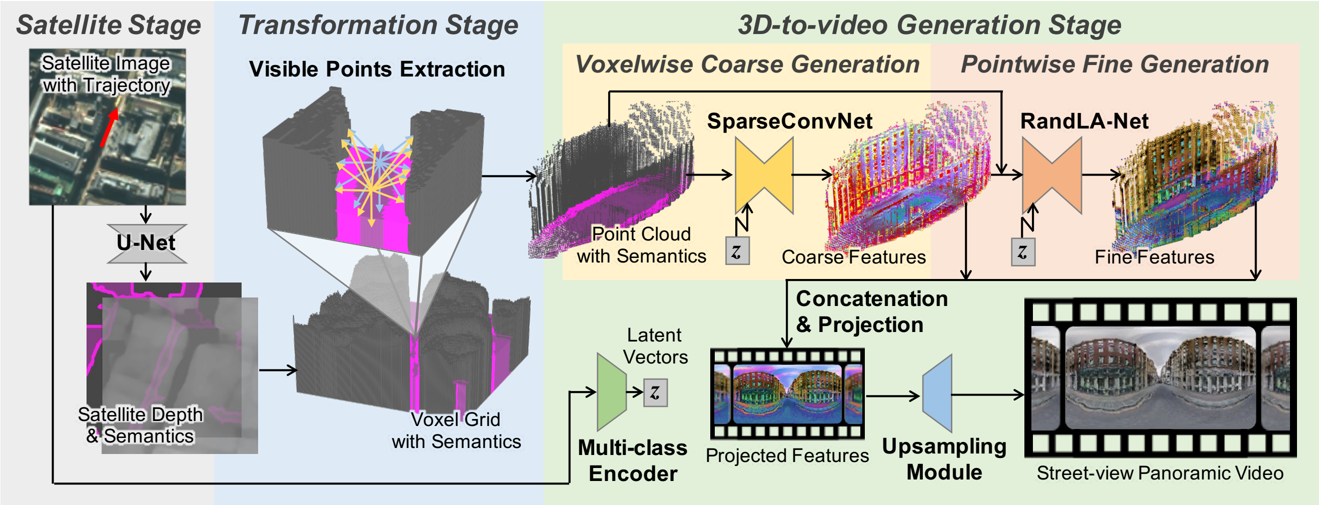
Satellite Stage
Transformation Stage
3D-to-video Generation Stage
Satellite Image with Trajectory
U-Net
Satellite Depth & Semantics
Visible Points Extraction
Voxel Grid with Semantics
Voxelwise Coarse Generation
Pointwise Fine Generation
SparseConvNet
RandLA-Net
Point Cloud with Semantics
Coarse Features
Fine Features
z
Concatenation & Projection
Latent Vectors
Multi-class Encoder
Projected Features
Upsampling Module
Street-view Panoramic Video

Figure 2: **Overview of our network architecture**. Our network consists of multiple sub-networks accounting for three processing stages which transform between different scene representations. These stages are: **Satellite Stage:** The input satellite image is processed by a 2D U-Net [30] to generate a 2.5D height map with corresponding semantics. **Transformation Stage:** To obtain a 3D representation, the semantic height map is converted into a semantic voxel occupancy grid. Visible points are then extracted according to the sampling points of the input trajectory. **3D-to-video Generation Stage:** A generator operating in the 3D domain infers features for each point from the semantics. The cascaded SparseConvNet [9] and RandLA-Net [11] both with hourglass structures act on coarse and fine generation successively. Rather than using a single seed as in [20], we use a multi-class texture encoder that computes multiple latent vectors from the input satellite image. Lastly, the point cloud with concatenated features is projected to each frame, which is finally upsampled with a light-weight network to double the resolution. **Note: (1)** The 3D-to-video generation stage is trained under the framework of BicycleGAN [47]; **(2)** Sky points are included in the pipeline but not visualized here; **(3)** Features are illustrated with pseudo colors.

with three stages: a satellite stage, a transformation stage, and a 3D-to-video generation stage. The satellite stage is similar to the current state-of-the-art method S2G [20] and estimates both depth map and semantics from an input satellite image. Different from the geo-transformation layer used in S2G [20] which transforms the satellite domain to the street view, we directly extract visible points from the constructed occupancy grid according to the given input trajectory. In the last 3D-to-video generation stage, two cascaded networks are utilized to generate a feature point cloud from semantics, followed by a projection to each video frame and a light-weight upsampling module. The second and third stages are detailed in the following subsections.

3.1. Visible Points Extraction

We first build a semantic voxel occupancy grid using the depth and semantic images from the satellite stage. Together with the sampling locations in the input trajectory, we create a point cloud with only visible points and build 3D-2D correspondences. This corresponds to finding the index of the point in the 3D space for each pixel in the video. Each pixel has a uniquely corresponding 3D point, and each point in the 3D space may correspond to multiple pixels. The same mapping will also be utilized for projecting the colored point cloud onto the video frames in the final step of the 3D-to-video generation stage.

Alg. 1 describes the detailed procedure for extracting visible points and building 3D-2D correspondences. The algorithm takes as input the voxelized occupancy grid V and ordered sampling locations $L \in \mathbb{R}^{T\times 2}$. Here, T denotes the number of sampling locations, which is equal to the number of video frames. The final outputs consist of an ordered set P_T saving the 3D coordinates (x, y, z) of all visible points and a mapping tensor $M \in \mathbb{R}^{T\times H\times W}$ for all 2D frame pixels. Each element M_{tpq} keeps an index value i if the frame pixel in position (t, p, q) corresponds to the i-th visible point in P_T. The ordered set of visible points and mapping matrix are iteratively computed. We assign value of 0 to all frame pixels that have no corresponding point in the point cloud P_t in the current iteration.

At each time step t, we first obtain a dense depth map $d \in \mathbb{R}^{H\times W}$ for the frame at location L_t by taking a $\mathrm{z-buffer}$ operation in the occupancy grid V. This processing step is identical to the geo-transformation layer proposed in S2G [20]. Then, a preliminary mapping $m \in \{0, 1, ..., |P_{t-1}|\}^{H\times W}$, which indicates the correspondence between the current frame pixels and the visible points set P_{t-1} so far, is calculated by the project function. $m_{pq} = i$ means that the i-th point in P_{t-1} is projected to the (p, q)-th pixel **and** the depth value is in a range of $d_{pq}(1 \pm \epsilon)$, otherwise $m_{pq} = 0$. In our experments, we set $\epsilon = 0.5\%$. For pixels without corresponding points, *i.e.*, $\{(p, q) \mid m_{pq} =$

$0\}$, we $\mathrm{unproject}$ them to the 3D space to obtain an additional ordered set P_a containing the incremental visible points and an additional mapping m_a saving the correspondences between these pixels and the incremental visible points. It should be noted that the incremental indices satisfy $m_a \in \{0, |P_{t-1}| + 1, ..., |P_{t-1}| + |P_a|\}^{H \times W}$, where a pixel with a corresponding point in P_{t-1} is assigned with 0 and pixels with correspondences in P_a have the indices offset by $|P_{t-1}|$. Hence, $m \odot m_a = \{0\}^{H \times W}$ always holds, where $\odot$ denotes the Hadamard product. Finally, we update the visible point set and mapping tensor by joining P_{t-1} with P_a and saving $m + m_a$ to M_t.

Algorithm 1: Visible Points Extraction

```
Input  : $V$ (occupancy grid), $L$ (locations)
Output: $P_T$ (point cloud), $M$ (point-pixel mapping)
Init    : $P_0 \leftarrow \varnothing$, $M \leftarrow \{0\}^{T \times H \times W}$
for $t \leftarrow 1$ to $T$ do
    $d \leftarrow \mathrm{z-buffer}(V, L_t)$
    $m \leftarrow \mathrm{project}(P_{t-1}, L_t, d)$
    $P_a, m_a \leftarrow \mathrm{unproject}(L_t, d, m, |P_{t-1}|)$
    $P_t \leftarrow P_{t-1} \bigcup P_a$
    $M_t \leftarrow m + m_a$
end
```

Since only the center frame has ground truth street-view RGB and to reflect the projection characteristics of the panorama image, the locations of the sampling points L are inputted to the algorithm in an order of $c, c+1, c-1, c+2, c-2, ...$, where c is the index of the center frame.

3.2. 3D Generator

In the 3D-to-video generation stage, we first infer features for the point cloud in the 3D space from the reprojected semantics. The semantics of the points is gathered from the satellite semantics according to each point's coordinates in the horizontal plane. Distant points are simply labeled as *sky*. The proposed 3D generator consists of a SparseConvNet [9] and a RandLA-Net [11], with a cascaded connection. Both networks are operating purely in the 3D domain and have an hourglass structure acting on coarse and fine generation successively. Finally, the points are projected to frames, which are further turned into the output video via a light-weight upsampling module.

The **coarse generation stage** is based on voxels. At the beginning of this stage, the point cloud is first voxelized according to the targeted voxel size. Multiple points sharing the same voxel will be averaged as the feature of that voxel. In our experiments, the voxel size is set to 3.125cm (32 voxels per meter). The SparseConvNet [9] only operates on the occupied area of the voxel grid avoiding unnecessary computations on free space and thus achieving time- and memory-efficient 3D convolutions. Finally, the output of the network is de-voxelized to a point cloud. Again, points sharing the same voxel will be assigned to the same feature. As depicted in Fig. 2, the visualized point cloud with intermediate coarse features already shows some characteristics of the building facade like windows.

The **fine generation stage** is based on the point cloud. The input of this stage is a concatenation of the intermediate coarse features and the original point semantics from the skip connection. RandLA-Net [11] is an efficient and lightweight state-of-the-art architecture designed for semantic segmentation of large-scale point clouds. We leverage this network to infer the fine features for each point. We set the number of nearest neighbors to 8, and the decimation ratio in its local feature aggregation module to 4.

Each pixel in the video frame then gathers both coarse and fine features from its corresponding point in the point cloud according to the point-pixel mapping M computed in the transformation stage. Finally, the **upsampling module** doubles the resolution and turns the frames with rich features into the output RGB video. In order not to break the consistency from the 3D space, the module is designed only with very few parameters.

The reason for using a cascaded architecture of these two networks rather than only using RandLA-Net is that its efficient setting makes the size of the network rather small, but the capacity may not be enough to support a scene generation. With the help of SparseConvNet which learns high-level features, RandLA-Net can better infer fine features from local information. We also conduct experiments on a generator with only RandLA-Net as detailed in Sec. 4.4.

3.3. Multi-class Encoder

S2G [20] follows BicycleGAN [47] to use a single latent vector when generating the whole scene. Instead, we use a multi-class texture encoder that computes several latent vectors per class to enrich the diversity of generated scenes.

The encoder in the BicycleGAN [47] used in our pipeline takes as input the ground truth street-view RGB, as well as the semantics of the center frame during training. The role of the semantics here is an indicator used for attentive pooling. After obtaining the feature map F of the entire image, the encoder does not directly perform average pooling but instead pools the features of pixels with the same semantic class to finally obtain multiple latent vectors. For a specific class c, its corresponding semantic map S^c is used for attentive pooling to finally obtain the latent vector v_c of this class, *i.e.*, $v_c = (\sum_{ij} S^c_{ij} F_{ij}) / (\sum_{ij} S^c_{ij})$, where i, j denote the spatial indices. The encoder for the satellite image is similar to the encoder in the BicycleGAN. During training, the goal is to make the generated latent vectors as similar as possible to what is generated by the encoder in the BicycleGAN. Since some of the classes, *e.g. sky*, and *sidewalk*, may not be able to infer from the satellite image, there is no loss on the latent vectors for these classes during training and they are directly given random vectors during inference.

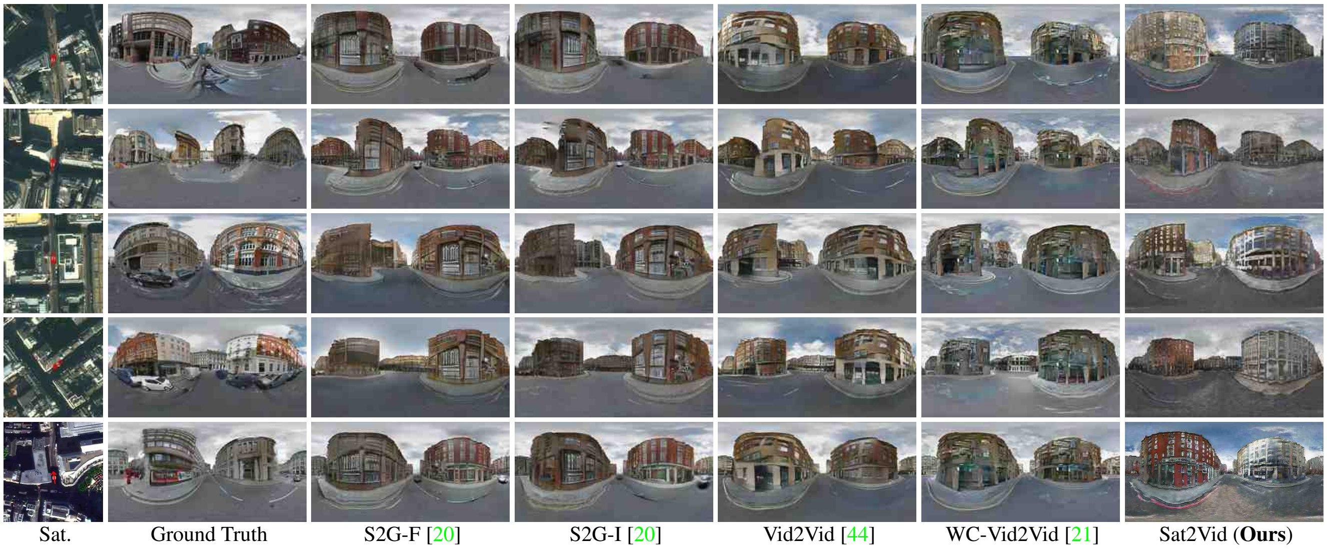

Figure 3: **Qualitative baseline comparison (*animations*)**. We show comparisons to state of the arts on a variety of examples. Ours generates more realistic videos with better temporal consistency and contains fewer artifacts.

4. Experiments

4.1. Ground Truth

To the best of our knowledge, there is currently no available dataset that provides both satellite images and corresponding street-view panorama videos. As the first work that sheds light on the task of street-view video synthesis from a single satellite image, we first produce a dataset that satisfies the requirements for the task. Specifically, we extend the London panorama dataset used in S2G [20] by generating the ground truth of street-view video snippets. The original dataset includes around 2K pairs of satellite images and corresponding street-view panoramas that are captured in the center position of the satellite images. The estimated depths (elevation) and semantics of the satellite image are also provided as ground truth. In brief, we interpolate the ground-truth street-view panorama videos in the 3D space via a point cloud, of which the geometry is calculated by the estimated depth of the available street-view panorama in the center position. We elaborate on the details as follows.

Sampling trajectory. Each single street-view panorama image provided in the London panorama dataset [20] is taken in the center of the satellite image and is associated with orientation. To generate the street-view panorama video surrounding the location of this image, we set the sampling paths in both training and inference in a total range of 7 meters straight ahead and back from the viewing center. Taking the interval step of 0.5 meters, a total number of 15 frames including the center frame are sampled to form a video. We denote the provided single street-view panorama image as the center frame for brevity.

Geometry. To generate panorama frames in novel positions, both the interpolation via a point cloud and simple warping require precise geometry of the scene. However, an accurate geometry is hard to be inferred from the satellite image considering its limited resolution and not accurate enough ground-truth elevation. Therefore, we infer the scene geometry from the available center frame instead of the satellite image. We first generate a dense depth map for the center frame using MiDaS [26], a state-of-the-art method for monocular depth estimation. Although the pretrained model used pinhole images, it still works well for panoramas. We normalize the depth map by ensuring that the height of the viewing center (standing point) is 3 meters. Then we unproject the central frame depth to generate a raw 3D point cloud and obtain the depths for other frames by reprojecting the point cloud into each frame. For the location without a valid projection, we infer its missing depth value by exploiting the OpenCV inpainting function. Through unprojecting each frame into the 3D space according to the depth, a final point cloud can be constructed.

Interpolation via point cloud. Only points unprojected from the center frame possess the exact RGB information. For other points in the point cloud, we complement their colors through the nearest neighbor search. Specifically, for each uncolored point, we search for its 32 nearest-neighbor center-frame points that have valid information and determine its RGB by a distance-based weighted average on these neighbors. Finally, by re-projecting all colored points back to the frames, we get a video of good quality. The generated ground-truth video examples can be seen in Fig. 3.

Semantics. Obtaining street-view semantic videos follows the procedure mentioned above. We first adopt DeepLab v3+ [3] with an Xception 71 [5] backbone and which is pre-trained on the Cityscapes [6] dataset to get the semantics of center frames. Compared to SegNet [1] utilized in

Method	PSNR↑	SSIM↑	Sharp Diff.↑	P_{Alex} ↓	P_{Sqz} ↓	P_{VGG} ↓
Pix2Pix [13]	- / 13.257	- / 0.313	- / 24.673	- / 0.606	- / 0.478	- / 0.629
Regmi *et al*. [27]	- / 13.305	- / 0.320	- / 24.560	- / 0.587	- / 0.443	- / 0.600
S2G-F [20]	14.110 / 14.146	0.347 / 0.346	25.851 / 25.861	0.530 / 0.528	0.422 / 0.422	0.626 / 0.626
S2G-I [20]	14.169 / 14.146	0.365 / 0.346	**26.137** / 25.861	0.520 / 0.528	0.404 / 0.422	0.594 / 0.626
Vid2Vid [44]	13.546 / 13.502	0.391 / 0.390	25.552 / 25.553	0.488 / 0.483	0.363 / 0.361	0.545 / 0.544
WC-Vid2Vid [21]	13.879 / 13.904	0.346 / 0.345	25.400 / 25.410	0.508 / 0.502	0.369 / 0.367	0.556 / 0.554
Sat2Vid (**Ours**)	**15.171 / 15.220**	**0.409 / 0.410**	26.068 / **26.060**	**0.482 / 0.478**	**0.342 / 0.342**	**0.535 / 0.533**

Table 1: **Quantitative baseline comparison**. For each entry we report two numbers indicating the evaluation on all frames and only on the center frame, respectively. Our method outperforms all baselines on most of the metrics.

S2G [20], DeepLab v3+ generates more accurate semantics. The semantics of other frames are again complemented by the nearest neighbors search described above via a voting strategy instead of the weighted average used for RGB.

4.2. Implementation Details

Our framework is implemented in PyTorch and run on a single Nvidia Tesla V100 GPU with 32GB memory. For the dataset, we keep an output resolution of 512×256 such that the point cloud size of each scene is around 200K. Both the voxel size of the occupancy grid and the sampling size in ray marching is 0.5m, which restricts the potential accuracy loss happening in constructing the correspondence mapping from the coarse voxel grid. During training, we use the geometry from the satellite depth to be consistent with the inference stage. For the network architecture, the default training settings of BicycleGAN [47] are employed, using 16 for the size of latent vectors and 64 for the size of intermediate features. The multi-noise encoder only takes as input the center frame. We further distinguish between left and right buildings in the semantic labels to achieve better diversity. For the 3D generator, we use the default provided U-Net [30] implementations under SparseConvNet [9] and RandLA-Net [11] frameworks which are originally used for point cloud semantic segmentation. The training takes ∼5 days from scratch while the inference takes ∼2.8s for generating a 15-frame video with the above-mentioned resolution. The training and validation of the satellite stage follow [20]. The mIoU and the accuracy of the semantic segmentation are 0.755 and 0.865 respectively, while the average relative and absolute errors of height estimation are 4.17% and 2.86m, respectively. More implementation details can be found in the supplementary material.

4.3. Baseline Comparison

Since we are the first to propose a method for generating street-view panoramic videos from single satellite images, we design two baseline methods by adapting state-of-the-art street-view panoramic image synthesis method S2G [20] for video generation: **(1) S2G-F**: each frame is generated individually but shares the same latent vector encoded from input satellite image; **(2) S2G-I**: only center frame is generated and other frames are interpolated by using the point cloud coloring procedure described in Sec. 4.1. Vid2Vid [44] and WC-Vid2Vid [21] are also included in the comparison, which are originally designed for video-to-video translation. We generate additional per-frame semantics and pixel correspondences (only for WC-Vid2Vid) to satisfy their input requirements. The comparison is conducted on the test set of London panorama [20].

For quantitative evaluation, we follow [20] and use PSNR, SSIM, and sharpness difference (Sharp Diff.) as low-level metrics to measure the per-pixel differences between the predicted frames and the ground-truth video. The high-level perceptual similarity is also taken into account. P_{Alex}, P_{Sqz}, P_{VGG} denote the evaluation results based on the backbone of AlexNet [15], SqueezeNet [12] and VGG [34].

In addition to the above two baselines, we compare to two image-to-image translation works, Pix2Pix [13] and Regmi [27], on the center frame generation. The quantitative results are shown in Tab. 1. For the video generation comparison, our improved performance may result from a better temporal consistency of our generated video, since all methods use the same geometry inferred from the input satellite image. Regarding the center frame comparison, we outperform all state-of-the-art methods on all metrics, which indicates superiority of our method in generating geometrically consistent single street-view panorama.

More qualitative results are presented in Fig. 3. We can see that the frames generated by our method are both temporally and geometrically consistent. Since each frame from S2G-F [20] is synthesized independently, the textures in different frames are nearly stationary and there is no consistent transition between them when the observation location changes. Vid2Vid [44] has better per-frame appearance but still suffers from the problem of stationary patterns. This may be due to an inaccurate optical flow estimation within their network. For S2G-I [20], we can see that the interpolation can ensure consistency of texture between frames since every frame's texture comes from center frame and is based on the geometry. Nevertheless, it is easy to find that the texture in frames which are far away from center frame is likely to be blurred, especially on building facades which are invisible in the center frame. WC-Vid2Vid [21] generally have good consistency since pixel correspondences are provided as input. However, their appearances, especially

Method	$\text{MSE}_{\text{RGB}} \downarrow$	PSNR↑	SSIM↑	Sharp Diff.↑	$\text{P}_{\text{Alex}} \downarrow$	$\text{P}_{\text{Sqz}} \downarrow$	$\text{P}_{\text{VGG}} \downarrow$	User Study
Vid2Vid [44]	21.605	21.764	0.774	30.950	0.116	0.077	0.211	9.3%
WC-Vid2Vid [21]	10.604	27.783	0.871	35.296	0.108	0.074	0.176	32.6%
Sat2Vid (**Ours**)	**1.668**	**43.982**	**0.997**	**50.748**	**0.006**	**0.007**	**0.021**	**58.1%**

Table 2: **Quantitative temporal self-consistency**. The evaluation is based on a u-turn-shaped trajectory. Our method outperforms all baselines since they do not handle long-range temporal consistency.

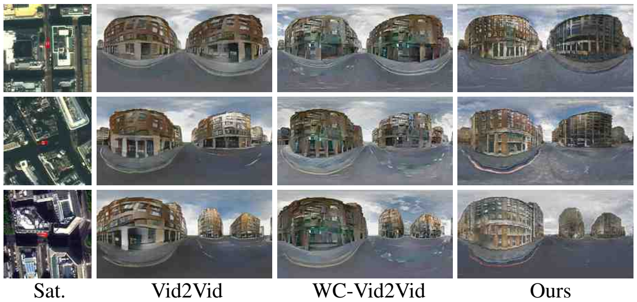

Figure 4: **Qualitative temporal self-consistency (*animations*)**. Videos are synthesized on a u-turn-shaped trajectory.

building facades, look similar across different examples.

Temporal self-consistency. To evaluate the temporal self-consistency of synthesized video frames between different methods, we designed an experiment based on a special u-turn-shaped trajectory, with a total of 60 frames. We then compute the pixel-wise difference between two frames of the same position in two directions. Such an evaluation is devised to assess the frame's temporal self-consistency in one consecutive synthesis. Beside metrics used in Tab. 1, we also compared the MSE value of RGB.

In addition, we conducted a user study, where we provided randomly selected 15 samples (including 10 forward motions and 5 u-turns) with results of Vid2Vid [44], WC-Vid2Vid [21], and ours. We asked 28 people to select only one result of the best naturalness and consistency for each sample, in a total of 420 votes. All evaluations of the temporal self-consistency as well as the voting ratios of user study are detailed in Tab. 2. We also present results of the u-turn trajectory in Fig. 4. Both quantitative and qualitative results indicate our method has significantly better self-consistency across frames than the two strong baseline methods. For more experimental results and comparison to the baselines, please refer to the supplementary material.

4.4. Ablation Study

To better evaluate the effectiveness of the individual components of our method, we also conduct an ablation study by incrementally adding components into our basic framework. More specifically, we focus on the following three components: **(1)** the SparseConvNet [9] used in the 3D generator; **(2)** the setting of multiple latent vectors; **(3)** the final upsampling module. We set the basic framework as the pipeline with only RandLA-Net [11] in the 3D generation stage, while our method possesses all components.

Tab. 3 shows quantitative evaluation results of the ablation study. The abbreviations of the method names in the table are defined as follows. **R**: the basic framework that uses RandLA-Net [11] and a global latent vector in the 3D generation stage; **R+S**: the coarse and fine generation framework by further incorporating SparseConvNet [9]; **R+S+M**: further using a multi-class encoder to the R+S setting. **R+S+M+U**: further adding the upsampling module which forms our final method with all components.

The effectiveness of each added component is shown by clear performance improvements of the PSNR, P_{Alex}, and P_{Sqz} metrics. Fig. 5 further shows a qualitative comparison of the results generated by the aforementioned methods. As illustrated, frames generated by the full framework show higher consistency and smoothness over time compared with the other ablation variants. Especially, the addition of SparseConvNet [9] **(R+S)** significantly improves the generation quality compared to the basic setting **(R)** that only uses RandLA-Net [11], which can only give the overall color and cannot restore the texture details, *e.g.*, building facade. We address the main reason as the explicit allocation of coarse generation and fine generation to two cascaded different networks respectively. This alleviates the struggle of RandLA-Net [11] in generating both coarse and fine textures. With introduction of multi-class encoder that generates multiple latent vectors **(R+S+M)**, the performance is further improved since it disentangles the latent vectors for different classes and enables more generation possibilities. The upsampling module (**R+S+M+U**) further doubles the resolution and makes frames much clearer and realistic.

We also tried directly warping satellite images as parts of the input of upsampling module to better utilize the input information. However, evaluation of this addition (**R+S+M+U+W**) yields lower PSNR and SSIM metrics, as well as perceptual similarity, which indicates that warped satellite images do not directly provide useful information. This is might due to the limited resolution but also due to artifacts like cast shadows which make it difficult to readily extract useful color information. Fig. 6 shows two examples of warped satellite images. In a few cases like the 1st example, roads and lane lines can be warped into street view, although very blurry. In most cases like the 2nd one, roads covered by shadows of neighboring buildings lead to a very dark warping result. This also illustrates the difficulty of cross-view video generation from a single satellite image.

Method	PSNR↑	SSIM↑	Sharp Diff.↑	P_{Alex} ↓	P_{Sqz} ↓	P_{VGG} ↓
R	13.686 / 13.739	**0.417 / 0.417**	25.726 / 25.736	0.584 / 0.580	0.443 / 0.443	0.621 / 0.619
R+S	14.551 / 14.590	0.402 / 0.403	25.493 / 25.479	0.561 / 0.564	0.404 / 0.402	0.572 / 0.568
R+S+M	14.655 / 14.714	0.385 / 0.391	25.811 / 25.823	0.551 / 0.546	0.403 / 0.399	0.576 / 0.572
R+S+M+U (**Ours**)	**15.171 / 15.220**	0.409 / 0.410	26.068 / 26.060	**0.482 / 0.478**	**0.342 / 0.342**	**0.535 / 0.533**
R+S+M+U+W	14.546 / 14.576	0.394 / 0.394	**26.341 / 26.349**	0.503 / 0.500	0.345 / 0.346	0.541 / 0.539

Table 3: **Quantitative ablation study**. For each method we report two numbers indicating evaluation on all frames and only on the center frame, respectively. In short, the ablations are: **R**: basic framework with RandLA-Net [11]; **+S**: adding SparseConvNet [9]; **+M**: multi-noise encoder; **+U**: upsampling module; **+W**: warped satellite information.

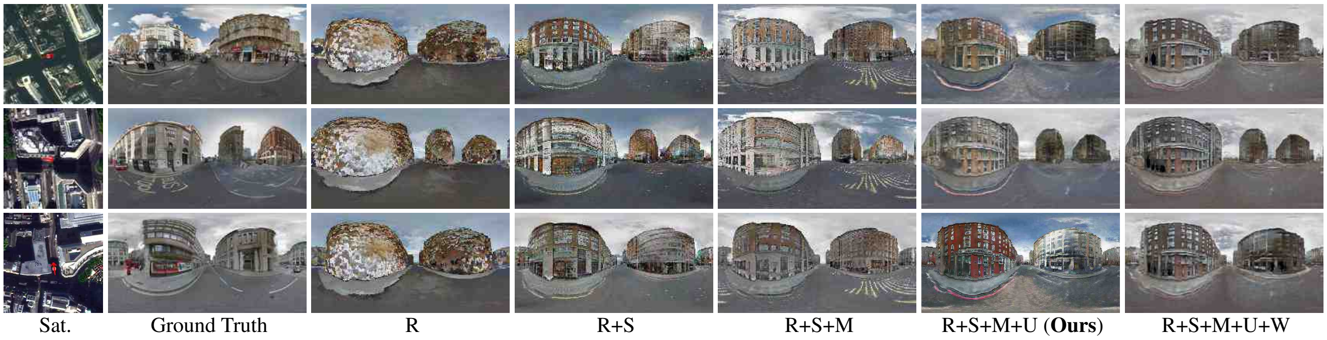

Figure 5: **Qualitative ablation study (*animations*)**. We present exemplary qualitative results for various ablations of our method. The synthesized videos visibly contain more details and achieve higher levels realism with our full method.

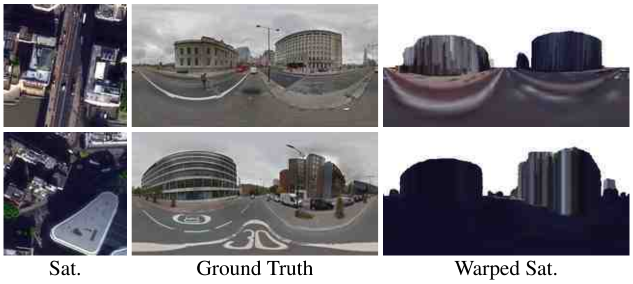

Figure 6: **Warped color satellite information**. The examples illustrate the low-quality of warped satellite images which often do not provide useful color information.

4.5. Limitation and Future Work

Since the proposed method builds on the height estimation and semantic segmentation of the satellite image, the final synthesized video may suffer from some geometric inconsistency for the potential inaccurate estimated height and semantics. Besides, our method also fails to handle buildings with overhanging structures such as overpasses or protruding roofs because they cannot be well represented in the 2.5D height map from the top view. Furthermore, the point cloud size will explode with increasing numbers of video frames, leading to potential memory problems in the 3D convolutions. This currently limits the attainability of super-resolution videos and the application to the longer trajectories navigation or large-scale scenarios. Although we can divide the area into multiple blocks and perform the generation individually, the texture consistency between blocks may not be well guaranteed, which can be further studied in the future. Future work could also incorporate a single street-view panorama as additional input to guide the generation and make the synthesized videos as real as possible. Moreover, the field of cross-view video synthesis could be generalizable for indoor scenes, such as the navigation video generation from a single floor plan layout for virtual house visiting.

5. Conclusion

We proposed a novel approach for cross-view video synthesis. In particular, we presented a multi-stage pipeline that takes as input a single satellite image with a given trajectory, and generates a street-view panoramic video with both geometrical and temporal consistency constrained in a 3D point cloud. Our experiments demonstrate that our method outperforms existing state-of-the-art cross-view generation or video translation approaches and is able to synthesize more realistic street-view panoramic videos in larger variability. We see our work as basic research to build more powerful 3D-aware generative networks. Compared to video translation methods, ours can generate photo-realistic videos without requiring hardly available, aligned per-frame inputs. To the best of our knowledge, we presented the first work that synthesizes videos under cross-view settings.

Acknowledgments. Zuoyue Li was supported by the Swiss Data Science Center Fellowship program. Zhaopeng Cui was affiliated with the State Key Lab of CAD&CG, Zhejiang University. Rongjun Qin was supported by the Office of Naval Research grant No. N000141712928. Martin R. Oswald was supported by Innosuisse grant No. 34475.1IP-ICT and a research grant by FIFA.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017. 5

[2] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5933–5942, 2019. 2

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 5

[4] Yang Chen, Yingwei Pan, Ting Yao, Xinmei Tian, and Tao Mei. Mocycle-gan: Unpaired video-to-video translation. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 647–655, 2019. 2

[5] Francois Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 5

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 5

[7] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *arXiv preprint arXiv:1605.07157*, 2016. 2

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 2

[9] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3, 4, 6, 7, 8

[10] Zekun Hao, Xun Huang, and Serge Belongie. Controllable video generation with sparse trajectories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7854–7863, 2018. 2

[11] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3, 4, 6, 7, 8

[12] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 6

[13] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017. 6

[14] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey, 2020. 2

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 6

[16] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Flow-grounded spatial-temporal video prediction from still images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 600–615, 2018. 2

[17] Xiaodan Liang, Lisa Lee, Wei Dai, and Eric P Xing. Dual motion gan for future-flow embedded video prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1744–1752, 2017. 2

[18] Elizaveta Logacheva, Roman Suvorov, Oleg Khomenko, Anton Mashikhin, and Victor Lempitsky. Deeplandscape: Adversarial modeling of landscape videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, August 2020. 2

[19] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016. 2

[20] Xiaohu Lu, Zuoyue Li, Zhaopeng Cui, Martin R. Oswald, Marc Pollefeys, and Rongjun Qin. Geometry-aware satellite-to-ground image synthesis for urban areas. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 3, 4, 5, 6

[21] Arun Mallya, Ting-Chun Wang, Karan Sapra, and Ming-Yu Liu. World-consistent video-to-video synthesis. In *Eur. Conf. Comput. Vis.*, 2020. 2, 5, 6, 7

[22] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015. 2

[23] Moustafa Meshry, Dan B Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural rerendering in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6878–6887, 2019. 2

[24] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Eur. Conf. Comput. Vis.*, 2020. 2

[25] Junting Pan, Chengyu Wang, Xu Jia, Jing Shao, Lu Sheng, Junjie Yan, and Xiaogang Wang. Video generation from single semantic label map. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2019. 2

[26] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 5

[27] Krishna Regmi and Ali Borji. Cross-view image synthesis using conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3501–3510, 2018. 1, 2, 6

[28] Krishna Regmi and Mubarak Shah. Bridging the domain gap for ground-to-aerial image matching. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2

[29] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *Eur. Conf. Comput. Vis.*, 2020. 2

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 234–241. Springer, 2015. 3, 6

[31] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *Proceedings of the IEEE international conference on computer vision*, pages 2830–2839, 2017. 2

[32] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3d photography using context-aware layered depth inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[33] Harry Shum and Sing Bing Kang. Review of image-based rendering techniques. In *Visual Communications and Image Processing 2000*, volume 4067, pages 2–13. International Society for Optics and Photonics, 2000. 2

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 6

[35] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2437–2446, 2019. 2

[36] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, pages 1121–1132, 2019. 2

[37] A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner, R. Pandey, S. Fanello, G. Wetzstein, J.-Y. Zhu, C. Theobalt, M. Agrawala, E. Shechtman, D. B Goldman, and M. Zollhöfer. State of the art on neural rendering. *Computer Graphics Forum (EG STAR 2020)*, 2020. 2

[38] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 2

[39] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018. 2

[40] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances in neural information processing systems*, pages 613–621, 2016. 2

[41] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, pages 835–851. Springer, 2016. 2

[42] Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The pose knows: Video forecasting by generating pose futures. In *Proceedings of the IEEE international conference on computer vision*, pages 3332–3341, 2017. 2

[43] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Bryan Catanzaro, and Jan Kautz. Few-shot video-to-video synthesis. *Advances in Neural Information Processing Systems*, 32:5013–5024, 2019. 2

[44] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*, 2018. 2, 5, 6, 7

[45] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. SynSin: End-to-end view synthesis from a single image. In *CVPR*, 2020. 2

[46] Menghua Zhai, Zachary Bessinger, Scott Workman, and Nathan Jacobs. Predicting ground-level scene layout from aerial imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 867–875, 2017. 2

[47] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, 2017. 3, 4, 6