2021 IEEE 10th Global Conference on Consumer Electronics (GCCE), Oct. 12-15, 2021.

An Aerial Cinematographer AI for Settlements in Minecraft–Toward Their Crowd Assessment

Hao Jia and Ruck Thawonmas
Graduate School of Information Science and Engineering
Ritsumeikan University
Kusatsu, Japan
ruck@is.ritsumei.ac.jp

Pujana Paliyawan
Research Organization of Science and Technology
Ritsumeikan University
Kusatsu, Japan
pujana.p@gmail.com

***Abstract*—This paper proposes an aerial cinematographer AI for shooting videos of settlements in Minecraft worlds. A supervised learning approach is adopted to train the AI using a decision tree with movement behavior data of a target player. Results on unseen data show the effectiveness of the proposed AI when its position is close to the settlement center.**

***Index Terms*—Aerial Cinematographer, Decision Tree, Generative Design in Minecraft Competition**

I. Introduction

The Generative Design in Minecraft Competition (GDMC)[1] is a competition where participants create AI programs (generators) that generate settlements on unknown Minecraft maps. The competition's main aim is to advance research in computational creativity and co-creativity. At present, generated settlements will be evaluated by human judges who are experts in relevant areas [1]. Judges will be asked to evaluate each generator in four categories: Adaptability, Functionality, Narrative and Aesthetics. To assess the generality of generators, the number of unknown maps should be large, but this would impose more burdens on judges, in particular with more and more submissions.

We argue that the above issue can be solved with crowd assessment. In this approach, a video concisely summarizing the contents of a settlement generated by a generator of interest on each unknown map will be created by another AI (cinematographer AI) that has been separately trained to imitate salient behaviors of each judge during their evaluation process. Such videos–in all possible combinations of generators, maps, and judges–will then be assessed by a large number of audiences.

As a first step toward this approach, this work focuses on the visual aesthetics criterion, the most subjective one among the four. In addition, for this criterion assessment, our work is limited to aerial cinematography of a target settlement, often used in introduction of settlements[2]. To allow for interpretability of learned judge behaviors, a decision tree method is used. Besides GDMC, the proposed cinematographer AI can be used by general users of Minecraft, or even adapted to other sandbox games, to introduce their worlds to others.

This work was supported in part by KAKENHI KIBAN (C) 19K12291.

[1]https://gendesignmc.engineering.nyu.edu/

[2]A settlement by our 2020 generator ICE_JIT: https://tinyurl.com/3z7t7dw9

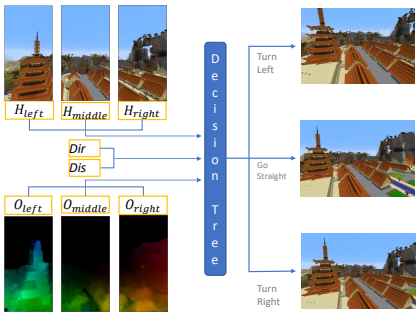

Fig. 1. Overview of the proposed cinematographer AI.

II. Existing Work

To the best of our knowledge, there is no work related to aerial cinematography in Minecraft using a supervised learning approach. Most related recent work targeting real environments used supervised learning [2], like ours, and reinforcement learning [3]. However, both used deep learning that lacks interpretability, a useful property to understand judges' behaviors during the evaluation process albeit it is beyond the scope of this demo paper.

III. Proposed System

Figure 1 shows an overview of the proposed cinematographer AI. It takes four types of input features and outputs a movement direction. The four input features are as follows: (1) the direction (Dir) from the current position and orientation toward the settlement center–having three possible discrete values for left, straight, or right–(2) the distance (Dis) between the current position and the settlement center–having a continuous value–(3) the color entropy of the current scene–split into three sub-scenes: left sub-scene (H_{left}), middle sub-scene (H_{middle}), and right sub-scene (H_{right})–and (4) the optical flow of the current scene–also split into three sub-scenes: left sub-scene (O_{left}), middle sub-scene (O_{middle}), and right sub-scene (O_{right}). The last two features have continuous values and are described in detail in Sec. III-A. One of the three movement directions will be decided by the AI as its output: turn left, go straight, or turn right; after rotating to the resulting direction, it moves forward 20 meters, 16 pixels per meter in Minecraft, before making another decision.

In this work, an area of 600x600 meters with the AI's initial position ($x_0 = 0, z_0 = 0$) in the middle is explored to find a settlement center. This area is divided into grids with a size of 60x60 meters. Information on objects in these grids is used to heuristically decide the settlement center and the altitude of the proposed AI. The location of the settlement center is set to the centroid of grids highly condensed with man-made objects on their surfaces, more precisely, grids with a total number of objects weighted by their object IDs above the mean value + (2 $\times$ the standard deviation). Note here that we consider Minecraft objects with higher IDs more likely to be man-made ones. The altitude of the proposed AI is empirically set to 10 meters above the highest object in the current grid. The orientation of the AI at its initial position is set toward the settlement center, and the initial altitude (y_0) is to 10 meters above surface.

A. Color Entropy and Optical Flow Features

1) Color Entropy: Color entropy [4] indicates the amount of information about the color of each image. Our idea behind using it as one of the features is that a player might want to proceed toward a place with a high color entropy value, e.g., turning left when the color entropy of the left sub-scene is highest. The color entropy of a given sub-scene H_x (for $x =$ *left, middle, right*) is calculated separately.

2) Optical Flow: Optical flow intensity measures the intensity of movement of each pixel, representing visual clutter. Our rationale for using this feature is that the player should be motivated to move toward a target based on visual clutter cues. Here, the Gunnar Farneback's algorithm [5] is used to effectively obtain the dense optical flow, the optical flow for all pixels in the image. Dense optical flow was also used in Huang et al. [2]. This algorithm is applied to a whole scene (screen), and it returns a 2-channel array containing intensity and direction vectors, as visually shown in Fig. 1 separately for each sub-scene. The sum of the intensity in each sub-scene is used as its feature (O_{left}, O_{middle} and O_{right}).

3) Implementation: We use the scikit-learn version of decision tree[3]. Our empirical findings suggest that players tend to have different movement behaviors according to the distance (d) between the current position and the settlement center. As a result, we deploy separate decision trees for three normalized distance ranges: $d < 1/3$ (short), $1/3 \leqq d < 2/3$ (medium), and $d \geqq 2/3$ (long). In actual use, a decision tree trained using data of a target player in a distance range of interest will be used to create the player's behavior in that range.

IV. System Testing

We tested the proposed system using a typical GDMC map, the top three generators in 2020, and three players. The three players were asked to assess all the generators (one assessment per generator), in a Latin square order, from the aesthetics perspective. During their evaluation, they used the same system as the proposed system, except that they had to decide one of the three directions (turn left, go straight, or turn right) on their own every 20 meters. Their data were collected and used in training and evaluating the proposed system as described below.

TABLE I
Comparison of the proposed system, using a decision tree method, with the baseline system, always moving toward the settlement center.

Distance Range	Method	Player 1	Player 2	Player 3
$d < 1/3$	Decision Tree	**0.75**	**0.72**	0.19
	Toward Center	0.00	0.56	**0.20**
$1/3 \leqq d < 2/3$	Decision Tree	**0.32**	**0.53**	0.44
	Toward Center	0.29	0.42	**0.63**
$d \geqq 2/3$	Decision Tree	0.33	0.33	0.75
	Toward Center	**0.75**	**1.00**	**1.00**

For each player, data collected during the 2nd assessment are used as testing data, while the rest as training data. The data are then further divided into three groups according to the distance range. The hyper-parameters of each decision tree, including a subset of features, are a combination of those that maximizes the average accuracy when the 1st-assessment data and the 3rd-assessment data are used in two-fold cross validation. Resulting hyper-parameters are summarized in our supplementary page[4] where the source code and data are also available.

Table I shows the accuracy on the test data of the proposed system and that of a baseline system that always moves toward the settlement center. It can be seen that, except for Player 3, the proposed system outperforms the baseline system in the short and medium distance ranges. For the long distance range, the results suggest that the baseline system should be used in stead, which makes sense. As a result, the proposed system has been modified such that the AI always moves toward the settlement center during the long distance range. The aforementioned supplementary page also lists videos by each player during their 2nd assessment, videos by the post-modification proposed system for respective settlements, and videos by the baseline system for respective settlements. Our future work includes improvement of the accuracy for the medium distance range.

References

[1] C. Salge et al., "The AI Settlement Generation Challenge in Minecraft: First Year Report," *KI-Künstliche Intelligenz*, vol. 34.1, pp. 19–31, 2020.

[2] C. Huang et al., "Learning to Film from Professional Human Motion Videos." In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 4244–4253, 2019.

[3] M. Gschwindt et al., "Can a Robot Become a Movie Director? Learning Artistic Principles for Aerial Cinematography," In *Proc. 2019 IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, pp. 1107–1114, 2019.

[4] J. Shen et al., "Optimal Illumination for Visual Enhancement Based on Color Entropy Evaluation," *Optics Express*, vol. 24, issue 17, pp. 19788–19800, 2016.

[5] G. Farnebäck, "Two-Frame Motion Estimation Based on Polynomial Expansion," *Lecture Notes in Computer Science 2749* (*Scandinavian conference on Image analysis*), pp. 363–370, 2003.

[3] https://scikit-learn.org/stable/modules/tree.html

[4] https://moss-j.github.io/aerial/