Sketch2Map: A Game Map Design Support System Allowing Quick Hand Sketch Prototyping

Tong Wang
Cygames Research
Cygames, Inc.
Tokyo, Japan
0000-0001-9698-7054

Shuichi Kurabayashi
Cygames Research
Cygames, Inc.
Tokyo, Japan
0000-0001-5967-7727

***Abstract*—Modern open-world games require a vast extent of land with attractive diversity. However, creating such terrain is one of the most challenging tasks due to its large data size and the number of iterations in development. This paper presents a sketch-based map design support system that can help designers to create a game world with simple sketches, by transferring hand-drawn sketches of the game world such as coastline contours and river outlines into 3D terrain. This unified sketch-to-level framework involves a two-stage generative model that consists of the cGAN to convert a sketch into an elevation bitmap and a deterministic algorithm to generate a level asset. The first cGAN process probabilistically generates wider variations of elevation bitmap, including diverse terrain elements exploiting its model trained against terrain data. The second deterministic algorithm interprets the bitmap to generate the practical terrain asset. This combination of the probabilistic model and the deterministic model is highly effective in reflecting designers' intension enough while generating variations. This framework allows designers for rapid-prototyping and trials-and-errors game assets creation by converting rough sketches to rich terrain assets at interactive speed. We use more than two thousand procedural game maps pairing with corresponding sketches created by spline interpolation as our training data. The prototype implementation of this framework shows the effectiveness and practicality of our two-stage terrain generation approach.**

***Index Terms*—Map generation, cGAN, Level Design Automation**

I. Introduction

With the excessive speed of market growth, video games are now booming in every corner of the world. Game companies and indie developers are inevitably compressing authoring time to adapt to this fast-paced industry. Game design literature such as [1] and [2] emphasized the importance of paper-pen system in prototyping phase. Many designers will need to constantly prototype ideas about the game world and level layouts with the pen-paper system during different phases in game development([3]).

Whiteboard illustration of the world map, terrain, or level layout often shines valuable design insights that will then be implemented as actual assets in the game engine for further test iterations. But such implementation efforts often cost time and effort, thus harm the efficiency and smoothness of the creation process. Prototyping aiding tools are becoming necessary to help alleviate such idea-to-implement efforts by directly convert sketches to rich game world representations.

This paper present a case study of such support system called *Sketch2Map* on automatically converting sketches that can be composed in seconds with a casual drawing of continent contours, islands, coast outline of the game world into game assets such as elevation map or terrain geometry by a conditional generative adversarial neural network (cGAN) in interactive to real-time speed. We also proposed a unified sketch-to-level framework that is adaptive to multiple use cases in designing schemes.

The main contribution of the paper is a game map authoring system that can support fast and precise sketch-based automatic prototyping. To generate a faithful prototype from a rough sketch, we constructed a two-stage generative model based on cGAN (conditional generative adversarial networks) that help to synthesize levels or terrains based on simple contour sketches.

The sketch2map framework provides advantages over both direct hand-design and randomized procedural generation: Compared with direct hand-design in-game engine: This system will help save designers' time by filling detailed level textures: heightmap in the case of game maps or terrains automatically with a carefully trained generative model. Compared with randomized procedural generation: This framework introduces strong human guidance in the creation process. The guidance scheme will show human experiences and insights in top-level creation with a macro perspective.

II. Background and Related Work

A. Game Map Generation as Level Design

The game world often serves as an environment for world-building or stage to hold scattered game events. Recent years world and terrain as macro-level design received considerable critical attention with some successful releases of free-roaming open-world games such as *The Legend of Zelda: Breath of the Wild*([4]). Experienced terrain designers can provide more valuable insights in open-world design with faster iteration with a sketch-based prototyping tool.

978-1-7281-4533-4/20/$31.00 ©2020 IEEE

B. Sketch-based Authoring System

A hand-drawn sketch is a universal expressing tool of human ideas, perspectives, and emotions. A sketch is the most intuitive and convenient prototyping work that can be composed by anybody at any time ([5]). The topic of using the sketch in aiding game content creation has been discussed in a fine-grained context, for example, in lower level game asset generation (*bit-level* PCG as discussed in [6]).

Numerous studies have attempted to use GAN(Generative Adversarial Network, [7]) in generative work. We used a modified pix2pix [8] structure in the pipeline to create connections between sketch and game world design. A recent work discussing sketch-based authoring of terrain with cGAN is [9], which provided an approach to build a sketch-based authoring system with simple user-inputs and high user control. Comparing with this work, we used a unique two-stage strategy that focused on the more precise reflection of map design such as coastline contour rather than terrain elements. [10] is another work that discussed the sketch-based generative model as a general UI/UX generation tool.

We also draw inspiration from ArchiGAN([11]), which used a simple authoring system to edit contour and then fill textures learned from indoor architects' apartment building design. Comparing with this work, we construct the pipeline from a game designer's perspective and make sketch-map pairs much more flexible to create.

III. Pipeline of Sketch2Map

Figure 1 shows the whole pipeline of the sketch2map system. *Map* will also be referred to as a special case of *level* in the following text. The functionality of each component will be explained as following:

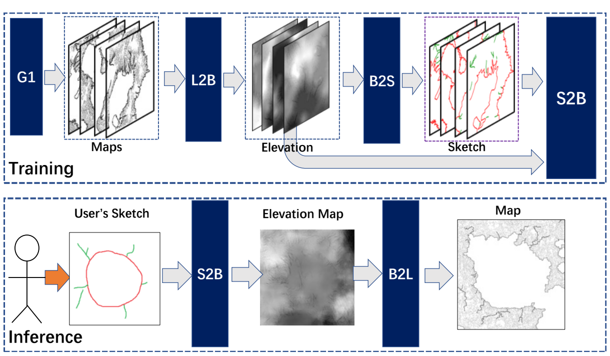


Fig. 1: Overview of the pipeline.

1) *G1* is the generator of training data. We will create the training data by a random game world map generator as well as from real world elevation data.
2) *Level2Bitmap(L2B)*: Levels/Maps will then be transformed into a more precise multi-channel 2D bitmap representation.
3) *Bitmaps2Sketch(B2S)*: Use a sketch to describe the corresponding bitmap to generate a sketch-level pair.
4) *Sketch2Bitmaps(S2B)*: A cGAN model that learns the mapping from sketches to level bitmaps by training with the sketch-bitmap pair dataset. We use a customized pix2pix framework([12]) here.
5) *Bitmaps2Level(B2L)*: A deterministic logic that transfers the bitmap syntax descriptions back to the actual level asset.

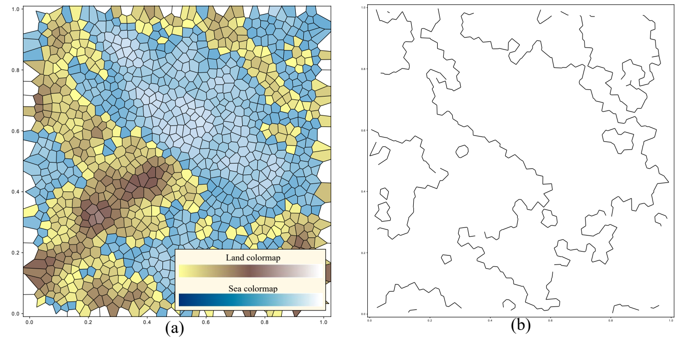


Fig. 2: Generate training data using procedural generation. (a). Procedural or real elevation mapped on a Voronoi tessellation in 2D space. (b). Get the contour line that separate land and sea.

In this paper, we will focus on the case study of game map generation. Users will draw the outline/contours of continents and general ground features such as rivers with the specified color. The following section will describe in detail about individual components that consist the Sketch2Map pipeline.

A. Dataset Generation: G1, L2B and B2S

G1: In the case of game map generation, we create a world map and its sketch to form a pair from the procedurally generated game world. Figure 2 shows the process of generating original game world. 2D space will be sampled and tessellated with blue-noise samples. Heightmaps generated with multiple Perlin noise will be applied to the Voronoi tessellation of blue-noise samples. Then we can get the outline of game world continents by extract edges that lie on sea level in the Voronoi diagram. Rivers and erosion effects ([9]) can be added accordingly as well. We also support using high accuracy map of real-world global terrain elevations for the training.

L2B: The Level2Bitmap component is in charge of the preparation and enhancement of training data by representing game world syntax in multi-channel discretized 2D images.

B2S: The Bitmap2Sketch component will be in charge of converting syntax bitmaps to a certain style of sketch representation automatically to generate training data. We use spline interpolation in our fantasy map generator case study to generate contour sketches from the level bitmap as training data. Figure 3 shows how spline interpolation with different parameters can be used to describe a complex outline from a terrain elevation bitmap generated from previous stages. We mix three sketch types with different smoothness in our training data set.

B. Learning: S2B

S2B: Now let's look at the core part of our pipeline: sketch2bitmap. It will use training data from the previous

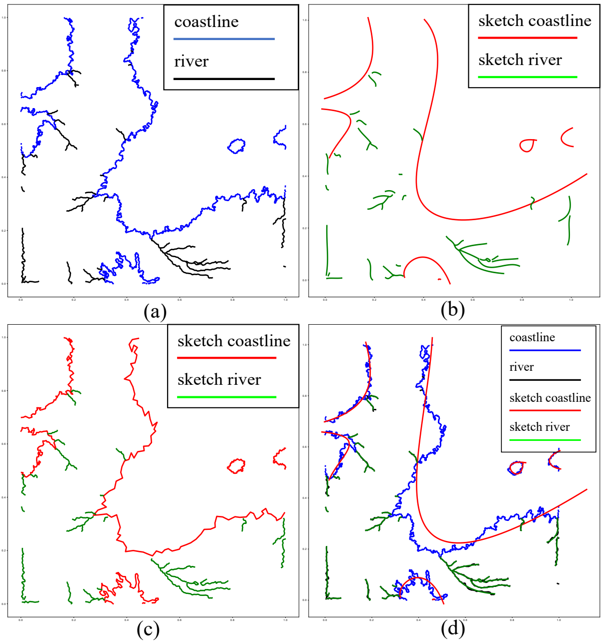


Fig. 3: Sketch data generated using spline interpolation from the generated contour. (a). the original continent contour line. (b). A smooth sketch generated by spline interpolation. (c). A more precise spline interpolation. (d). overlay comparison.

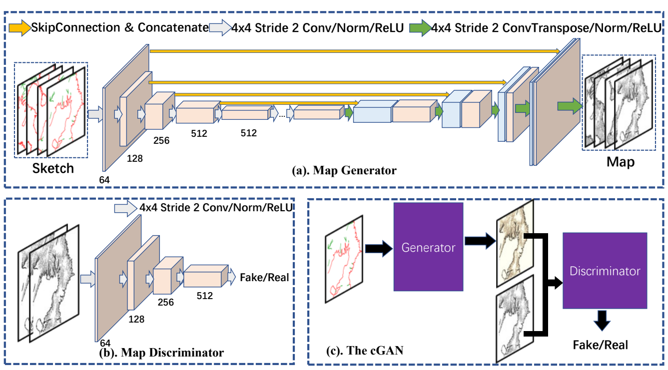


Fig. 4: Architecture of the cGAN as map generator. The generator is a U-net structure as described in [13]. Discriminator is a PatchGAN similar to [12].

stages and then feed the sketch-level pair into a conditional GAN as is shown in Figure 4.

To faithfully translate the designer's level design idea into actual game map assets, we added another segmentation cGAN stage to enhance the results. We observed that in most game map designs, the faithfulness of conveying design ideas would often be a top priority. Also, note that heightmap in sea area will often be ignored in actual implementation. With such observation bear in mind, we first feed sketches into a segmentation cGAN in order to generate a proper segmentation image that separates land and sea much more precisely. It will then be used as input for the next cGAN stage to generate heightmap as textures inside the separation. The two-stage will be trained separately in order to generate more genuine results. See the comparison of our two-stage architecture with baseline in section IV.

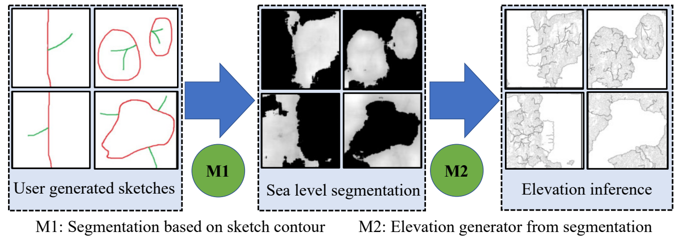


Fig. 5: The two stage strategy. M1 is the cGAN that mapping sketch and sea-land segmentation. M2 will convert the segmentation to elevation.

IV. Results and Comparison

Here we show some results generated with sketch2map. Figure 6 shows the results generated with a simple sketch in Figure 6(a). The sketch depicting a rough idea of how the world will present by coastline contour circles in red lines and several river indications in green lines. (b) is the 2D map generated with sketch2map with a rendering technique from [14]. (c) is the elevation map rendered in 3D with a customized terrain color map and (d) is the perspective view of the generated map which can be tested directly in a game engine.

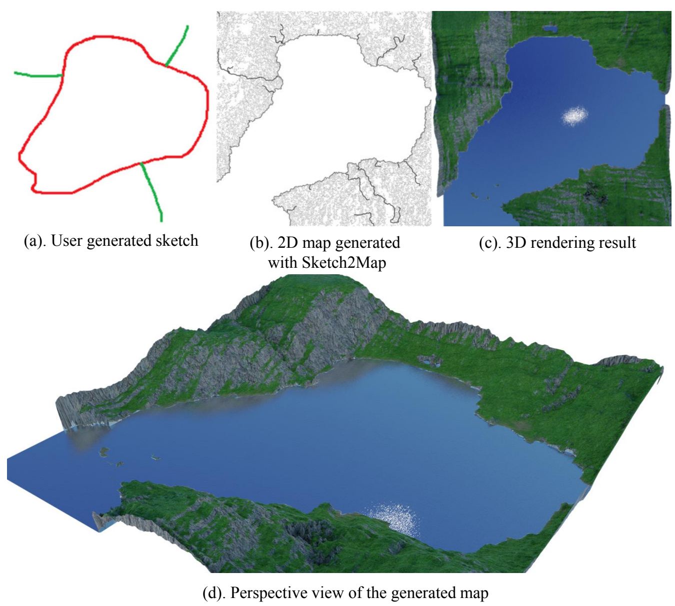
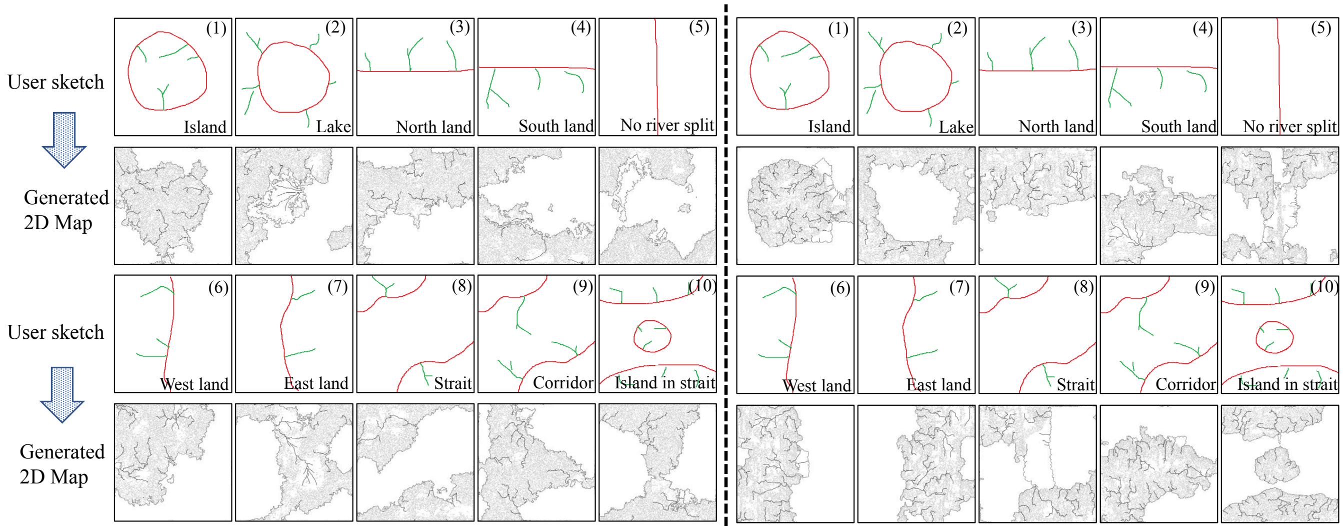


Fig. 6: Sketch2map converts user sketch to 2D map and 3D terrain.

We compared our two-stage method with the baseline one stage pix2pix cGAN framework by testing ten corner case sketches. Each sketch represents a typical continent layout in a local region of a fantasy world map. Notice how our method can represent user ideas more precisely for almost every case.



(a). One-stage (b). Our two-stage framework

Fig. 7: Comparison of our two-stage results vs. one stage result. (a). is generated with one stage. and (b). our two-stage strategy which is more faithful in reflecting designer's perspective than (a).

V. Conclusion and Limitation

In this paper, we proposed sketch2map, a generative framework that aims to faithfully translate the designer's simple contour sketch representing coastlines and rivers to a game world map that can be tested in game directly. Each component of this framework is highly customizable and adaptable to different demands in sketch-based world map or terrain creation workflow.

One limitation is that designers need to be trained to convey level syntax in a certain style. Another limitation is that the current framework lacks good support of lakes, that the network tends to expand water level till map edges. Some cases are hard for the network to create, for example, the *No river split* case shown in Figure 7. Adaptation to different sketch styles and more faithful translation with the dataset that contain different kinds of realistic geography will also be possible future work.

References

[1] J. Feil and M. Scattergood, *Beginning game level design*. Thomson Course Technology, 2005.

[2] T. Castillo and J. Novak, *Game Development Essentials: Game Level Design*. Delmar Learning, 2008.

[3] C. W. Totten, *Architectural Approach to Level Design*. Boca Raton, FL, USA: CRC Press, 2019.

[4] S. T. Takuhiro Dohta Hidemaro Fujibayashi. (Mar. 2017, Accessed Jan 15, 2020). Change and constant: Breaking conventions with the legend of zelda: Breath of the wild, [Online]. Available: https://www.gdcvault.com/play/1024562/Change-and-Constant-Breaking-Conventions.

[5] P. Xu, T. M. Hospedales, Q. Yin, Y.-Z. Song, T. Xiang, and L. Wang, "Deep learning for free-hand sketch: A survey and a toolbox," 2020. arXiv: 2001.02600 [cs.CV].

[6] M. Hendrikx, S. Meijer, J. Van Der Velden, and A. Iosup, "Procedural content generation for games: A survey," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 9, no. 1, Feb. 2013.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[9] É. Guérin, J. Digne, E. Galin, A. Peytavie, C. Wolf, B. Benes, and B. Martinez, "Interactive example-based terrain authoring with conditional generative adversarial networks," *Acm Transactions on Graphics (TOG)*, vol. 36, no. 6, pp. 228–241, Nov. 2017.

[10] A. Robinson, *Sketch2code: Generating a website from a paper mockup*, 2019. arXiv: 1905.13750 [cs.CV].

[11] S. Chaillou. (Jul. 2019, Accessed on Feb 3, 2020). Archigan: A generative stack for apartment building design, [Online]. Available: https://devblogs.nvidia.com/archigan-generative-stack-apartment-building-design/.

[12] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[14] M. O'Leary. (Aug. 2016, Accessed on Feb 6, 2020). Generating fantasy maps, [Online]. Available: https://mewo2.com/notes/terrain/.