CGI2024 manuscript No.
(will be inserted by the editor)

Seasonal Terrain Texture Synthesis via Köppen Periodic Conditioning

Toshiki Kanai · **Yuki Endo** · **Yoshihiro Kanamori**

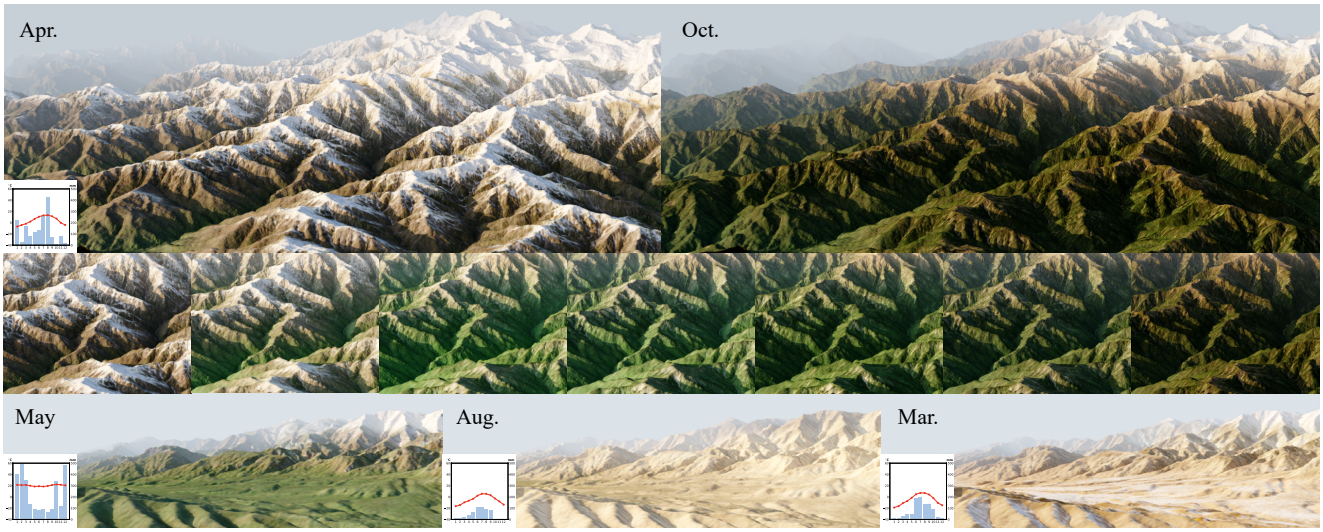

Fig. 1 3D terrain models textured using our method. Our method can generate terrain textures that reflect climatic and seasonal changes using a heightfield and monthly temperature, precipitation, and insolation as inputs. (Top) Textures for April and October in the frigid zone. (Middle) Monthly changes from April to October. (Bottom) 3D models (with a different heightfield) generated using typical textures for (left) torrid, (center) subarctic, and (right) frigid zones, respectively.

Abstract This paper presents the first method for synthesizing seasonal transition of terrain textures for an input heightfield. Our method reproduces a seamless transition of terrain textures according to the seasons by learning measured data on the earth using a convolutional neural network. We attribute the main seasonal texture transition to vegetation and snow, and control the texture synthesis not only with the input heightfield but also with the annual temperature and precipitation based on *Köppen's climate classification* as well as insolation at the location. We found that month-by-month synthesis yields incoherent transitions, while a naïve conditioning with explicit temporal information (e.g., month) degrades generalizability due to the north-south hemisphere difference. To address these issues, we introduce a simple solution – *periodic conditioning* on the annual data without explicit temporal information. Our experiments reveal that our method can synthesize plausible seasonal transitions of terrain textures. We also demonstrate large-scale texture synthesis by tiling the texture output.

Keywords Deep learning · GAN · Texture synthesis · Terrain

T. Kanai
E-mail: kanai.toshiki.as@alumni.tsukuba.ac.jp
University of Tsukuba

Y. Endo
E-mail: endo@cs.tsukuba.ac.jp
University of Tsukuba

Y. Kanamori
E-mail: kanamori@cs.tsukuba.ac.jp
University of Tsukuba

1 Introduction

We can see a wide range of earth landscapes due to terrain and climate variations. The landscape appearance can also change dramatically with seasons, even in the same location. For example, in regions with large

temperature differences between summer and winter, such as subarctic regions, trees are in full leaf in summer and snow in winter. Suppose we can reproduce these appearance differences due to terrain, climate, and seasonality in a 3D terrain model. We can then explore a wide range of applications in fields such as video and gaming productions and in visualizing the impact of global warming and other climate change effects on the earth. However, existing studies on terrain and landscape in computer graphics have mostly focused on the generation of 3D terrain geometry [19,44,20, 41,11,24,35,18], with relatively little attention paid to terrain texture synthesis [42,43]. In particular, to the best of our knowledge, there are currently no methods for generating terrain textures that take into account transitions due to climate and season.

The terrain texture variations due to climate and seasonality are mainly caused by vegetation and snow. Regarding vegetation, *Köppen's climate classification* [34] focuses on differences in vegetation, which uses temperature and precipitation as explanatory variables to classify climate. For snow, the terrain geometry affects snow accumulation and melting; for example, snow tends to accumulate more and melt more slowly in highlands. Also, in the north hemisphere for example, south-facing slopes receive more sunlight, so snow accumulates more slowly and melts more quickly than on north-facing slopes. Other factors that affect climate and season include, for example, latitude. However, various places with different latitudes have similar climates due to the influence of ocean currents. Therefore, latitude is not an appropriate control parameter for climate and season. Based on these considerations, we propose a method for generating terrain textures that reflect climate and season using a heightfield and monthly temperature, precipitation, and insolation for one year as inputs. By using a convolutional neural network (CNN) trained with our novel measured dataset, we can synthesize terrain textures in all seasons for the input terrain heightfield (see Figure 1).

This study aims to address the following two issues. First, the generated terrain textures should not depend on month information, such as "*February is winter and August is summer,*" because seasons may differ even in the same month, for example, in the northern and southern hemispheres. Furthermore, suppose a location on a hemisphere has a terrain geometry and annual temperature and precipitation similar to those on the other hemisphere with a half-year delay for the annual data. These locations should have similar texture transitions; as proof, Köppen's climate classification does not depend on month information. Second, there is no available dataset of heightfields with ground-truth terrain textures that include temperature and precipitation information, which is necessary for the supervised training of our network.

This paper offers the following simple solutions. For the first issue, we control the temporal texture transition by ordering the input monthly data, i.e., temperature, precipitation, and insolation. Namely, we synthesize terrain textures for one year month-by-month while changing the order of the annual data periodically so that the target month's data come first. We refer to our technique as *Köppen periodic conditioning*, which yields better generalizability across the north and south hemispheres. We offer smooth inter-month transition via feature map interpolation. For the second issue, we construct a novel dataset combining measured data from multiple data sources such as space shuttles and satellites. As no competitive methods exist, we validate the effectiveness of our method through an ablation study comparing several variants with different inputs, such as latitude and explicit month information. We further demonstrate large-scale texture synthesis by tiling texture outputs.

2 Related Work

3D terrain generation. A number of terrain generation methods have been proposed to support the production of 3D terrain models [14], with a particular focus on heightfield generation. For example, Guérin et al. [19] trained a CNN on the space shuttle 3D terrain data (NASA SRTM) to generate a realistic heightfield from a sketch. In a recent study by Guérin et al. [20], a multigrid method can also be used to interactively generate heightfields by converting gradient manipulations from sketch input to elevation values in real time. However, these methods do not generate terrain textures corresponding to the heightfields.

Terrain appearance simulation. Vegetation and snow cover are two of the most important elements for representing seasonally changing landscapes. For vegetation, computer graphics has a long history of research on plant growth simulation and modeling [17,27,4,36, 22,5]. Makowski et al. [25] focused on the relationship between temperature and precipitation as a means of controlling models of vegetation ecosystems and incorporated these as parameters in their simulations. Palubicki et al. [29] built on these geographic parameters to allow for more microscopic vegetation simulations. In snow cover simulations, solar radiation is also used as an environmental information other than temperature and precipitation [13,10]. A similar study [3] used solar

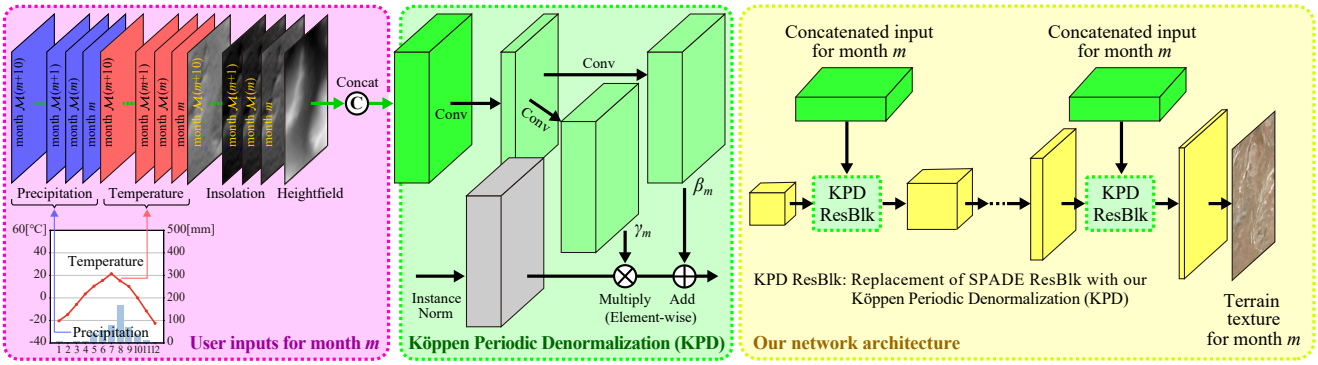

Fig. 2 Overview of our method. To output a terrain texture for month m (where $m = 1, 2, \ldots, 12$), one-year temperature, precipitation (both represented as constant maps), and insolation maps are ordered (where $\mathcal{M}(i) = (i \mod 12) + 1$), concatenated with the target heightfield (left), and fed to the *Köppen Periodic Denormalization* (*KPD*) layer (middle). The concatenated input for month m is injected repeatedly at multiple scales (right). One-year textures are generated by repeating the whole process for each month m. Note that heightfields in this paper are visualized using linear normalization for better display.

irradiance in glacier simulations. Argudo et al. [2] proposed a method to generate a variety of ground surface conditions, taking into account both vegetation and snow cover. Their method constructs a coherent multi-layer dictionary that represents relationships among parameters such as topographic elevation, vegetation density, and irradiance. These simulation methods can produce photorealistic scenes with snow or vegetation. However, they cannot handle seasonal changes over an annual cycle, taking into account both vegetation and snow cover. In addition, although our work is similar to theirs in that they use various geographic data, our problem setting differs from their works in that our goal is to obtain topographic textures.

Terrain texture synthesis. Terrain textures are also used to represent large-scale topography and for geospatially based artworks such as panoramic maps [37, 7]. The existing study on terrain texture generation by Spick et al. [42] can generate four-channel textured heightfields via supervised learning using CNNs. Zhu et al. [45] proposed a method to generate satellite images from the labeled map data. Panagitou et al. [30] presented a framework for generating both terrain textures and heightfields using generative adversarial networks (GANs). However, these methods do not account for seasonal transitions in terrain textures and do not allow user control over the generated heightfields and terrain textures.

Dachsbacher et al. [12] proposed a method for generating terrain textures using geographic data on temperature, precipitation, and solar radiation, in addition to heightfields. This method classifies each pixel as rock, vegetation, water, or snow based on the given geographic data, and colors it according to the classification result. However, such a method suffers from the problem of representing colors that continuously change with the seasons. In addition, this method requires pixel-by-pixel temperature and precipitation maps, but it is costly for the user to prepare such input. Although such parametric or rule-based texture generation methods are often effective for generating textures at a specific time instance, they are difficult to control for continuous transition. Our method, on the other hand, can generate smooth time-varying terrain textures with a simple control using temperature and precipitation.

Image-to-image translation. Image-to-image translation techniques have evolved rapidly with the advent of U-net [39], GANs [15] and diffusion models [21]. A wide variety of methods [31] exist, such as pix2pix [23], SPADE [32], and Palette [40] for supervised learning of transformations between domains, CycleGAN [46] for unsupervised learning, and BicycleGAN [47] and Swapping Autoencoder [33] for multimodal transformations. These methods have also been used to reproduce seasonal changes such as summer to winter. However, in order to create terrain textures that account for various climates and seasons, users need to prepare multiple reference images in advance. In our study, by focusing the extensibility of the SPADE's denormalization layer [32], we extend it and introduce a novel denormalization layer that processes heightfields and monthly temperature, precipitation, and insolation for one year. This allows the generation of a variety of terrain textures reflecting climates and seasons without the use of reference images.

3 Our Method

Figure 2 summarizes our method. Our inputs are a heightfield and monthly temperature, precipitation, and insolation data for one year. We train a CNN in a supervised manner to synthesize terrain textures reflecting the corresponding climate and season. As the CNN architecture, we extend the *spatially denormalization layer* (SPADE) by Park et al. [32] for our purpose. Whereas the original SPADE block accepts a semantic mask to enforce spatial layout for photorealistic semantic image synthesis, there are several extensions of denormalization layers for handling geometric information [26, 38], optical flows [26], and shadow removal from terrain landscapes [43]. We adopt a denormalization layer to inject information on heightfield, temperature, precipitation, and insolation at multiple scales. For training, because there is no dataset for our purpose, we create a novel dataset using the geospatial analysis platform Google Earth Engine [16]. Hereafter we explain the details of conditioning with denormalization layers and dataset construction.

3.1 Conditioning in Denormalization Layer

We aim to synthesize a terrain texture for each month from the inputs (and later employ inter-month interpolation in Section 3.2). A straightforward way would be to feed just one month of information and output a texture month-by-month, just like an ordinary image-to-image translation. Unfortunately, this approach yields temporally inconsistent textures, as we demonstrate later in Section 4.2. Instead, we feed one year of information to our network to account for the annual transition. In this case, a natural question is how we can signal the target month's information to the network. A naïve approach would be to feed the target month's index to the network explicitly, but this approach does not generalize across the north and south hemispheres, as explained in Section 1.

Our solution is to rotate the one-year data periodically so that the target month's data come to the front (Figure 2, left). More formally, suppose we want to synthesize a texture for month m (where $m = 1, 2, \ldots, 12$). We order each of the annual temperature, precipitation, and insolation according to the sequence of $\{m, \mathcal{M}(m), \mathcal{M}(m+1), \ldots, \mathcal{M}(m+10)\}$, where $\mathcal{M}(i) = (i \mod 12) + 1$. For example, in the case of October (i.e., $m = 10$), the sequence becomes $\{10, 11, 12, 1, \ldots, 9\}$. The sequences of temperature and precipitation are converted into constant maps of the same size as the input heightfield, concatenated together with the insolation maps and heightfield in the channel direction, and then fed to our *Köppen Periodic Denormalization* (KPD) block (Figure 2, middle).

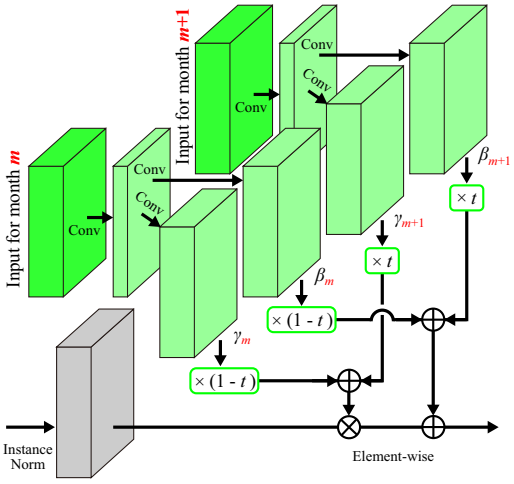

Fig. 3 Feature map interpolation between consecutive months. The feature maps $\{\beta, \gamma\}$ for months m and $m+1$ are linearly interpolated with parameter $t \in [0, 1]$ and then fed to the denormalization block.

3.2 Network Design and Extensions

Network architecture and loss functions. Our network architecture is a simple extension of that of SPADE [32]. Namely, we replace the SPADE residual blocks (or *ResBlk*) with our KPD ResBlk (Figure 2, right), where the SPADE blocks are replaced with our KPD blocks. We use the same loss functions as those of the SPADE network, except that we omit the loss for the variational autoencoder (VAE) because we do not use VAE.

Feature map interpolation for inter-month texture transition. So far, we can generate monthly terrain textures, but it is important to be able to handle continuous seasonal changes between months when considering applications such as video production. Our method can obtain continuously changing terrain textures by interpolating the feature maps. Specifically, during inference, the feature maps $\{\beta_m, \gamma_m\}$ and $\{\beta_{m+1}, \gamma_{m+1}\}$ in our KPD block for months m and $m+1$ are linearly interpolated with parameter $t \in [0, 1]$ (Figure 3).

$$\beta' = (1-t)\,\beta_m + t\,\beta_{m+1}, \quad \gamma' = (1-t)\,\gamma_m + t\,\gamma_{m+1}. \tag{1}$$

The interpolated $\{\beta', \gamma'\}$ are then used in our KPD block.

3.3 Dataset

We construct a dataset for training using the geospatial analysis platform *Google Earth Engine* [16]. Our dataset consists of four types of observational data and two types of derived data. The observational data are heightfields, corresponding monthly temperature and precipitation data for one year, and terrain textures, all observed in 2020 and publicly available. The heightfields are obtained from *NASADEM* [28], and the spatial resolution is 30 meters per pixel. For temperature and precipitation, we use the monthly averaged data of *ERA5* [9]. Our terrain textures are the Level-2 product of the Sentinel-2 land observation satellite [1], in which atomospheric effects (e.g., faint clouds) and terrain shadings are removed to some extent. The spatial resolution is 10 meters per pixel. Our derived data are cloud masks and insolation maps. Cloud masks are binary masks to specify clouds and cloud shadows in the terrain textures; although the satellite images were taken by two satellites every five days, cloudy sites might often be covered by clouds with their shadows. We thus synthesize terrain textures from one-month satellite images with minimum clouds and shadows, which are extracted as cloud masks [6]. To eliminate the effects of clouds and shadows during training, we exclude them in calculating loss functions using the cloud masks. Before subdivided into 256×256 tiles, the insolation maps are calculated by averaging the direct solar radiation for every hour for the 15th day of every month using ray casting, considering the solar positions at specific longitudes[1].

Our dataset includes 97 sites randomly sampled from land territories while excluding artificial structures. The sampled sites are categorized according to Köppen's climatic divisions as follows: A (torrid) 13 sites, B (arid) 32 sites, C (temperate) 27 sites, D (subarctic) 15 sites, and E (frigid) 10 sites. A total of 10 sites, two randomly selected from each of the five climate categories, were used as test data, and the remaining 87 sites were used as training data. Each site was sampled at 0.5 degrees latitude and longitude, and the resulting image data has a resolution of $5,500 \times 5,500$ pixels and was divided into small tiles of 256×256 pixels. We also excluded the following sites; 1) sites with elevations less than 100 meters, which tend to contain artificial structures, and 2) sites with missing areas in the terrain texture that exceeded five percent of 256×256 pixels (e.g., the gray boxes in the ground-truth (GT) textures in Figure 6). Note that the black pixels in the GT textures of Figures 6 and 7 are missing areas due to clouds. Consequently, the dataset contains 414K terrain textures and 41K heightfields at a resolution of 256×256 pixels. We matched the resolutions of heightfields and insolation maps to that of terrain textures (i.e., 256×256) using linear interpolation, whereas we used averaged temperature and precipitation values within each heightfield.

Note that all the climate categories to which each site belongs are based on the study by Chen et al. [8]. Because temperature and precipitation vary somewhat from year to year, the combinations of climate divisions and temperature/precipitation in this paper do not necessarily correspond to the definition of each climate division.

[1] We used *pvlib* for calculating the solar altitude from the latitude.

4 Results

4.1 Experimental Setting

We implemented our method using Python and PyTorch and ran on an NVIDIA RTX A6000 for training and inference. The Adam optimizer was used with a learning rate of 0.0002 and $(\beta_1, \beta_2) = (0, 0.999)$, respectively. The batch size used was 32, and the model was trained for 20 epochs, which took approximately four days. The results reported are based on the test data and not the training data. Some values of temperature and precipitation in the dataset are fictional.

To validate our method, as there are no existing methods with the same objective, we conducted an ablation study by comparing it with different conditions. We denote temperature as $\mathbf{T}$, precipitation as $\mathbf{P}$, month index as $\mathbf{M}$, latitude as $\mathbf{L}$ (where $\mathbf{T}$, $\mathbf{P}$, $\mathbf{M}$, and $\mathbf{L}$ are represented as constant maps whose resolution is 256×256), and insolation maps as $\mathbf{S}$. We omit heightfield symbols to shorten the descriptions because all conditions require heightfields. In the following experiments, the resolution of output textures is 256×256 pixels and the spatial resolution is 10 meters per pixel.

- ***Condition* 1** ($\mathbf{T}'_{12} + \mathbf{P}'_{12} + \mathbf{M}$): We fed a heightfield, fixed-ordered temperature $\mathbf{T}'_{12}$ and precipitation $\mathbf{P}'_{12}$ for one year, and month index $\mathbf{M}$.
- ***Condition* 2** ($\mathbf{T}_1 + \mathbf{P}_1$): We fed a heightfield, one-month temperature $\mathbf{T}_1$ and precipitation $\mathbf{P}_1$.
- ***Condition* 3** ($\mathbf{L} + \mathbf{M}$): We fed a heightfield, latitude $\mathbf{L}$, and month index $\mathbf{M}$. The latitude $\mathbf{L}$ is an approximation of average insolation.
- ***Condition* 4** ($\mathbf{T}_{12} + \mathbf{P}_{12}$): We fed a heightfield, and periodically-rotated temperature $\mathbf{T}_{12}$ and precipitation $\mathbf{P}_{12}$ for one year (as explained in Section 3.1), without insolation maps $\mathbf{S}$.

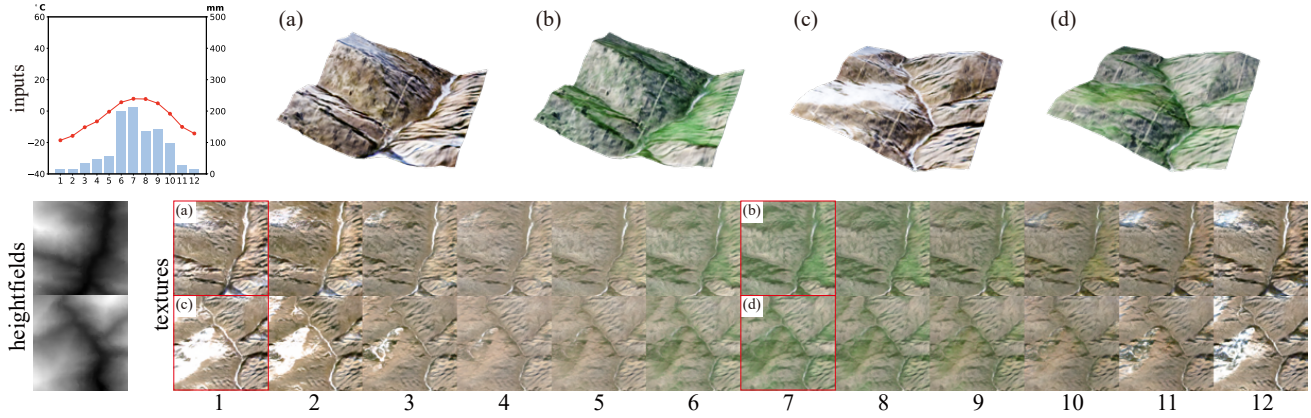

Fig. 4 Qualitative comparison of terrain textures generated by changing only the input heightfields. We generated the 3D terrain models using the terrain textures in the red rectangles and the input heightfields. The result texture patterns differ depending on the terrain shapes.

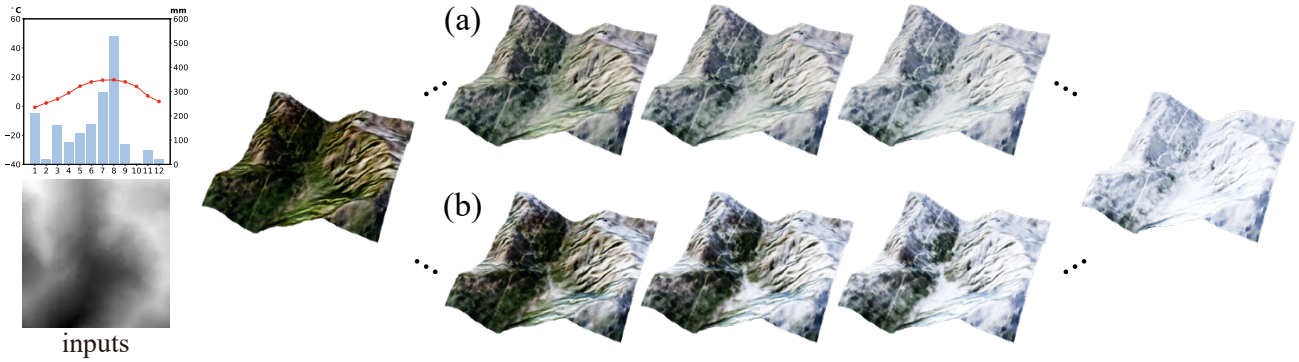

Fig. 5 Qualitative comparison of terrain textures obtained via (a) output image interpolation and (b) feature map interpolation. The terrain turns uniformly white in (a), while the snow increases gradually from top to bottom in (b).

Ours ($\mathbf{T}_{12} + \mathbf{P}_{12} + \mathbf{S}_{12}$): We fed a heightfield, and periodically-rotated temperature $\mathbf{T}_{12}$, precipitation $\mathbf{P}_{12}$, and insolation $\mathbf{S}_{12}$ for one year.

The input maps are concatenated in the channel direction.

4.2 Qualitative Evaluation

Comparison of Conditions 1-4 and Ours. Figure 6 shows a qualitative comparison of *Conditions* 1-4 and our method. The top results show that *Conditions* 1 and 3 are poorly reproducible throughout the year. *Condition* 2 exhibits an extreme decrease in vegetation from March to June. *Condition* 4 and our method produced textures that reflected consistent seasonal changes that were close to the ground truth throughout the year. In the lower results, *Condition* 1 produced dry textures. In *Condition* 2, the change from June to September is extreme. In *Conditions* 3 and 4, the seasonal changes are continuous but poorly reflect summer vegetation. Our method produced textures that were closer to the ground truth than the other conditions. The black area in the ground truth is the missing area where the image of the ground surface could not be obtained.

Verifying adaptability to input periodicity. We verified that our method depends not on month indices but only on the temperature and precipitation transitions to reflect seasonal texture differences. Figure 7 compares *Condition* 1 ($\mathbf{T}'_{12} + \mathbf{P}'_{12} + \mathbf{M}$) and *Condition* 4 ($\mathbf{T}_{12} + \mathbf{P}_{12}$), feeding temperature and precipitation with a half-year delay. *Condition* 1 relies on month indices and cannot account for the temperature and precipitation periodicity to generate photorealistic textures, exhibiting almost blue-or-white with grid-like artifacts. *Condition* 4 successfully handles the same temperature and precipitation transitions despite the half-year delay. Note that the quantitative assessment in Table 1 does not reflect these extreme differences because there is less land in the southern hemisphere and, therefore, less southern hemisphere data in the dataset.

Table 1 Quantitative comparison of the different input conditions. The bold and underlined values show the best and second-best scores.

	RMSE ↓	SSIM ↑	LPIPS ↓	∂AD↓	∂SSIM↓
Cond. 1 ($\mathbf{T}'_{12} + \mathbf{P}'_{12} + \mathbf{M}$)	64.3	**0.57**	0.47	123.3	<u>0.13</u>
Cond. 2 ($\mathbf{T}_1 + \mathbf{P}_1$)	75.7	0.52	0.49	124.8	0.16
Cond. 3 ($\mathbf{L} + \mathbf{M}$)	66.8	<u>0.55</u>	<u>0.46</u>	**116.9**	0.14
Cond. 4 ($\mathbf{T}_{12} + \mathbf{P}_{12}$)	<u>61.1</u>	0.52	**0.44**	<u>120.8</u>	**0.11**
Ours ($\mathbf{T}_{12} + \mathbf{P}_{12} + \mathbf{S}_{12}$)	**59.3**	0.52	**0.44**	123.3	**0.11**

Verifying input dependency. Figure 8 shows qualitative comparisons with varying input temperature and precipitation, respectively. The continuous changes in the terrain textures with the varying input temperature and precipitation confirm that our method can synthesize textures that reflect climate and seasons. For example, in July, the cooler the temperature and the higher the precipitation, the greener the vegetation. In October and November, colder temperatures and more precipitation lead to more snow accumulation.

Figure 4 shows qualitative comparisons with varying input heightfields. Even with the same temperature and precipitation input, the resultant textures differ for different terrain shapes. It can be seen that heightfields play an important role in texture synthesis.

Verifying feature map interpolation. Figure 5 shows a comparison between the results of linear interpolation at (a) the output texture level and (b) the feature map level (explained in Section 3.2), to obtain continuous seasonal transition between two adjacent months in the winter season with snow cover. (a) The output texture interpolation does not take elevation into account and shows only a simple hue change. Contrarily, (b) the feature map interpolation shows the plausible progression of snow cover from higher elevations.

4.3 Quantitative Evaluation

For quantitative evaluation with reference to the ground-truth textures, we adopt RMSE, SSIM, and LPIPS as evaluation metrics. To evaluate the validity of the seasonal transitions, we also define novel metrics, i.e., ∂AD and ∂SSIM, which are defined as follows:

$$\partial\mathrm{AD} = \mathbb{E}\left[\,||G_{m+1} - G_m| - |T_{m+1} - T_m||\,\right], \tag{2}$$

$$\partial\mathrm{SSIM} = \mathbb{E}\left[\,|\mathrm{SSIM}(G_{m+1} - G_m) - \mathrm{SSIM}(T_{m+1} - T_m)|\,\right], \tag{3}$$

where m is a month index, G is a ground-truth texture, and T is an inferred texture.

Table 1 shows the result of quantitative comparison with different input conditions. As can be seen in the table, *Ours* won the most frequently. Although *Conditions* 1 and 3 scored the best for SSIM and ∂AD, respectively, these conditions show much worse RMSE, which consequently results in low-quality textures, as discussed in Section 4.2. Note that the inconsistent score rankings among the evaluation metrics in Table 1 stem from the different characteristics of the metrics and resultant images, which is common in evaluations of machine learning techniques. Further investigation is left for future work.

5 Discussions

This study aims to generate seasonal changes in mountainous regions, taking into account vegetation and snow cover. Our method uses temperature and precipitation as environmental information based on the Köppen's climate classification and also considers insolation. Insolation contributed to the accuracy, especially when generating snowy textures for subarctic and arctic zones distributed at high latitudes. This may be due to the fact that, in high-latitude regions, the annual sun elevation changes significantly, and when the sun is low, there is a clear distinction between slopes exposed to direct sunlight and those that are not, resulting in differences in ground surface temperature.

We discuss the controllability of our method. Our method can generate various terrain textures from the corresponding user inputs. Heightfields can be obtained from existing DEM data or created using existing tools. The temperature and precipitation data can be selected from typical data of climatic divisions with additional modification if necessary. The insolation maps can be calculated automatically, as mentioned in Section 3.3. We do not rely on pixel-level temperature/precipitation maps because we have to employ physics simulations to obtain such maps, considering other factors, such as wind and pressure, which drastically deteriorates usability. We can also handle large-scale scenes by seamlessly connecting adjacent texture tiles, as demonstrated in Figure 1 (see Appendix A for the details).

6 Conclusions and Future Work

In this paper, we have for the first time addressed the terrain texture generation that can control seasonal transitions for an input heightfield. To achieve this, we proposed a supervised learning framework using a CNN that includes monthly temperature, precipitation, and insolation as conditional signals. Our method can appropriately reflect seasonal transitions of the generated terrain texture by periodically changing the input order of monthly temperature, precipitation, and insolation.

In addition, we created a novel dataset based on actual observations from space shuttles and satellites to train the CNN. The experiments showed that our method can generate realistic terrain textures corresponding to the input signals.

As a future challenge, we would like to improve the accuracy of terrain texture generation in areas with heavy rainfall, in particular, the torrid zones. Some satellite images for the torrid zones contain thin clouds that could not be removed. Higher quality datasets produced through the further development of remote sensing technology will improve the resultant quality. Our inputs and outputs have a one-to-one correspondence and have no randomness. This design was made because simply adding randomness would degrade the quality of the generated terrain textures. To account for various factors not considered in the input conditions, it might be useful to introduce some randomness to enrich the output variations. In particular, an interesting avenue of future work is to integrate our periodic conditioning into the modern architectures of diffusion models to explore output variations.

Acknowledgement

The authors are grateful to Prof. Kenichi Ueno for his professional feedback on climate science.

References

1. Agency, E.S.: Sentinel-2 MSI level-2a BOA reflectance (2023). URL `https://sentinels.copernicus.eu/web/sentinel/technical-guides/sentinel-2-msi/level-2a-algorithms-products`. [accessed 22-December-2023]
2. Argudo, O., Andujar, C., Chica, A., Guérin, E., Digne, J., Peytavie, A., Galin, E.: Coherent multi-layer landscape synthesis. Vis. Comput. **33**(6–8), 1005–1015 (2017)
3. Argudo, O., Galin, E., Peytavie, A., Paris, A., Guérin, E.: Simulation, modeling and authoring of glaciers. ACM Trans. Graph. **39**(6) (2020)
4. Benes, B., Andrysco, N., Stava, O.: Interactive Modeling of Virtual Ecosystems. In: Eurographics Workshop on Natural Phenomena, pp. 9–16. The Eurographics Association (2009)
5. Beneš, B., Millán, E.U.: Virtual climbing plants competing for space. In: Proceedings of the Computer Animation, CA '02, pp. 33–42. IEEE Computer Society, USA (2002)
6. Braaten, J.: Sentinel-2 cloud masking with s2cloudless. `https://developers.google.com/earth-engine/tutorials/community/sentinel-2-s2cloudless` (2020). [accessed 18-October-2023]
7. Brown, S.A., Samavati, F.: Real-Time Panorama Maps. In: Non-Photorealistic Animation and Rendering. Association for Computing Machinery, Inc (ACM) (2017)
8. Chen, D., Chen, H.W.: Using the Köppen classification to quantify climate variation and change: An example for 1901–2010. Environmental Development **6**, 69–79 (2013)
9. Copernicus Climate Change Service: Era5-land monthly averaged data from 2001 to present (2019). URL `https://cds.climate.copernicus.eu/doi/10.24381/cds.68d2bb30`. [accessed 22-December-2023]
10. Cordonnier, G., Ecormier, P., Galin, E., Gain, J., Benes, B., Cani, M.: Interactive generation of time-evolving, snow-covered landscapes with avalanches. Comput. Graph. Forum **37**(2), 497–509 (2018)
11. Cordonnier, G., Jouvet, G., Peytavie, A., Braun, J., Cani, M.P., Benes, B., Galin, E., Guérin, E., Gain, J.: Forming terrains by glacial erosion. ACM Trans. Graph. **42**(4) (2023)
12. Dachsbacher, C., Bolch, T., Stamminger, M.: Procedural reproduction of terrain textures with geographic data. In: Proceedings of Vision Modeling and Visualization, pp. 105–112 (2006)
13. Foldes, D., Benes, B.: Occlusion-based snow accumulation simulation. In: Proceedings of the Fourth Workshop on Virtual Reality Interactions and Physical Simulations, VRIPHYS 2007, pp. 35–41. Eurographics Association (2007)
14. Galin, E., Guérin, E., Peytavie, A., Cordonnier, G., Cani, M.P., Benes, B., Gain, J.: A Review of Digital Terrain Modeling. Computer Graphics Forum **38**(2), 553–577 (2019)
15. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, vol. 27, pp. 2672–2680. Curran Associates, Inc. (2014)
16. Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., Moore, R.: Google earth engine: Planetary-scale geospatial analysis for everyone. Remote Sensing of Environment **202**, 18–27 (2017)
17. Greene, N.: Voxel space automata: Modeling with stochastic growth processes in voxel space. In: Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '89, p. 175–184. Association for Computing Machinery, New York, NY, USA (1989)
18. Grenier, C., Guérin, E., Galin, E., Sauvage, B.: Real-time terrain enhancement with controlled procedural patterns. Computer Graphics Forum p. e14992 (2023)
19. Guérin, É., Digne, J., Galin, E., Peytavie, A., Wolf, C., Benes, B., Martinez, B.: Interactive example-based terrain authoring with conditional generative adversarial networks. ACM Trans. Graph. **36**(6), 228:1–228:13 (2017)
20. Guérin, É., Peytavie, A., Masnou, S., Digne, J., Sauvage, B., Gain, J., Galin, E.: Gradient terrain authoring. Comput. Graph. Forum **41**(2), 85–95 (2022)
21. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems, vol. 33, pp. 6840–6851. Curran Associates, Inc. (2020)
22. Hädrich, T., Benes, B., Deussen, O., Pirk, S.: Interactive modeling and authoring of climbing plants. Computer Graphics Forum **36**(2), 49–61 (2017)
23. Isola, P., Zhu, J., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, pp. 5967–5976. IEEE Computer Society (2017)

24. Lochner, J., Gain, J., Perche, S., Peytavie, A., Galin, E., Guérin, E.: Interactive authoring of terrain using diffusion models. Computer Graphics Forum **42**(7), e14,941 (2023)
25. Makowski, M., Hädrich, T., Scheffczyk, J., Michels, D.L., Pirk, S., Palubicki, W.: Synthetic silviculture: multi-scale modeling of plant ecosystems. ACM Trans. Graph. **38**(4), 131:1–131:14 (2019)
26. Mallya, A., Wang, T., Sapra, K., Liu, M.: World-consistent video-to-video synthesis. In: ECCV, *Lecture Notes in Computer Science*, vol. 12353, pp. 359–378. Springer (2020)
27. Měch, R., Prusinkiewicz, P.: Visual models of plants interacting with their environment. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '96, p. 397–410. Association for Computing Machinery, New York, NY, USA (1996)
28. NASA JPL: NASADEM Merged DEM Global 1 arc second V001 (2020). URL `https://lpdaac.usgs.gov/products/nasadem_hgtv001`. [accessed 18-October-2023]
29. Palubicki, W., Makowski, M., Gajda, W., Hädrich, T., Michels, D.L., Pirk, S.: Ecoclimates: climate-response modeling of vegetation. ACM Trans. Graph. **41**(4), 155:1–155:19 (2022)
30. Panagiotou, E., Charou, E.: Procedural 3d terrain generation using generative adversarial networks. In: Workshops of the 11th EETN Conference on Artificial Intelligence 2020 co-located with the 11th EETN Conference on Artificial Intelligence (SETN 2020), *CEUR Workshop Proceedings*, vol. 2844, pp. 9–14. CEUR-WS.org (2020)
31. Pang, Y., Lin, J., Qin, T., Chen, Z.: Image-to-image translation: Methods and applications. IEEE Trans. Multim. **24**, 3859–3881 (2022)
32. Park, T., Liu, M., Wang, T., Zhu, J.: Semantic image synthesis with spatially-adaptive normalization. In: CVPR 2019, June 16-20, 2019, pp. 2337–2346. Computer Vision Foundation / IEEE (2019)
33. Park, T., Zhu, J.Y., Wang, O., Lu, J., Shechtman, E., Efros, A., Zhang, R.: Swapping autoencoder for deep image manipulation. In: Advances in Neural Information Processing Systems, vol. 33, pp. 7198–7211. Curran Associates, Inc. (2020)
34. Peel, M.C., Finlayson, B.L., McMahon, T.A.: Updated world map of the Köppen-Geiger climate classification. Hydrology and Earth System Sciences **11**(5), 1633–1644 (2007)
35. Perche, S., Peytavie, A., Benes, B., Galin, E., Guérin, E.: Authoring terrains with spatialised style. Computer Graphics Forum **42**(7), e14,936 (2023)
36. Pirk, S., Niese, T., Hädrich, T., Benes, B., Deussen, O.: Windy trees: Computing stress response for developmental tree models. ACM Trans. Graph. **33**(6) (2014)
37. Premoze, S.: Computer generation of panorama maps. In: 3rd ICA Mountain Cartography Workshop. Citeseer (2002)
38. Richter, S.R., Alhaija, H.A., Koltun, V.: Enhancing photorealism enhancement. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(2), 1700–1715 (2023)
39. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015, Proceedings, Part III, *Lecture Notes in Computer Science*, vol. 9351, pp. 234–241. Springer (2015)
40. Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., Norouzi, M.: Palette: Image-to-image diffusion models. In: ACM SIGGRAPH 2022 Conference Proceedings, SIGGRAPH '22. Association for Computing Machinery, New York, NY, USA (2022)
41. Schott, H., Paris, A., Fournier, L., Guérin, E., Galin, E.: Large-scale terrain authoring through interactive erosion simulation. ACM Trans. Graph. **42**(5) (2023)
42. Spick, R.J., Walker, J.A.: Realistic and textured terrain generation using GANs. In: European Conference on Visual Media Production, 3, pp. 1–10. ACM (2019)
43. Takahashi, H., Kanamori, Y., Endo, Y.: 3D terrain estimation from a single landscape image. Computer Animation and Virtual Worlds **33**(6), e2119 (2022)
44. Zhao, Y., Liu, H., Borovikov, I., Beirami, A., Sanjabi, M., Zaman, K.: Multi-theme generative adversarial terrain amplification. ACM Trans. Graph. **38**(6), 1–14 (2019)
45. Zhu, J., Kelly, T.: Seamless satellite-image synthesis. Comput. Graph. Forum **40**(7), 193–204 (2021)
46. Zhu, J., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: IEEE International Conference on Computer Vision, ICCV 2017, pp. 2242–2251. IEEE Computer Society (2017)
47. Zhu, J., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E.: Toward multimodal image-to-image translation. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, pp. 465–476 (2017)

A Large-scale Texture Synthesis by Tiling Outputs

To obtain a large terrain texture corresponding to a heightfield larger than the input/output resolution (i.e., 256×256) of our texture generation CNN, we seamlessly connect adjacent output textures using alpha blending. We divide an input heightfield into small tiles of 256×256 pixels with a stride of 128 pixels and feed them to the texture generation CNN to obtain small texture tiles corresponding to each heightfield tile. To blend the neighboring texture tiles, we blend the quarters (i.e., 128×128 pixel regions) of each texture tile, which are covered by four neighboring tiles except for the heightfield boundaries. By applying an alpha map whose alpha values are defined by normalized distances from the center pixel, we can obtain a large terrain texture without noticeable seams. Figure 1 shows the rendered result with an alpha-blended texture corresponding to a heightfield of approximately $5,500 \times 5,500$ pixels. For the temperature and precipitation in generating large-scale terrain textures, the user can choose whether to specify them on a per-tile basis or to use a constant value for the entire area, depending on the quality desired by the user. In the result of Figure 1, the temperature is varied by $-0.6°\mathrm{C}$ per 100m elevation difference (standard temperature reduction rate) between the entire elevation average and the per-tile elevation average. While we used the same value for precipitation for simplicity, high-quality textures were obtained.

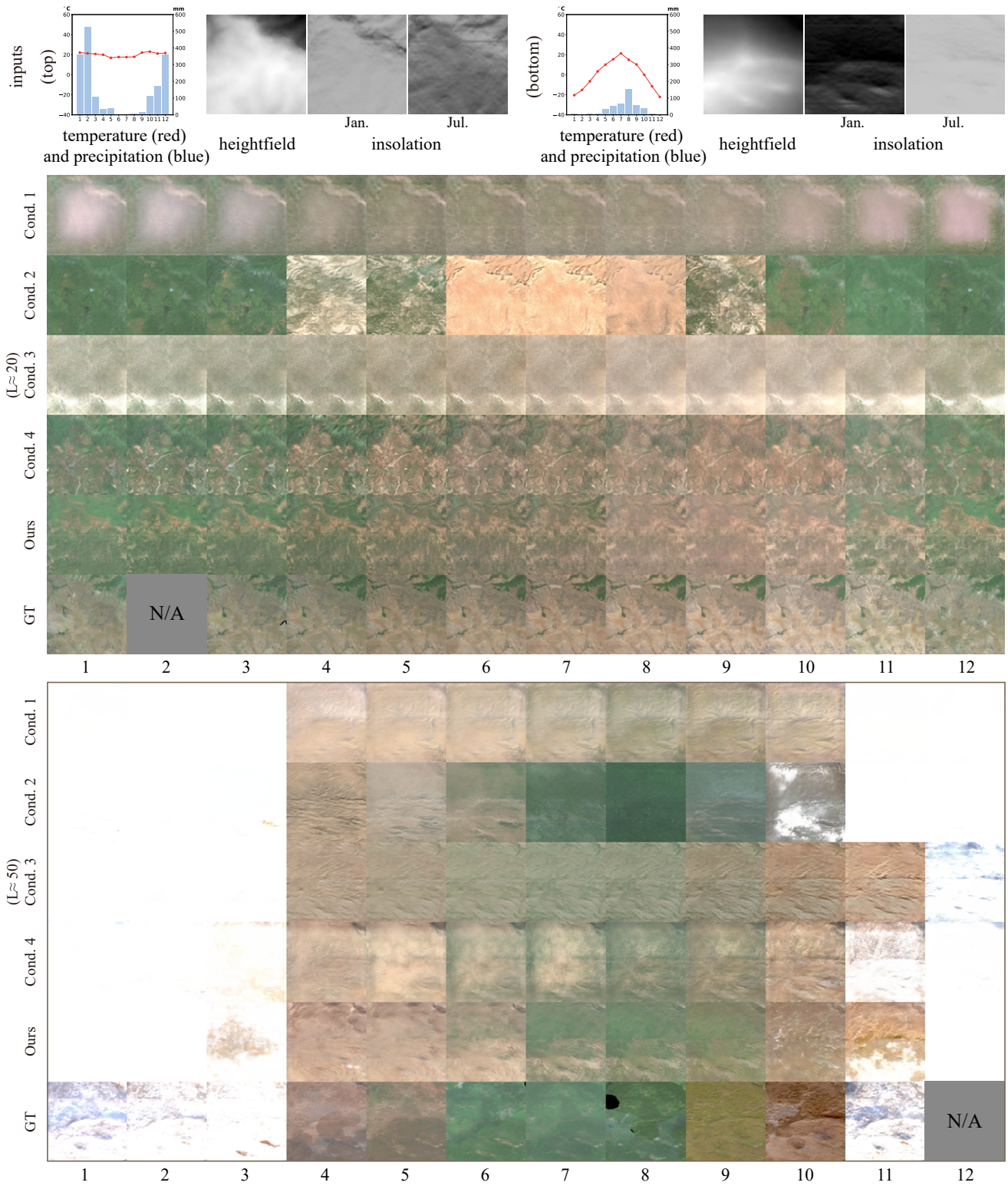

Fig. 6 Qualitative comparison of terrain textures obtained by *Conditions* 1-4 and our method. The black pixels in the GT texture are missing areas due to clouds. The grey boxes with "N/A" indicate that the textures were excluded from the dataset due to the high deficiency level. See Section 3.3 for the details.

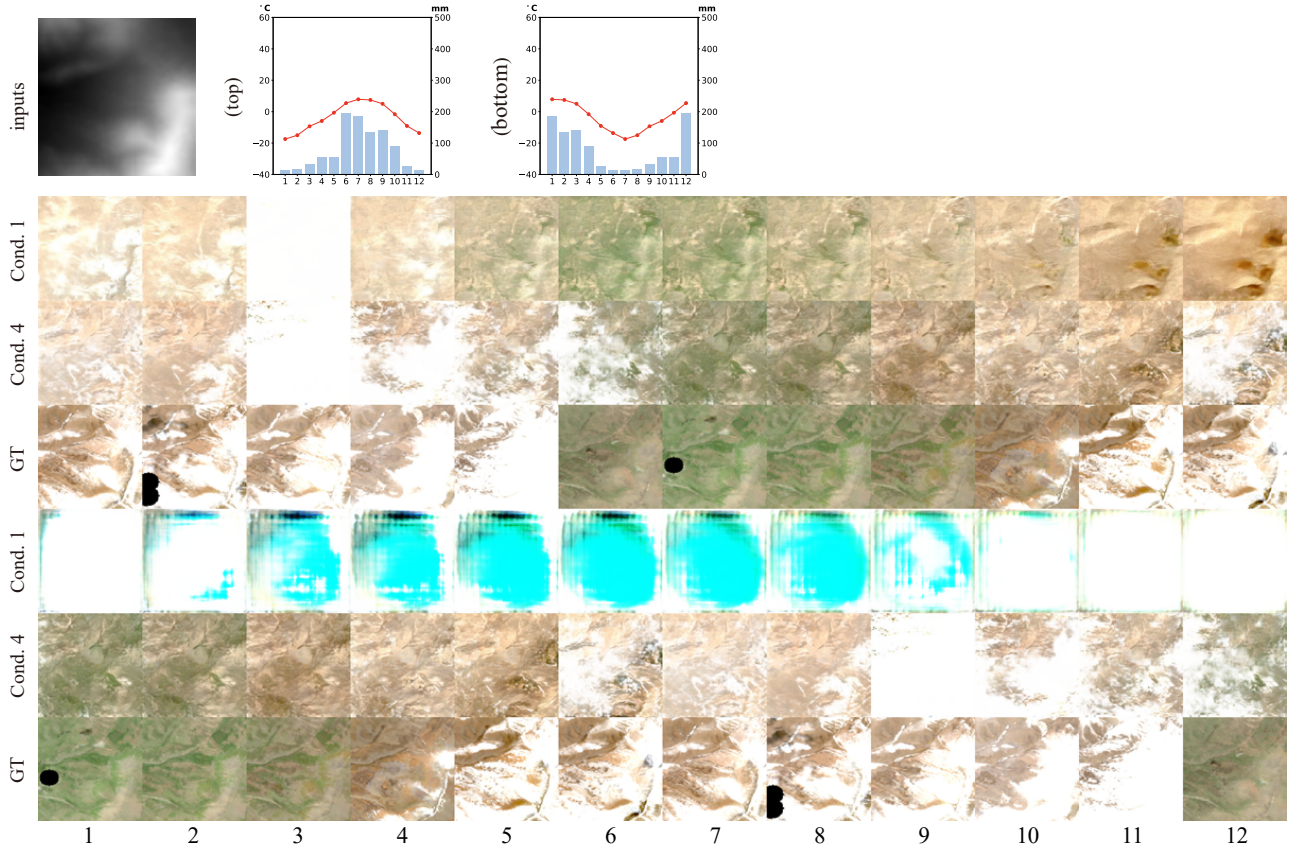

Fig. 7 Qualitative comparison of terrain textures obtained by *Conditions* 1 and 4 for contrasting the conditioning using month index maps (Condition 1) and our periodic conditioning (Condition 4). The black pixels in the GT texture are missing areas due to clouds.

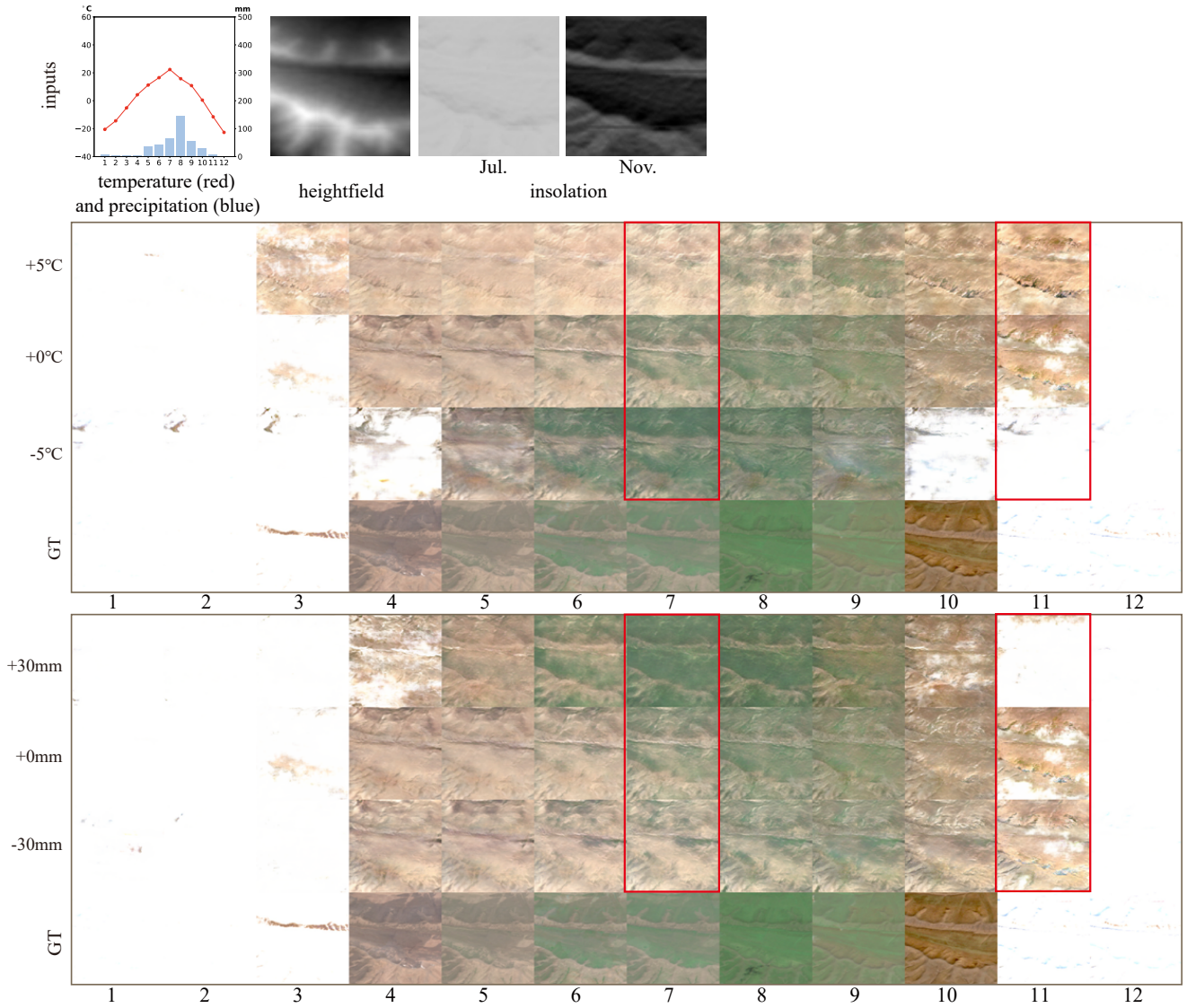

Fig. 8 Qualitative comparison of terrain textures generated by changing only the input temperature or precipitation. As shown in the red rectangles, the lower temperatures and higher precipitation increase the white areas due to snow cover in early spring in April. The higher precipitation also makes the green colors of the vegetation deeper in August, in summer.