EUROGRAPHICS 2022/ N. Pelechano and D. Vanderhaeghe
Short Paper

Scene Synthesis with Automated Generation of Textual Descriptions

J. Müller-Huschke, M. Ritter*, and M. Harders

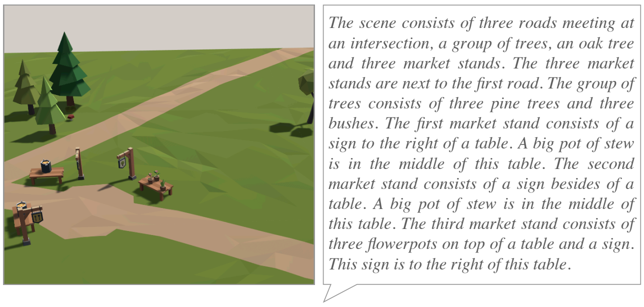
Interactive Graphics and Simulation Group at the Department of Computer Science, University of Innsbruck, Austria.
*corresponding author

Abstract

Most current research on automatically captioning and describing scenes with spatial content focuses on images. We outline that generating descriptive text for a synthesized 3D scene can be achieved via a suitable intermediate representation employed in the synthesis algorithm. As an example, we synthesize scenes of medieval village settings, and generate their descriptions. Our system employs graph grammars, Markov Chain Monte Carlo optimization, and a natural language generation pipeline. Randomly placed objects are evaluated and optimized by a cost function capturing neighborhood relations, path layouts, and collisions. Further, in a pilot study we assess the performance of our framework by comparing the generated descriptions to others provided by human subjects. While the latter were often short and low-effort, the highest-rated ones clearly outperform our generated ones. Nevertheless, the average of all collected human descriptions was indeed rated by the study participants as being less accurate than the automated ones.

CCS Concepts

• ***Computing methodologies*** → ***Computer graphics; Natural language generation;***



The scene consists of three roads meeting at an intersection, a group of trees, an oak tree and three market stands. The three market stands are next to the first road. The group of trees consists of three pine trees and three bushes. The first market stand consists of a sign to the right of a table. A big pot of stew is in the middle of this table. The second market stand consists of a sign besides of a table. A big pot of stew is in the middle of this table. The third market stand consists of three flowerpots on top of a table and a sign. This sign is to the right of this table.

Figure 1: *(Left:) Example of procedurally generated 3D scene. (Right:) Automatically generated description with our framework.*

1. Introduction

Describing the spatial world around us is a common task for humans. When describing entities in computer-based environments, automated methods attempt to mimic the quality and accuracy of such descriptions. Since virtual worlds are often created by algorithms [STBB14], one could leverage metadata and intermediate representations produced during such processes to improve automated descriptions. Thus, we have created a pipeline to automatically generate 3D scenes of medieval villages; and combined it with a natural language generator to add a textual description (see example in Fig. 1). Our main contributions are the integration of the proof-of-concept framework, leveraging scene synthesis and natural language generation (NLG) techniques, as well as a pilot study to assess its usability. A limitation of our current implementation is the necessary manual definition of the scene graph grammars. Our system represents a base for further development; possible future applications could, for instance, be automatic high level annotations of virtual worlds for visually impaired users as well as procedural generation of narrations of virtual gaming characters. Note that in contrast to image captioning, our system is not bound to a single viewpoint. Below, we first outline the framework – 3D scene content and spatial relationships between objects are initially formalized in a relationship graph (see example in Fig. 2); the scene layout is then produced using Metropolis Hastings sampling. In parallel, a basic textual description is synthesized using an NLG pipeline. In the pilot user study we then compare human and machine descriptions of the 3D scenes. We investigate whether the combined approach can produce descriptions that are perceived as accurate regarding scene objects and their relationships and are possibly even preferred over human descriptions. Further details on the system and study are available in the supplementary material.

2. Related Work

Some initial work has been carried out on automatically creating textual descriptions for generated 3D scenes. A closely related domain is captioning of images (see a recent survey in [HSSL19]), where projections of (real or virtual) 3D scenes are denoted. In NLG, referring expression generation has been focused recently. It targets the unambiguous description of individual objects in specific contexts. For objects in images, often deep learning is utilized [MHT*16]. Further, 3D scene synthesis by arranging objects has received quite some attention; with recent approaches being data-driven, e.g. using large datasets of indoor scene layouts as priors [WLW*19, ZYM*20]; but also other priors, such as factor graphs [YYW*12] or combinations of hardcoded and example scene extracted constraints [YYT*11]. Finally, other recent work allows users to provide an input text to control scene layout generation [MZP*18] or image generation [RPG*21].

© 2022 The Author(s)
Eurographics Proceedings © 2022 The Eurographics Association.

DOI: 10.2312/egs.20221026

delivered by
EUROGRAPHICS DIGITAL LIBRARY
www.eg.org diglib.eg.org

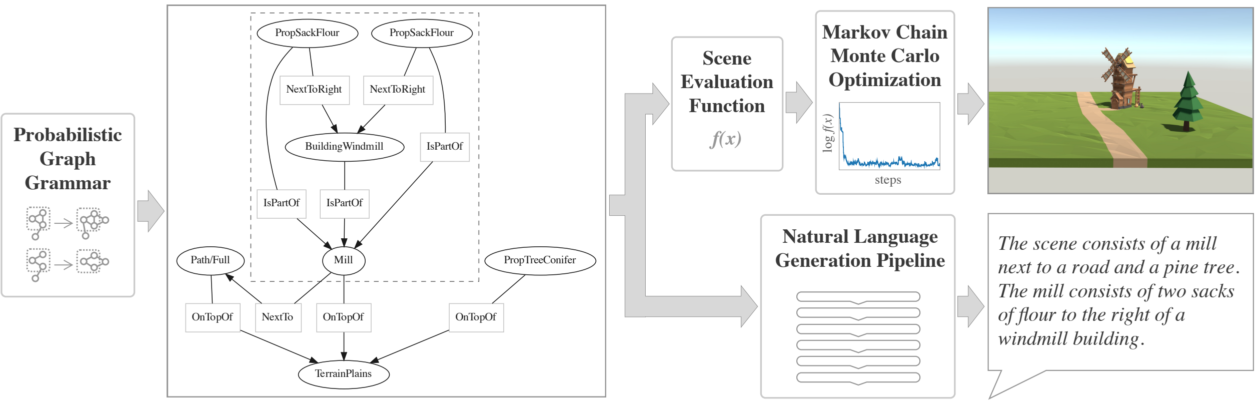

Probabilistic Graph Grammar
PropSackFlour
PropSackFlour
NextToRight
NextToRight
BuildingWindmill
IsPartOf
IsPartOf
IsPartOf
Path/Full
Mill
PropTreeConifer
OnTopOf
NextTo
OnTopOf
OnTopOf
TerrainPlains
Scene Evaluation Function
$f(x)$
Markov Chain Monte Carlo Optimization
$\log f(x)$
steps
Natural Language Generation Pipeline
The scene consists of a mill next to a road and a pine tree. The mill consists of two sacks of flour to the right of a windmill building.

Figure 2: *Overview of the process to generate a scene layout and its textual description from a scene relationship graph (the intermediate representation). The latter is initially derived from a probabilistic graph grammar.*

3. Generation Framework

Matching scene layouts and textual descriptions are derived by sharing the same scene relationship graph as intermediate representation. It contains all objects in the scene as nodes and their spatial relationships as edges. Relationships are in general formulated viewpoint-independent. Relative directions (e.g. left/right) are only used for objects with a clearly identifiable front. This graph is derived from a graph grammar [EHK92], which is non-deterministic and has probabilities assigned to its individual rules. The grammar was manually tuned to produce graphs that result in believable scenes. This was done by taking artist created scene layouts with the same assets as inspiration, deriving semantic rules from them (e.g. that sacks of flour are usually next to a mill) and encoding them as graph grammar rules. For each object type in the graph there are one or more 3D models, each annotated with metadata. The latter may contain textual object descriptions, connecting trails (i.e. paths), and collision volumes. The scene layout is produced by encoding the spatial relationships of the graph into an evaluation function, which then is optimized using a Markov Chain Monte Carlo (MCMC) method. A matching textual description is produced by feeding the same graph into a purpose-built NLG pipeline. An overview of all steps is provided in Figure 2.

3D Scene Generation: The process starts with instantiating a 3D model for each graph node and random placement in the scene. If an object is marked to be on top of another one, its y-coordinate (i.e. elevation) is set according to the supporting object. This leaves the x- and z-coordinates, as well as the rotation around the y-axis to be optimized. In addition, objects can be assigned to trails, with an optional connection point as well as a set of control points to be optimized. Special path objects are designed to provide either a main road or an intersection within the scene. To obtain a graph evaluation function yielding values in $[0,1]$, each edge is turned into a factor using logical connectives ([YYW*12]):

$$\begin{aligned} \mathit{Equals}(x,y,\sigma^2) &= \mathcal{N}(0,\|x-y\|,\sigma^2)/\mathcal{N}(0,0,\sigma^2); \\ \mathit{Greater}(x,y,h) &= \mathrm{Sig}(x-y,h); \\ \mathit{Less}(x,y,h) &= \mathrm{Sig}(y-x,h); \\ \mathit{Range}(x,y_{\min},y_{\max},h) &= \mathit{Greater}(x,y_{\min},h)\mathit{Less}(x,y_{\max},h), \end{aligned} \tag{1}$$

with Gaussian $\mathcal{N}(x,\mu,\sigma^2)$ and sigmoid function $\mathrm{Sig}(x,h) = 1/(1+e^{-hx})$; h and σ control steepness (we use $h = 3.0$, $\sigma^2 = 0.1$). Next, spatial relationships v_i are set, e.g., A "*NextTo_NorthOf*" B:

$$v_{NT\_NO}(A,B) = \mathit{Less}(d,2)\cdot\mathit{Equals}(\alpha,0), \tag{2}$$

with d the minimal separation between the collision volumes of A and B, and α the angle between the x-axis and the vector pointing from B to A. The product of all such factors v_i finally forms the scene evaluation function $f(S)$ for a configuration S; taking a scene layout as input and returning a value in $[0,1]$. Additional factors of f are designed to encourage the generation of feasible trails and to avoid overlapping objects. Five geometric helper functions and 19 object relation functions have been employed and are provided in the supplementary material. Also note that the computation of f is done in log-space to avoid floating point underflows.

Thus, the practically employed energy function becomes $f = e^{-v}$, with $v = v_r + v_c + v_p + v_q$, with v consisting of four different constraint types: v_r – encoding spatial relationships between objects; v_c,v_p – avoiding collisions between objects or with pathways; v_q – encouraging generation of believable pathways. The scene layout is then optimized by maximizing f. MCMC is applied using Metropolis Hastings with parallel tempering. This denotes simulating multiple Markov chains with stationary distributions matching a distribution with a probability density function proportional to f. Metropolis Hastings requires sampling a jumping distribution $G(S'|S)$, that proposes at each Markov chain step a new scene layout S' depending on the current one S. S' is accepted with probability $\min\left(1, f(S')/f(S)\right)$ as the new layout, otherwise S remains current. The jumping distribution is sampled by randomly performing one of the following actions:

- Resample position of single object or cluster (one unit is one meter in the scene): $x' \sim \mathcal{N}(x,0.5^2)$, $z' \sim \mathcal{N}(z,0.5^2)$;
- Resample angle around y-axis of single object or cluster (one unit is one degree rotation): $\varphi' \sim \mathcal{N}(\varphi,10^2)$;
- Resample trail control points;
- Randomly reconnect trail to different trail.

Note that optimization is first applied in separate to smaller object clusters in the scene relationship graph, before applying it to

© 2022 The Author(s)
Eurographics Proceedings © 2022 The Eurographics Association.

the whole scene. This accelerates the production of good scene layouts. The clustering is obtained via special container objects ("*IsPartOf*" relationships) in the graph. Further, the chains are parameterized to explore the search space at different speeds (temperature of 1.3^i for the i-th chain); and to randomly swap information with neighboring chains, such that scene layouts with the highest value of f move to the lowest temperature chain. Overall, good scene layouts are typically obtained after $N = 100,000$ steps, i.e. $20,000$ simulation steps for each of the five Markov chains. In our unoptimized, single-threaded implementation this takes about one minute.

Textual Description Generation: The text is created via an NLG pipeline, governed by hardcoded rules, following the standard stages in [GK18]. The rules were designed according to a pre-study, in which 10 human annotators provided 30 textual descriptions of 3 different scenes, generated using the methodology described above. Since humans tend to group objects together, we employ the same clusters of the scene generation to achieve a hierarchical textual description. Input to the pipeline is the scene relationship graph, a set of fixed expressions, and some measure of importance assigned to each object in the scene. The stages of the pipeline are:

1. **Content determination**: Take the set of graph nodes as entities of the textual description; and the set of edges as a list of relationships that should be described in the text. Filter out entities without expressions (and their connected relationships) or low importance, as well as relationships without expression.
2. **Text structuring**: Construct a list of sentence prototypes: add one sentence prototype describing which entities and clusters are in the scene; add one prototype per cluster describing its entities; add one prototype for each relationship to be described. Order sentences according to cluster and object importance.
3. **Sentence aggregation**: Merge certain sets of sentence prototypes, without losing information; e.g. if two sentences describe the same spatial relationship of two distinct entities to a third one, then the second sentence can be integrated into the first.
4. **Lexicalization**: Select words from the expressions assigned to objects and from a table of expressions for spatial relationships.
5. **Referring expression generation**: Both subject and object of each sentence prototype can consist of one or more entities that need to be referred to. Referring expressions are constructed by using the first expression that unambiguously describes a set of objects, from an ordered list of possibilities: identification by assigned expression, by cluster, or by numbering.
6. **Linguistic realization**: The sentence prototypes are finally transformed into a list of actual sentences, adhering to English grammar and inflection rules, using SimpleNLG [GR09].

The output is an automatic description of the 3D scenes synthesized in the previous section (see example in Fig. 1). Note that the results are deterministic with no textual variety, i.e. the same scene relationship graph always produces the same textual description.

4. Pilot User Study

We conducted a pilot user study to assess the quality of the generated scenes and the matching descriptions, as well as to compare to human-generated ones. Twenty-nine subjects (20 males, 9 females, age 21–66) participated in the unpaid study. All had normal or corrected vision and were non-native English speakers.

The study consisted of two sessions and lasted about half an hour each. In the first session, subjects were first asked about five 3D scenes in general. These were randomly selected from a pool of twenty previously generated scenes of varying complexity. Users could explore the virtual world. They were then asked to rate on a 5-point Likert scale their agreement with the two following statements: 1) *“This scene is plausible and does not contain any obvious errors in the placement of objects.”* and 2) *“This scene is aesthetically pleasing.”* Additionally the participants provided their own textual descriptions of the scenes, answering to the prompt *“Describe what can be seen in this scene in about five sentences.”* Thereafter, participants took part in the second session, on average 14 days later. In it the human descriptions obtained in the previous session were leveraged, by pitting them against machine-generated descriptions. A subset of twelve scenes from the initial set was randomly selected and presented, each along with two textual descriptions. In eight of the cases, a human and a machine text were shown, in four both descriptions were by humans. The order of presentation was randomized and counter-balanced. For each description participants were asked to rate on a 5-point Likert scale their agreement with the statement *“The text describes the scene accurately.”* In addition, they had to reply to a 2AFC question, indicating their preferred description; answering to the prompt *“Which textual description would you prefer?”* All collected data was anonymized.

5. Results

Regarding the quality of the generated scenes, as assessed in the first session, about 65% of the participants *strongly agreed* or *agreed* that the scenes are plausible, while 21% *disagreed* or *strongly disagreed*. A similar amount agreed that the scenes are aesthetically pleasing, albeit in this case only 7% were in disagreement. There was a positive correlation between the ratings of a scene’s plausibility and its aesthetics (Pearson, $\rho = 0.85$, $p < 0.01$). For the textual descriptions, investigated in the second session, the machine descriptions were perceived as slightly more accurate on average than user descriptions (t-test, $p < 0.01$). Figure 3 (top) provides a visual illustration of the results. About 80% of participants *strongly agreed* or *agreed* that a machine description was accurate (greenish bars), while only 60% said so for the user descriptions; moreover, 9% *disagreed* or *strongly disagreed* for machine descriptions, compared to 10% for user descriptions (reddish bars). However, if for each scene only the highest rated user description were to be considered, then all users *strongly agreed* or *agreed* that it was more accurate (bottom). Next, when participants were asked about their preference of either a random user description or a machine-generated description, the latter was cho-

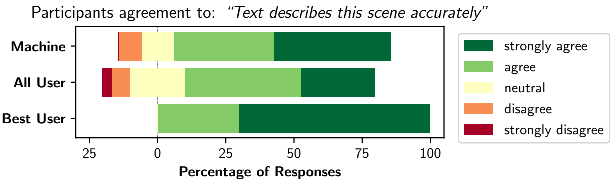


Figure 3: *Likert scale ratings of scene description accuracy (as percentages of responses, with common zero point), separated by source. Results are shown for machine-generated as well as all user descriptions; and for the respective best (rated as most accurate) user description for a specific scene.*

© 2022 The Author(s)
Eurographics Proceedings © 2022 The Eurographics Association.

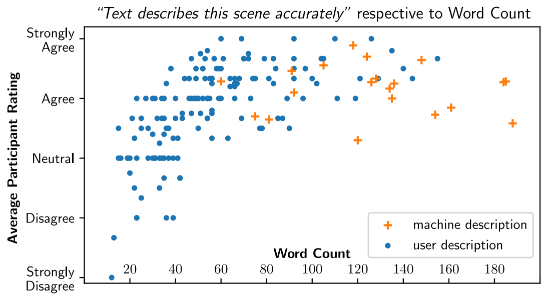


Figure 4: *Correlation of word count & average accuracy rating of scene descriptions – automated (red) vs. human-generated (blue).*

sen in 51% of the cases. No statistically significant differences were found regarding the preference (t-test, $p = 0.62$); both when choosing between two random user descriptions as well as between a user description and a machine description. Thus, machine generated descriptions were of similar popularity as the average user provided ones. Notably, in 10% of cases participants preferred a description, which they previously had judged to be less accurate. In our study, the textual descriptions provided by users consisted on average of 56 words, while machine descriptions were longer at 127 words. In general, short descriptions were usually perceived to not be very accurate (see Figure 4). For the user descriptions, word count correlates with perceived accuracy (Pearson, $\rho = 0.62$, $p < 0.01$); however, that was not found for the machine descriptions (Pearson, $p = 0.83$). Still, the length gives machine descriptions an advantage. In fact, normalizing the word count of both groups to the average (91 words), by selecting only the shortest machine descriptions and longest user descriptions, yields a higher accuracy rating for the latter. Most user descriptions were shorter than asked for (≤ 5 sentences). In post-hoc questionnaires, many participants stated that they were able to tell which description was machine-generated; however, sometimes they still preferred the latter.

6. Discussion & Conclusion

The scenes produced can be plausible and aesthetically pleasing, however, their variety and quality is limited by complexity of the graph grammar employed. Failure cases exist, and scenes are usually problematic due to objects intersecting one another or the path layout being implausible or just different than planned in the intermediate representation. In contrast to state of the art scene synthesis works that rely on machine learning [ZYM*20, WLW*19], our method is not able to leverage existing human knowledge about proper object placement, it all must be encoded in the graph grammar. Constructing the grammar and annotating the objects with collision volumes and language expressions is time-consuming, however it allows for scene layout synthesis in domains where no large datasets exist yet and is more time-efficient than creating a new dataset. Taking a qualitative look at the least accurate machine descriptions reveals that these tend to belong to scenes where the layout violated one or more constraints of the scene relationship graph. In contrast, poorly rated user descriptions tend to contain spelling and grammatical errors, and be somewhat shallow, e.g. only enumerating objects, but not their relation. On the other hand, highly rated user descriptions often describe the scenes in fewer sentences than the machine descriptions; still, they contain broad information on individual scene objects and their spatial relationships. The limitation to objects, spatial relationships between them, and clusters in our intermediate representation makes generated descriptions inherently weaker than very accurate human descriptions, since they can infer additional context from scene elements. The synthesized texts often appear as more lengthy and tedious than human ones.

We have presented a proof-of-concept system which integrates several established techniques to generate 3D scenes of medieval villages as well as matching automated textual descriptions. In a pilot user study a majority of the generated scene layouts were found aesthetically pleasing and believable by participants. The textual descriptions were deemed accurate in 80% of the cases and performed better than the average description participants themselves gave for a scene during the study, somewhat caused by the users giving shorter descriptions than asked for. Machine descriptions are not competitive with the accuracy of the best human descriptions given in the study; and are easily identified as machine-generated.

Source Code at https://github.com/JulianMH/EG22-Scene-Text

References

[EHK92] Ehrig H., Habel A., Kreowski H.-J.: Introduction to graph grammars with applications to semantic networks. *Computers & Mathematics with Applications 23*, 6-9 (Mar. 1992), 557–572. 2

[GK18] Gatt A., Krahmer E.: Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research 61* (Jan. 2018), 65–170. 3

[GR09] Gatt A., Reiter E.: SimpleNLG: A realisation engine for practical applications. In *Proc. of the 12th EU Workshop on Natural Language Generation* (2009), Assoc. for Comp. Linguistics, p. 90–93. 3

[HSSL19] Hossain M. Z., Sohel F., Shiratuddin M. F., Laga H.: A comprehensive survey of deep learning for image captioning. *ACM Comput. Surv. 51*, 6 (Feb. 2019). 1

[MHT*16] Mao J., Huang J., Toshev A., Camburu O., Yuille A. L., Murphy K.: Generation and comprehension of unambiguous object descriptions. In *Proceedings of IEEE CVPR* (June 2016). 1

[MZP*18] Ma R., Zhang H., Patil A. G., Fisher M., Li M., Pirk S., Hua B.-S., Yeung S.-K., Tong X., Guibas L.: Language-driven synthesis of 3D scenes from scene databases. *ACM Trans. on Graphics 37*, 6 (Dec. 2018). 1

[RPG*21] Ramesh A., Pavlov M., Goh G., Gray S., Voss C., Radford A., Chen M., Sutskever I.: Zero-shot text-to-image generation. In *Proc. Intl. Conf. on Machine Learning* (2021), PMLR, pp. 8821–8831. 1

[STBB14] Smelik R. M., Tutenel T., Bidarra R., Benes B.: A Survey on Procedural Modelling for Virtual Worlds. *Computer Graphics Forum 33*, 6 (Sept. 2014), 31–50. 1

[WLW*19] Wang K., Lin Y.-A., Weissmann B., Savva M., Chang A. X., Ritchie D.: PlanIT: planning and instantiating indoor scenes with relation graph and spatial prior networks. *ACM Trans. on Graphics 38*, 4 (July 2019). 1, 4

[YYT*11] Yu L.-F., Yeung S.-K., Tang C.-K., Terzopoulos D., Chan T. F., Osher S. J.: Make it home: Automatic optimization of furniture arrangement. *ACM Trans. on Graphics 30*, 4 (July 2011). 1

[YYW*12] Yeh Y.-T., Yang L., Watson M., Goodman N. D., Hanrahan P.: Synthesizing open worlds with constraints using locally annealed reversible jump MCMC. *ACM Trans. on Graphics 31*, 4 (July 2012). 1, 2

[ZYM*20] Zhang Z., Yang Z., Ma C., Luo L., Huth A., Vouga E., Huang Q.: Deep Generative Modeling for Scene Synthesis via Hybrid Representations. *ACM Trans. on Graphics 39*, 2 (2020). 1, 4

© 2022 The Author(s)
Eurographics Proceedings © 2022 The Eurographics Association.