TU Delft

Delft University of Technology

A generalized semantic representation for procedural generation of rooms

Balint, J. Timothy; Bidarra, Rafael

DOI
10.1145/3337722.3341848

Publication date
2019

Document Version
Final published version

Published in
FDG'19s

Citation (APA)
Balint, J. T., & Bidarra, R. (2019). A generalized semantic representation for procedural generation of rooms. In F. Khosmood, J. Pirker, T. Apperley, & S. Deterding (Eds.), *FDG'19s: Proceedings of the 14th International Conference on the Foundations of Digital Games* (pp. 1-8). Article 85 (ACM International Conference Proceeding Series). ACM. https://doi.org/10.1145/3337722.3341848

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' – Taverne project

https://www.openaccess.nl/en/you-share-we-take-care

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

A Generalized Semantic Representation for Procedural Generation of Rooms

J. Timothy Balint
Delft University of Technology
jbalint@stonehill.edu

Rafael Bidarra
Delft University of Technology
r.bidarra@tudelft.nl

ABSTRACT

Procedural generation of rooms aims to create virtual environments that mimic common patterns found in real-world indoor locations, like offices or bedrooms. Graph-based models (e.g. factor graphs or Bayesian networks) have often been used to represent typical location's objects and their occurrence likelihood (nodes), as well as their inter-relationships (edges). Previous methods have struggled to represent object semantics in their graph nodes; specifically, they fail to fully and effectively support notions as *abstractions* (e.g. generic seat instead of chair) and replication (e.g. cups instead of cup). We propose a generalized representation and use for object semantics that overcomes the above limitations of graph-based models in the procedural generation of rooms. This node representation handles semantics as attributes, and clearly distinguishes the contribution of the attributes on the node from the potential effects of the node on the whole graph. We illustrate the additional expressive power of the resulting graph-based model for room generation, and show that it subsumes previous models as particular cases.

CCS CONCEPTS

• **Computing methodologies** → **Semantic networks**; • **Applied computing** → *Computer games*.

KEYWORDS

Procedural Content Generation, Data Representation, 3D Content Generation

ACM Reference Format:
J. Timothy Balint and Rafael Bidarra. 2019. A Generalized Semantic Representation for Procedural Generation of Rooms. In *The Fourteenth International Conference on the Foundations of Digital Games (FDG '19), August 26–30, 2019, San Luis Obispo, CA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3337722.3341848

1 INTRODUCTION

A *Procedurally Generated Room (PGR)* is a location generated from a representation of the objects and their relationships commonly found in that location. PGRs are also sometimes called synthesized indoor scenes. Creating PGRs requires Procedural Content Generation (PCG) methods that sensibly distribute the appropriate content over the available space. Each PGR instance should provide the impression of the distribution while being dissimilar in both the total of objects used and the way in which they are laid out.

Procedural room generation is concerned not only with *where* objects should be placed in a setting, but also with *how many* objects, as well as with the visual and non-visual ways in which the objects interact. We call this distribution of objects that describe a room and their relationships the *motif* of the location. For example, the motif of an office would contain one or more desks, with one or more chairs related in proximity and orientation to that desk.

There are two main methods used to generate rooms: grammars [16, 28] and sampling from a graph-based data structure [3, 4, 22]. In essence, sampling from a graph-based data structure selects a subset of the graph and generates a room from the selected content and relationships. The data structures used in sampling are learned from examples of that location, where each node in the model is a learned probability for a given object (and relationship, in the case of factor graphs) to exist. Links in these models denote a causal relation between them. Factor graphs, which further define relationship nodes in addition to content nodes, can be thought of as having *semantic links*, where additional meaning is given to the relationships between objects. The content and structure of those graphical models therefore embody the semantics of a location.

Graph structures for rooms are also useful in that they allow similarity between rooms to be defined [5]. However, semantics that are only probabilistically related to the distribution of objects (such as the frequency of multiple instances of the same object being located in the same room or different possible styles) are internodal properties and not easily represented in this configuration. Thus, while graphs are a powerful representation of motif, there are several semantic properties that they cannot encapsulate.

Two specific properties that we focus on in this work are *abstractions* and *replications*. Abstractions allow for the selection of one object to generate from a possible group. Replication allows for the generation of multiple objects given a single selection. These semantics can also be used together to create homogeneous and heterogeneous objects, therefore expanding the variety of possible generation.

While semantics can expand a room generator's ability, both the data structure and the generator force the motif to be simpler. This means object properties like replication and abstractions can be lost in the motif, as well as never sampled by the generator. The distribution of generated objects and their configuration is known as the *visible distribution* of a motif. When the visible distribution is unable to generate the entire distribution of objects found in the motif, the overall variability of PGRs decreases. As a result, the expressive power of a room generation system is hampered by both the representation it uses and its generation system.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
FDG '19, August 26–30, 2019, San Luis Obispo, CA, USA
© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-7217-6/19/08…$15.00
https://doi.org/10.1145/3337722.3341848

To better capture objects and their associated semantics, we propose a motif representation based on the novel node representation of a *Content Chunk*. Using a content chunk, semantics for an object are now contained within a single node, instead of being distributed over the graph. When paired with current generation models, the motifs using content chunks are better equipped to express objects' semantics, and therefore more fully represent the motif.

2 RELATED WORK

Data-driven PGRs attempt to populate and layout objects based on a learned pattern. Of the large body of work available, many [6, 9, 19] use a probabilistic graph representation (such as a Bayesian network or factor graph) in order to store and generate rooms. Rooms are generated as subsets of the motif, choosing objects either by selecting sections of the graph or through graph search. One way to increase the variability of generated rooms is to incorporate semantics into the objects. Work such as Savva et al. [21] determines semantics on objects such as the material of an object and type categorizations. These semantics are used for scene synthesis queries like those used in SceneSeer [2]. For these approaches, semantics is used as a selection mechanism instead of a motif, requiring more human intervention. While improvements such as that by Ma et al. [18] have sub-motifs, the semantics from Savva et al. [21] is not used in the sub-motif learning, signifying that the integration of semantics in motifs is still an open problem.

Other semantics, such as abstraction and replication of objects in a motif, has been explicitly captured as part of the graph data-structure. The method of encoding these semantics generally falls into two categories: using a *special connection* between independent nodes in the graph and *enumerating all possible semantic connections*. A generic example of these two methods can be found in Figure 1.

Special connections for replication treat each content as independent, but provide the ability for them to be related. An abstract representation can be seen in Figure 1a. Work such as Fisher et al. [4], Kermani et al. [11], and Liang et al. [14] treat replication of objects as independent objects, in a process known as *Binarization*. Each of these encoding methods also contain one or more special graph connections to state that these objects are related. However, creating a graph in this fashion means that each object may (and oftentimes does) have differing links between the same object, which may lead to inaccuracies such as missing relationships between them. Therefore, using this method provides a more realistic representation for what objects exist in the world, but is incomplete in how they are configured or controlled.

Counter to using a special connection, some work enumerates each possible combination of a set of objects as a given singular node, and describes how the objects in a given combination relate to one another. The example in Figure 1b enumerate four different combination of glasses and mugs on a table. Works such as Scenesuggest [20] and Sceneseer [2] enumerate items in a set. In these works, the probability of an item having multiple of that same object in a given environment becomes its own, disjoint probability. While other work such as Ma et al. [17] does not specifically use an object-relation motif formulation, they do learn a probability of occurrence for objects given the cooccurance of objects already placed in the environment. This is analogous to having a single

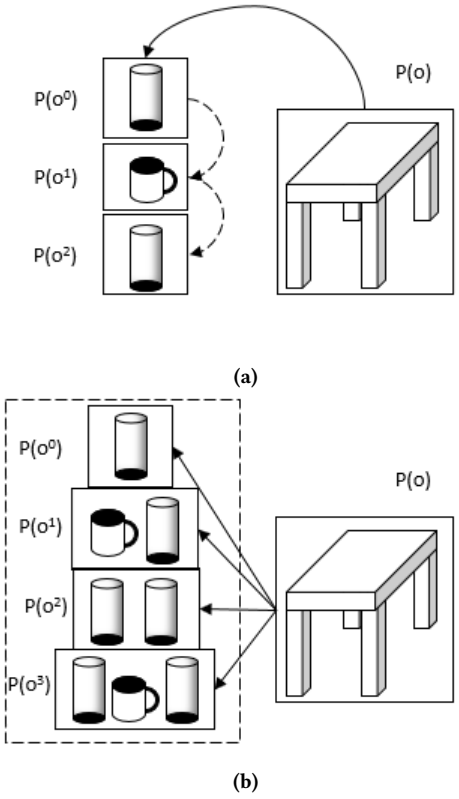

Figure 1: Two competing graph based representations to place one to two cups and zero to one mugs on a table, based on the likelihood of probability $P(o)$. (a) Special connection graph methods would have a probability for each individual object, which would be resolved individually. The special connections are shown as dotted lines. (b) Enumerating methods would have a probability for each individual grouping of objects. In this figure, there is a box around the enumerated objects.

occurs relationship between possible objects in the environment. Their later work [18] adapts a sub-scene model, incorporating heterogeneous sets with specified relationships as sub-motifs with a given probability. The sub-motifs in Ma et al., as well as the singular objects in other work, allow precise control of the configuration of content inside that node, which is an improvement over the control mechanisms of the special connection methods. However, the enumeration of possible abstractions becomes cumbersome as the number of potential ways to replicate and abstract grows. This leads to unrealistic representations of how the nodes interact, as the number of nodes tend to overwhelm generation systems to only generate a subset of the motif.

The tricks and techniques used for graph-based representations of rooms either rely on having independent representations for all objects or enumerating all possible combinations of a set of objects. Each of these methods captures and assumes different meaning in the relationship between objects in a location. In order

to capture both replications and abstractions (as well as control for possible semantics not yet used) in the same graph, a hybrid graph representation is needed.

3 GENERALIZED REPRESENTATION OF GRAPH NODES

In Section 2, we described the two most common ways of representing semantics such as abstractions and replications of objects in graph-based motifs. However, previous graph-based models do not contain a generalized way of representing semantics in a node. Therefore, we define a new representation of objects in motifs, called a *content chunk*. The content chunk has self-contained semantics (such as the minimum and maximum replication of an object allowed in the motif), each with an associated probability that the semantic is a part of the content's context. The instantiation of object semantics during generation is then performed on the content chunk (instead of the whole graph). This is guided by a heuristic function called the *Internal Probability Function* (IPF). In addition to the IPF, we include an *External Probability Function* (EPF) to control when the node is selected from the motif during generation. An example of a content chunk can be seen in Figure 2.

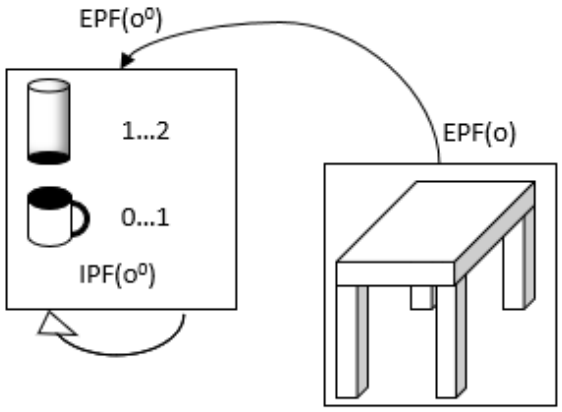

Figure 2: A motif using content chunks for the example graph from Figure 1. The content chunk treats each individual object as an *attribute*, whose likelihood of representing that node is the IPF. Much like $P(o^{\#})$ in Figure 1, we have a probability of the node being generated, denoted as the EPF.

In addition to object semantics, a content chunk can have *inter-relationships* between objects generated through replication of the chunk. An inter-relationship is a relationship that applies to all objects generated by the content chunk (i.e. a loop) when the relationship is selected by the generator. In Figure 2, the inter-relationship on the content chunk of glasses and mug would create a relationship between all generated mugs and glasses. As motifs are represented as graphs, a content chunk can be thought of as a fully connected section of the graph.

Semantics in a content chunk are represented as a set of tunable parameters called *attributes*; examples can be the different abstractions of objects or the range of objects that can be reasonably replicated in a given room. A single attribute is a real-valued set representing one semantic in a content chunk, comparable to the use of *units* for semantic entities in other work [12]. As attributes can be inter-related, it is more important during generation to consider the full set of attributes for a single content chunk.

The need for an attribute set becomes apparent when examining content chunks that contain both abstraction and replication, such as the one seen in Figure 2. In the figure, this content chunk can generate mugs or cups. There are two possibilities that may occur in this scenario: either we have a homogeneous set (consisting only of cups) or heterogeneous sets (a combination of mugs and cups). In a homogeneous set, only one abstraction may be valued anything other than zero, denoting a set for that abstraction. Furthermore, each abstraction can have a different range. In this example, the system has learned that more cups may exist in the generated room than a set of mugs. A heterogeneous set would allow both abstractions to exist in a scene, with the same relationships between all of them.

3.1 Probability Functions

As content chunks contain semantics in the node, each node becomes much more expressive than in other graph-based motifs. However, during generation, the variability of the semantics sets must be instantiated. To do so, content chunks have two functions, one that controls the likelihood of the instantiated attribute set representing the node and one that controls the interaction of the node in the graph.

The *Internal Probability Function (IPF)* is a heuristic function used to describe the overall probability of a given combination of semantics during generation. It allows the semantics in the content chunk to be sampled, so that different configurations may exist. For example, disjoint attributes (such as only having homogeneous objects) will generate an IPF of zero if the semantics of a content chunk create a heterogeneous set. Having more complex functions represent each attribute, as is done in [15], can then become a cost function, which can be sampled to find well defined semantic sets. Finally, it should be noted that a content chunk that contains only a single object will have a delta Dirac function centered at 1 as an IPF.

Graph structures that encode replication, such as the Bayesian Network structure of Fisher et al. [4], have a starting probability to begin adding in a set. Furthermore, many other motif definitions [11, 17, 20] contain a probability for that node appearing in a scene. This is different than (albeit can be related to) the likelihood of a given attribute set. Therefore, we also formulate a content chunk with an *External Probability Function* which controls how the content chunk exists within the motif.

The *External Probability Function (EPF)* describes the likelihood that the instantiated content chunk should be selected during generation. The EPF may rely on the IPF and encoded set of attributes to determine its probability. Therefore, the EPF can rely on the results of sampling the semantic set, and should be determined after the IPF is determined. Furthermore, much like IPFs for singular semantics, when a content chunk only contains a single attribute

and single value, the EPF of that content chunk should be a single value.

The IPF and EPF are, thus, an effective solution for handling multi-object relationships when using any graph-based model.

4 EXAMINATION

4.1 Reduction to Content Chunks

Content chunks subsume other graph-based models of motifs, allowing semantics of objects such as abstractions and replications to be encapsulated within each object node in the graph. By subsuming several other graph-based representations, content chunks should be able to overcome the individual limitations that each other data structure suffers from. We refer to this subsumption as a *reduction* from one data structure to another.

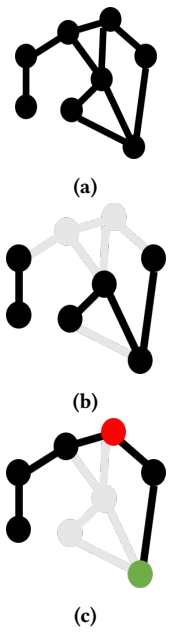

Figure 3: A high level example of the sub-graphs created by our two test algorithms. From (a) the motif, (b) Kermani et al. select a subset of the relationships. Any object connected to a selected relationship is also selected. In contrast, (c) SceneSeer performs a search from a starting selected node (in red) to an end node (in green). The objects and relationships not selected are shown in gray.

When converting from one data structure that was developed in combinationwith a generation method, both the structure and the effects of the generator must be taken into account. While the underlying distribution is contained in their motif, the *visible distribution*, which is the distribution that is expressed through generation, is based on the generator. One crucial aspect of this work is that the visible distribution may not match the underlying distribution that is captured in the motif. As the motif represents the distribution of the room, deviations caused by generation should not exist (or those deviations should be minimized). Therefore, we will describe when the reduction to content chunks allows the motif to better align to the visible distribution.

4.1.1 Special Connection Reduction. The first (and simplest) reduction that we perform is with Kermani et al.'s factor graphs [11]. The factor graphs are a kind of special connection graph; therefore each content chunk in the factor graph contains one object with a probability of existence, and relationships are represented as a joint probability between objects. Kermani et al. has one relationship, the *symmetry* relationship[1], that in fact represents object semantics (duplication of objects). This relationship requires three objects, a set of two identical objects and a third object which acts as the line of symmetry. It is with this example that we reduce their factor graphs into factor graphs with content chunks.

Rooms are generated by Kermani et al. by selecting relationships to create a subgraph from the motif (seen in Figure 3b), using a Monte Carlo Markov Chain (MCMC) method. Each node and link contribute to the overall likelihood of those objects and that configuration existing, either as the probability or compliment of that probability. Creation is controlled by specifying the desired number of objects, meaning that the size of the used graph has a direct effect on the variation of the generated rooms, and contributes to the difference between the motif and visible distribution. In the following reduction, we focus on representing the underlying distribution instead of the visible one.

The two set objects that can be compressed into a content chunk have independent probabilities of existence. However, for a set of objects, there will be one object that has a larger probability than the others, with an equivalent or decreasing probability of existence for each additional object in the set. To generate the IPF, we first notice that the probabilities for duplicated items can be sorted into decreasing order. Therefore, we sort the probabilities for each item, and then compute a scan of the product for all probabilities. That is, for the ith item in the set, the probability for that to exist is the multiplication of all the previous probabilities in the set. The IPF is then the normalized value for all calculated probabilities of the same object $1 \ldots n$. Similarly, the EPF, which describes how likely the set is to be in the graph, is the non-normalized scan probability for each set calculation.

Content chunks combine all set items from the factor graph into a single node. As the sampling function evaluates the probability of the graph for each iteration, we can calculate the IPF and EPF for each iteration. This causes the reduced graphs to produce more sets, as creating a set is not solely reliant on those special connections and does not compete with other connections during generation.

4.1.2 Enumeration Reduction. The second reduction that we perform is to reduce a graph that stores enumerated object sets to a graph using content chunks. For this, we choose the method of SceneSeer [2], which is also found in SceneSuggest [20]. These methods store their motifs as a table of both existence and joint probabilities. One component of this set of work is that they use a one level taxonomy, which they employ when the learned data has a low probability. This means that the captured information ignores any specificity in the abstraction, creating heterogeneous

[1]As shown in their additional materials, located at https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12976.

sets. Furthermore, they learn a separate probability of existence for each count of a set of items, making their methods enumeration methods. SceneSeer uses a graph-growing generation method (seen in Figure 3c), which selects two objects from the motif and grows the scene through a search algorithm between the two objects. Additional objects not directly in the path between the two selected objects may also be included.

Much like the previous reduction, the IPF and EPF of this reduction relies on probability tables. Each count in a set has an independent probability calculated for it. These probabilities are used to create an IPF in the following manner: the internal probability of one or both objects existing is the normalized probability between each calculated set of objects. As abstraction information is lost during the learning stage, the IPF calculates the likelihood from the table based on the total number of objects, and not the number of objects in each abstraction. Unlike the previous reduction, each relationship in this graph is a calculated static prior. So the EPF can be represented as the singular value between nodes, similar to the original.

4.2 Application to Room Generation

As stated in Section 2, either type of graph-based model (both special connection and enumeration) captures information that the other is unable to. As content chunks subsume both models, they can be used to transfer the learned features from one method to another. For room generation purposes, this means that heterogeneous sets learned for SceneSeer can be used in conjunction with the Monte Carlo Markov Chain method of Kermani et al.

We begin by implementing the learning methods of both Kermani et al. and SceneSeer. After creating the motifs of a bedroom from the SUNCG data-set [1, 23], we generate several examples of each using their generation method, and select the sub-motif that has the median number of objects and relationships between them. An example of each result can be seen in Figure 4. Next, we modify each motif to use content chunks using the reductions described in Section 4.1. We then generate using these modified motifs. The results can be seen in Figure 5.

When nodes are converted into content chunks, the learned code for Kermani et al. generates, on average, the same number of content chunk objects and relationships as in their data structure. In other words, the underlying behavior of the algorithm remains unchanged. However, when considering the sets of generated objects as separate objects, the total number of generated objects is greater than when simply using factor graphs. With an internal probability function to determine sets, the likelihood that a set is selected greatly increases. This is due to the underlying set relationships (*symmetry*) having a low probability of existence, especially when compared to other, singular objects. As the generation algorithm is optimized for the number of objects, as the possible number of items in a motif grows, the likelihood of duplication decreases when the optimum number of objects is less than the total variety of objects. By extracting that semantic into a content chunk, it no longer becomes a competing control that ignores portions of the motif. Therefore, converting their representation allows their selection algorithm to focus on the underlying relationships between objects, meaning the visible distribution more closely matches the underlying distribution.

(a)

(b)

Figure 4: Examples of generated bedrooms using the motif and generation methods of (a) Kermani et al. and (b) SceneSeer.

The selection method of SceneSeer, for two disjointed sets of objects, can produce varying effects depending on the full distance

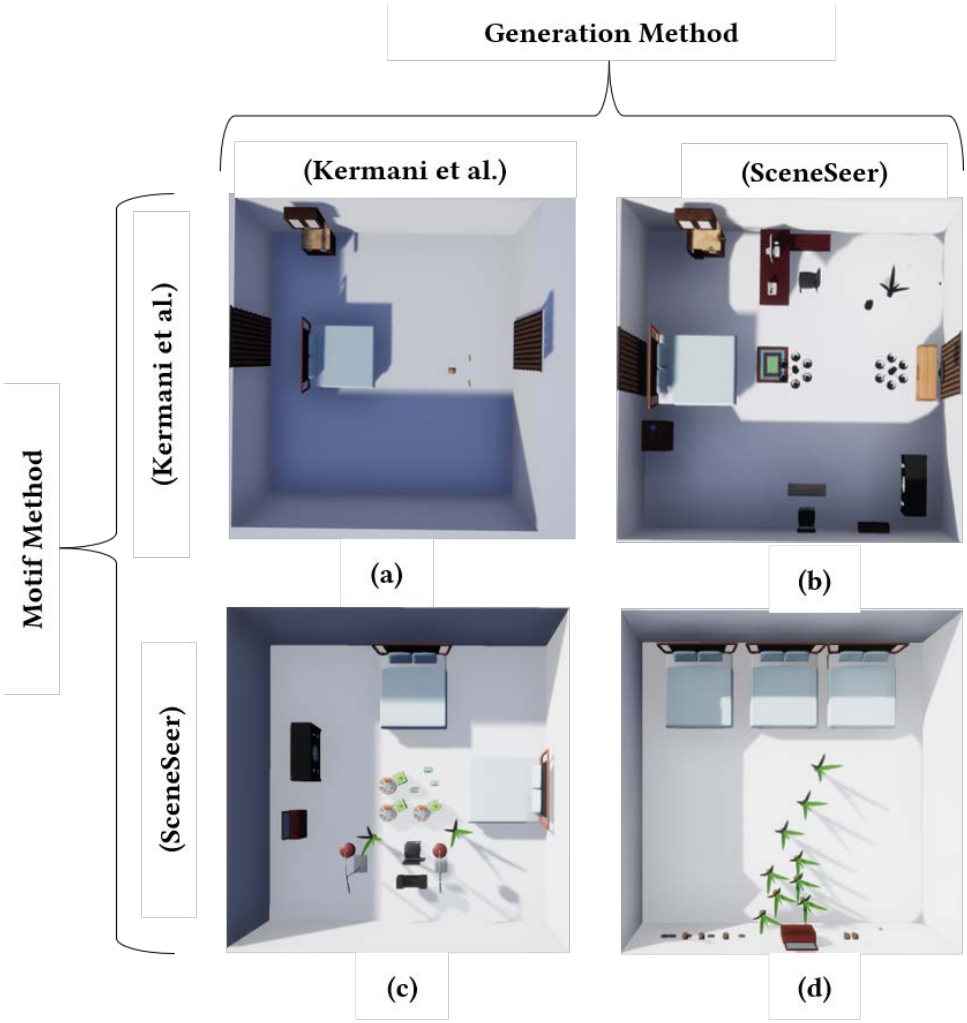

Figure 5: Examples of generated bedrooms from content chunks reduced from the motifs of Kermani et al. and SceneSeer. Note that more objects are generated using SceneSeer's generation method (independently of the motif method). Furthermore, more sets are stored in the motif method of SceneSeer than of Kermani et al..

between objects in the graph. The further the objects, the more likely that many additional objects will be added to the scene. We see this effect in both the generalized representation and the original probability graph. However, due to the decrease in the number of nodes considered when using content chunks, growing a room on the graph has less overall nodes and paths to consider. In Figure 5d, having fewer paths and nodes means that objects are less constrained (when compared to Figure 4b, keeping the set semantics while being able to prune some of the search space.)

One interesting consequence of using content chunks is that more combinations from the motif are expressed, such as the three beds that appear in Figure 5d. While three double beds may seem improbable for a room, their existence in the motif means that there is a statistically significant (although low) chance that those objects can exist in the location. When sets are not an attribute, the global section of three beds is far less likely, as search stops once one bed node is selected. This is due to the ordering of like nodes in the motif, and therefore makes it much more difficult for those algorithms to express rooms that are known to be possible (due to their existence in the motif).

The reductions from nodes to content chunks have a similar effect on the total size of the motif. For our two test data-sets, this reduction effect is approximately 45% for Kermani et al.'s generated motif and 50% for Saava's. Therefore, both methods capture a lot of replicated objects in the SUNCG data-set, which are not expressed to the same extent in their generation methods.

5 DISCUSSION AND CONCLUSIONS

We introduced content chunks, a generalized node representation for objects in graph-based representations of motifs. By using it, procedural room generation can be broken up into what is learned and what is generated, allowing for a more in-depth comparison

between different methods. The compactness of motifs using content chunks has a positive effect on room generation. For example, in our tests, this increased the likelihood of replicated objects.

Most graph-based PGR algorithms grow the graph sequentially, testing the addition and configuration of single objects one at a time, using a generate and test approach [25]. For the generation of a room containing only a few objects, the placement chunk does not provide much information. Furthermore, slight changes to traditional methods, such as group based MCMC [15], show that adding and configuring a single object is neither efficient nor sufficient. Instead, using content chunks allows for group-aware addition by accounting for object sets.

Due to the generality afforded by content chunks, it is not immediately apparent how the described structure can be learned from real data-sets. Based on the reductions presented in Section 4, content chunks can subsume many different representations. A cursory examination of the generated motifs shows that these representations capture different information, even though they mine the same location data. One possible approach, rather than designing one algorithm to learn content chunks, is to discover how to aggregate the information gleamed from other approaches. Doing so would also allow us to more thoroughly compare the node representation on room generation, as the data and generation method used can be the same for all comparison methods and the generation method can be the same as well. Furthermore, to the best of our knowledge, there are so far no quantitative metrics to precisely compare the expressive power of motif and generation methods. Developing such metrics is an important advancement to the field, but is outside the scope of this work.

Beyond graph-based representations, deep neural network methods have recently been used to procedurally generate rooms [13, 27]. These methods provide both a generation pipeline as well as new representation schemes. In future work, it would be interesting to assess how the information conveyed by these new methods compares to that of the generalization proposed here to graph-based methods. Furthermore, it would be also interesting to explore possible uses of the functions that are learned by these deep neural networks for improving the interior and exterior probability functions of content chunks. Finally, graph-based models are being used to represent other aspects of a game, such as mechanics and rule-sets [7, 8], or richer game level semantics [10, 24, 26]. These are all semantic representations, and therefore it would be worth exploring whether content chunks could be extended with even richer semantics.

ACKNOWLEDGMENTS

We would like to thank Shib Duman for donating compute time for generating the bedrooms. We would also like to thanks Jessica Randall and Chris Cook for their suggestions and edits. This work is part of the VESP project (Virtual E-Coaching and Storytelling Technology for Post-Traumatic Stress Disorder), financed by the Netherlands Organization for Scientific Research (gr. nr. 314-99-104).

REFERENCES

[1] J. Timothy Balint and Rafael Bidarra. 2019. Mined Object and Relational Data for Sets of Locations. (Feb. 2019). https://doi.org/10.4121/uuid:1fbfd4a0-1b7f-4dec-8097-617fea87cde5

[2] Angel X Chang, Mihail Eric, Manolis Savva, and Christopher D Manning. 2017. SceneSeer: 3D Scene Design with Natural Language. *arXiv preprint arXiv:1703.00050* (2017), 10.

[3] Kate Compton and Michael Mateas. 2006. Procedural Level Design for Platform Games.. In *the Second Artificial Intelligence and Interactive Digital Entertainment Conference*. AAAI Press, Marina del Rey, California, 109–111.

[4] Matthew Fisher, Daniel Ritchie, Manolis Savva, Thomas Funkhouser, and Pat Hanrahan. 2012. Example-based synthesis of 3D object arrangements. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 135.

[5] Matthew Fisher, Manolis Savva, and Pat Hanrahan. 2011. Characterizing Structural Relationships in Scenes Using Graph Kernels. *ACM Trans. Graph.* 30, 4 (July 2011), 34:1–34:12. https://doi.org/10.1145/2010324.1964929

[6] Qiang Fu, Xiaowu Chen, Xiaotian Wang, Sijia Wen, Bin Zhou, and FU Hongbo. 2017. Adaptive synthesis of indoor scenes via activity-associated object relation graphs. *ACM Transactions on Graphics* 36, 6 (2017), 13.

[7] Michael Cerny Green, Ahmed Khalifa, Gabriella AB Barros, Tiago Machado, Andy Nealen, and Julian Togelius. 2018. AtDELFI: automatically designing legible, full instructions for games. In *the 13th International Conference on the Foundations of Digital Games*. ACM, Malmo, Sweden, 17:1–17:10.

[8] Matthew Guzdial and Mark Riedl. 2016. Toward game level generation from gameplay videos. *arXiv preprint arXiv:1602.07721* (2016), 8.

[9] FrederickW.P. Heckel and G.Michael Youngblood. 2011. Contextual Affordances for Intelligent Virtual Characters. In *Intelligent Virtual Agents*, HannesHÃűgni VilhjÃąlmsson, Stefan Kopp, Stacy Marsella, and KristinnR. ThÃąrisson (Eds.). Lecture Notes in Computer Science, Vol. 6895. Springer Berlin Heidelberg, 202–208.

[10] DaniÃńl Karavolos, Anders Bouwer, and Rafael Bidarra. 2015. Mixed-Initiative Design of Game Levels: Integrating Mission and Space into Level Generation.. In *the 10th International Conference on the Foundations ofDigital Games (FDG 2015)*. ACM, Pacific Grove, CA, 8.

[11] Z. Sadeghipour Kermani, Z. Liao, P. Tan, and H. Zhang. 2016. Learning 3D Scene Synthesis from Annotated RGBâĂŘD Images. *Computer Graphics Forum* 35, 5 (2016), 197–206. https://doi.org/10.1111/cgf.12976

[12] Jassin Kessing, Tim Tutenel, and Rafael Bidarra. 2012. Designing semantic game worlds. In *Proceedings of the The third workshop on Procedural Content Generation in Games*. ACM, 2.

[13] Manyi Li, Akshay Gadi Patil, Kai Xu, Siddhartha Chaudhuri, Owais Khan, Ariel Shamir, Changhe Tu, Baoquan Chen, Daniel Cohen-Or, and Hao Zhang. 2019. GRAINS: Generative Recursive Autoencoders for INdoor Scenes. *ACM Trans. Graph.* 38, 2 (Feb. 2019), 12:1–12:16. https://doi.org/10.1145/3303766

[14] Yuan Liang, Fei Xu, Song-Hai Zhang, Yu-Kun Lai, and Taijiang Mu. 2018. Knowledge graph construction with structure and parameter learning for indoor scene design. *Computational Visual Media* 4, 2 (2018), 123–137.

[15] Yuan Liang, Song-Hai Zhang, and Ralph Robert Martin. 2018. Learning guidelines for automatic indoor scene design. *Multimedia Tools and Applications* (2018), 1–21.

[16] Tianqiang Liu, Siddhartha Chaudhuri, Vladimir G. Kim, Qixing Huang, Niloy J. Mitra, and Thomas Funkhouser. 2014. Creating Consistent Scene Graphs Using a Probabilistic Grammar. *ACM Trans. Graph.* 33, 6 (Nov. 2014), 211:1–211:12. https://doi.org/10.1145/2661229.2661243

[17] Rui Ma, Honghua Li, Changqing Zou, Zicheng Liao, Xin Tong, and Hao Zhang. 2016. Action-driven 3D Indoor Scene Evolution. *ACM Trans. Graph.* 35, 6 (Nov. 2016), 173:1–173:13.

[18] Rui Ma, Akshay Gadi Patil, Matthew Fisher, Manyi Li, SÃűren Pirk, Binh-Son Hua, Sai-Kit Yeung, Xin Tong, Leonidas Guibas, and Hao Zhang. 2018. Language-driven Synthesis of 3D Scenes from Scene Databases. *ACM Trans. Graph.* 37, 6 (Dec. 2018), 212:1–212:16. https://doi.org/10.1145/3272127.3275035

[19] Siyuan Qi, Yixin Zhu, Siyuan Huang, Chenfanfu Jiang, and Song-Chun Zhu. 2018. Human-centric indoor scene synthesis using stochastic grammar. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Salt Lake City, Utah, 5899–5908.

[20] Manolis Savva, Angel X Chang, and Maneesh Agrawala. 2017. Scenesuggest: Context-driven 3d scene design. *arXiv preprint arXiv:1703.00061* (2017).

[21] Manolis Savva, Angel X Chang, and Pat Hanrahan. 2015. Semantically-enriched 3D models for common-sense knowledge. In *the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. Boston, MA, 24–31.

[22] Thomas Smith, Julian Padget, and Andrew Vidler. 2018. Graph-based Generation of Action-Adventure Dungeon Levels using Answer Set Programming. In *Proceedings of FDG 2018*. Malmo, Sweeden, 10.

[23] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. 2015. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *the IEEE conference on computer vision and pattern recognition*. 567–576.

[24] Adam J Summerville, Morteza Behrooz, Michael Mateas, and Arnav Jhala. 2015. The learning of zelda: Data-driven learning of level topology. In *the 10th International Conference on the Foundations of Digital Games*. Pacific Grove, CA, 5.

[25] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne. 2011. Search-Based Procedural Content Generation: A Taxonomy and Survey. *IEEE Transactions on Computational Intelligence and AI in Games* 3, 3 (Sept. 2011), 172–186. https://doi.org/10.1109/TCIAIG.2011.2148116

[26] Roland Van der Linden, Ricardo Lopes, and Rafael Bidarra. 2013. Designing procedurally generated levels. In *Proceedings of the second workshop on Artificial Intelligence in the Game Design Process*. AAAI Press, AAAI Press, Palo Alto, CA, 41–47.

[27] Kai Wang, Manolis Savva, Angel X. Chang, and Daniel Ritchie. 2018. Deep Convolutional Priors for Indoor Scene Synthesis. *ACM Transactions on Graphics* 37, 4 (Aug. 2018), 70:1 – 70:14.

[28] Yi-Ting Yeh, Lingfeng Yang, Matthew Watson, Noah D. Goodman, and Pat Hanrahan. 2012. Synthesizing Open Worlds with Constraints Using Locally Annealed Reversible Jump MCMC. *ACM Trans. Graph.* 31, 4 (July 2012), 56:1–56:11. https://doi.org/10.1145/2185520.2185552