GOS: A LARGE-SCALE ANNOTATED OUTDOOR SCENE SYNTHETIC DATASET

Mingye Xie, Ting Liu, Yuzhuo Fu

School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China

ABSTRACT

Scene editing has attracted increasing research interests owing to its valuable applications in the field of photography and entertainment. With style-based GAN being proposed, images can be reasonably edited on specific semantics by manipulating in latent space of the generator. However, existing datasets cannot satisfy the demands of large amounts of diverse data and rich semantic annotations at the same time, which makes the existing method difficult to edit on the content of outdoor scene images. To address these problems, we propose a large-scale, diverse synthetic dataset called "GOS dataset" generated based on a video game, which contains fine-grained semantic annotations. Extensive experiments show that utilizing the features obtained from the annotations of our dataset achieves better performance in outdoor scene editing, especially for distance and viewpoint of scenes, which indicates the extracted features have a certain generalization capability.

***Index Terms*—** Scene editing, synthetic data generation, generative adversarial network (GAN)

1. INTRODUCTION

We live in a colorful world with diverse scenes changing continuously, and there is a natural need to edit the photos of the scenery being taken. Cameramen will try to correct lens distortion and perspective distortion that appears in the photograph. Entertainment users want to experience rotating and zooming objects of scenes in Augmented Reality (AR).

Early methods of scene editing simply modify contrast, brightness, and saturation of images [1], or change style based on CNN [2]. Generative Adversarial Networks (GANs) [3] have achieved astonishing success in realistic image synthesis. There appears many GAN-based methods on scene editing, such as image-to-image translation [4], 3D-aware scene manipulation [5], image synthesize [6].

Style-based generators [7, 8] can better learn the distribution of training data and make latent space less entangled. By identifying feature vectors in latent space and attach to human-understandable semantics, we can move the latent code towards a certain direction, changing the semantics in the synthesized image without re-training the generator. Recent works use the method to achieve camera movements [9], face editing [10], and indoor scene editing [11].

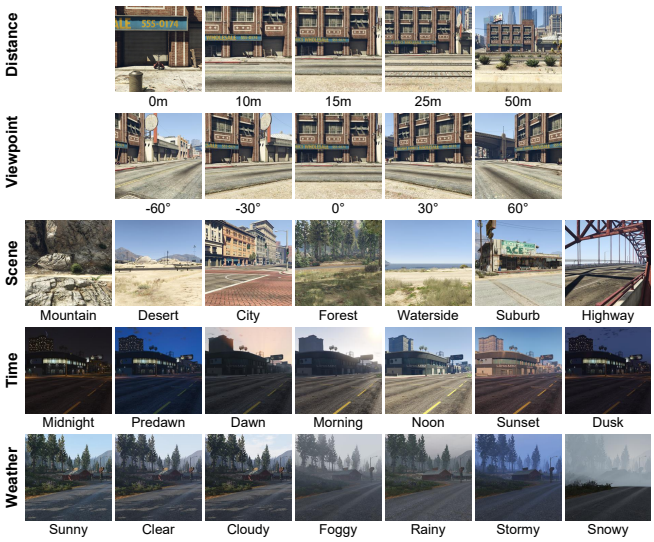

Fig. 1. Overview of several attributes in GOS dataset.

The method above primarily relies on adequate datasets applied to training high-quality generators, and rich semantic labels used for determining editing direction. There are several scene datasets existing, some focus on outdoor landscape [12, 13, 14], some are streetscapes captured by vehicles [15, 16, 17]. However, these datasets can only satisfy one of the adequate semantic labels or plentiful and diverse images. In addition, there is currently no dataset that included annotations related to the structure of the scene, such as the viewpoint and distance of the camera when collecting.

Annotating the ground-truth of a large amount of raw data need to spend plenty of time and labor. Moreover, it is difficult to avoid manual errors in the labeling process. Several methods [18, 19, 20] about the generation and annotation of synthetic datasets based on video games or open-source engines are proposed, which can effectively serve the tasks.

In order to solve the challenges above, we construct a synthetic outdoor scene dataset called "**G**TA V **O**utdoor **S**cene" ("**GOS**" for short) dataset with the help of a game engine. Compared with existing datasets, GOS has several advantages: (1) easy collection and annotation; (2) larger

This work was supported by the National Natural Science Foundation of China (Grant No.61977045).

Table 1. Comparison of real-world and synthetic outdoor scene related datasets. #ctg means category.

Dataset	#image	Resolution	#ctg	#view	#dist
SUN Attr. [12]	14,340	600×450	717	–	2
LSUN [13]	10,000,000	376×256	10	–	–
Places365 [14]	1,803,460	256×256	365	–	–
Streetview [15]	62,058	1280×1024	–	5	–
Cityscape [16]	25,000	2048×1024	–	–	–
Waymo [17]	250,000	1920×1160	–	5	†
SYNTHIA [18]	13,400	960×720	–	4	†
VIPER [19]	254,064	1920×1080	–	–	†
GOS (Ours)	4,632,500	1920×1080	7	10	5

† Indirect annotates, need to analysis related data (e.g. depth map).

data volume and more diversified scenes; (3) more abundant annotations (such as viewpoint and distance of scene).

After obtaining the dataset, we use annotations in GOS training the attribute predictor to get attribute boundaries, then apply editing on images to make its attributes change in specific needs. The proposed predictor can improve the performance in editing the structure of outdoor scene images based on the style-based generator.

In summary, the major contributions in the paper are as follows: 1) We build an extensive and diverse synthetic outdoor scene dataset using the self-developed tool, which contains 4,632,500 images with fine-grained annotations. 2) We exploit the annotations in GOS training attribute predictor, the extracted feature can achieve a better editing effect on several outdoor scenes comparing other methods, especially in viewpoint and distance.

2. GTA V OUTDOOR SCENE DATASET

Grand Theft Auto V (GTA V) is an action-adventure video game. The open-world design lets players immersed in board countryside and fictional city, and the proprietary engine makes its gameplay scenes comparable to the real world. We develop a tool for outdoor scene collecting in GTA V based on Script Hook V[1], which allows using GTA V script native functions in custom plugins. We also utilize a tool[2] to collect depth and stencil maps corresponding to gameplay scene via intercepting the data of the game rendering pipeline.

GOS is constructed using the above tools in the following steps: 1) sets up the scene environment, 2) captures scene images and annotates the semantics, 3) removes failure cases. Coming up is the detailed introduction of each part.

2.1. Scene construction

GTA V contains 226 main streets distributed in cities and rural areas, whose start points, endpoints, and waypoints can be obtained by digging game data. We set up collection points every 15 meters on each street to ensure the collected scenes will neither be missed nor excessively repetitive.

In order to collect more structure-related semantics for the dataset, which rarely exists in others, we arrange multiple cameras with different viewpoints and distances at each collection point. To enrich the attribute of illumination for the dataset, various time and weather conditions are also introduced. Fig. 2(a) shows scene construction configuration.

The collection tool will build the scene environment based on preset configuration, collect stable scene images (including color images, depth, and stencil maps), and annotate the attributes of the scenes. The whole process is fully automated.

2.2. Collection & Selection

When the camera moves away from the collection point, it may be blocked by objects beside the street, or enters buildings or mountains, which makes collected scenes not meet the requirements. The following are some typical scenarios.

Scene occlusion. Foreground objects appear unexpectedly in the middle of the camera and the background of the scene, such as street lights, fences, and trees, causing collected scenes to occluded.

Render missing. The camera moves into a building or mountain, but the related texture is not rendered in some cases, making the upper part of the scene stay normal, but the lower part shows patterns of sky or water, which is usually roads, makes the whole scene appear unnatural.

We also place a large object at the collection point, which can be used to indicate the depth of the location, making obtained depth and stencil map able to filter unqualified images. Fig. 2(b) shows selection process with some failure cases.

2.3. Dataset Properties

GOS consists of 4,632,500 images, contains 231,625 scenes. Detailed information and samples about the dataset is available online at `https://myronxie.github.io/GOS/`. We summarize the attributes in GOS into following aspects:

Place. 15,370 places with 7 major scene categories, contains *mountain*, *city*, *forest*, *desert*, *waterside*, and so on.

Viewpoint. 5 different types of viewpoint for each side of place: every 30° from $-60° \sim 60°$.

Distance. 5 different types: *0m*, *10m*, *15m*, *25m*, *50m*.

Time. 8 different kinds: *midnight*, *predawn*, *dawn*, *morning*, *midday*, *afternoon*, *sunset* and *dusk*.

Weather. 8 different types: *sunny*, *clear*, *cloudy*, *foggy*, *overcast*, *rainy*, *stormy*, and *snowy*.

Table 1 compares the properties of GOS and existing datasets, including data volume, image resolution, annotation types, which can reflect its advantages. On the whole, these aspects together make GOS a rich dataset for research.

[1] `http://www.dev-c.com/gtav/scripthookv`

[2] `https://github.com/umautobots/GTAVisionExport`

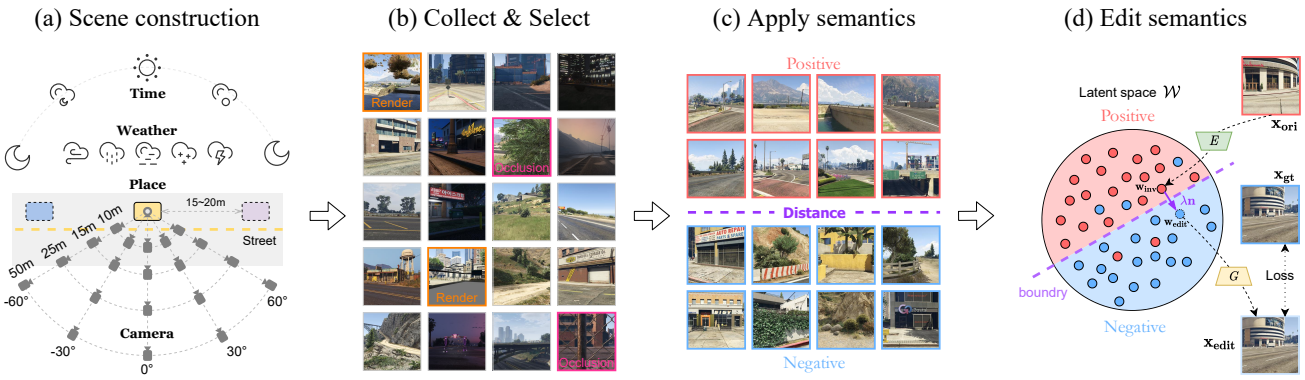

Fig. 2. The framework of GOS construction and image semantics editing based on style-based generator.

3. METHOD

3.1. Edit image on style-based generator

Fig. 2(d) shows the process of editing on the style-based generator. Style-based generator G maps the latent code $\mathbf{w}$ to the image $\mathbf{x}$ by learning the distribution of the dataset. The generated realistic images and the decoupling of latent space make image semantics editing possible.

In order to edit the real images, an encoder E is introduced to map the real image $\mathbf{x_{ori}}$ back to latent code $\mathbf{w_{inv}}$, which is also known as GAN inversion. E is trained based on pre-trained G which is expected to faithfully reconstruct images.

Each image $\mathbf{x}$ contains several different kinds of semantic annotations. A semantic predictor S can be constructed based on certain semantics with the specific dataset, then attach the semantics to the fake image $\mathbf{x}'$ generated by the generator G. Finally, the corresponding latent code $\mathbf{w}'$ can be assigned scores related to specific semantics.

We can divide the samples based on the semantic scores, then learn the direction $\mathbf{n}$ in which semantic score changes the most obvious in the latent space. Then latent code can be manipulated as: $\mathbf{w_{edit}} = \mathbf{w_{inv}} + \lambda\mathbf{n}$, where λ represents the hyper-parameter of step length. $\mathbf{w_{edit}}$ is then fed into the G to obtain final edited image $\mathbf{x_{edit}}$. It will make the image look more positive on such semantics with $\lambda > 0$, vice versa.

3.2. Build semantic predictor

Among the above method, semantic predictor S plays an important role, which generates the semantic score, determines semantic boundary, and emerges editing direction.

In order to take advantage of the rich semantic annotations provided by GOS, we use these training attribute predictors. Due to space limitations, we mainly focus on distance editing of the scene. We use *distance* annotation in GOS training on ResNet-101 [21]. The furthest distance (*50m*) designated as the highest score, the nearest (*0m*) regarded as the lowest score. Fig. 2(c) shows the process of semantics division.

For better comparison, we introduce several ways related to the depth of the scene to construct semantic predictor:

- Semantic annotations: We employ an attribute predictor forecasts 102 scene attributes in SUN attribute [12], and use *no horizon* label as distance distinction.
- Depth estimation: We use MiDaS [22] to generate pseudo depth of images, and treat the average depth of the overall image as the semantic score of distance.
- Unsupervised: We extract unentangled eigenvectors using SeFa [23] on the pre-trained generator, and pick the one that can change the semantics most obviously.

For semantics division, we use the method based on [10], where latent codes can be divided into positive and negative samples based on average semantic score, and then search for the decision boundary in the latent space $\mathcal{W}$ using SVM. Specifically, we choose the unit vector $\mathbf{n}$ orthogonal to the decision boundary as the editing direction.

4. EXPERIMENTS

4.1. Settings

Dataset. For LSUN [13], we use 126,527 images of *church-outdoor* category. For Places365 [14], we extract outdoor-related images subjectively containing about 1 million images. For Streetview [15], we extract all side views contains 41,372 images. For GOS, we use the data of *day* and *sunny* attributes. All images are cropped and scaled to 256 × 256.

Implementation. We train StyleGAN2 on the above datasets individually with config-f and no modification of hyper-parameters[3]. The iteration number is set to 400k with a batch size of 8. For image inversion, we use ReStyle [24] over pSp [25] with 200k iteration. For image editing, all methods use the same λ for fairness. We use layer-wise editing (mainly lower layers of G) to better focus on the distance and viewpoint editing effect. All experiments are conducted on a workstation with one NVIDIA GeForce RTX 3090 GPU.

[3] https://github.com/rosinality/stylegan2-pytorch

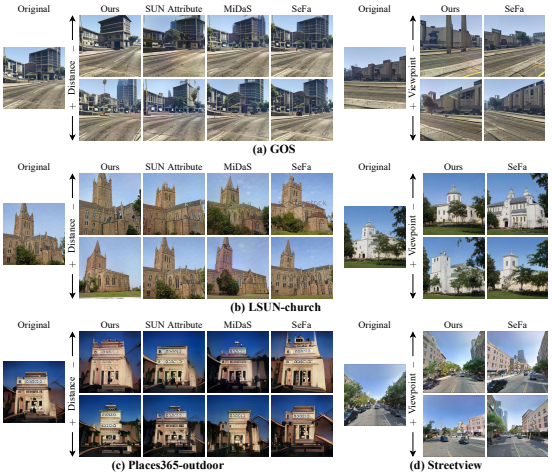

Fig. 3. Overview of distance and viewpoint editing on generated images in several outdoor scene datasets.

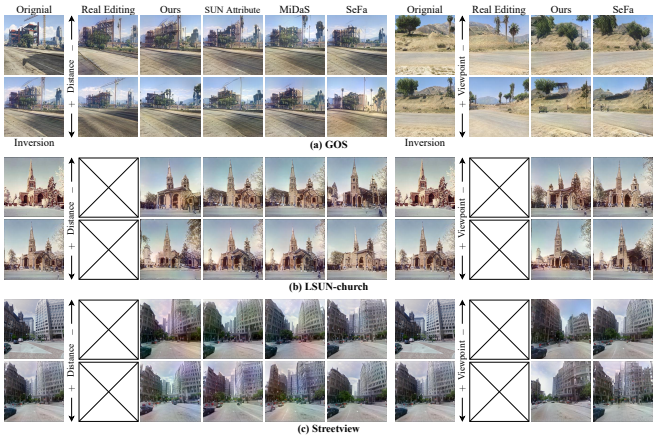

Fig. 4. Overview of distance and viewpoint editing on inversion images in several outdoor scene datasets. Only GOS has ground-truth of different distances and viewpoints.

4.2. Evaluation

Quantitative Evaluation. Since GOS has ground-truth at various distances, we can judge the degree of editing based on the semantic score difference between the edited image and the original image like [11]. Table 2 shows quantitative results editing in inversion images, the closer the value is to reality, the better. The result shows our method can not only significantly outperform other methods by a large margin in our predictor, but also achieve a certain effect in other metrics.

Qualitative Evaluation. We show visual results of editing on generated images in Fig. 3. Some salient objects in the scene such as buildings have more obvious changes, but the overall scene such as roads are not obviously changed, which is more like zooming of the objects in the scene, explains it is difficult to learn about the features of the distance of overall scene in the existing outdoor scene datasets. Our method can achieve better editing performance in multiple datasets.

Although there exist differences in latent code distribution between the inversion and generated images, which makes the inversion images distorted, Fig. 4 shows the boundary learned based on generated images still achieves a certain effect on inversion images, reflects a certain degree of generalization.

User Study. To perceptually evaluate the editing results, we also conduct a user study in Table 3. We collect about 900 votes from 15 volunteers. Each test set contains original image and images edited by different methods whose sequence is randomly arranged. Participants need to vote for an image with the best editing effect of certain semantics. The user study shows our method outperforms baselines by a large margin in GOS and other datasets.

Table 2. Quantitative evaluation of distance editing on GOS. $S(\cdot)$ means semantic predictors trained on different methods. "+", "−" means positive and negative editing directions.

Method	S(GOS)		S(MiDaS)		S(SUNA)	
	+	−	+	−	+	−
Real editing	0.128	0.068	2.243	12.25	0.491	0.618
Semantic(SUNA)	0.025	-0.047	**2.571**	2.269	**0.415**	0.208
Depth estimate	0.035	0.018	0.633	0.166	-0.001	0.198
Unsupervised	0.054	-0.064	1.073	0.945	-0.018	0.015
Semantic(GOS)	**0.099**	**0.085**	5.516	**5.909**	0.119	**0.245**

Table 3. User study of distance editing on several datasets.

Method	GOS	LSUN	Streetview
Semantic(SUNA)	19.0%	10.5%	19.0%
Depth estimate	3.0%	14.5%	33.0%
Unsupervised	11.3%	23.3%	17.0%
Semantic(GOS)	**66.7%**	**51.8%**	**41.0%**

5. CONCLUSION

In this paper, we construct a large-scale diverse dataset GOS for the first time with fine-grained annotations based on GTA V. Through exploiting the synthetic data, we train semantic predictors using annotations of viewpoint and distance in GOS. Then we manipulate the latent code based on learned semantic predictors on the styled-based generator to edit the semantics of images. The abundant experiments show that our method achieves the best editing effect on different outdoor datasets compared with other methods, which indicates the features learned from the proposed dataset have a certain generalization. In the future, we will keep focusing on exploring more realistic inversion methods for outdoor scene images and general editing strategies that can achieve better editing effects span on different outdoor scene datasets.

6. REFERENCES

[1] William B Thompson, Peter Shirley, and James A Ferwerda, "A spatial post-processing algorithm for images of night scenes," *Journal of Graphics Tools*, vol. 7, no. 1, pp. 1–12, 2002.

[2] Leon A Gatys, Alexander S Ecker, and Matthias Bethge, "Image style transfer using convolutional neural networks," in *CVPR*, 2016, pp. 2414–2423.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *NeurIPS*, 2014, vol. 27, pp. 2672–2680.

[4] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017, pp. 1125–1134.

[5] Shunyu Yao, Tzu Ming Harry Hsu, Jun-Yan Zhu, et al., "3d-aware scene manipulation via inverse graphics," in *NeurIPS*, 2018, vol. 31, pp. 1887–1898.

[6] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *CVPR*, 2019, pp. 2337–2346.

[7] Tero Karras, Samuli Laine, and Timo Aila, "A style-based generator architecture for generative adversarial networks," in *CVPR*, 2019, pp. 4401–4410.

[8] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila, "Analyzing and improving the image quality of stylegan," in *CVPR*, 2020, pp. 8110–8119.

[9] Ali Jahanian, Lucy Chai, and Phillip Isola, "On the "steerability" of generative adversarial networks," in *ICLR*, 2020.

[10] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou, "Interpreting the latent space of gans for semantic face editing," in *CVPR*, 2020, pp. 9243–9252.

[11] Ceyuan Yang, Yujun Shen, and Bolei Zhou, "Semantic hierarchy emerges in deep generative representations for scene synthesis," *IJCV*, pp. 1–16, 2021.

[12] Genevieve Patterson and James Hays, "Sun attribute database: Discovering, annotating, and recognizing scene attributes," in *CVPR*. IEEE, 2012, pp. 2751–2758.

[13] Fisher Yu, Ari Seff, Yinda Zhang, and other, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015.

[14] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba, "Places: A 10 million image database for scene recognition," *TPAMI*, vol. 40, no. 6, pp. 1452–1464, 2017.

[15] Amir Roshan Zamir and Mubarak Shah, "Image geo-localization based on multiplenearest neighbor feature matching usinggeneralized graphs," *TPAMI*, vol. 36, no. 8, pp. 1546–1558, 2014.

[16] Marius Cordts, Mohamed Omran, Sebastian Ramos, et al., "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016, pp. 3213–3223.

[17] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, et al., "Scalability in perception for autonomous driving: Waymo open dataset," in *CVPR*, 2020, pp. 2446–2454.

[18] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *CVPR*, 2016, pp. 3234–3243.

[19] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun, "Playing for benchmarks," in *ICCV*, 2017, pp. 2213–2222.

[20] Suncheng Xiang, Yuzhuo Fu, Guanjie You, and Ting Liu, "Unsupervised domain adaptation through synthesis for person re-identification," in *ICME*. IEEE, 2020, pp. 1–6.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[22] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *TPAMI*, 2020.

[23] Yujun Shen and Bolei Zhou, "Closed-form factorization of latent semantics in gans," in *CVPR*, 2021.

[24] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or, "Restyle: A residual-based stylegan encoder via iterative refinement," in *ICCV*, October 2021.

[25] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or, "Encoding in style: a stylegan encoder for image-to-image translation," in *CVPR*, June 2021.