Geometry-Guided Street-View Panorama Synthesis From Satellite Imagery

Yujiao Shi, Dylan Campbell, Xin Yu, and Hongdong Li

Abstract—This paper presents a new approach for synthesizing a novel street-view panorama given a satellite image, as if captured from the geographical location at the center of the satellite image. Existing works approach this as an image generation problem, adopting generative adversarial networks to implicitly learn the cross-view transformations, but ignore the geometric constraints. In this paper, we make the geometric correspondences between the satellite and street-view images explicit so as to facilitate the transfer of information between domains. Specifically, we observe that when a 3D point is visible in both views, and the height of the point relative to the camera is known, there is a deterministic mapping between the projected points in the images. Motivated by this, we develop a novel satellite to street-view projection (S2SP) module which learns the height map and projects the satellite image to the ground-level viewpoint, explicitly connecting corresponding pixels. With these projected satellite images as input, we next employ a generator to synthesize realistic street-view panoramas that are geometrically consistent with the satellite images. Our S2SP module is differentiable and the whole framework is trained in an end-to-end manner. Extensive experimental results on two cross-view benchmark datasets demonstrate that our method generates more accurate and consistent images than existing approaches.

Index Terms—Novel view synthesis, satellite imagery, street-view imagery

1 Introduction

Given a satellite image, such as Fig. 1a, what would one see when standing at the location of the image center? In this example, one would reason that there is a fork in the road, a tree inside the fork, and grass elsewhere. This satellite to street-view image synthesis task aims to generate an omni-directional street-view panorama captured at a location corresponding to the center of the given satellite image. Our goal in this work is to synthesize a street-view panorama with scene structures that are as geometrically consistent with the satellite image as possible, while preserving visual similarity with the ground-truth panorama.

Structure-preserving street-view panorama synthesis is useful for several downstream tasks. For instance, it has been shown to be helpful for cross-view image localization [1], [2]. For example, using both synthesized street-view images and the original satellite images can help to increase geo-localization performance [1]. Combining satellite-to-street-view image synthesis with cross-view geo-localization in a unified framework further improves the performance of both tasks [2]. Furthermore, satellite imagery now covers the entire world and is easily accessible almost everywhere. In contrast, street-view images are expensive to collect and are not available everywhere. Synthesizing ground-level images from satellite images helps to enrich media content for regions that are hard or expensive for humans or vehicles to access.

As a special case of novel view synthesis (NVS), the satellite to street-view image synthesis is remarkably challenging because: (1) the significant change in viewing angle results in minimal field-of-view (FoV) overlap; and (2) there are radical differences in image appearance since the imaging modalities are highly distinct and they may be captured at different times of day, seasons, and weather conditions.

Conventional methods for solving this problem are often based on conditional generative adversarial networks (GANs) and they approach this problem as a pure image generation task. In their implementations, a powerful generator is employed to map the satellite images to the ground-level viewpoint by playing a min-max game with a discriminator.

Although a deep neural network is theoretically able to learn any transformation, neglecting the significant domain discrepancy between satellite and street-view images would lead to inferior performance. To be specific, satellite images show the top of objects in an overhead view by parallel projection, while street-view panoramas capture scenes at ground level with a spherical equirectangular projection. The transformations between satellite and street-view images are far more complex: they are not only content-dependent but also geometry-dependent.

In this paper, we develop a novel approach to explicitly establish the geometric correspondences between satellite and street-view images, recover the spatial layout information of scene objects in street-view, and handle occlusions.

- *Yujiao Shi and Hongdong Li are with Australian National University, Canberra, ACT 0200, Australia. E-mail: {Yujiao.Shi, Hongdong.Li}@anu.edu.au.*
- *Dylan Campbell is with the University of Oxford, Oxford OX1 2JD, U.K.. E-mail: dylan@robots.ox.ac.uk.*
- *Xin Yu is with the University of Technology Sydney, Ultimo, NSW 2007, Australia. E-mail: xin.yu@uts.edu.au.*

Manuscript received 25 Aug. 2020; revised 16 Dec. 2021; accepted 26 Dec. 2021. Date of publication 7 Jan. 2022; date of current version 3 Nov. 2022.
This work was supported in part by ARC-Discovery under Grant 190102261. Yujiao Shi is a China Scholarship Council (CSC)-funded PhD student to ANU and Dylan Campbell is grateful for support from Continental AG.
(Corresponding author: Yujiao Shi.)
Recommended for acceptance by J. Hays.
Digital Object Identifier no. 10.1109/TPAMI.2022.3140750

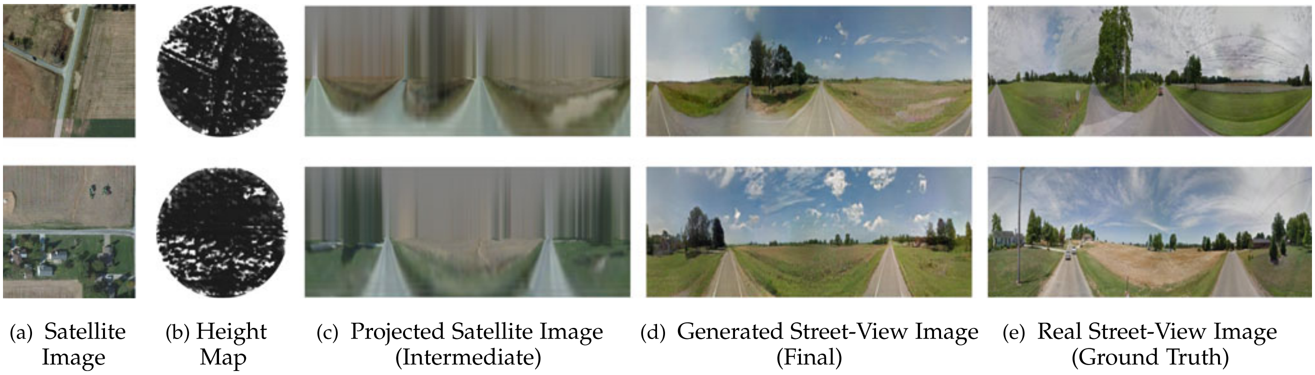

(a) Satellite Image
(b) Height Map
(c) Projected Satellite Image (Intermediate)
(d) Generated Street-View Image (Final)
(e) Real Street-View Image (Ground Truth)

Fig. 1. Given a satellite image (a), our method first estimates the height distribution (b), where lighter is higher, and then differentiably projects the satellite image to the ground-level viewpoint (c) according to the estimated height distribution. Conditioned on the projected image, our generator synthesizes a realistic street-view panorama (d) that is geometrically consistent with the satellite image, and very similar to the real street-view image (e).

We achieve this goal by developing a satellite to street-view projection (S2SP) module. Our S2SP module first estimates height maps of the satellite images and uses these to project satellite image pixels to the street view.

Specifically, our S2SP module estimates the height probability distribution for a given satellite image at a fixed set of heights, and constructs a satellite-view multiplane image (MPI) across the discretized heights to model the 3D scene. To generate the street-view panorama, the S2SP module then transforms the satellite-view MPI to a street-view MPI by unrolling and stretching a set of concentric cylinders. The street-view image is then rendered from the street-view MPI by employing an *over* alpha compositing technique [3] in a back-to-front order. In this manner, our approach handles occlusions between scene objects in a differentiable way. Figs. 1b and 1c provide two examples of the estimated height maps and the projected images from our S2SP module. Conditioned on the projected images, we use a generator network to synthesize realistic panoramic images and in-paint the missing textures.

Furthermore, due to the issue of GPS drift, it is difficult to collect location-aligned satellite and street-view image pairs, where the location of the street-view camera exactly corresponds to the center of a satellite image [4]. In this paper, our S2SP module also provides a method to align the collected satellite and street-view image pairs, providing clean training signals and accurate evaluation measurements for satellite to street-view synthesis.

This is an original submission that is not based on any published conference papers. The main contributions of this paper are summarized as follows:

- a new geometry-aware framework for satellite to street-view image synthesis, which explicitly establishes the geometric correspondences between the satellite and street-view images and allows a generator to focus on learning scene content dependent transformations (i.e., the visual appearance transformation of scene objects between the overhead view and the street view);
- a novel and differentiable satellite to street-view image projection module, which provides a way to estimate the height map for satellite images and the visibility of objects in the street view, without explicit supervision; and
- a novel mechanism for alleviating cross-view image pair misalignments, which not only helps to obtain clean satellite to street-view image synthesis training pairs but also identifies the location shift between the street-view camera and the satellite image center.

2 Related Work

2.1 Novel View Synthesis

Traditional novel view synthesis addresses the problem where only a small camera movement exists between source and target views. Liu *et al.* [5] tackled the problem of single image novel view synthesis. They approximated the real world scene as a set of planes with different surface normals, and learned to predict the homography of these planes, with which the input view can be transformed to the target view. Zhou *et al.* [6] proposed the multiplane image representation (MPI) to extrapolate views from stereo images with narrow baselines. They modeled the scene as a number of image planes at a fixed range of depths with respect to a reference camera coordinate frame. Based on this representation, Flynn *et al.* [7] explored a learned gradient method to estimate an MPI from a set of sparse camera viewpoints, and Tucker and Snavely [8] introduced a scale-invariant view synthesis mechanism to generate an MPI from a single-view online video.

Our method for satellite to street-view image synthesis is also built upon the multiplane image representation. As the viewpoint change is very large in our task, it is very difficult to directly render the target street-view panorama from an overhead-view MPI. Therefore, we integrate the overhead-view MPI as an intermediate output of our network rather than the final output, and employ a subsequent generator conditioning on the projected street-view images to inpaint missing textures and synthesize realistic images.

2.2 Satellite-View and Street-View Synthesis

The cross-view image synthesis task is extremely challenging between overhead and ground views since the visual appearance changes significantly. Zhai *et al.* [9] proposed to learn a linear transformation matrix between satellite and street-view semantics so that one can predict the street-view semantic layout by a matrix multiplication of the transformation and the satellite semantics. Regmi and Borji [10] investigated employing conditional GANs for cross-view synthesis. Instead of a single image, their networks regressed the target view image and its semantics jointly, with the semantic branch providing an additional supervision signal for image synthesis. They further extended their work by using a homography to map the images based on the common area between the views [11]. This provided more realistic details to the input image of conditional GANs. In contrast to a single homography on the ground plane, our method models geometric correspondences for all pixels between the views. Tang *et al.* [12] used street-view semantics and the satellite image to synthesize a street-view image. In contrast, our approach does not require a target view semantic segmentation during training and testing, thus reducing the requirements of the dataset annotation (or a dataset used to pre-train a semantic segmentation network). Instead, we exploit the two-view geometric constraints as a source of information with which to condition the generator. Lu *et al.* [4] also exploited geometric cues for satellite to street-view image synthesis, but they required ground-truth height and semantic supervision for the satellite images, and cannot train their network in an end-to-end manner due to discretized operations in the satellite-view to street-view projection. On the contrary, our method solves the problem in an end-to-end fashion under a more challenging and general setting, where the satellite map height supervision is not available.

2.3 Cross-View Image Geo-Localization

Satellite and street-view image pairs are also frequently used for the task of image geo-localization. Different from satellite to street-view image synthesis, the cross-view image geo-localization is a deep metric learning problem which aims to learn discriminative feature representations for scenes at different locations. In this task, a query image captured by a ground-level camera is matched against a database with geo-tagged satellite images to determine the ground camera's location.

Workman and Jacobs [13], [14] and Vo and Hays [15] pioneered this task by investigating a family of deep learning methods for the cross-view geo-localization. Hu *et al.* [16], Sun *et al.* [17] and Cai *et al.* [18] focused on designing powerful networks or losses to achieve better results. Based on a simple deep architecture, Liu and Li [19] used orientation information of street-view and satellite images to assist geo-localization. Regmi and Shah [1] employed a conditional GAN to generate satellite images from street-view panoramas so as to bridge the domain gap for cross-view image matching.

Our previous works are based on cross-view image geo-localization, a different but related task to satellite to street-view image synthesis. For the cross-view image geo-localization, we first proposed an optimal feature transport module that aligned cross domain features to facilitate similarity matching [20], and later showed that a polar transform could be used to establish approximate geometric correspondences between satellite-view and street-view images and thus boost the localization performance [21], [22]. The polar transform is a simple approximation of the nonlinear transformation between satellite-view and street-view images; in this work we instead use the true transformation from two-view geometry, which does not distort the scene geometry, for the image synthesis task.

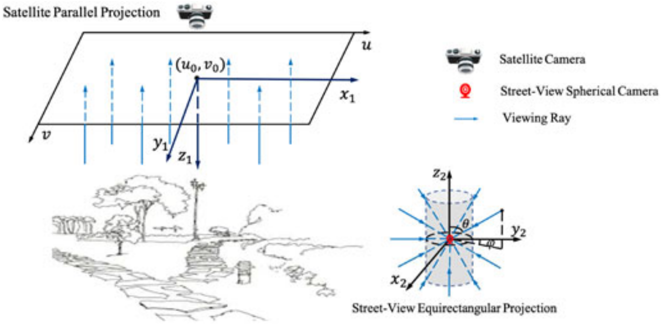

Satellite Parallel Projection
u
(u_0, v_0)
x_1
v
y_1
z_1
Satellite Camera
Street-View Spherical Camera
Viewing Ray
z_2
θ
y_2
ϕ
x_2
Street-View Equirectangular Projection

Fig. 2. Visualization of different projection approaches by a satellite camera and a street-view spherical camera. The former is a parallel projection in the overhead view, and the latter is an equirectangular projection at ground level.

3 Satellite and Street-View Geometry

3.1 Parallel Projection of a Satellite Camera

As shown in Fig. 2 (left), we denote the satellite camera coordinates as (x_1, y_1, z_1) and the satellite image coordinates as (u, v). The projection between the satellite camera coordinate system and satellite image coordinate system is approximated as a parallel projection, which maps the point (x_1, y_1, z_1) to the satellite image point (u, v) as

$$\begin{aligned} u &= u_0 + s x_1 \\ v &= v_0 + s y_1, \end{aligned} \tag{1}$$

where (u_0, v_0) is the satellite image center, and s is the scale factor between the satellite image coordinates and the world coordinates.

3.2 Perspective Projection of an Omnidirectional Street-View Camera

The projection method of an omnidirectional street-view camera is illustrated in Fig. 2 (right). We use a cylinder to represent the image plane, but the pixels are parameterized under a spherical coordinate system. Let (x_2, y_2, z_2) be the camera coordinates, and (θ, ϕ) be the street-view image coordinates. The omnidirectional street-view camera maps the point (x_2, y_2, z_2) to the panorama image point (θ, ϕ) by an equirectangular projection

$$\begin{aligned} \theta &= \begin{cases} \operatorname{atan2}(\sqrt{x_2^2 + y_2^2}, z_2) & z_2 \neq 0 \\ \pi/2 & z_2 = 0 \end{cases} \\ \phi &= \begin{cases} \operatorname{atan2}(x_2, y_2) & y_2 \neq 0 \\ \pi/2 \cdot \operatorname{sign}(x_2) & y_2 = 0 \end{cases}. \end{aligned} \tag{2}$$

3.3 Satellite-View and Street-View Geometry

For the satellite to street-view image synthesis, we illustrate the geometric correspondences between satellite and street-view images in Fig. 3. As shown in the figure, the street-view camera is set at the location corresponding to the satellite image center. We set the origin of the world coordinate to the ground camera location, where its x-axis is parallel to the v direction of the satellite image coordinate, y-axis is parallel to the u direction of the satellite image coordinate, and z axis points upward. For a pixel p_{sat} in the satellite image which is visible from the street view, the transformation between p_{sat}

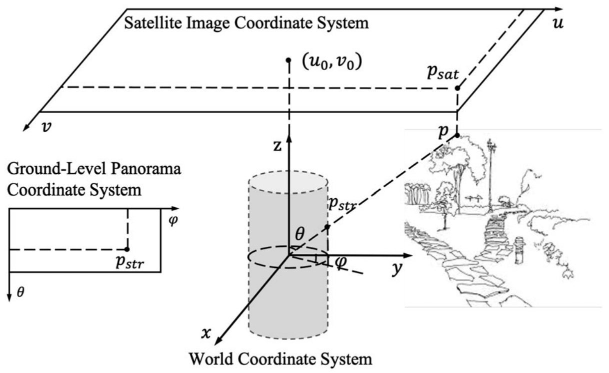

Satellite Image Coordinate System
u
(u_0, v_0)
p_{sat}
v
p
z
Ground-Level Panorama Coordinate System
φ
p_{str}
θ
p_{str}
θ
φ
y
x
World Coordinate System

Fig. 3. Corresponding pixels in a pair of satellite and street-view images. When a point p in the world coordinate is visible in both the satellite and street-view images, there is a deterministic geometric mapping between the projected pixels p_{sat} and p_{str}.

and its projected location at the street-view panorama p_{str} is deterministic given the height z, expressed as

$$\theta = \begin{cases} \text{atan2}(\sqrt{(v-v_0)^2+(u-u_0)^2}, sz) & z \neq 0 \\ \pi/2 & z = 0 \end{cases}$$
$$\phi = \begin{cases} \text{atan2}(v-v_0, u-u_0) & u \neq u_0 \\ \pi/2 \cdot \text{sign}(v-v_0) & u = u_0 \end{cases}. \tag{3}$$

In order to establish these geometric correspondences, one could estimate a pixel-wise height map for the satellite image and then project the corresponding 3D points into the street-view image plane. A standard approach would be to sort the projected 3D points along each viewing ray by depth (using a z-buffer) and selecting the closest point to color the corresponding pixel. However, this forward mapping approach has several problems. (1) Poor occlusion modeling: this approach only models the top surface of the scene, and so the occlusion in the global world is hard to be modeled. For example, points that lie behind a tree (viewing at the ground level) will not be occluded, because only the tree canopy is treated as an occlusion surface. (2) Forward mapping artifacts: the significant mismatch in resolution between the satellite and street-view images leads to many missing pixels in the output projected image. (3) Closest point selection: hard selection from the z-buffer is non-differentiable near the visibility boundary, because a small change in the estimated height can lead to a discontinuous change in the output projection, as points shift from visible to invisible and vice versa. Moreover, it leads to sparse gradient signals, since there is zero gradient for non-visible points. This is sub-optimal for points that should be visible but are not, due to a slightly incorrect height estimate.

Considering these issues, we instead propose to model the scene with a dense volumetric representation rather than sparse 3D points, and use an inverse mapping to obtain the street-view projection. We use multiplane images (MPIs) [6] to achieve this goal.

An MPI is a set of fronto-parallel image planes $\{I_1, I_2, \dots, I_N\}$ at a fixed range of depths with respect to a reference coordinate frame, where N is the number of the depth planes. Each image plane I_i in an MPI includes three color channels $C_i \in \mathbb{R}^{H\times W\times 3}$ and an alpha channel $\alpha_i \in \mathbb{R}^{H\times W\times 1}$ to encode the transparency, where H and W are the height and width of the image planes. This multiplane image structure is able to represent the geometry and texture of scenes including occluded elements. In the next section, we will present the technical details on the MPI construction and its usage in our proposed differentiable satellite to street-view image projection.

4 Street-View Synthesis Guided by Two-View Geometry

An overview of our network is shown in Fig. 4. Our method establishes the geometric correspondences between the satellite and street-view images by a differentiable Satellite to Street-view image Projection (S2SP) module, and then employs a generator to produce realistic and geometrically consistent street-view panoramas with respect to input satellite images.

4.1 Satellite to Street-View Image Projection

Height Estimation. Given a satellite image, we first estimate its pixel-wise height probability distribution for a fixed range of heights, given by

$$D = f_{\text{height}}(I_{\text{sat}}) \quad \text{s.t.} \ \textstyle\sum_{i=1}^{N} D_{h,w,i} = 1, \tag{4}$$

where $D \in \mathbb{R}^{H^{\text{sat}}\times W^{\text{sat}}\times N}$ denotes the estimated probability distribution of the N discretized heights, $f_{\text{height}}(\cdot)$ is the height estimation network, $I_{\text{sat}} \in \mathbb{R}^{H^{\text{sat}}\times W^{\text{sat}}\times 3}$ denotes the RGB satellite image, and h, w and i are the indices for the image height, image width and height (elevation) dimensions, respectively. Estimating the height probability of pixels instead of a single height map enables us to model the 3D scene in a dense and relatively continuous manner.

Satellite-View MPI Construction. Given the estimated height probability distribution, we construct a satellite-view MPI across the discretized heights to model the global scene. As

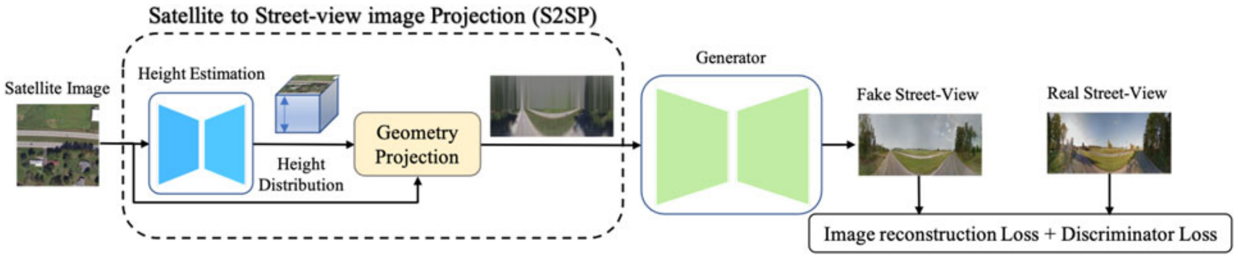


Fig. 4. Flowchart of the proposed framework. We propose a novel satellite to street-view image projection module to transform the satellite images to the street viewpoint and a projection-conditioned generator to synthesize realistic street-view panoramas that are geometrically consistent with the satellite images.

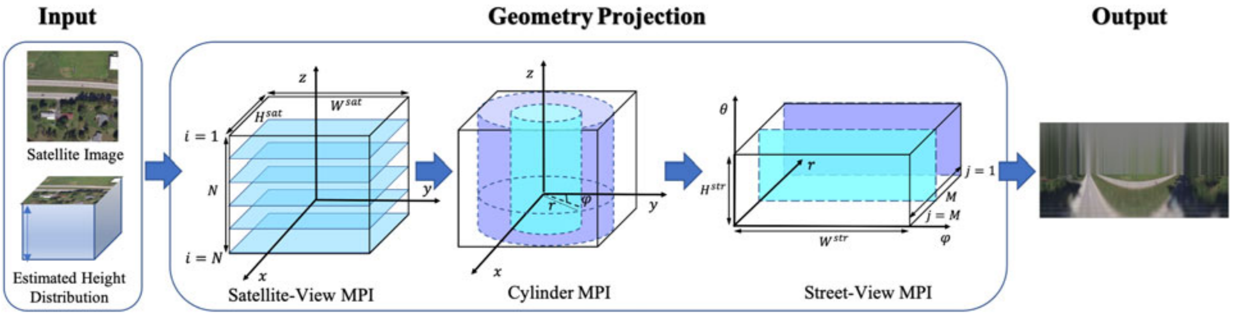


Fig. 5. Illustration of the geometry projection block in our S2SP module. We first construct an overhead-view MPI according to the given satellite image and the estimated height probability distribution, and then convert it to the street viewpoint by unrolling and stretching each concentric cylinder. The final street-view image is rendered in a back-to-front order from the street-view MPI.

illustrated in Fig. 5, an satellite-view MPI is composed of a set of image planes parallel to the satellite image with different heights.

Pixels in a satellite image correspond to the topmost points of scene objects. Purely modeling the 3D points corresponding to the satellite image pixels is insufficient for solving the occlusion problem at the street view, since it leaves the points below the top structures undetermined. Therefore, we also model the lower points by assuming that these points are also opaque. This is achieved by setting the transparency channel for each image plane $\alpha_i^{\text{sat}} \in \mathbb{R}^{H^{\text{sat}} \times W^{\text{sat}} \times 1}$ to the cumulative height probability distribution

$$\alpha_i^{\text{sat}} = \sum\nolimits_{k=1}^{i} D_{\cdot,\cdot,k}, \tag{5}$$

where plane $i = N$ is the ground plane. For the color channels of the satellite-view MPI, we set them to the input satellite image, with $C_i^{\text{sat}} = I_{\text{sat}}$.

*Street-View MPI Projection.*As our target is to synthesize street-view panoramas, our next step is to determine the order of these points along viewing rays at the street viewpoint. To do so, we first decompose the overhead-view MPI to a set of concentric cylinders with uniformly sampled radii, as illustrated in Fig. 5 (middle), and then project the cylinders to the street-view panorama (spherical) image coordinates. The result is a set of image planes at the street viewpoint with uniformly sampled depths (along the z-axis), which we refer to as the street-view MPI.

Let (u, v, z) denote the coordinates of points in the overhead-view MPI and (θ, ϕ, r) represent the coordinates of projected points in the street-view MPI, where r is the concentric cylinder radius and $r = \sqrt{(u-u_0)^2 + (v-v_0)^2}$. The transformation between the source points in the original satellite MPI and the target points in the street-view MPI is

$$\begin{aligned} u &= u_0 + sr\cos\phi \\ v &= v_0 + sr\sin\phi \\ z &= \begin{cases} r/\tan\theta & \theta \in [0, \pi/2) \cup (\pi/2, \pi] \\ 0 & \theta = \pi/2 \end{cases}. \end{aligned} \tag{6}$$

With this transformation, the points are sorted from far to near along viewing rays at the street view, and the satellite to street-view image projection is solved by an inverse mapping.

Street-View Image Rendering. Denote $C_j^{\text{str}} \in \mathbb{R}^{H^{\text{str}} \times W^{\text{str}} \times 3}$ and $\alpha_j^{\text{str}} \in \mathbb{R}^{H^{\text{str}} \times W^{\text{str}} \times 1}$ as the color and alpha channels of the street view MPI, respectively, where $j \in [1, M]$ is the index of street-view image planes, M is the plane number of the street-view MPI, H^{str} and W^{str} are the height and width of target street-view images, respectively. The street-view image is then rendered from the street-view MPI using *over* alpha compositing [6] in a back-to-front order

$$I_{\text{comp}}^{j} = \begin{cases} \alpha_1^{\text{str}} \cdot C_1^{\text{str}} & \text{if} j = 1 \\ \alpha_j^{\text{str}} \cdot C_j^{\text{str}} + (1 - \alpha_j^{\text{str}}) \cdot I_{\text{comp}}^{j-1} & \text{otherwise} \end{cases}, \tag{7}$$

where $j = 1$ indicates the furthest image plane and $j = M$ corresponds to the closest image plane. The final composited image is thus I_{comp}^{M}. Points with higher α values will have higher weights in the composited image and thus dominate the color of a pixel. By using this approach, all the pixels in the original satellite image contribute to the final rendered street-view image, and thus it allows dense gradients during training. Since all the operations in our S2SP module are differentiable, our network can be trained in an end-to-end manner.

4.2 Network Architecture

As illustrated in Fig. 4, our network includes a Satellite to Street-view image Projection (S2SP) module and a generator to synthesize geometrically consistent and realistic street-view panoramas. We employ the Pix2Pix [23] network as our generator backbone. For the height estimation block in the S2SP module, we employ the same architecture as the Pix2Pix network but with the number of output channels in each layer reducing by $1/16$. Since ground-truth height maps of the satellite images are not available, there is no explicit supervision for the height estimation block in our pipeline. Instead, implicit supervision is provided by enforcing the two-view geometric constraints and propagating the error signal back from the output of the generator. In particular, if the estimated heights of pixels deviate from ground-truth values, the projected 3D points will be reordered when viewed at the ground level, changing the rendered images. Thus, the error between the final generated image and the real target image may be larger when the heights are estimated incorrectly. Backpropagating this signal through the differentiable cross-view projection layer allows the height estimation network parameters to be updated without explicit supervision.

4.3 Adversarial Learning From Corresponding Satellite and Street-View Image Pairs

Following recent state-of-the-art approaches [10], [12], we use a generative adversarial network and adopt an adversarial training procedure. The generator G is trained to map satellite images to their street-view counterparts by playing a min-max game with the discriminator network D. The generative adversarial objective function, minimized by the generator and maximized by the discriminator, is given by

$$\begin{aligned}\mathcal{L}_{GAN}(G, D) =&\mathbb{E}_{I_{\text{str}}\sim p_{str}(I_{\text{str}})}[\log D(I_{\text{str}})]\\&+\mathbb{E}_{I_{\text{sat}}\sim p_{sat}(I_{\text{sat}})}[\log(1 - D(G(I_{\text{sat}})))],\end{aligned}\quad(8)$$

where I_{sat} is the input satellite image, I_{str} is the real street-view image, and $G(I_{\text{sat}})$ is the generated street-view image.

Since the database street images do not strictly correspond to their satellite counterparts, e.g., the satellite images are captured in summer while the street-view panoramas depict scenes in winter or autumn, we adopt the perceptual loss $\mathcal{L}_{\text{per}}$ [6] along with the $\mathcal{L}_1$ loss between the real and generated street-view images to evaluate their feature similarity and their pixel-wise color similarity. Overall, the image reconstruction loss is given by

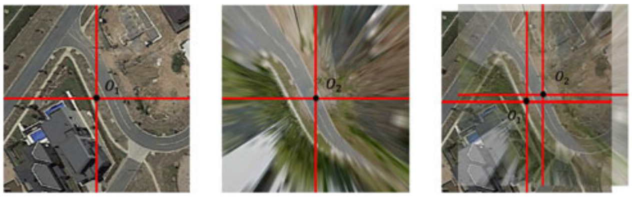
$$\begin{aligned}\mathcal{L}_{\text{rec}}(G) =&\mathcal{L}_1(G) + \mathcal{L}_{\text{per}}(G)\\=&\|I_{\text{str}} - G(I_{\text{sat}})\|_1 + \sum\nolimits_l \lambda_l\|d_l(I_{\text{str}}) - d_l(G(I_{\text{sat}}))\|_1,\end{aligned}\quad(9)$$

where $d_l(\cdot)$ indicates a set of feature representations of an image from a VGG-19 network [24] (conv1_1, conv2_2, conv3_2, conv4_2 and conv5_2), $\|\cdot\|_1$ is the L_1 distance and λ_l represents the corresponding weight hyperparameters. Following the work of Zhou *et al.* [6], we set the weight hyperparameters to the inverse of the number of neurons in each layer. The overall objective of our network, minimized by the generator and maximized by the discriminator, is

$$\mathcal{L}(G, D) = \mathcal{L}_{\text{GAN}}(G, D) + \mathcal{L}_{\text{rec}}(G).\quad(10)$$

5 Aligning Satellite and Street-View Image Pairs

Due to the GPS positioning error, it is hard to collect strictly location aligned satellite and street-view image pairs for satellite to street-view image synthesis, where the camera location of street-view panorama corresponds exactly to the center of the satellite image. Figs. 6a and 6d show an example satellite and street-view image pair from the CVACT dataset, which is tagged as "matching" but is not strictly aligned.



(a) Satellite Image (b) Projected Street-View (c) Overlay

Fig. 7. Visualization of satellite and street-view image pair misalignment: (a) original satellite image with image center O_1; (b) projected street-view image in the overhead view with image center O_2; and (c) overlaid image of (a) and (b). There is a location shift between O_1 and O_2.

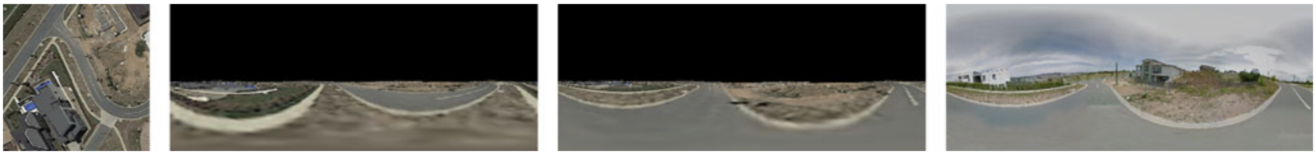
This misalignment is detected using the following procedure. We first project the street-view panorama to the satellite view by exploiting the geometric correspondences illustrated in Section 3. In this process, we assume that all the pixels lie on the ground plane. The results are presented in Fig. 7b. We then overlay it on the original satellite image (Fig. 7a), as shown in Fig. 7c. There is a shift between the original satellite image center O_1 and the projected street-view image center O_2.

To correct this misalignment automatically, we exploit the satellite to street-view image projection. Since the misalignment is small, we select 40×40 points in a central region of the original satellite image, corresponding to 11.25×11.25 meters, to model the potential shifts between them. We next exhaustively project the satellite image into the street viewpoint at each of the points, and compare the similarity (SSIM value) between the projected images and the original street-view image. The one with the maximum similarity is selected and its corresponding "shift" is adopted to align the cross-view image pair.

We perform geometric satellite to street-view image projection under the assumption (for this part only) that all the pixels lie on the ground plane. Hence there are no trainable parameters and we can use this directly as a pre-processing step to create clean satellite and street-view training pairs. Figs. 6b and 6c show a comparison of the projected images before and after our correction. It can be seen that the projected image after correction is aligned with the corresponding street-view image from the dataset.

6 Benchmarks for Satellite to Street-View Image Synthesis

There are currently two large-scale and publicly available cross-view datasets, namely, CVACT [19] and CVUSA [9]. These two datasets has been widely used as benchmarks for cross-view image based geo-localization [1], [9], [16], [17],



(a) Satellite Image (b) Projected Satellite Image (Original) (c) Projected Satellite Image (Corrected) (d) Street-View Image

Fig. 6. Example of satellite and street-view image misalignment and correction: (a) original satellite image, (b) projected satellite image according to the original satellite image center, (c) projected satellite image after mitigating the satellite and street-view image misalignment, and (d) corresponding street-view image in the database. The height maps are set to zero for this visualization.

[18], [19], [20], [21], [22]. In this paper, we instead employ them for satellite to street-view image synthesis.

Both CVACT and CVUSA contain 35,532 satellite and street-view image pairs for training and 8,884 image pairs for testing. As both of the two datasets are introduced for image geo-localization, they are allowed to have slightly location shift between matching satellite and street-view image pairs. However, in the satellite to street-view image synthesis task, the goal is to synthesize a street-view panorama as if it is captured at the same geographical location as the satellite image center. Therefore, it is necessary to have exactly location aligned cross-view image pairs, especially for performance evaluation.

For CVACT, we propose a new split for the problem of satellite to street-view image synthesis, with 26,519 training pairs and 6,288 testing pairs, and augment the dataset with translation offsets to provide ground-truth alignment. This new split is necessary because many of the ostensible ground-truth image pairs are not aligned in the original dataset, that is, the translation offset between the centre of the satellite image and the camera position is unknown. This is highly problematic for the cross-view synthesis task. To generate this split, we automatically computed the translation offsets for the image pairs, using the image alignment method proposed in Section 5. However, we are unable to compute the offset in two situations: (1) where the translational misalignment is too large, typically more than 16 meters, and (2) where there is severe cross-class occlusion in the vertical direction. The latter refers to situations where, for example, the satellite view can only see a tree canopy, while the street view sees the road surface underneath. For these cases, our alignment method, and human annotators, are unable to estimate the translation offset, and so these image pairs are filtered from the split. The split and translation offsets will be made publicly available.

The street-view panoramas in CVUSA dataset were cropped at the top and bottom by Zhai *et al.* [9] to reduce the fraction of sky and ground. It is not clear whether they have been cropped uniformly across the dataset, so we cannot apply the same misalignment rectification method to the CVUSA dataset. During training and testing, we approximate the street-view panoramas in the CVUSA dataset as having a 90-degree vertical field of view (FoV) with the central horizontal line corresponding to the horizon. The CVACT dataset contains panoramas with a 180-degree vertical FoV [19]. Note that the panoramic images in both datasets have a 360-degree horizontal FoV.

Lu *et al.* [4] also proposed a cross-view dataset with ground truth semantics and height maps for satellite images (which are not available in CVACT and CVUSA dataset). However, this dataset has not been released. Regmi and Shah [1] also introduced the OP dataset for fine-grained cross-view geo-localization. However, this dataset has significant and uncorrelated position and orientation misalignments and is an order of magnitude smaller than CVACT and CVUSA, making it unsuitable for the image synthesis task. Therefore, we use the CVACT and the CVUSA datasets for evaluation.

7 Experiments

7.1 Implementation Details

In our implementation, we resize the input satellite image to 256×256 pixels and set the output size of a street-view panorama to 128×512 pixels. We approximate the height of the street-view camera as 2 meters with respect to the ground plane. The maximum height modeled by our multiplane image representation is 8 meters. We set the number of satellite-view MPI planes N to 64 in our experiments, with a half meter interval between planes. The number of street-view MPI planes M is also set to 64. The network is trained in an end-to-end manner with a batch size of 4. We follow Pix2Pix's [23] use of the Adam optimizer [25] with a learning rate of 0.0002 for both the generator and discriminator, and $\beta_1 = 0.5$, $\beta_2 = 0.9999$. The source code is available at https://github.com/shiyujiao/Sat2StrPanoramaSynthesis.git.

The street-view image rendering complexity is $O(H^{\text{str}} W^{\text{str}} M)$. The flops in $\alpha^{\text{str}} \cdot C^{\text{str}}$ are $3H^{\text{str}} W^{\text{str}} M$, and the flops in $(1-\alpha^{\text{str}}) \cdot I_{\text{comp}}$ are $4H^{\text{str}} W^{\text{str}} M$. Using an RTX 2080 TI, the maximum batch size is 12 for training and 64 for testing. The training time is around 12 hours for the CVACT dataset and 16 hours for the CVUSA dataset. On average, the computation time is 0.2s per synthesized image.

7.2 Evaluation Metrics

In this paper, we adopt various evaluation metrics for quantitative assessment. The low-level similarity measures includes root-mean-square error (RMSE), structure similarity index measure (SSIM), peak signal-to-noise ratio (PSNR), and sharpness difference (SD). They evaluate the pixel-wise similarity between two images. However, in the cross-view synthesis task, it is hard to synthesize exactly the same target view image as the ground truth, due to the minimal overlap and seasonal change between input and target view images. The colors between the generated and ground truth street-view images may be different, but they depict the same location. Thus, we further adopt the high-level perceptual similarity [26] for the performance evaluation. The perceptual similarity evaluates the feature similarity of generated and real images. We employ the pretrained AlexNet [27] and Squeeze [28] networks as backbones for the evaluation, denoted as P_{alex} and P_{squeeze}, respectively.

Additionally, we employ a pre-trained semantic classifier to measure the semantic difference between the real and generated images. The semantic classifier is trained on the CityScapes dataset [29] and fine-tuned on the CVUSA dataset by using the lightweight RefineNet [30] network. We report the pixel-wise accuracy (Acc.) and mean intersection over union (mIoU), as in the work [31].

7.3 Comparison With Existing Methods

We compare our method with Pix2Pix [23] and XFork [10]. Pix2Pix is a well-known GAN-based network for image-to-image translation and has been widely used as the baseline for cross-view image synthesis [10], [12]. XFork, proposed by Regmi and Borji [1], used semantic information as additional network guidance during training, generating the target image and semantic map simultaneously with a weight-shared decoder. The authors also proposed another network, XSeq, which stacked two generators together to generate the target image and semantic map sequentially. We compare with XFork, which was shown to outperform XSeq [10]. Selection GAN [12] is another recent work on cross-view image synthesis. However, it assumed that a

TABLE 1
Quantitative Comparison With Existing Algorithms on the CVACT (Aligned) and CVUSA Datasets

	Method	RMSE↓	SSIM↑	PSNR↑	SD↑	Acc.↑	mIoU↑	P_{alex} ↓	P_{squeeze} ↓
CVACT (Aligned)	Pix2Pix [23]	49.15	0.3733	14.47	16.06	0.8318	0.2137	0.4506	0.2921
	XFork [10]	50.81	0.3701	14.17	15.89	0.8227	0.2123	0.4408	0.2932
	Ours	**48.23**	**0.4212**	**14.65**	**16.33**	**0.8353**	**0.2145**	**0.4099**	**0.2701**
CVUSA	Pix2Pix [23]	56.11	0.2952	13.35	15.90	0.7742	0.2042	0.5037	0.3774
	XFork [10]	56.77	0.2926	13.25	15.80	0.7722	0.2039	0.5144	0.3836
	Ours	**53.67**	**0.3408**	**13.77**	**16.27**	**0.7831**	**0.2060**	**0.4824**	**0.3577**

semantic segmentation of the street-view panorama was available during testing, which is different to our problem setting. Our goal in this paper is to synthesize a street-view panorama from a satellite image without any information from the target domain. Lu *et al.* [4] also exploited geometric correspondences for satellite to street-view image synthesis. However, their work needs explicit height and semantic supervision for satellite images, which is not available for current accessible datasets (CVUSA and CVACT). Therefore, we cannot meaningfully compare with these two methods.

Table 1 presents the quantitative comparison results. As indicated in the table, our method achieves consistently better results on all quantitative evaluation metrics. Fig. 8 provides some qualitative visualizations from the CVACT (aligned) and CVUSA datasets. As indicated by the results, our pipeline generally produces more natural images, which can be observed from the first two examples in Fig. 8a and the first example in Fig. 8b, where the images generated by Pix2Pix and XFork confuse some regions and inpaint them with artifacts. In particular, our method generates street-view panoramas that are more geometrically-consistent with respect to the input satellite images.

As shown in the first example of Fig. 8a, there is a roundabout in the input satellite image, which can also be observed from the corresponding street-view panorama (Ground Truth). Our method successfully projects the estimated scene geometry and recover this structure. The fully black-box networks Pix2Pix and XFork fail to learn the geometric transformations and generate unnatural synthesized images. The same phenomenon can be observed in the second example in Fig. 8a. Notably, our method can recover the lane line structure from the satellite image, as shown in the third and fourth example in Fig. 8a. Such lane line structure is challenging for a fully black-box model to generate, since different locations will have different structures and lane lines occupy a relatively small region in both input and output images.

Fig. 8b presents some challenging cases with a variety of complex scene structures from the CVUSA dataset. It can be seen that our method recovers the geometric structures from satellite images, while generating diverse and appropriate visual features. In contrast, the generated images from Pix2Pix and XFork are much more uniform for the first two examples, with two main roads uniformly distributed and other regions in-painted with grass. For a more complex case, the third example in Fig. 8b, Pix2Pix and XFork fail entirely while our method is more successful at recovering the road structure from the input satellite image. Preserving the correct road topology is very important in many applications, such as autonomous driving. The last example in Fig. 8b presents the most challenging case. The geometric structure of the input satellite image is complicated, and this kind of image (urban jungle) is rare in the training set. Pix2Pix and XFork hallucinate the ground-level scene, while our approach generates street-view panoramas that are more geometrically consistent with the ground truth. More specifically, our method restores (roughly) the ground and building geometry from the satellite image.

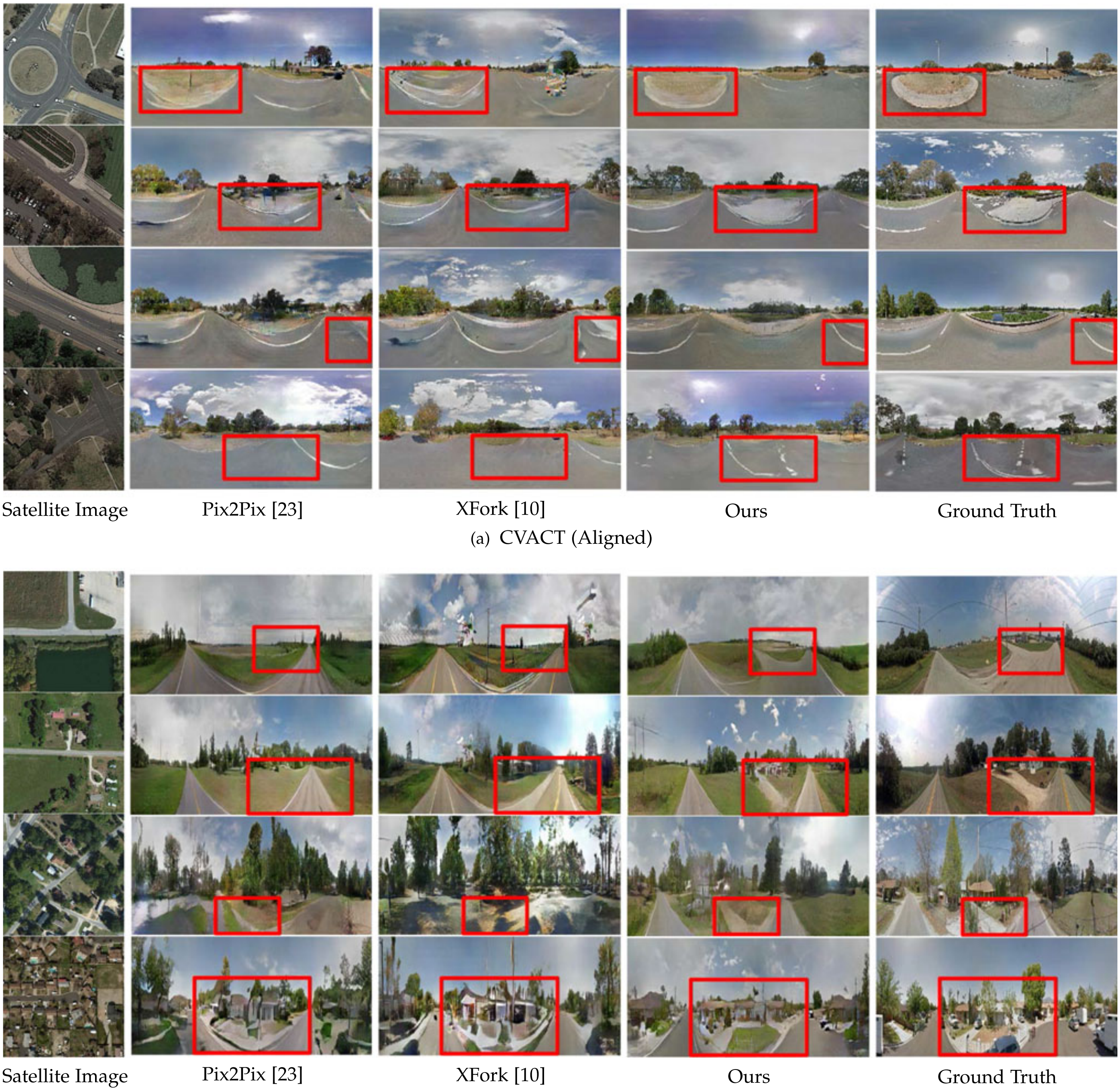
7.4 Comparison on Aligned or Unaligned Datasets

In this section we evaluate the performance of algorithms on the aligned and unaligned CVACT test sets with models trained on the aligned and unaligned training sets. We first consider the results on the aligned test set, which are more reliable measures of image synthesis performance. As shown in Table 2, all methods perform worse when trained on unaligned image pairs compared to aligned image pairs. This is expected since the supervision signal is noisier during training. Our method is the least sensitive, likely due to the explicit assumption in the projection module that the viewpoint of the synthesized image is strictly at the center of the satellite image.

We also evaluate the models on the unaligned test set. All models perform worse on this test set compared to the aligned test set, regardless of whether training data is aligned or not, since the “ground truth” is not correct or predictable. The performance of the Pix2Pix and XFork models decreases when trained on unaligned data, which is consistent with the results on the aligned test set. However, the magnitude of the decrease is much smaller. Since these models do not have an explicit alignment rule, the alignment of the training data does not affect the evaluations on unaligned data too much. In contrast, the performance of our method is inferior when trained on aligned data compared to unaligned data, because the explicit alignment rule in our method reduces the network’s capacity to learn any misalignment bias.

7.5 Ablation Study

In this section, experiments are carried out to validate the importance of each component in our framework. We divide our experiments into two groups: handcrafted methods and deep learning methods. For the handcrafted features, we compare with the polar transform [21] and our satellite to street-view image projection without height estimation, denoted as “Polar” and “Projection”, respectively. Both of these methods assume that pixels in the satellite



(a) CVACT (Aligned)

(b) CVUSA

Fig. 8. Qualitative comparison of synthesized images for the CVACT (Aligned) and CVUSA datasets.

TABLE 2
Performance Comparison on the Aligned and Unaligned CVACT Dataset (Both Training and Testing)

	Test	Aligned				Unaligned			
	Train	RMSE↓	SSIM↑	PSNR↑	SD↑	RMSE↓	SSIM↑	PSNR↑	SD↑
Pix2Pix [23]	Unaligned	50.21	0.3704	14.29	16.03	51.79	0.3563	14.03	15.88
	Aligned	49.15	0.3733	14.47	16.06	51.39	0.3544	14.10	15.91
	Change (%)	*2.122*↓	*0.7704*↓	*1.321*↓	*0.1874*↓	*0.7730*↓	*0.5100*↑	*0.4762*↓	*0.2044*↓
XFork [10]	Unaligned	51.36	0.3638	14.10	15.80	53.06	0.3485	13.83	15.65
	Aligned	50.81	0.3701	14.17	15.89	53.01	0.3497	13.82	15.71
	Change (%)	*1.068*↓	*1.744*↓	*0.5450*↓	*0.5903*↓	*0.0830*↓	*0.3340*↓	*0.0736*↑	*0.3745*↓
Ours	Unaligned	48.37	0.4210	14.62	16.27	49.82	0.4014	14.38	16.24
	Aligned	48.23	0.4212	14.65	16.33	50.66	0.3901	14.24	16.07
	Change (%)	*0.2910*↓	*0.0411*↓	*0.2481*↓	*0.3379*↓	*1.701*↑	*2.813*↑	*0.9244*↑	*1.011*↑

Here, the underlined number indicates the performance change (in percent) between a model trained on unaligned image pairs the same model trained on aligned image pairs. The arrow "↓" indicates performance decrease and "↑" indicates the performance increase.

TABLE 3
Ablation Study on the CVACT (Aligned) and the CVUSA Dataset

	Method	Params	CVACT (Aligned)				CVUSA			
			RMSE↓	SSIM ↑	PSNR ↑	SD ↑	RMSE ↓	SSIM ↑	PSNR ↑	SD ↑
Handcrafted	Polar	-	82.08	0.1503	9.941	13.83	74.50	0.1427	10.79	14.46
	Projection	-	118.0	0.1955	6.725	17.18	118.2	0.0742	6.737	16.22
Deep	Ours w/o S2SP	33.46M	48.63	0.4050	14.57	16.15	54.05	0.3197	13.70	16.08
	Ours w/o height (polar)	33.46M	49.46	0.3962	14.42	16.13	54.78	0.3122	13.57	16.00
	Ours w/o height (projection)	33.46M	49.08	0.4068	14.48	16.18	54.49	0.3301	13.63	16.11
	Ours w/o projection	33.62M	49.47	0.4074	14.43	16.14	53.96	0.3216	13.72	16.12
	Ours w/o $\mathcal{L}_1$	33.62M	48.76	0.4174	14.55	16.20	59.25	0.1512	12.79	10.56
	Ours w/o $\mathcal{L}_{\text{per}}$	33.62M	50.96	0.3973	14.18	16.10	55.43	0.3230	13.47	16.10
	Ours	33.62M	**48.23**	**0.4212**	**14.65**	**16.33**	**53.67**	**0.3408**	**13.77**	**16.27**

image lie on the ground plane. "Polar" is a simple approximation for cross-view image alignment, while "Projection" establishes the real geometric correspondences for pixels which have ground-level height in a satellite and street-view image pair.

For deep learning approaches, all of the comparison algorithms employ the Pix2Pix network as the generator backbone. We first remove the whole S2SP module from our pipeline, denoted as "Ours w/o S2SP". In this pipeline, only the generator backbone is retained and the generator is conditioned on the original satellite image. Next, we remove the height estimation block from our pipeline and assume that all the pixels have the same height as the ground plane (zero height). Two types of generator conditionings are investigated with this baseline: the polar-transformed images and the perspective-projected images, denoted as "Ours w/o height (polar)" and "Ours w/o height (projection)", respectively. We also investigate the usefulness of the geometry projection block by replacing it with a simpler approach: a soft argmax operation to convert the estimated height probability distribution into a single height map, which is concatenated with the satellite image to condition the generator, denoted as "Ours w/o projection". Finally, we study the influence of the $\mathcal{L}_1$ and $\mathcal{L}_{\text{per}}$ losses by removing them from the total image reconstruction loss, denoted as "Ours w/o $\mathcal{L}_1$" and "Ours w/o $\mathcal{L}_{\text{per}}$", respectively.

The ablation study is presented in Table 3. The handcrafted methods perform significantly worse than the deep learning methods, indicating that the satellite to street-view transformation is too complex to be modeled by simple approximations. For the deep learning methods, our whole pipeline consistently achieves the best results on all evaluation metrics, indicating that every component is important for the success of the model.

Regarding the $\mathcal{L}_1$ and $\mathcal{L}_{\text{per}}$ losses, we found that using $\mathcal{L}_{\text{per}}$ significantly improves the quality of the synthesized images, as indicated by the last two rows of Table 3. This loss provides higher-level feature similarity guidance to the network in addition to the pixel-wise $\mathcal{L}_1$ color difference during training. This is especially important in the cross-view synthesis task, since the ground truth target-view images in the training set are not captured at the same time as the input-view images. Purely using the $\mathcal{L}_1$ color loss makes it difficult for the network to learn the significant cross-view transformations. However, discarding the $\mathcal{L}_1$ loss also impairs the performance. This can be observed from the third last row of Table 3, where the performance of "Ours w/o $\mathcal{L}_1$" drops significantly on the CVUSA dataset. The synthesized images of "Ours w/o $\mathcal{L}_1$" on the CVUSA dataset are very blurry, suggesting that the $\mathcal{L}_1$ loss is responsible for producing sharper images.

Height Estimation. The ablation labelled as "Ours w/o height (projection)" is a variant of our method with a single MPI plane ($N = 1$) corresponding to the ground plane. Generally, "Ours w/o height (projection)" recovers the ground structure of a scene from a satellite image, while scene objects with heights higher than the ground plane are hallucinated. This phenomenon can be easily observed from Fig. 10. As shown in the two examples, the trees annotated in the satellite images have similar colors to their surrounding region. "Ours w/o height (projection)" does not recognize the trees and the corresponding regions in the synthesized images are in-painted with grass. In contrast, our whole pipeline with height estimation successfully distinguishes objects with different heights and the generated images are more geometrically consistent with the input satellite images. Fig. 11 presents some examples on a special case where the input satellite images are of low quality. As "Ours w/o height (projection)" tries to hallucinate the scenes for objects with heights higher than the ground plane, its generalization ability on these low quality input images is poor and the generated images have many artifacts. Instead, our whole pipeline with the guidance of height estimation has more geometric clues, and thus the synthesized images are more natural.

In Fig. 9, we provide some additional qualitative results of our method, including the estimated height maps and the projected images from the S2SP module. We crop the height maps to the inscribed circle, since information outside that region is never used and receives no supervision. Fig. 12 provides more examples of estimated height maps for complex scenes. We highlight some regions of interest with rectangles, especially areas with greater heights.

The estimated height maps are coarse and sometimes inaccurate, due to the lack of explicit height supervision (with objects behind occluders having no supervision at all) and the use of an approximated camera height. However, misestimated heights can be tolerated by our subsequent generator network. Generally, our height estimation block is generally able to learn the statistical height distribution of different types of objects, e.g., trees are relatively higher

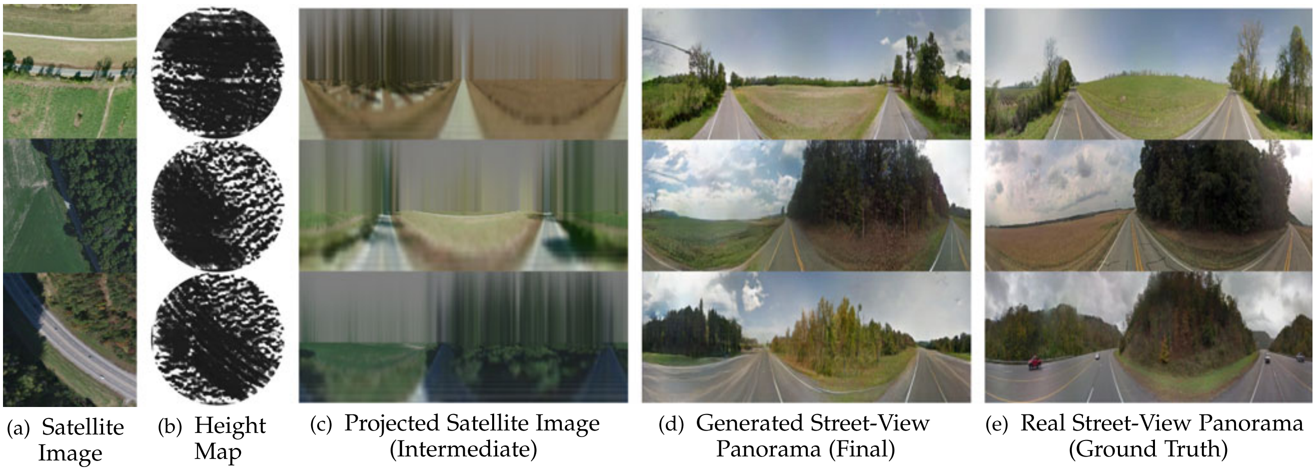

(a) Satellite Image
(b) Height Map
(c) Projected Satellite Image (Intermediate)
(d) Generated Street-View Panorama (Final)
(e) Real Street-View Panorama (Ground Truth)

Fig. 9. Additional qualitative results with height maps (lighter is higher), projected satellite images generated by our S2SP module, and synthesized images generated by our entire pipeline.

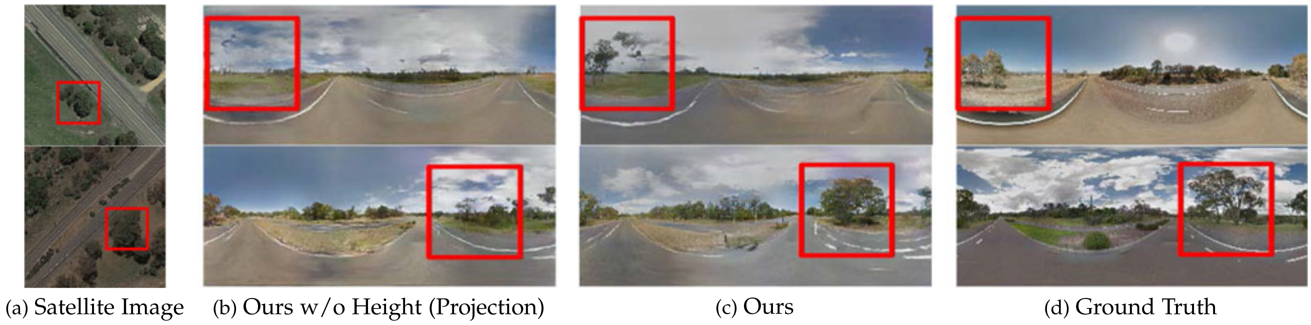


Fig. 10. Qualitative comparison of images synthesized by our method with learned height maps ('Ours') and with height maps fixed to zero ('Ours w/o Height (Projection)').

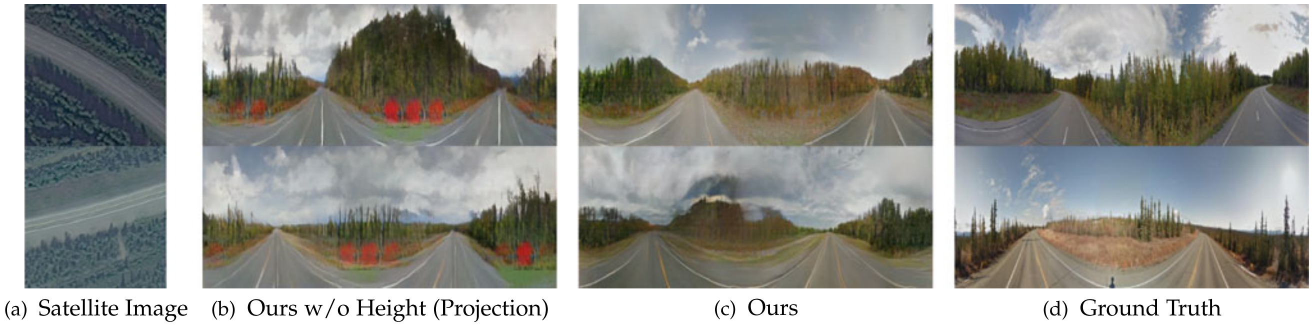


Fig. 11. Qualitative comparison of images synthesized from low-quality satellite images.

than roads. The camera heights in the CVUSA and CVACT are approximated as 2 meters in our implementation.

Influence of the Number of Height Planes. We investigate the performance of our method with different numbers of planes N in the satellite-view MPI. The results are presented in Table 4. While increasing the number of height planes above one increases performance, further increases do not provide significant improvements. This may be because there is no explicit height supervision for the height estimation block and so the network learns the statistical height distribution of particular objects. Therefore, beyond a certain point, the density of sampling along the height dimension may not make a difference. We expect that a larger value of N would be more advantageous when height supervision is available.

7.6 Other Satellite to Street-View Image Projection Methods

In our S2SP module, we convert the satellite-view MPI to a street-view MPI by unrolling and stretching the concentric cylinders. In this section, we propose and discuss another satellite to street-view image projection method: directly projecting each of the image planes in the satellite-view MPI to the street viewpoint (from Figs. 13a, 13b and 13c).

Let (u, v, z) denote the source coordinates of points in the image plane of the satellite-view MPI (z is constant for each plane), and (θ, ϕ) denote the coordinates of panorama projection rays viewed at the ground level. The projection between source and target coordinates for each plane in Fig. 13a can be expressed by Eq. 3. As indicated by the equation, for fixed θ, the distance between the street-view camera

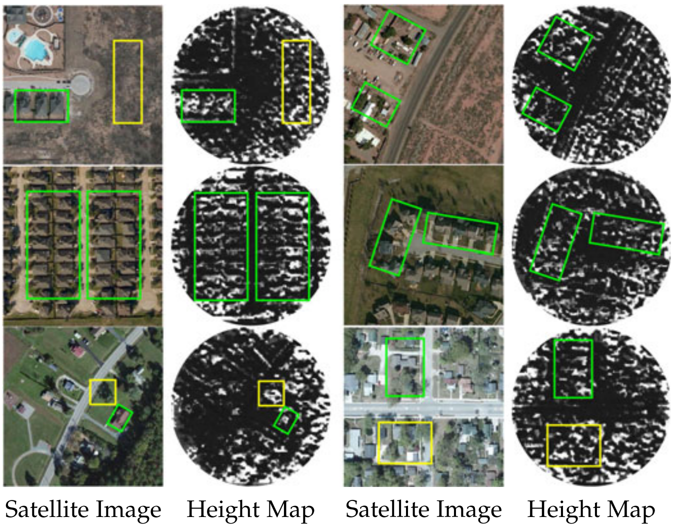


Fig. 12. Additional estimated height maps for complex scenes. Regions with greater heights (i.e., buildings and trees) are annotated with rectangles (buildings as green and trees as yellow).

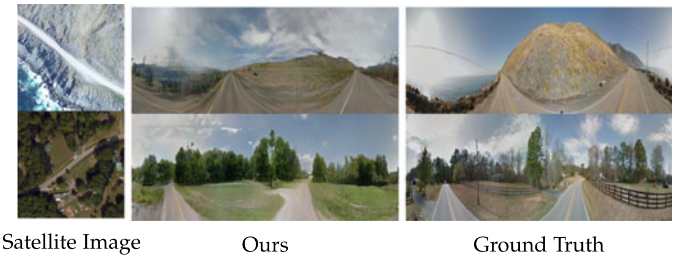


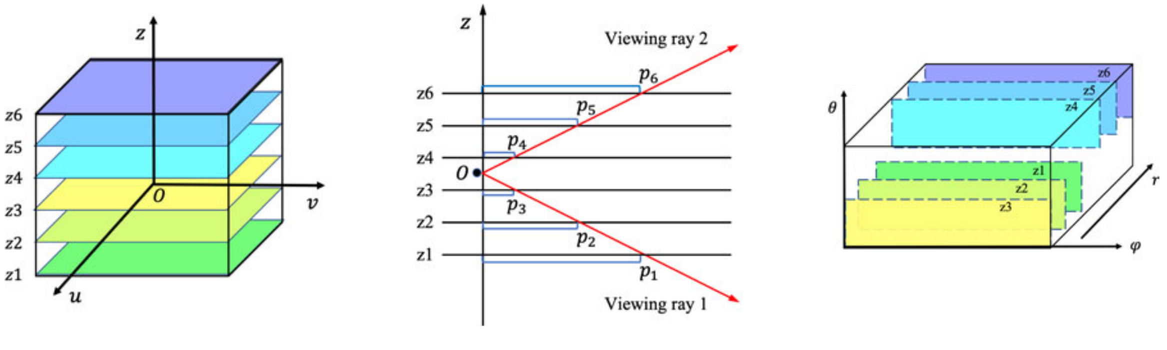
Fig. 14. Failure cases. First row: the height of the hill is hard to estimate accurately from a single satellite image. Second row: the fence is difficult to be visible from above and is ignored.

and a 3D point (θ, ϕ, z) in the target coordinates increases as z increases for $z > 0$ and decreases as z increases for $z < 0$. Fig. 13b provides an intuitive illustration. The projected points from plane $z3$ will be nearer than those from planes $z1$ and $z2$, and the projected points from plane $z4$ will be nearer than those from planes $z5$ and $z6$. All points with $z < 0$ will be projected to the bottom half of the street-view panorama and those with $z > 0$ will be imaged to the top half. Therefore, simply adjusting the order of projected planes will help to sort points from far to near along viewing directions, as shown in Fig. 13c. After that, Equation 7 can be adopted to render the projected street-view panorama. Note that the projected volumetric scene representation in Fig. 13c is not an MPI since the depth of image planes are not uniformly sampled.

We denote this projection as “height-wise” and our MPI-to-MPI projection as “depth-wise”. Furthermore, the projected volumetric scene representation, the street-view MPI in Fig. 5 and the reordered planes in Fig. 13c, can be employed directly as an input to the generator instead of the rendered image. Therefore, we investigate the performance of different satellite to street-view image projection methods with different network conditions. The results are presented in Table 5. As indicated by the results, there is negligible difference in performance between the different approaches, while all of them outperforms existing methods (results presented in Table 1). This demonstrates that establishing the geometric correspondences between the two view images is indeed useful for the satellite to street-view image synthesis. For the difference among different projection methods, using volumetric generator inputs requires slightly more model parameters and longer training time. For the sake of better interpretability, we select the “depth-

TABLE 4
Performance Comparison as the Number of Planes N Varies on the CVACT (Aligned) and the CVUSA Dataset

N	Params	CVACT (Aligned)				CVUSA			
		RMSE↓	SSIM↑	PSNR↑	SD↑	RMSE↓	SSIM↑	PSNR↑	SD↑
1	33.46M	49.08	0.4068	14.48	16.18	54.49	0.3301	13.63	16.11
16	33.62M	**47.89**	0.4178	**14.71**	16.29	53.73	0.3316	13.76	16.21
32	33.62M	48.91	0.4140	14.52	**16.40**	54.58	0.3249	13.61	16.10
64	33.63M	48.23	**0.4212**	14.65	16.33	**53.67**	**0.3408**	**13.77**	**16.27**



(a) Satellite-view MPI (b) Viewing rays (c) Plane reorder

Fig. 13. “Height-wise” satellite to street-view image projection method. We directly project each of the image planes in an satellite-view MPI (a) to the street viewpoint, and then adjust the order of projected planes according to the principle illustrated in (b). The result is a volumetric scene representation that sorts points from far to near along viewing rays (c).

TABLE 5
Performance Investigation With Different Satellite to Street-View Projection Approaches on the CVACT (Aligned) Dataset

		Params	CVACT (Aligned)				CVUSA			
			RMSE↓	SSIM↑	PSNR↑	SD↑	RMSE↓	SSIM↑	PSNR↑	SD↑
height-wise	volume	33.82M	48.00	**0.4231**	14.68	16.35	53.80	0.3276	13.74	16.18
	image	33.63M	**47.72**	0.4134	**14.73**	16.17	54.15	0.3301	13.69	16.12
depth-wise	volume	33.82M	48.23	0.4222	14.64	**16.40**	53.93	0.3337	13.73	16.17
	image	33.63M	48.23	0.4212	14.65	16.33	**53.67**	**0.3408**	**13.77**	**16.27**

wise” projection and use the rendered image as the generator input.

8 Discussion and Limitations

For image generation tasks, skip connections between the encoder and decoder of a UNet-type generator can be useful for recovering fine detail. However, their presence or absence made no difference to the performance of our method and the baseline methods (i.e., Pix2Pix and XFork). For the baselines, this may be due to the significant differences in satellite and street-view image modalities and resolutions. Our approach partially alleviates these differences with the S2SP module, but skip connections contribute to the final performance marginal.

The main limitation of our method is that it hallucinates the façades of objects since a satellite image only views their top surfaces. This can be seen in the trees of the second row of Fig. 8b. In fact, this is a common limitation for all the existing satellite to street-view synthesis approaches. Given a satellite image, there are many possibilities regarding the street-view appearance of objects (e.g., various building façade structures, different colors, and different seasons). It would be interesting to reformulate the task as a one-to-many problem, with a latent vector encoding the desired property in the target view image. Our method also fails when the heights are misestimated, such as the hill in the first row being taller than expected, and when objects are too small (indistinguishable) in the satellite images, such as the fence in the second row of Fig. 14.

Another limitation is the rendering speed: our approach takes 0.2s to synthesize one image, whereas Pix2Pix and XFork only take 0.02s. For time-critical applications, the baseline methods (Pix2Pix and XFork) could be used when the distribution of cross-view image pairs is uniform (e.g., the variation of scene structures at different locations is small), since the geometric correspondences can be learned statistically during training. However, our model (with slower rendering) is needed when the variation of scene structures is large, since the inductive bias encoded in our framework makes it easier for the network to learn the significant cross-view transformations.

9 Conclusion

In this paper, we have proposed a novel geometry-aware method for synthesizing street-view panoramas from satellite images. In contrast to existing methods that adopt black-box training procedures, our algorithm explicitly establishes the geometric correspondences between satellite and street-view images. The central innovation of this paper is the novel and differentiable satellite to street-view image projection module, which exploits the two-view geometry of the setup for accurate and occlusion-aware image synthesis. Quantitative and qualitative experimental results demonstrate that our method is able to generate geometrically-consistent street-view panoramas from satellite images. We expect the key idea of this paper (explicitly exposing the geometric correspondences and implicitly estimating the height map) to be useful for solving other related problems, such as estimating the height/depth maps of satellite/street-view images and novel view synthesis given two-view image pairs.

Acknowledgments

The authors would like to thank AE and all reviewers for their time and patience in reviewing this paper and their constructive suggestions.

References

[1] K. Regmi and M. Shah, “Bridging the domain gap for ground-to-aerial image matching,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 470–479.

[2] A. Toker, Q. Zhou, M. Maximov, and L. Leal-Taixé, “Coming down to earth: Satellite-to-street view synthesis for geo-localization,” 2021, *arXiv:2103.06818*.

[3] T. Porter and T. Duff, “Compositing digital images in SIGGRAPH comput,” *ACM SIGGRAPH Comput. Graph.*, vol. 18, no. 3, pp. 253–259, 1984.

[4] X. Lu, Z. Li, Z. Cui, M. R. Oswald, M. Pollefeys, and R. Qin, “Geometry-aware satellite-to-ground image synthesis for urban areas,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 859–867.

[5] M. Liu, X. He, and M. Salzmann, “Geometry-aware deep network for single-image novel view synthesis,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4616–4624.

[6] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, “Stereo magnification: Learning view synthesis using multiplane images,” *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–12, 2018.

[7] J. Flynn *et al.*, “Deepview: View synthesis with learned gradient descent,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2367–2376.

[8] R. Tucker and N. Snavely, “Single-view view synthesis with multiplane images,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 551–560.

[9] M. Zhai, Z. Bessinger, S. Workman, and N. Jacobs, “Predicting ground-level scene layout from aerial imagery,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4132–4140.

[10] K. Regmi and A. Borji, “Cross-view image synthesis using conditional GANs,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3501–3510.

[11] K. Regmi and A. Borji, “Cross-view image synthesis using geometry-guided conditional GANs,” *Comput. Vis. Image Understanding*, vol. 187, 2019, Art. no. 102788.

[12] H. Tang, D. Xu, N. Sebe, Y. Wang, J. J. Corso, and Y. Yan, “Multi-channel attention selection GAN with cascaded semantic guidance for cross-view image translation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2417–2426.

[13] S. Workman, R. Souvenir, and N. Jacobs, "Wide-area image geolocalization with aerial reference imagery," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3961–3969.

[14] S. Workman and N. Jacobs, "On the location dependence of convolutional neural network features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2015, pp. 70–78.

[15] N. N. Vo and J. Hays, "Localizing and orienting street views using overhead imagery," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 494–509.

[16] S. Hu, M. Feng, R. M. H. Nguyen, and G. Hee Lee , "CVM-Net: Cross-view matching network for image-based ground-to-aerial geo-localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7258–7267.

[17] B. Sun, C. Chen, Y. Zhu, and J. Jiang, "GeoCapsNet: Aerial to ground view image geo-localization using capsule network," 2019, *arXiv:1904.06281*.

[18] S. Cai, Y. Guo, S. Khan, J. Hu, and G. Wen, "Ground-to-aerial image geo-localization with a hard exemplar reweighting triplet loss," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8390–8399.

[19] L. Liu and H. Li, "Lending orientation to neural networks for cross-view geo-localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5617–5626.

[20] Y. Shi, X. Yu, L. Liu, T. Zhang, and H. Li, "Optimal feature transport for cross-view image geo-localization," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11990–11997.

[21] Y. Shi, L. Liu, X. Yu, and H. Li, "Spatial-aware feature aggregation for image based cross-view geo-localization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 10090–10100.

[22] Y. Shi, X. Yu, D. Campbell, and H. Li, "Where am i looking at? Joint location and orientation estimation by cross-view matching," 2020, *arXiv:2005.03860*.

[23] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[25] D. P. Kingma and J. Ba, "Adam: A methodfor stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, vol. 9, 2015.

[26] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 586–595.

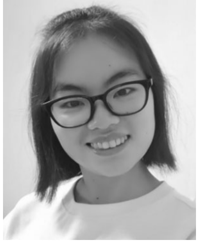
[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[28] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and$<$ 0.5 MB model size," 2016, *arXiv:1602.07360*.

[29] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3213–3223.

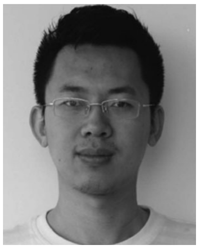
[30] V. Nekrasov, C. Shen, and I. Reid, "Light-weight refinenet for real-time semantic segmentation," 2018, *arXiv:1810.03272*.

[31] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.



Yujiao Shi received the BE and MS degrees in automation from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2014 and 2017, respectively. She is currently working toward the PhD degree with the College of Engineering and Computer Science, Australian National University. Her research interests include satellite image based geo-localization, novel view synthesis, and scene understanding.

Dylan Campbell received the BE degree in mechatronic engineering from the University of New South Wales and the PhD degree in engineering from Australian National University in 2018. He was a research assistant with Cyber-Physical Systems Group, Data61–CSIRO. He is currently a research fellow with Visual Geometry Group, University of Oxford. He was a research fellow with the Research School of Computer Science, Australian National University and the Australian Research Council Centre of Excellence in Robotic Vision. His research interests include computer vision and machine learning topics, including visual geometry, and differentiable optimization.



Xin Yu received the BS degree in electronic engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2009, the PhD degree with the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2015, and the PhD degree with the College of Engineering and Computer Science, Australian National University, Canberra, Australia, in 2019. He is currently a senior lecturer with the University of Technology Sydney. His interests include computer vision and image processing.

Hongdong Li is currently a professor with ANU and the founding chief investigator for the Australia Centre of Excellence for Robotic Vision. Before 2010, he was with NICTA working on the "Australia Bionic Eyes" project. His research interests include 3D vision reconstruction, structure from motion, multiview geometry, and applications of optimization methods in computer vision. He is an associate editor for *IEEE Transactions on Pattern Analysis and Machine Intelligence*, the guest editor for *International Journal of Computer Vision*, and the area chair of ICCV, ECCV, and CVPR conferences. He was the program chair for Australia Conference on Robotics and Automation 2015 and the program co-chair for Asian Conference on Computer Vision 2018. He was the recipient of a number of paper awards in computer vision and pattern recognition, the CVPR 2012 Best Paper Award, the ICCV Marr Prize Honorable Mention in 2017, and a shortlist of the CVPR 2020 Best Paper Award.