Command2Vec: Feature Learning of 3D Modeling Behavior Sequence—A Case Study on "Spiral-stair"

Wen Gao[1](✉), Xuanming Zhang[2](✉), Weixin Huang[1](✉), and Shaohang Shi[1](✉)

[1] School of Architecture, Tsinghua University, Beijing 100084, China
{gaow19,shish19}@mails.tsinghua.edu.cn, huangwx@tsinghua.edu.cn
[2] School of Software, Tsinghua University, Beijing 100084, China
zhang-xm20@mails.tsinghua.edu.cn

Abstract. In this study, we applied machine learning to mine the event logs generated in modeling process for behavior sequence clustering. The motivation for the study is to develop cognitively intelligent 3D tools through process mining which has been a hot area in recent years. In this study, we develop a novel classification method Command2Vec to perceive, learn and classify different design behavior during 3D-modeling aided design process. The method is applied in a case study of 112 participate students on a 'Spiral-stair' modeling task. By extracting the event logs generated in each participate student's modeling process into a new data structures: 'command graph', we classified participants' behavior sequences from final 99 valid event logs into certain groups using our novel Command2Vec. To verify the effectiveness of our classification, we invited five experts with extensive modeling experience to grade the classification results. The final grading shows that our algorithm performs well in certain grouping of classification with significant features.

Keywords: Process mining · Design cognition · 3D-modeling · Machine learning

1 Introduction

Since Computer Aided Design introduced into design industries, 3D-modeling aided design has become one of the most important ways for design creation nowadays. Through advanced 3D-modling design tools, designers could create, edit, test and delete 3D geometries freely and rapidly. As a 'visual reasoning' process [1], cognitive process of design has been greatly augmented for 3D tool is able to provide accurate 3D visual feedbacks for designers to efficiently optimize their designs [2]. With deep permeation of 3D-modeling, design thinking is at the same time evolving from human behavior to human-3D tool integrated behavior. Hence, a more cognitively intelligent human-computer interactive system becomes an urgent quest from nowadays designers. Studies on application of artificial intelligence technologies to design have always been hot area [3]. Although, many recent studies on applying machine learning (ML) in design process

P. F. Yuan et al. (Eds.): CDRF 2021, *Proceedings of the 2021 DigitalFUTURES*, pp. 45–54, 2022.
https://doi.org/10.1007/978-981-16-5983-6_5

are positive responses to the quest [4, 5], researches on application of ML in cognitively intelligent 3D tool are still insufficient. Designing is believed a unique process [6]. That to develop ML which especially adapts to 3D-modeling aided design process should draw more attention.

In order to build an intelligent 3D tool to be able to perceive, learn and classify design representations similarly as designers, classification is a primary ability for computer which facilitates evaluating, comparing and decision making. In this study, we proposed a novel embedding algorithm Command2Vec (available on https://github.com/initxuan/Command2Vec) for 3D tool to identify and classify different modeling processes. For the purpose of letting 3D tool firstly be able to 'perceive and learn' the process, a structure of data 'command graph' was developed. In form of directed graph, 'command graph' was extracted from event log and could reflect relationships between commands and objects during modeling. Command graph was then embedded into vectors based on Word2Vec, a natural language processing algorithm, for further clustering by K-mean++. The whole algorithm formed our Command2Vec. In this study, an experiment was conducted with 112 junior students participating in a 'spiral-stair' modeling task. This experiment is to explain and test the performance of Command2Vec. A popular software Rhinoceros 3D (Rhino) is chosen as the 3D tool in this study. An illustration is given in Fig. 1 to show our research workflow.

After the experiment, 99-participant's data were screened as valid data for study. By constructing command graph from event logs with a total of 4728 nodes, 10059 edges, 72 different commands, final 99 command graphs were obtained without repetition. Through Command2Vec, 6 groups were finally achieved as clustering results. Five

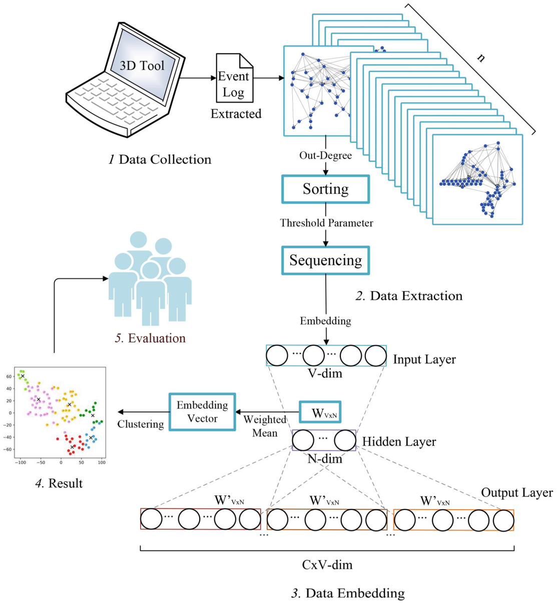


Fig. 1. Work flow of research

modeling experts with extensive design experience were invited to rate the classification. High score was got in groups with significant features. In the end, the results were quantitatively analyzed and discussed (Fig. 1).

2 Related Work

Our study falls into the research area of process mining in process discovery. “The goal of process discovery is to learn a model based on an event log” [7]. Event log is a kind of behavioral record happening ubiquitously in our digital world. However, that to learn and predict human behavior through process discovery is a very complicated subject of research. Process mining in design industries based on event log starts to attract attentions in recent years. Tao et al. pointed out that more attention should be paid on how to use the big data of cyber-physical models generated in product designing process to better serve the design management in the entire product lifecycle [8]. In the field of design, some research works have been done by mining event log generated by the cyber-physical models. Yarmohammadi et al. attempted to characterize the performance of modeler based on time from BIM log using sequence based mining algorithm [9]. Pan and Zhang used BIM log as data source to explore designers’ preference as well as productivity pattern to improve managing efficiency. They proposed a novel clustering model for the exploration [10]. Other works done by Pan, Y., et al. include knowledge discovery and designers grouping using clustering algorithm based on BIM log [11, 12].

3 Methodologies

3.1 Data Preparing

In order to let computers ‘understand’ human modeling behavior, a novel data structure was proposed. In this study, we adopted directed graph to represent the behavior sequence in 3D modeling aided design. This data structure is extracted from event log generated by user modeling process. Figure 2 illustrates the proposed data structure. The nodes in the directed graph are of two kinds: command nodes and object nodes. The naming method for the command node is the sequential number of the command with command name operated by user. The naming method for object nodes is the sequential number in which the 3D model objects are generated. The command node and the object node are connected by a directed line which represents the cause-and-effect relation. For example, as shown in Fig. 2(a), the first command generates the first object; first object causes second command’s operation which then generates second and third objects, and so on. After obtaining such a directed graph, the computer can find the command-path of any generated object, as well as object-path to any command. This form of data structure can largely reflect the cognitive activity during 3D modeling (Fig. 2).

In the case study, we observed that when people were asked how to complete a modeling task, they tended to answer the question by describing the operation process with key commands. On this basis, we simplify the command-object graph (Fig. 2(a)) by extracting subgraph of command nodes, and obtain the final command graph (Fig. 2(b)). During the extracting process, subgraph of deleted object as well as command with no object generation are excluded. The command graph becomes our data source for feature learning and clustering later.

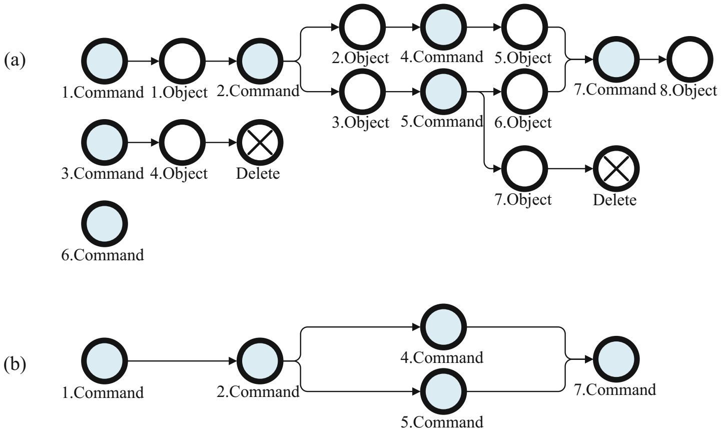


Fig. 2. The process of event log extraction into command graph (a) command-object graph, (b) extracted subgraph: command graph.

3.2 Embedding

3.2.1 Algorithm Comparison

After getting the command graph which represent modeling behavior of designer, a novel embedding algorithm based on Word2Vec: Command2Vec, was developed to learn feature of command graph.

We first applied Node2Vec [14] and Graph2Vec [15], two graph-based embedding algorithms, to embed the command graph respectively. However, the clustering results are not explainable both within classified groups or in between them. Due to the reason that these two algorithms are only sensitive to the topology of the graph, but not the semantics and sequence of it, we turned to the Word2Vec embedding algorithm applied in natural language processing (NLP).

However, the results of learning the entire network path by Word2Vec was still inexplicable by evaluators. An assumption was the highly frequent "noise commands" in the modeling process such as gumball transform, drag, join that appeared in almost every participant's commands set. We further process the input data, command graph, by extracting the top 7 out-degree ranking commands and form into key-command sentences. The key-command sentences performed well that proved assumption. Following are the specific algorithm description.

3.3 Command2Vec

Let G $= \{G_1, G_2, \ldots, G_n\}$ represent the set of all command graphs, where $G_i = (V_i, E_i)$ represents the i$^{\text{th}}$ person's command graph, V_i represents node and E_i represents edge. First, use E_i to calculate the out-degree OD_{iv_j} of each command node $v_j \in V_i$ in the command graph G_i. When the nodes have the same out-degree, sort them in ascending order according to the sequence of occurring of the nodes, and obtain an ordered array N_i containing all command nodes v_j. The same processing is performed on all command graphs, and the ordered array $N = \{N_1, N_2, \ldots N_n\}$ corresponding to all command graphs is obtained, where N_i is the ordered array corresponding to the ith command graph.

N_i is screened according to the incoming threshold parameter θ, and the first θ command nodes $v_j(j \in [1, \theta])$ in N_i are selected to form a new sequence $N_i^{'}$. In order to avoid interference with the final result, we will filter out the two commands “Redo” and “Delete” during the screening process. All new sequences form a new set $N^{'} = \left\{N_1^{'}, N_2^{'}, \ldots N_n^{'}\right\}$.

Since each command node in the command graph represents a specific command in 3D-tool, all elements in the set N' are sequences composed of specific meaning. When we regard commands with the same name appearing in different positions as the same command, the number of different commands that constitute N' is limited.

Let C $= \{c_1, c_2, \ldots c_m\}$ represent the set of different commands that constitute N', where $c_i(i \in [1, m])$ represents a specific commands. Let $f : C \rightarrow \mathbb{R}^d$ be the mapping function from command to feature representation, where d is a parameter that specifies the dimension of the feature vector. Equivalently, f is a matrix with $|C| \times d$ parameters. The vector in the ith row of this matrix is the embedding vector of the command c_i. We use the Skip-gram [16] architecture to learn the matrix f based on the set N'. After learning the embedding vector corresponding to each command, we will finally obtain the embedding vector of the ith personal command graph according to the commands contained in $N_i^{'}$, using the weighted average method of out-of-degree (Eq. (1)).

$$Vec_i = \frac{1}{\sum_{c_j^{'} \in N_i^{'}} OD_{ic_j^{'}}} \left(\sum_{c_j^{'} \in N_i^{'}} OD_{ic_j^{'}} \times f\left(c_j^{'}\right) \right) \tag{1}$$

Where $OD_{ic_j^{'}}$ represents the degree of output of the command $c_j^{'}$ corresponding to the command node in the ith command graph.

3.4 Clustering

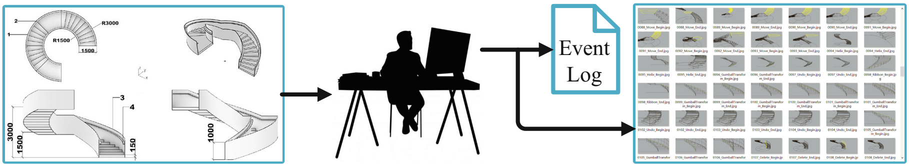
The purpose of using clustering algorithm is to test how effective our novel embedding method is. K-mean++ clustering algorithm [17] and t-SNE [18] were applied to cluster and visualize results.

4 Experiment

An experiment was conducted with 112 participants of junior students majored in architecture on a ‘spiral-stair’ modeling task. This experiment was to collecting event logs by different individuals on same goal. The chosen 3D-modeling software was Rhino in version 6. In this experiment, a requirement description for the task is shown in Fig. 3(a). ‘Spiral-stair’ is a well-defined architectural problem with certain design constraints [13]. The majority of participants were beginners to Rhino. This study chose to define clear and short modeling tasks in order to obtain not too complex and comparable modeling process data. Beginners were chosen to ensure the authenticity and diversity of the data for sophisticated modelers tend to have similar solutions on modeling.

Using Rhino API, a backstage plugin to record both instructional and geometric event log was developed to run on Rhino 6. Event logs were collected based on one by one Rhino commands while modeling (Fig. 3(b)). Each command's records included the basic information such as command's name, beginning and ending time, command's result. Also, all Rhino objects' GUIDs happened in command's history were also recorded. For instance, command 'Circle' (center, radius) requires a center coordinates and a radius number to generate a circle curve in Rhino. If user choose a center by clicking on another existing object in Rhino, then the clicked object's GUID would be recorded. In this way, relationships between command and its related 3D objects can be 'perceived' by computer.

In addition to event log, screenshot of active viewport on Rhino per user-instruction (rhino command) was saved simultaneously during each modeling process. The screenshots were collected as a sequence of images named in command's index and command's name (Fig. 3(c)).



(a) Task (b) Modeling (c) Event Log & Screenshots

Fig. 3. Requirement information of the experiment (a) 'spiral-stair' task (b) participant modeling (c) records of event log and sequence of screenshots of the active viewports

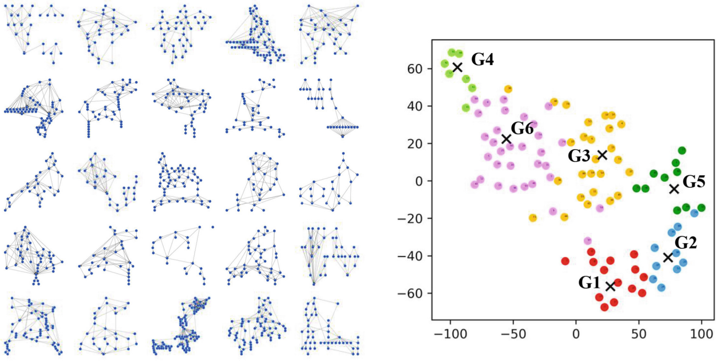
5 Results

5.1 Experiment Results

In this case study, 99 out of 112 total participants were screened valid for further analysis. The rules for data screening are: 1. Data that has not been recorded from the beginning or has not been recorded; 2. Data not modelled from an empty file; 3. Incomplete data due to various other reasons. According to our proposed method, a total of 4728 nodes, 10059 edges, 72 different commands were extracted from event logs. Finally, 99 command graphs were created with no repetition. Figure 4(a) shows the variety of command graph examples. Through our novel embedding algorithm: Command2Vec, the behavior sequence of modelling process from 99 participants were characterized into 6 groups as shown in Fig. 4(b).

5.2 Evaluation

In this study, external analysis method was used for effectiveness evaluation of clustering results. Five architectural designers with more than 5-year design experience were



(a) (b)

Fig. 4. (a) examples of randomly picked command graphs in the experiment and (b) clustering result of all embedded command graphs from 99 effective data.

invited to evaluate the result. The evaluation criteria are divided into two dimensions: the similarity within current evaluating group and the difference of current group to rest of the groups. Based on the seven-point grading system, the final rating form was show in Table 1. Each participant's full screenshots in the experiment were compiled into videos for evaluators to watch and rate. By random sampling, each evaluator only needs to watch a portion of the videos to cover the whole number of videos, while with overlapped samples to other evaluators. This method greatly reduced time of evaluation.

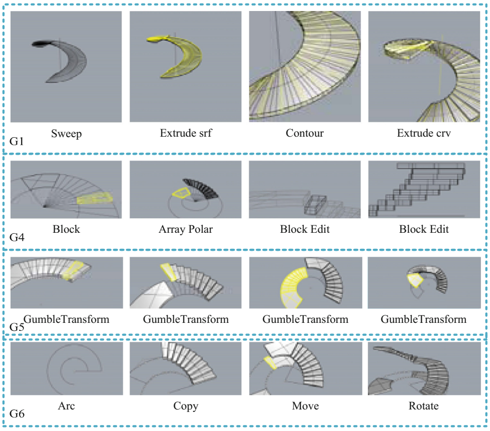


Fig. 5. Typical example screenshots with key commands in G1, G4, G5 and G6.

According to evaluation result, high evaluations grading of five experts appear in G1, G4, G5 and G6 on dimension of 'similarity within group' (Fig. 5). Group1 shows significant feature in steps generation by using contour to swept surface. Group4 modeling behavior featured in using 'Block' to edit stair-step. Group5 shows significant feature in constructing steps one by one in ascending procedure. Participants in G6 show strong similarity in construction phasing which are from all ‘stair steps’ to all ‘platforms’ to all ‘fences’ to all ‘handrails’. This proves from the side that Command2Vec has learned some significant features. However, group 2 and group 3 are more variously featured than the other four groups.

Table 1. Returned evaluation.

Expert 1: ✖ Expert 2: □ Expert 3: ✚ Expert 4: ○ Expert 5: ✱							
Clustering Group		G1	G2	G3	G4	G5	G6
Grade One		*similarity within current group*					
Strong	7	□✚			□✚	□✚	□✚
	6	✖	○	✚	✖○	✖	○
	5	○	□✚✱	□		○	
To	4		✖	○			
	3			✱		✱	✖
	2	✱		✖	✱		✱
weak	1						
Grade Two		*difference of current group to rest groups*					
Strong	7	□✖✱			□✚✖		
	6	✚		✚○		□	□✚
	5		□✚○	□	○	✚✖	○
To	4		✱			○	
	3	○		✖			
	2		✖	✱	✱	✱	✖✱
weak	1						
G: group ▨: section of high grading							

Co-existing commands (Table 2) and their out-degree rankings (Fig. 6) in each of the group were analyzed. Top 3 coexisting commands in G1 and G4 are featured in modeling behavior with key commands like ‘sweep1’ and ‘block’. The median of each group’s command’s out-degree ranking was also around 3 for the top 3 frequently co-current commands but with wide ranges.

Table 2. Top 3 frequently coexisting commands per group.

Top 3	G1	G2	G3	G4	G5	G6
1	ExtrudeCrv	Arc	Split	Block	Copy	Circle
2	Helix	ExtrudeSrf	Circle	Circle	Divide	Polyline
3	Sweep1	ExtrudeCrv	Polyline	BlockEdit	ExtrudeSrf	ArrayPolar

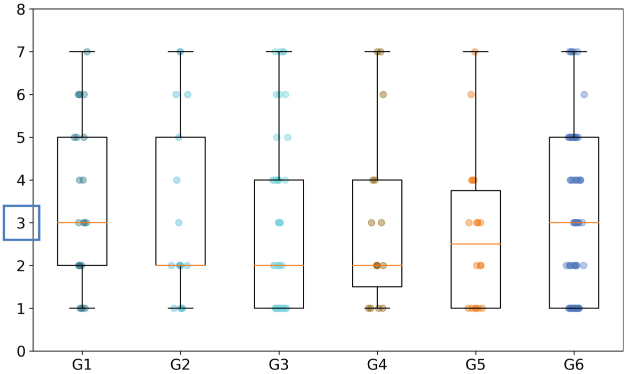


Fig. 6. Box-plot of out-degree rankings for each group's top 3 frequently coexisting commands.

6 Conclusion and Discussion

This study investigated the application of machine learning to empower 3D tool on perceiving, learning, and classifying modeling behaviors. An experiment was conducted to collect 3D-modeling event logs from 112 junior students. Information of commands and objects were retrieved from event logs to form command graphs. A novel algorithm Command2Vec was developed based on Word2Vec for graph embedding. Finally, 6 groups were clustered. An evaluation was done by five experienced designers by watching videos compiled from individual entire modeling screenshots and grading. Comparing to the refined input data, evaluating video included sub-process of trial-and-error and 'noise commands' which more realistically restored modeling scenario. This increased the credibility of the external evaluation results in this study.

We chose feature learning in NLP for data mining because we observed that modeling operations are similar to human languages in certain extent, such as the sequence and hidden structure of commands operated during modeling. Command2Vec performed well in feature learning of key commands and procedures on the 'spiral-stair' task.

Command-object graph has a higher data dimension than the command graph used in this study. The problem of how to embed binary command-object graph with both semantic and geometric information waits to be studied in the future. In terms of evaluation, the sample size of five experts are little, and grading process by watching many videos is very time-costing. A better sampling method to allow larger number of experts in evaluation is thought as an improving direction.

References

1. Goldschmidt, G.: Serial sketching: visual problem solving in designing. Cybern. Syst. **23**(2), 191–219 (1992)
2. Won, P.H.: The comparison between visual thinking using computer and conventional media in the concept generation stages of design. Autom. Constr. **10**(3), 319–325 (2001)
3. Hehua, Z., Ming, G.: Research and application of Intelligent BIM model checking tools. J. Inf. Technol. Civil Eng. Archit. **10**(2), 1–6 (2018)

4. Huang, W., Zheng, H.: Architectural drawings recognition and generation through machine learning (2018)
5. Chen, Z., Huang, W., Luo, Z.: embedGAN: a method to embed images in GAN latent space. In: Yuan, P.F., Yao, J., Yan, C., Wang, X., Leach, N. (eds.) CDRF 2020, pp. 208–216. Springer, Singapore (2021). https://doi.org/10.1007/978-981-33-4400-6_20
6. Cross, N.: Designerly ways of knowing. Des. Stud. **3**(4), 221–227 (1982)
7. Van Der Aalst, W.: Process mining. Commun. ACM **55**(8), 76–83 (2012)
8. Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H., Sui, F.: Digital twin-driven product design, manufacturing and service with big data. Int. J. Adv. Manuf. Technol. **94**(9–12), 3563–3576 (2017). https://doi.org/10.1007/s00170-017-0233-1
9. Yarmohammadi, S., Pourabolghasem, R., Castro-Lacouture, D.: Mining implicit 3D modeling patterns from unstructured temporal BIM log text data. Autom. Constr. **81**, 17–24 (2017)
10. Zhang, P., Chen, R.P., Wu, H.N.: Real-time analysis and regulation of EPB shield steering using Random Forest. Autom. Constr. **106**, 102860 (2019)
11. Pan, Y., Zhang, L., Li, Z.: Mining event log for knowledge discovery based on adaptive efficient fuzzy Kohonen clustering network. Knowl. -Based Syst. **209**, 106482 (2020)
12. Pan, Y., Zhang, L., Skibniewski, M.J.: Clustering of designers based on building information modeling event log. Comput.-Aided Civil Infrastruct. Eng. **35**(7), 701–718 (2020)
13. Kitchener, K.: Cognition, Metacognition, and Epistemic Cognition: A Three-Level Model of Cognitive Processing. Hum. Dev. **26**(4), 222–232 (1983). http://www.jstor.org/stable/26764585
14. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864, August 2016
15. Narayanan, A., Chandramohan, M., Venkatesan, R., Chen, L., Liu, Y., Jaiswal, S.: graph2vec: learning distributed representations of graphs. arXiv:1707.05005 (2017)
16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. arXiv:1310.4546 (2013)
17. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. Stanford (2006)
18. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(11), 2579–2605 (2008)