MixedViewDiff - A Mixed View based Diffusion Model for Panorama Synthesis

Albert Peng[1] David Wang[1] Dijsktra Liu[1] Sunny Yuan[1]

[1]Washington University in St. Louis

Abstract

The task of panorama synthesis often suffers from challenges related to spatial and texture inconsistencies due to sparse or misaligned data sources. Prior work addressed these issues using nearby panoramas [12] or by estimating depth from satellite images [4], but faced limitations in handling structure-texture mismatches. In this work, we propose a novel approach that leverages depth estimation from satellite imagery to extract accurate structural information and style/texture guidance from nearby panoramas. Our approach synthesizes target-location panoramas by jointly learning structural consistency through satellite and panoramas mix viewed-based depth estimation and texture consistency through nearby street-level panoramas. By combining these complementary inputs, our approach maintains geometric faithfulness while preserving texture style across locations, ensuring both spatial and texture alignment in synthesized panoramas. This method overcomes the limitations of previous models, by ensuring both spatial and stylistic fidelity in diverse environments.

1. Introduction

The increased availability of street-level panoramas within map and navigation services has been a crucial advancement in the digital age. However, captures of such panoramas across locations can often be difficult and costly. Thus, *cross-view synthesis* arose naturally as a task that seeks to recover street-level panoramas from corresponding satellite images. Such works have been explored using geometry techniques [8], convolutional neural networks [13], and diffusion models [4].

Recently, [12] proposed *Mixed-view synthesis* as shown in Fig. 1: a task that uses nearby street-level panoramas as additional inputs to the original cross-view problem. This fits naturally into the status quo of panorama-availability, as image coverage is dense but additional interpolation images could be useful for users during navigation. In particular, [12] uses a diffusion-based background to generate imagery, guided by additional information extracted through attention models.

2. Related Work

2.1. ControlNet

Stable Diffusion [7] is renowned for its strong capabilities in generating high-quality images with guidance. To make controlling the diffusion model more versatile, ControlNet [14] was introduced to fine-tune Stable Diffusion for task-specific conditioning. In this work, we implemented ControlNet to achieve street view panorama synthesis.

Typically, Stable Diffusion accepts textual prompts as the sole control over the generation process. With ControlNet, additional controls can be incorporated without retraining the entire Stable Diffusion model. These controls can include drawing sketches, depth maps, or even high-dimensional features extracted by pretrained models. ControlNet enhances the image generation process by tuning the U-Net of Stable Diffusion. Specifically, it replicates the structure and parameters of the original encoder blocks and the middle block of Stable Diffusion, using them as a deep, robust, and powerful backbone network to learn diverse control conditions for image generation. By adding trainable layers, ControlNet learns to influence image synthesis, enabling stronger and task-specific guidance.

Notably, this process does not require retraining the Stable Diffusion model, which was trained on a significantly larger dataset (approximately 5 billion images using the LAION-5B dataset). Despite the added layers, the required GPU memory for generation does not significantly exceed that of the original model, allowing us to fully utilize the generative capabilities of Stable Diffusion while maintaining computational efficiency.

2.2. Cross-view and Mixed-view Synthesis

We take heavy inspiration from [12] and [4] and hope to achieve desirable results on mixed-view synthesis with a more efficient architecture. We noticed that in cross-view synthesis tasks, some of the biggest obstacles lie in localizing and recreating details of roads and buildings viewed

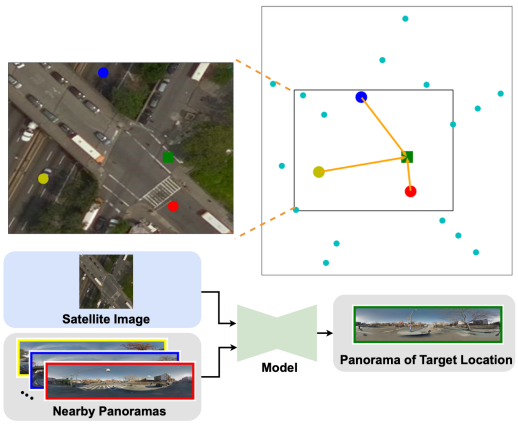


Figure 1. Mixed-view panorama synthesis proposed in [12]

from a satellite. Many works focus on different depth estimation and texturing techniques to solve this problem. Given the additional information from nearby street-level panoramas, we focus on efficient extraction mechanisms of road and building details. For example, with the height information available from the satellite image, we can estimate the projection structure view as the additional information guiding the generation of panoramas.

3. Approach

An initial high-level overview is presented in Fig. 2, which incorporates structure estimations with nearby panoramas. A geo-spatial attention encoder is introduced to compute the cross attention between the positional encoded target with near structures and panoramas, which helps to embed the input conditions. The detail implementation of geo-spatial attention encoder is introduced in the later part.

3.1. Data

Our dataset is based on the complete regions of Brooklyn and Queens [9, 10], with specific modifications to meet our project's requirements. For each street-view coordinate, we created a 512 × 512 px image centered on this location by merging all intersecting overhead satellite images and height maps. The merged images were then cropped to a 512 × 512 px size, as illustrated in Fig. 3. This process allows for more comprehensive structural estimates, as larger satellite coverage facilitates better structural estimation for objects further from the camera. For comparison, a 256 × 256 satellite image would be insufficient to capture distant structures. The corresponding height labels were also merged and cropped to 512 × 512 px, serving as ground truth for structural estimation.

Since overhead satellite images are not always captured continuously, some of the merged images contain gaps or blank areas. To address this, we calculated the coverage rate of each satellite image and filtered out images with low coverage. This ensured that only high-quality images were retained for the dataset.

During model training, multiple street-view panoramas were utilized within each satellite image. One panorama was selected as the target, while the others served as conditioning images, along with their corresponding structural estimations.

3.2. Data Preprocessing

As previously described, the satellite images were stitched and cropped to create the 512 × 512 px images. We calculated the coverage rate of these stitched images and visualized it by smoothing the frequency histogram using Kernel Density Estimation (KDE). As shown in Fig. 4 (a), the majority of the data had a coverage rate above 0.8. Consequently, we filtered out images with coverage below this threshold to maintain data quality. Additionally, we analyzed the relationship between coverage and geographic location, focusing on data from Brooklyn, as illustrated in Fig. 4 (b). The analysis revealed that low-coverage images predominantly occurred at the boundaries of the collection area. Apart from this geographic factor, there were no other significant coverage patterns. As a result, removing low-coverage images is unlikely to impact the model's generalization ability.

Finally, street-view images located within the same satellite image were grouped together along with their estimated structural data. This approach allows the data loader to read all input elements and the corresponding target from a single directory, facilitating efficient batch processing during training.

3.3. Structure Information

Data Preparation. To estimate the structure, we started by loading the height data in .npz format and represent them as a two-dimensional numpy array, where each element corresponds to a height value at a specific geographic coordinate. The geographic extent of the dataset is defined by two diagonal points that establish the boundary boxes of the satellite image, enabling accurate mapping of geographical coordinates to array indices.

Camera Position Transformation. The camera position, provided in geographical coordinates (latitude and longitude), is transformed into pixel indices within the height data array. This transformation is done by calculating the pixels per degree of latitude and longitude based on the total number of pixels in the image's height and width and the geographical extents it covers. This conversion ensures accurate positioning of the camera within the height data array we get in the previous step.

Panorama Generation. The core of the methodology

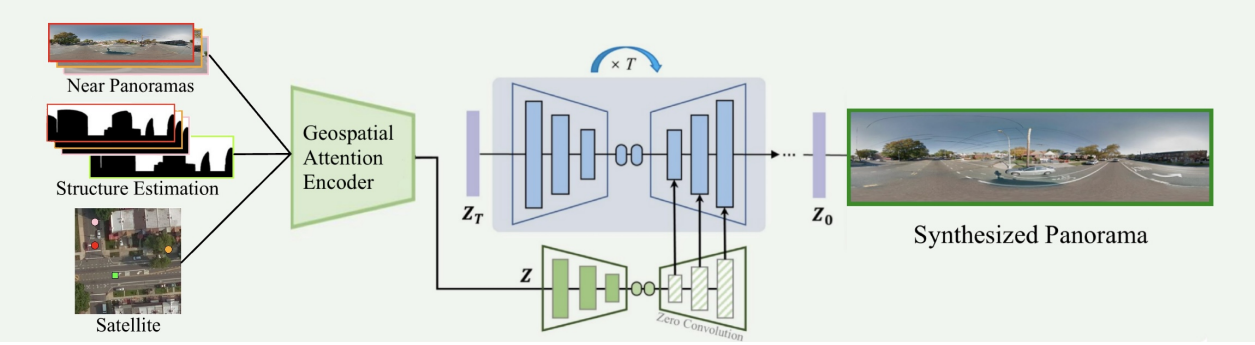


Figure 2. High-level overview of the our MixedViewDiff based on the structure of CrossViewDiff [4])

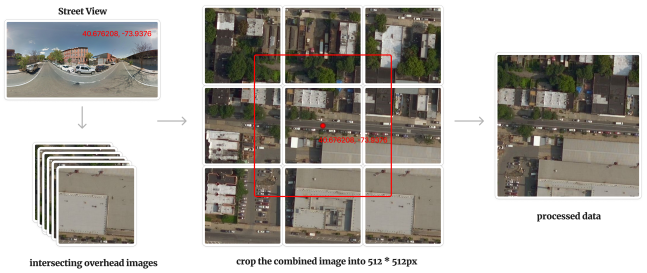
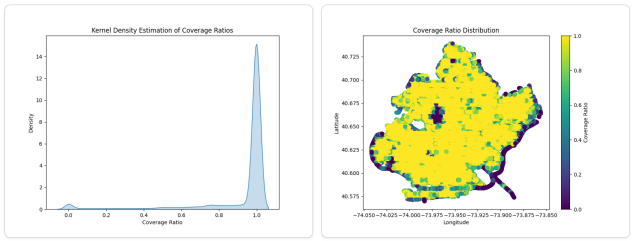


Figure 3. The data processing pipeline



a) Kernel Density Estimation of Coverage Ratios b) Geographical Distribution of Coverage Ratios

Figure 4. coverage ratios distribution

is the generation of a 360-degree panoramic image. The panorama is constructed by iterating through each degree of a full circle, adjusting for any initial rotation specified. For each degree, a ray extends from the camera position in the direction determined by the current angle. Heights are sampled along this ray until the edge of the image is reached.

To simulate perspective, a distance-based scaling factor is applied to each sampled height value. This factor diminishes the perceived height of objects as their distance from the camera increases, mimicking the way distant objects appear smaller to the human eye. The perspective effect is governed by the equation $\frac{1}{1+kd}$, where k is a tunable parameter that controls the rate of size reduction with distance, and d is the distance from the camera to the object.

Visualization. The final panorama is constructed by mapping the maximum observed heights, after perspective scaling, to a gray scale image where each column corresponds to a different angle of view and the height values fill from the bottom of the image.

Additionally, we add a constant sky space to our output panorama and apply a Gaussian filter to smooth the heights of the final array of maximum heights. We also further scale the dimensions of the output structure map to be the same as the street view panorama images. The goal is to make our final output look as close as the corresponding street view image, which will help improve accuracy in future steps.

Our approach is able to achieve accurate structure prediction based on depth/height input, as shown in Fig. 5. Comparing with previous approach that generates 3d voxel from depth images then projects voxel information to 2d structure map [12], our approach significantly simplifies the process and reduces computation time.

3.4. Model Building

3.4.1 Initial ControlNet Setup (Overhead Only)

We implemented an initial pipeline using the official ControlNet code base [2]. The setup simply used concatenated 512×512 satellite images as conditional inputs to ControlNet, with a constant text prompt, "A street view panorama." We tested Stable Diffusion v1-5 integrated with ControlNet, successfully establishing a multi-GPU environment for training. However, despite running for 3 days on 2 GPUs, the naive training process did **not** show convergence (see Fig. 8). This outcome prompted us to design an advanced embedding strategy that incorporates structural, positional, and satellite information for improved generating performance.

Figure 5. Visual demonstration of structure map. *Top.* Expected street view panorama image. *Bottom.* Output structure map generated from depth.

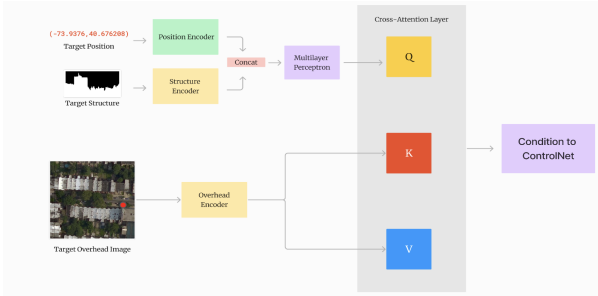


Figure 6. Our proposed Cross-view Model Architecture

3.4.2 Cross-view Control

Figure 6 demonstrates our proposed architecture to better leverage information provided by the overhead image. In particular, we use a pre-trained ResNet-18 [1] as a feature encoder to structure depth map and target overhead image. This approach effectively encodes information while also enabling the model to converge more quickly.

In order to incorporate positional information into our cross-view model, we use a position encoder similar to NeRF [5]. Given longitude and latitude coordinate information $p = (x, y)$, we use high-frequency sine and cosine encoder γ defined by

$$\gamma(p) = \big(\sin(2^0\pi p), \cos(2^0\pi p), ..., \sin(2^{L-1}\pi p), \cos(2^{L-1}\pi p)\big). \quad (1)$$

The encoded positional information is added onto the encoded structural information, and provided to a multi-headed attention as query. Meanwhile, the encoded features of the target overhead image is given to the multi-headed attention layer as the key and value. The output of the attention layer is provided to ControlNet as a condition, with information of structural, position, and target embedded.

3.4.3 Mixed-view Control

Expanding upon overhead image and structure, we leverage a wider range of structural and textural information using relative positions and panoramas of nearby locations, as shown in Figure 7. The query input to the multi-headed attention is maintained as a combination of encoded structure and positional encoding. We encode nearby structures using the same pre-trained ResNet-18 network and combine them with the corresponding positional encodings, following the same approach as described above. The resulting features are then used as keys for the attention layer. Features of panoramas from nearby locations are then encoded in another pre-trained ResNet-18 network and used as values to the multi-headed attention. This enhanced attention module is able to effectively extract structural information from nearby locations to better inform our model.

The original target overhead image is concatenated to the attention module's output, and passed through a fully-connected layer. This final module combines information extracted from the nearby panoramas with the overhead image, and provides this is a condition to the ControlNet.

4. Results

The Cross-view and Mixed-view models were trained on a dataset of 40,920 and 20,460 data points respectively. We used two A100 GPUs over the course of several days. Specifically, the Cross-view model converged in 2 days, while the Mixed-view model required 4 days to converge.

4.1. Qualitative and Visual Evaluation

The performances of our Cross-view diffusion model and Mixed-view diffusion model on testing datasets are shown in Figure 9 and 10 respectively. Additionally, Figure 11 shows the training result, which will be used to evaluate the over-fitting of the model. **Note**: The datasets for cross-view and mixed-view tasks were constructed differently, so the we did not used the same target images for comparison. Generally, the number of data points available for the cross-view task is significantly larger than for the mixed-view task. This is because several images in the mixed-view task are utilized as near panoramas and cannot be used as targets to prevent data leakage.

We observed that both models captured the structure and some texture of the ground truth images to some extend. The generated images' structure clearly align with

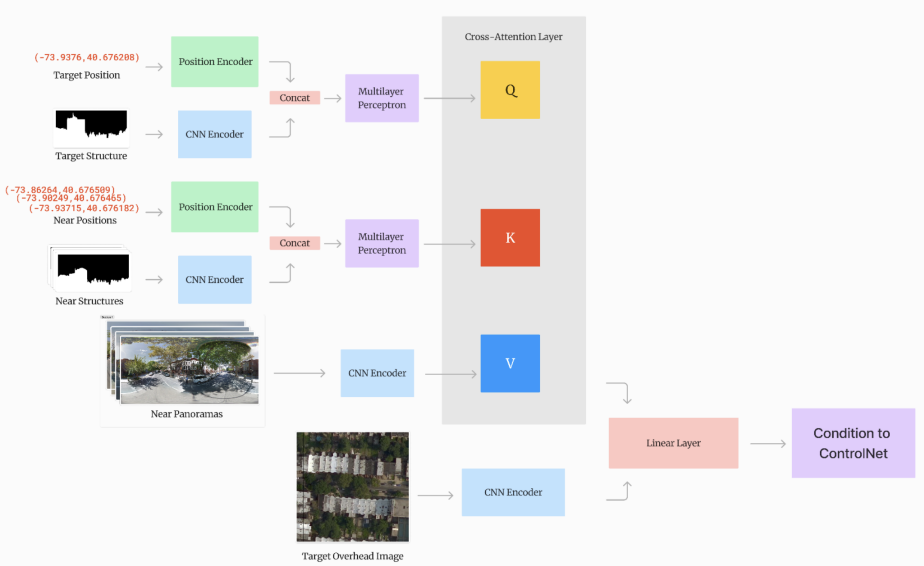


Figure 7. Our proposed Mixed-view Model Architecture

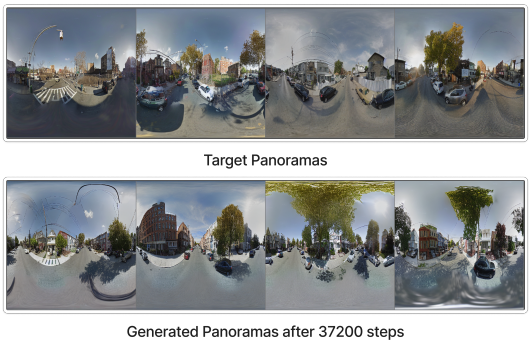


Figure 8. Naive Model Training Result

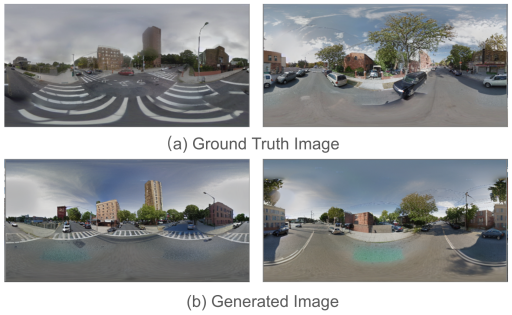


Figure 9. Cross-View Model Testing Result

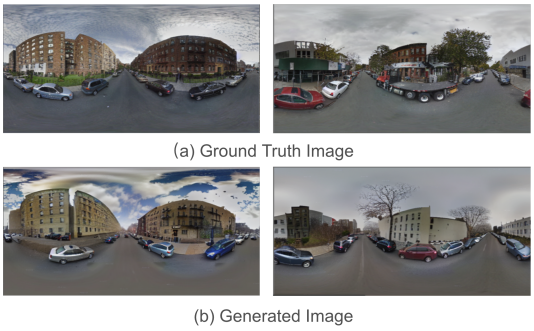


Figure 10. Mixed-View Model Testing Result

the ground truth, particularly in the representation of architectural features and trees. Mixed-view model better captures the texture details like windows than the cross-view model did. While the colors are not aligned quite well, but Mix-view model is still comparably better compare with the Cross-view model in terms of texture capturing.

However, further refinement is needed in areas such as sky color, the number and size of trees, and the appearance of the road. The visual result of the training set outperform the testing result especially in term of the texture and color captured. These discrepancies may be attributed to overfitting and not large enough dataset for this relatively large mixed-view model.

4.2. Quantitative Evaluation

We evaluate our results by comparing ground truth panorama images with our generated images after scaling both of them to 512 $\times$ 1024 px on 500 generated samples. We calculated metrics including PSNR and SSIM scores. The results are displayed in Table 1.

Our Cross-view diffusion model and Mixed-view diffusion model achieve similar PSNR and SSIM scores, with the PSNR score of Mixed-view diffusion model slightly higher than Cross-view diffusion model. Both models out-

(a) Ground Truth Image

(b) Generated Image

Figure 11. Mixed-View Model Training Result

Table 1. Comparison of baseline methods to our method. Note that we use different datasets than baseline methods, so the results are not directly transferable.

Method	**PSNR**(↑)	**SSIM**(↑)
Pix2pix [3]	11.93	0.0950
PanoGAN [11]	13.10	0.2981
Sat2density [6]	13.39	0.4325
GeoDiffusion [12]	14.14	0.4329
CrossDiff **(Ours)**	12.57	0.4116
MixedDiff **(Ours)**	12.59	0.4113

perform Pix2pix [3], with SSIM scores close to state-of-the-art Sat2density [6] and GeoDiffusion [12]. Note that we did not run the other baselines. We directly used their proposed performance metrics, but we used the same approach to do the evaluation.

5. Discussion and Conclusions

Progressing from the Overhead-only model to Cross-view and ultimately to Mixed-view model, our results show a gradually better performance with more information being incorporated when generating street view images.

The Mixed-view diffusion model has show excellent results on the training set, showing the ability to learn from near panoramas and panorama structure information, and other details like trees or bridges. This success is attributed to the information provided by the extra surrounding panoramas. However, there is still room for improvement in addressing the issue of overfitting, where we see there is a great discrepancy between the testing and training result. Currently, the model is relatively too large and complex, a which is a 10 GB model, with a parameter size that is disproportionately large compared to the training set data, leading to an excessive focus on the training set details and thus causing over-fitting. Future improvements should focus on simplifying the model to avoid overfitting.

6. Statement of Individual Contribution

Please see the submitted zip file for source code.

6.1. Albert Peng

1. Setup controlNet pipeline
2. Design and debug the condition embeddings
3. Code contribution: cldm.py, mix_view_diff.py

6.2. David Wang

1. Designed and implemented mixed-view diffusion model.
2. Ran training trials.
3. Data pre-processing: generate structures images on large scale, restructure dataset according to our need.
4. Code: cldm.py, mix_view_diff.py, dataloader.py, generate_target.py

6.3. Dijsktra Liu

1. Designed and implemented mixed-view diffusion model.
2. Implemented the data processing pipeline.
3. Created scripts for data analysis
4. Code: cldm.py, mix_view_diff.py, concat.py, find_boxes.py

6.4. Sunny Yuan

1. Designed and implemented structure maps from depth images pipeline.
2. implemented the position encoder.
3. Optimize structure maps for training.
4. Code: evaluation.ipynb, npz_to_structure.ipynb, mix view diff.py

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[2] Illyasviel. Controlnet. https://https://github.com/lllyasviel/ControlNet, 2022. Accessed: 2024-10-15. 3

[3] 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

[4] Weijia Li, Jun He, Junyan Ye, Huaping Zhong, Zhimeng Zheng, Zilong Huang, Dahua Lin, and Conghui He. Crossviewdiff: A cross-view diffusion model for satellite-to-street view synthesis, 2024. 1, 3

[5] 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, 2020. 4

[6] Ming Qian, Jincheng Xiong, Gui-Song Xia, and Nan Xue. Sat2density: Faithful density learning from satellite-ground image pairs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3683–3692, 2023. 6

[7] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1

[8] Yujiao Shi, Dylan Campbell, Xin Yu, and Hongdong Li. Geometry-guided street-view panorama synthesis from satellite imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):10009–10022, 2022. 1

[9] Scott Workman, Muhammad Usman Rafique, Hunter Blanton, and Nathan Jacobs. Revisiting near/remote sensing with geospatial attention. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 2

[10] Scott Workman, Menghua Zhai, David Crandall, and Nathan Jacobs. A unified model for near and remote sensing. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. Acceptance rate: 28.9%. 2

[11] Songsong Wu, Hao Tang, Xiao-Yuan Jing, Haifeng Zhao, Jianjun Qian, Nicu Sebe, and Yan Yan. Cross-view panorama image synthesis. *IEEE Transactions on Multimedia*, 25:3546–3559, 2022. 6

[12] Zhexiao Xiong, Xin Xing, Scott Workman, Subash Khanal, and Nathan Jacobs. Mixed-view panorama synthesis using geospatially guided diffusion, 2024. 1, 2, 3, 6

[13] 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. 1

[14] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, pages 3836–3847, 2023. 1