Toward Content-Aware 3D Scene Retrieval using Natural Language

Elizabeth Bradley

Advisor: Professor Thomas Funkhouser

Submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Science in Engineering

Department of Computer Science

Princeton University

June 2017

I hereby declare that I am the sole author of this thesis.

I authorize Princeton University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Elizabeth Bradley

I further authorize Princeton University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Elizabeth Bradley

Abstract

This thesis establishes a methodology for building natural language query-based retrieval systems for large datasets of 3D scenes. We present our results through the introduction of a proof-of-concept search engine, SUNCG Search, indexed on the SUNCG dataset of approximately forty-five thousand richly-annotated scenes. We find that for the retrieval tasks of dataset exploration and subset identification, our novel approach of identifying binary spatial relationships between objects greatly decreases the time users spend on these tasks. Additionally, we introduce unique solutions to information retrieval's prototypical difficulties – result visualization and query interface design – with specific application to the 3D scene use case.

Acknowledgements

Sola Deo gloria.

This thesis is the culmination of a most unlikely journey. I arrived to Princeton as a potential Politics major and am leaving as an engineer. This is an outcome only God could orchestrate. I want to communicate my warmest thanks to the following:

My thesis advisor and repeated rescuer **Professor Thomas Funkhouser**. Despite my misfortune in losing two advisors midstream to career changes, Tom swooped in on both occasions to welcome me into the Graphics Group fold. I am grateful for his guidance in finding this fascinating field of scene retrieval and the generosity with which he always shared his knowledge and time.

The inimitable post docs **Angel Chang** and **Manolis Saava**. Their investment and excitement in this thesis gave me the boost I needed when it seemed that the database would never be efficient. Their patience and willingness to contribute their knowledge and resources to this project made it easily ten times better than it would have been without them.

Professor Christiane Fellbaum, my linguistics consultant.

My dear friends **Erynn Kim**, **Marisa Salazar**, **Ruby Shao**, **Sofia Gallo**, and **Annie Lu** whose support these last four years has made me a better person. The whole of the **Princeton Evangelical Fellowship**. and my mentors **Debbie Boyce**, **Danielle Sallade**, **Allie Harjo**, and **Anna Megill**.

Adam Gallagher, my supporter through all of this.

My family. **Mom**, **Dad**, and **Jeremy**. You have always been the best team.

> For the love of Christ controls us, because we have concluded this: that one has died for all, therefore all have died; and he died for all, that those who live might no longer live for themselves but for him who for their sake died and was raised. – 2 Corinthians 5:14-15

To Adam

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Image retrieval is a popular problem in computer vision, attracting solutions from both academia and commercial entities. There are two broad approaches to image retrieval: content-aware and content-unaware. Content-unaware image retrieval relies on metadata, e.g. text from the surrounding webpage or human annotations – which can be misleading or otherwise incorrect. The content-aware approach tackles the challenge of identifying salient features within the candidate images themselves that match a given query.

In this project, we adapt this content-aware approach to the related field of 3D scene retrieval. This field – which tasks itself with searching and retrieving complicated three-dimensional structures – has as of yet been only preliminarily explored due to the relative dearth of large 3D scene datasets and the inherent complexity of 3D scene representation.

In some ways, scene retrieval is a more straightforward task than image retrieval. 3D scene datasets often arrive highly-structured and richly-annotated with pre-categorized objects and rooms. Yet, unless a proper approach is chosen, 3D scene retrieval can be an intractable task given that scenes can be arbitrarily more complicated than an image (due to the increased dimensionality) and involve the unique

challenge of viewpoint dependency. In other words, unlike content-aware image retrieval algorithms which can search between images without altering their representation, the corresponding scene retrieval algorithm must be able to adapt to arbitrary views, including users who, hoping to find scenes with "doorbells to the left of doors," presuppose a canonical view of a home (a view from the exterior) which the algorithm must identify and adapt it accordingly.

One challenge posed to both image and scene retrieval systems is the inherent difficultly of mapping natural language, which is by nature discrete, to the continuous geometries of images and scenes. The key research novelty of this thesis is the introduction of a robust procedure to identify binary spatial relationships which have both specific geometric descriptions and one-to-one mappings with English prepositions. Additionally, since these relationships are binary, their storage costs are constant and their querying well-defined. These features reduce some of the difficulties presented by any large dataset: the inherent inability to compute on-line.

This approach further capitalizes on the human understanding of scenes as inherently ordered. This ordering manifests itself in indoor scenes in the following way: scenes contain floors that contain rooms that contain objects. In fact, the traditional scene representation is a scene graph in which parents' nodes are determined by another binary spatial relationship: containment. In short, humans understand scenes by defining relationships within them. Therefore, a natural language search engine that understands and leverages the power of relationships will be more successful in yielding expected results to a human user.

This thesis is organized as follows:

First, in Chapter 2, we survey the field of information retrieval and the solutions that image retrieval and text retrieval offer to the archetypical challenges of the field. Additionally, we enumerate the small population of the commercial and academic 3D scene retrieval systems. In Chapter 3, we broadly introduce our approach to-

ward constructing a 3D scene retrieval system. In Chapter 4, we exhibit SUNCG Search – our proof-of-concept scene retrieval engine – and describe both technical and generalizable decisions made in its development. In Chapter 5, we discuss both the qualitative and quantitative evaluations of SUNCG Search through the lens of a user study. In Chapter 6, we survey the research implications of SUNCG Search, and in Chapter 7, we discuss the limitations (both inherent and logistic) of this thesis and offer potential avenues for improvement and exploration in the promising field of 3D scene retrieval.

Chapter 2

Related Work

This paper is among the first in the subfield of content-aware 3D scene retrieval on large datasets. Therefore, this section will reflect work that highlights the inherent difficulties for information retrieval tasks and explore various design decisions made in the fields of image retrieval and text retrieval. We will explore the implications that these difficulties pose for scene retrieval in Chapter 4.

Additionally, we explore previous work in in the geometric defining of spatial relationships in 3D spaces and content-unaware 3D scene retrieval.

2.1 Information Retrieval

2.1.1 Result Visualization

Text retrieval systems have long adopted the *search snippet* as the de facto visualization of results. A *search snippet* is a query-dependent subset of text on the page such that a user may verify at a glance the suitability of the result to their needs without further exploration, e.g. without navigating to the web page. Generally, image retrieval systems eschew the *search snippet* in favor of returning the entire image, as the image itself is often the most effective communicator of a result's suitability for

the user's use case.

The content of the snippets is essential to the user experience of an information retrieval system. In their historic paper in the field which predates the founding of major commercial search engines such as Google, Tombros and Sanderson (1998) found that users of internet search engines rely almost exclusively on snippets when performing relevance judgments of results, navigating to the webpage of potentially suitable results so rarely as to make such navigation only be a final measure when deciding between a small set of similar high-quality options. From this, we can conclude that if the *search snippet* is poorly formed or otherwise misleading in its apparent suitability, users will effectively never consider the corresponding result, regardless of its quality.

Additionally, the snippets must be well-ordered, as O'Brien and Keane (2006) discovered when they found through users studies that most users only consider snippets returned on the first page of results with a heavy bias toward reading fully only the first three or four results. If the snippets considered are deemed unsuitable, users will more often attempt a new query rather than scrolling down or navigating to another screen. Therefore, since users of information retrieval systems expect excellence from their results, figuring that few unsuitable results are the result of a poor query rather than just less suitable than lower ranked results, the ranking of these results must be done with careful consideration of features deserving of high ordering.

As Cutrell and Guan (2007) demonstrated through their eye-tracking studies, the structure and detail of the *search snippet* affects the effectiveness with which users can accomplish tasks. They found that a longer, more detailed query-dependent snippet yields greater performance for users tasked with asking informational queries but negatively impacts their performance on navigation queries due to the extra bloat. For example, many users, when tasked with looking for specific websites, skim over relevant results due to the perceived glut of the snippet.

Most text retrieval systems employ a variant of summarization or query-dependent text selection from the result's source document for their *search snippet*. However, for our purposes – as viewpoint-selection and therefore "scene summarization" was outside the scope of this work – we were interested in early research that challenged this paradigm by presenting snippets as distribution diagrams. In her historic paper, Hearst (1995) introduces a system called TileBars where the *search snippet* is a heatmap of query-matching text in the source document. Using TileBars, users can understand at a glance whether their query is the topic document (i.e. distributed uniformly) or merely a subject (i.e. distributed in a cluster). We ultimately chose to emulate such systems through image highlighting as a means of communicating the distribution of a given query in a result scene (see Section 4.3.2 for further discussion).

2.1.2 Content-Aware vs. Content-Unaware

Although no work has been done in the field of content-aware scene retrieval, researchers have long been working in the related field of 3D object retrieval. In their introduction of shape-based search methods, Funkhouser et al. (2003) conclude that the techniques applied to text retrieval cannot be naively applied to 3D retrieval tasks. They state that relying on human annotations is imperfect at best and incorrect at worst, citing three possible scenarios of failure: "when all related keywords are so common that the query result contains a flood of irrelevant matches (e.g., searching for faces: i.e., human not polygonal), when relevant keywords are unknown to the user (e.g., objects with misspelled or foreign labels), or when keywords of interest were not known at the time the object was annotated."

2.2 Spatial Relationships

The concept of geometric descriptors for spatial relationships has been a historic area of research in the fields of computer graphics as well as linguistics.

Early work from computer scientists includes Abella and Kender (1993) who defined a small set of relationships – including *near*, *above*, and *below* – by applying a set operators to the bounding boxes of objects. This approach, although simple and computationally inexpensive, proved to be too imprecise for this thesis. The small set of relationships able to be defined was limiting. Additionally, calculations using bounding boxes also proved to be misleading as bounding boxes fail to accurately describe irregular, non-rectangular shapes. This results in collision false positives, situations in which bounding boxes intersect when the objects are merely close – e.g. a chair pushed in under a table.

Retz-Schmidt (1988), a key inspiration in defining the vocabulary this thesis uses to discuss spatial relationships, identifies the inherent difficulties variance poses for certain viewpoint-dependent relationships, e.g. *right of* and left of. This ambiguity invalidates geometric descriptors for such relationships in the general case, and for this reason we omit viewpoint-dependent relationships from our set.

A fascinating research direction that embraces the ambiguity and imprecision of identifying relationships is presented by Gapp (1994) and Abella and Kender (1993). These papers introduce the notion of *fuzzifying* geometric descriptors by applying Monte Carlo simulations to determine whether an object matches the descriptor. Such approaches acknowledge the often vague and imprecise definitions of English prepositions, an fitting application to natural language that we hope to emulate.

2.2.1 Spatial Relationships in Scene Understanding

Several researchers have approached the task of defining spatial relationships in 3D scenes. In Fisher et al. (2012), the set of 3D scenes are relatively small and complicated – e.g. the surface of a desk – and it is therefore concluded that only *support* and *attachment* relationships are descriptive enough for scene generation. Savva et al. (2017) arrives at the same conclusion. However, since our dataset is comprised of much larger and sparser scenes, we assert that broader relationships are descriptive of functional designs present in our dataset that would not be in smaller spaces, so for example, chairs *facing* a table is indicative of a functional space for eating.

In Chang et al. (2014), Fisher et al. (2012), and Savva et al. (2017), the geometric descriptors for spatial relationships are calculated via heatmaps. However, because of the simplicity and restrictive viewpoint of these calculations, this approach is too ambiguous to yield consistent results, so we propose a more robust and specific solution via object-to-object measurements that allows for a wider range of spatial relationships.

In Hu et al. (2015) and Hu et al. (2016), researchers found that the functionality of objects is a substantive descriptor for an object, i.e. function distinguishes one object class from another. This conclusion impressed on this project the importance of indexing function as humans design their 3D spaces with functional use in mind. We elected to capture functionality through our set of spatial relationships and broad definition of potential object pairs (up to three-meters).

Finally, this project's conceit of performing scene retrieval via annotated scene graphs was directly inspired by Johnson et al. (2015) which applies annotated scene graphs to a varied set of image retrieval tasks. The paper illustrates the importance of having precise content-aware annotations when querying large datasets.

2.3 Previous Attempts at Scene Retrieval

Most other attempts – both academic and commercial – at large-scale 3D scene retrieval are ultimately variations on tag-based metadata searches.

2.3.1 Academic

Koutsoudis et al. (2012) proposes a scene retrieval system that is a searchable catalog of world heritage sites. Although the system elects to represent search results as annotated scene graphs, searching is performed via traditional text retrieval methods on manually defined metadata – e.g. field reports from experts. This approach works for the paper's application, but the approach is not generalizable since extensive text descriptions are too work-intensive for user on large datasets.

In a different vein, Zhao et al. (2014) proposes a scene retrieval system that indexes objects in 3D scenes via a new topological descriptor they call the "Interaction Bisector Surface" which precisely describes the functional use of a single object. The innovation of such a descriptor and the precision with which it can return similar objects illustrate the power inherent to capturing function when indexing. However, the "Interaction Bisector Surface" is many-parameter topology that neither maps to natural language nor to any visual representation except a precise 3D surface. Querying via this system is truly a chicken-and-egg problem: users query for scenes containing an object class with a specific function by providing an object from some scene with that specific function. The issue is clear: how does a user find an object suitable to query for their task without a means of searching for such objects in the dataset?

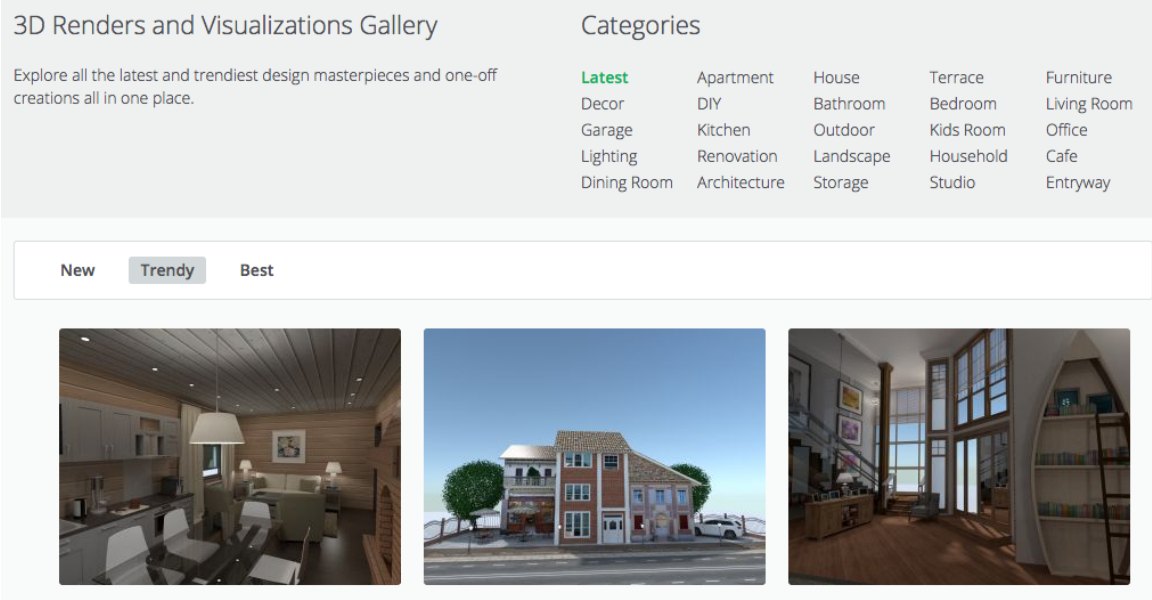

Figure 2.1: Screenshot of the Planner5D home design software search interface

2.3.2 Commerical

The home design software company Planner5D offers a 3D scene retrieval system for users to view scenes created through the Planner5D platform. In this retrieval system (see Figure 2.1), 3D scenes are labeled at creation with one of twenty-five category labels – e.g. studio, office, renovation. Users of the search feature can select from among these categories (or elect to view all categories at once) and then order the results either by recency of creation, recent popularity amongst other users, or overall popularity. Such a system is effective for users looking only for the the best-crafted scenes but would fail users seeking to gain an comprehensive understanding of the overall dataset. Additionally, this system does not allow users to perform recall tasks since the order of results in two of the three modes are liable to change over time as relatively popularity shifts and in the third mode excludes all but the newest scenes.

A fansite for the popular video gaming franchise *The Sims* called "The Sims Resource" boasts a mature 3D scene retrieval system. Users of the site upload building plans and in-game images of their often meticulously detailed scenes. This repository

of scenes can then be searched using a parameterized search which can filter via metadata such as lot area, number of levels in the structure, and the number of items in the scene. Additionally, the scene retrieval system allows for full-text search of user-provided explanations. Despite the increased feature set over Planner5D, this is still at its core a metadata search and not content-aware. Additionally, this system makes no attempt at solving the problem of query-specific ranking, instead preferring to return scenes in the order of popularity (number of likes) given by other users.

Chapter 3

Approach

The goal of this thesis is to present the design decisions and considerations of a proof-of-concept system for the exploration of large 3D scene datasets. Although we detail a specific implementation, we seek to present generalizable principals, applicable to researchers developing systems for related 3D scene tasks.

The novel element of this thesis is the introduction of efficient methods upon which to index and query the *content* of scenes instead of the *context*, e.g. manually generated metadata. With the recent advent of truly large 3D scene datasets, this ability is essential as the cost of analyzing scenes becomes too expensive to perform at runtime.

Specifically, we introduce binary spatial relationships that possess both specific geometric descriptors and one-to-one mappings with English prepositions. The benefits of such relationships are manifold:

- **Functional Descriptors** We can infer the functional relationship of two objects from the presence of a spatial relationships. For example, if object x is *supported by* object y, we implicitly know that object y is a supporter of x. If object x is a *near* object y, we implicitly know that the two belong to the same functional region.

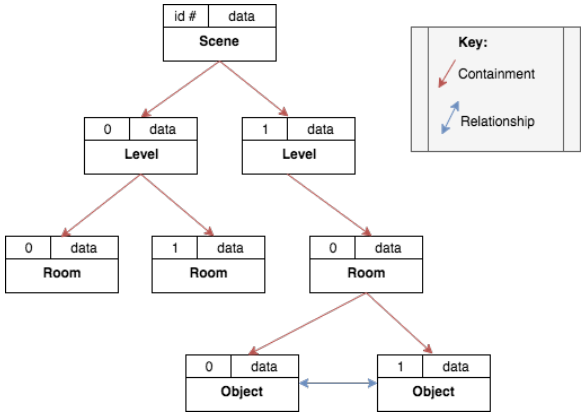

Figure 3.1: A generic scene graph with relationship annotations.

- **Storage Efficiency** Since each spatial relationships is binary, meaning that a pair of objects is *either* defined by the relationship or is not, it requires only one bit of storage per pairwise object.

- **Extends Scene Graphs** In their standard representation, scenes are trees in which arcs represent containing relationships. As Figure 3.1 demonstrates, scenes recursively contain components that contain other components. Pairwise object relationships can be represented as edges between leaf nodes of the scene graph. Therefore, we can leverage historical work in scene representation with only minor modifications for these annotations.

- **Native to Natural Language** As each spatial relationship corresponds uniquely to an English preposition, they are a human-first way of concisely describing content in a scene.

To expand on that final benefit, emphasizing spatial relationships as an efficient proxy for encapsulating scene content allows a querying system to easily leverage natural language instead of a specialized query language. Realizing the potential for a significant natural language element in this database exploration system, we permit users to use natural language instead of a specialized query language. This choice immediately allows our system to be approachable by non-experts as it emulates the

type of commercial search engine with which a user would be familiar. This natural language component is able to be extend into other supplementary features, namely autosuggestion in a context-aware manner.

Finally, we recognize that querying 3D scenes has an implicit representation problem: how to assure users in the correctness of their searches. With our goal of creating a dataset exploration system, we want the bar for entry to be low, meaning results ought to be clear and informative to users. This goal rules out merely returning a text list of string IDs. It is apparent that since a 3D scene can be arbitrarily large and complicated, returning a result view that merely presented a god's-eye view of a scene would be overwhelming and uninformative to a user. Therefore, we implement query-specific visualizations and orderings that allow users to get a feel for why the results were what they were. In the same way that traditional search engines use "snippets" of results to aid users in selecting the optimal option, so too do these enhanced visualization aid users to gauge at a glance the success of their querying.

Chapter 4

Implementation

As stated in Chapter 3, the goal of this thesis is to present the design decisions and considerations of a proof-of-concept system for the exploration of large 3D scene datasets. We do not enumerate all the technical details of this system but instead explain generalizable choices made in response to the unique challenges of 3D scene retrieval tasks. We hope that this mixed theoretical and practical approach will satisfy two types of readers: those implementing their own scene retrieval systems and those interested in the theoretical problems presented by scene retrieval.

This section is structured as follows:

First, we introduce the dataset upon which the proof-of-concept system – SUNCG Search – is designed to query. Then, we present technical specifics of this SUNCG Search. Finally, we explore more generally three challenges inherent to scene retrieval tasks and the design decisions made in SUNCG in response to them.

4.1 The Dataset: SUNCG

The SUNCG dataset (introduced in Song et al. (2017)) contains approximately forty-five thousand distinct 3D scenes. Each scene has been manually created by a user of

the Planner5D home design software[1]. A wide variety of scene types are represented, including single-family homes, office buildings, movie theaters, and holiday resorts. Excepting a few outliers, the scenes have realistic layouts and are quite dense. The average scene contains nine rooms and 124 objects.

A key advantage of SUNCG's synthetic scenes is that they arrive richly-annotated: all rooms have an assigned type (e.g. bedroom, garage) and all objects have two class labels, one general (e.g. kitchen appliance) and one specific (e.g. mixer). Additionally, the precise-to-the-millimeter-accuracy of object locations enabled the precision of the geometric descriptors of spatial relationships (which will be formally introduced in Section 4.3.1).

SUNCG is the largest publicly available dataset of its kind. However, its size motivates the need for 3D scene dataset exploration systems. Although most scenes are realistic, some are anomalous (e.g. a 10m by 10m chessboard) and/or poorly formed (e.g. rooms without doors). Although one could apply preprocessing rules to remove such scenes, the variety of the dataset makes the definition of such rules a rabbit hole-esque task without a means to locate these scenes other than manual inspection. An exploration system would enable researchers to understand the limitations of such a large dataset *before* their training groups are polluted with non-desirable scenes.

4.2 The System: SUNCG Search

As shown in Figure 4.1, our system borrows its design aesthetic from commercial search engines:

- a minimalist search bar wherein users input their queries in natural language
- an accurate count of results

[1]See https://planner5d.com.

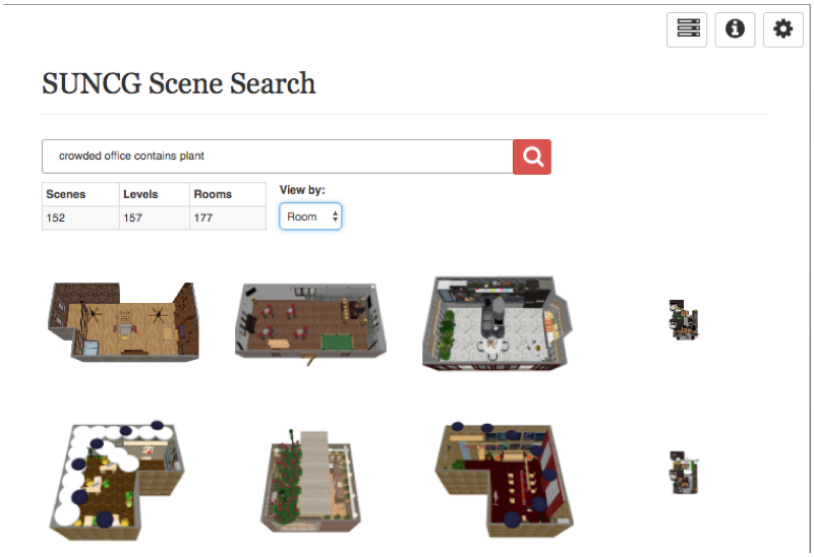

Figure 4.1: A screenshot of the SUNCG Search interface.

- an ordered list of results with a query-dependent "snippet" from which to gauge the result's suitability for the user's task

Notably, SUNCG Search allows users to alter the appearance of the "search snippet" via a selector (highlighted in Figure 4.1). Users can view matching results either as whole scenes, as individual levels, or as individual rooms. This option allows users to modify the snippet according to their task – e.g. some researchers only need to perform learning tasks on rooms, not whole scenes – without navigating to the built-in individual image viewer.

4.2.1 Server Design

As seen in Figure 4.2, SUNCG divides its computational tasks among three servers: one for parsing the natural language query, one for communicating with the database, and one for performing modifications to result images. This division allows for increased parallelization as well as simplifying the task of server maintenance and improvement for the system developer.

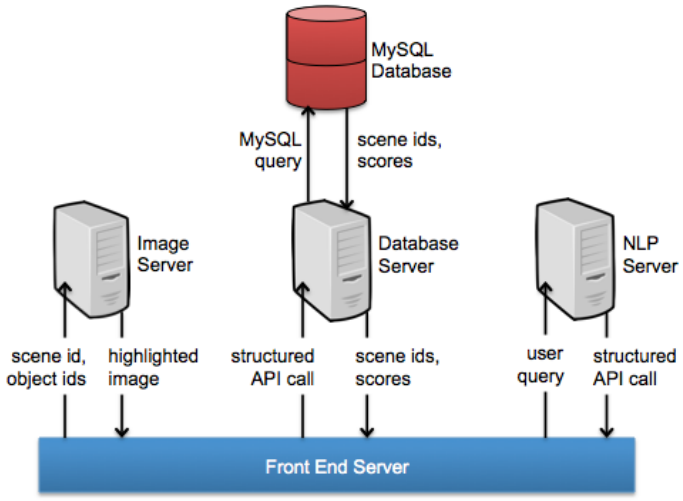

Figure 4.2: SUNCG Search server diagram.

4.2.2 Database Design

As visible in Figure 4.3, our database schema was designed to emulate scene graphs in its construction. There is an instance table for each type of non-leaf node – roots (scenes), levels, rooms[2]– and an instance table for each relationship annotation, meaning that each object-object relationship corresponds to a unique row in the database.

Since implementing a generative system of SQL composition was outside the scope of this project, we elected to handwrite all potential SQL queries as stored procedures on the database.

4.2.3 Query Space

With the decision to handwrite all SQL queries, it was apparent that the query space, the range of natural language queries users can pose to the system, required formal definition for completeness. I interviewed researchers familiar with the SUNCG dataset about potential queries that would aid their daily tasks. This set was aug-

[2]But note that a large amount of denormalization was performed for efficiency.

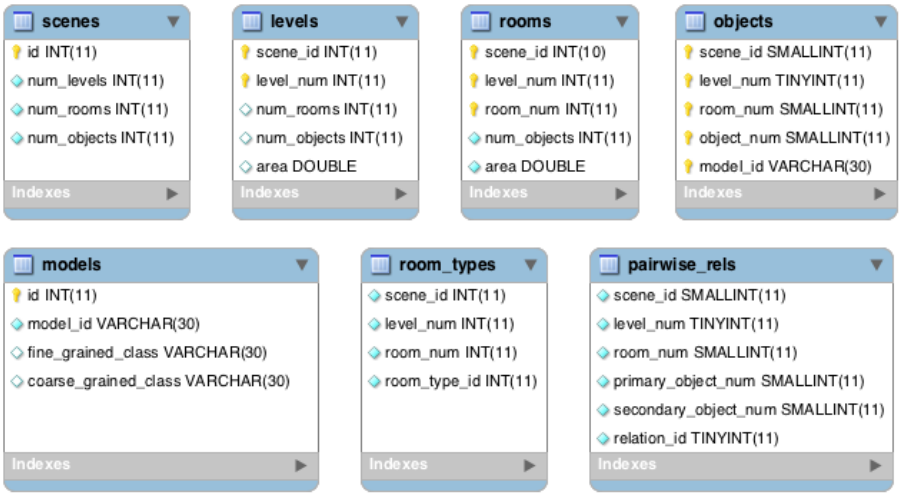

Figure 4.3: Simplified SUNCG database schema

mented by queries that they suggested a first-time user might pose about the dataset.

The suggested queries fell into three broad categories:

- **Subset Identification** queries to isolate a specific scene or set of scenes given some number of characteristics.
- **Outlier Identification** queries to locate outliers.
- **Dataset exploration** queries to lend a user broad understanding of distribution and frequency of facts in the dataset.

After these inquiries, the query space was defined as follows:

Definition. *Queries may be either property queries about the data of a non-object node or binary queries about the existence of a relationship between any ancestor-child pair or an object-object pair.*

See Table 4.1 for a summarization of the query types that map to this query space.

Additionally, with the goal of allowing users the ability write complex queries within this limited query space, we included a logic for chaining queries using conjunction and disjunction. Users can link successive valid queries with conjunctive

Query type	Parameter 1	(Parameter 2)	(Parameter 3)
Adjective	location	too many/many/few/no	object class
Relationship	object class	relationship	object class
Stats	location	small/large/dense/sparse	
Contains	object class		

Table 4.1: Query types and their required parameters

techniques (e.g. using the token "and" or grammar to indicate a multi-part query) or disjunctive techniques (e.g. using the tokens "without" or "but not") to filter the result set by these additional parameters. To illustrate, the query "small scene" and the query "scene with door" can be coalesced into the two-part query "small scene with door" – which yields the set of small scenes intersected with the set of scenes containing doors.

4.3 The Design Decisions

4.3.1 Spatial Relationships

As described in Section 3, this thesis introduces binary spatial relationships as a technique to efficiently index and query the *content* of scenes instead of the *context*. Each spatial relationship is defined by a specific geometric descriptor and a one-to-one mapping with an English preposition.

Defining a Set of Relationships

Even initially our set of relationships included those prepositions that map simply to orthographic directions on a bounding box – *above* and *below* – and those which map to a calculation of distances between closest points – *near* and *touching*. By compositing these prepositions, another set of prepositions can be defined, *hanging*, *supports*, and *supported by*. To illustrate, object x can be said to *support* object y if y is both *above* x and *touching* x.

A further set of relationships arrives from the leveraging of canonical faces that are defined for each object model. The insight here is that people design rooms with function in mind, requiring rooms to enable certain activities that each require specific furniture arrangements. For example, people eat at dining room tables, an activity that requires chairs to be near to the table and oriented facing it, i.e. the chairs' *front* faces are angled toward the table. It follows that our set of relationships can be defined solely by the most semantically meaningful faces, the *front* and the *back*, and this yields the following relationships: *faces*, *in front of*, *faces away*, and *behind*.

We rejected several additional sets of relationships for a variety of reasons. Some prepositions were semantically evocative but lacked a specific geometric descriptor, e.g. *against*. Others were too dependent on textual context, e.g. *on*, because a "plate *on* the table" has a very different geometric interpretation than "light switch *on* the wall." Finally, we did not explore avenues of relationships involving more than two objects (e.g. *between*, which requires three objects) due to our already defined query space which mandates that all relationship queries must be evaluating an annotation (an edge either original or added) in the scene graph. See Section 4.2.3 for discussion of this definition.

Finding Relationships

Initially, we explored the potential of using a heatmap approach (inspired by Fisher et al. (2012)) in which the secondary object in every every object-object pair was transformed to the canonical orientation of the primary object and its location drawn on an xy-surface. These diagrams could then be categorized geometrically into different relationships. However, because of the simplicity of these calculations (as well as the restrictive viewpoint), this approach proved too ambiguous to yield consistent results.

We then explored geometric descriptors that relied on bounding boxes. This ap-

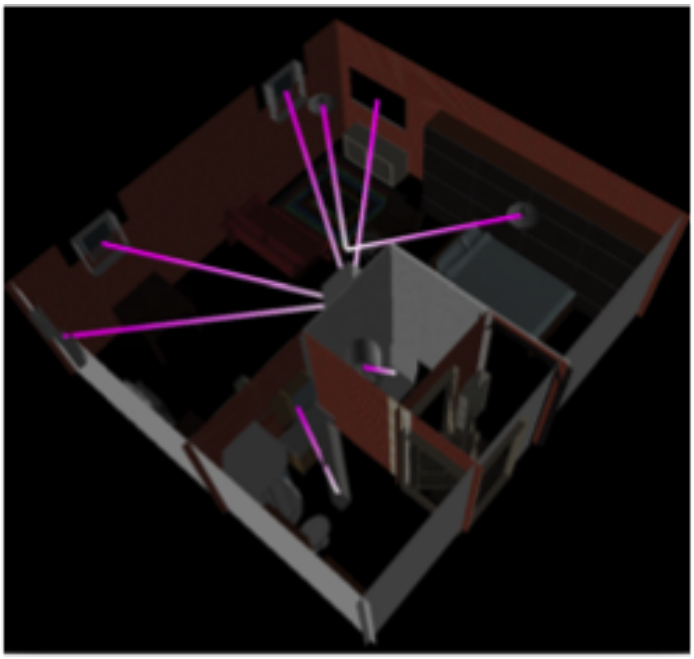

Figure 4.4: Example of all the *hanging* relationships in a room

Relationship	Definition
above	Contained in the x-y column, greater min z value than max z
behind	x is contained in the projection of y's backward face surfaces
below	Contained in the x-y column, less max z value than min z
faces	y is contained in the projection of x's front face surfaces
faces away	y is contained in the projection of x's backward face surfaces
hanging	y is *below* and *touching* x
in front of	x is contained in the projection of y's front face surfaces
supported by	x is *above* and *touching* y
supports	y is *above* and *touching* x
touching	The closest points of x and y have a distance of 0
near (1m)	The closest points of x and y have a distance $\leq$ 1m
near (2m)	The closest points of x and y have a distance $\leq$ 2m
near (3m)	The closest points of x and y have a distance $\leq$ 3m

Table 4.2: Geometric descriptors for relationships between a primary object x and a secondary object y

proach provided more clarity than heatmaps but ultimately proved to be too restrictive, allowing only seven possible geometric relationships. Since one of the expressed goals of this project is expressibility, we pursued another system that would allow for a wider variety of relationships to be identified.

Ultimately, we elected to define the geometric descriptors for the spatial relationships as a function of a specific collection of measurements between object-object pairs. Each object-object pair (x,y) was measured for thirty-five different parameters including distance between the two closest points, a sampled distribution of points on x with a larger z-coordinate than y, and a sampled distribution of the distance from points on x from points on y.

The viability of this solution was made possible by the specificity of the SUNCG dataset, namely the precision with which the location of objects is defined. For example, some relationships – e.g. *above*, *below*, *faces* – require measurements to be taken from the projection of surfaces, an action that requires specific knowledge of the geometry of objects. Therefore, the specificity and variety of relationships achieved in this project may not be feasible for less defined datasets.

4.3.2 Result Visualization

The challenge of effectively displaying search results for 3D scene retrieval lies in their complex nature. Since a scene can be arbitrarily large and complicated, the solution of image retrieval engines, which merely returning the entire result as a snippet would not be viable. Our goal is the same as an image retrieval system, namely to return a result that is understandable as a good result at a glance. An image is just a data structure that is understandable at a glance. On the other hand, an scene, however, which can be arbitrily complicated, may hide what a user wants to see. For example, a fine-grained search like "kitchen with sofa" would yield untenable results if the visualized result on presented an exterior view of the scene wherein neither the

kitchen nor the sofa was visible. Therefore, we focused on a human-changeable system where users could view the outside view (e.g. for an outside query), an inside view (e.g. for an inside query), or a zoomed-in inside view (to the room-level). We did not however, make this selection for users but instead allow them to make a viewpoint selection via a selector.

Ranking

One important element of visualization is the ranking of results. To illustrate the importance of a proper ordering, imagine querying your favorite commercial web-page retrieval system for "pancakes" and merely being returned all matching web addresses in alphabetical order. Filtering works for small datasets, for which manually inspecting a subset is a viable, but it is unhelpful to the point of uselessness on larger datasets. Therefore, we established a means of ranking the results of scenes by a notion of *relevance* where relevance was calculated as follows:

$$scene.rank = \sum_{q \in queries} \frac{value(q, scene)}{\sum_{s \in scenes} value(q, s)} \tag{4.1}$$

$$value(query, scene) = \begin{cases} query.type \text{ relationship} & \#\text{ satisfying relationships} \\ query.type \text{ property } p & scene.p \end{cases} \tag{4.2}$$

Seen in Equation 4.1, the rank calculation is normalized over all queries. Seen in Equation 4.2, we count the number of satisfying relationships for a given query instead of merely returning a true or false. This prevents the "pancakes" problem detailed above. We chose to be opinionated: users searching for "scenes with tables" will be returned scenes with the highest number of tables as we judge those scenes to be the most relevant to the query.

Result Rendering

As previously discussed, SUNCG Search allows users to alter the appearance of the "search snippet" via a selector. Users can view matching results either as whole scenes, as individual levels, or as individual rooms. This option allows users to modify the snippet according to their task (e.g. some researchers only need to perform learning tasks on rooms, not whole scenes) without navigating to the built-in individual image viewer. Additionally, we introduce a query-dependent feature that highlights the satisfying objects in a query. For example, given a relationship-type query such as x *above* y, the resulting image would have each instance of both type x and type y in a satisfying pair highlighted (each type in a different color for clarity). As shown in Figure 4.5, containment-type queries also highlight satisfying object instances. For multi-part queries, each subquery (of appropriate type) is assigned a different color palette so as to maintain distinct information. The inspiration for this feature arrives from the related work discussed in Section 2.1.1. Given the importance of the result rendering to a user judging the suitability of a result, we elect to explicitly communicate the matching elements to a user.

4.3.3 Natural Language Processing

As introduced above, we elected to use natural language as our mode of user-interaction for two reasons: (1) maintaining extensibility and (2) simplifying composite and disjoint multi-part queries.

Extensibility. The alternative, pre-defined query-building interfaces, excels within limited sample spaces. Through explicit listing of query types, these interfaces make the range of selections available apparent to users at a glance. This simplicity enables novice users to quickly adapt to the system and additionally never experience frustration with how to word queries. However, in deriving the query-space (as outlined in the Section 4.2.3), we knew that we had over twenty unique query-types with

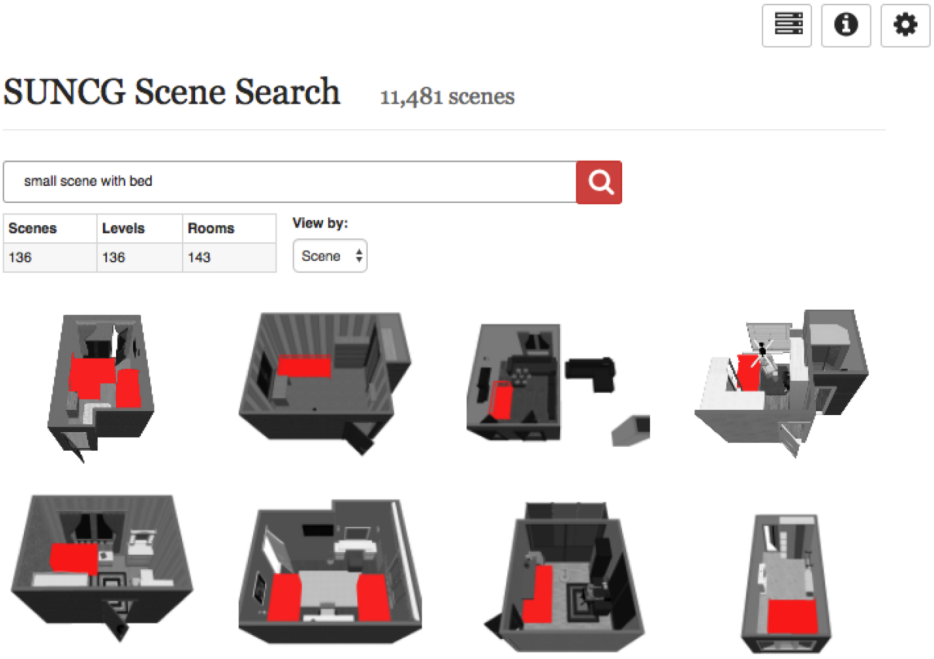

Figure 4.5: Screenshot of results with query-specific image highlighting.

large amounts of variation within those types. Therefore, a natural language interface would remain as simple as ever, not succumbing to the glut of a more explicit interface.

Simple multi-part queries. For the complex data filtering tasks implicit in single scene location or subset identification, it is apparent that a search engine for a large dataset requires support for multipart queries. In prototyping pre-defined query-building interfaces, it became apparent that the complexity of design would be too unwieldy for human use when compared to the simplicity of natural language. We elected to support conjunctive and disjunctive subqueries, signaled by the tokens "and" and "but not," respectively.

What follows are explorations of decisions made while implementing the NLP system.

DFA

The natural language processing step went through multiple iterations throughout the development of this project. One major limitation that presented itself early on was that all the database queries could not be generated on the fly and therefore occupy only a very specific query space (see Section 4.2.3 for further discussion on this limitation). Therefore, our NLP solution needed to direct user-inputted queries into very specific buckets (i.e. the predetermined database queries).

We elected to implement a DFA-inspired system (see Figure 4.6). Unlike a traditional DFA, our system would propagate information on each transition (e.g. the name of the roomtype) and on every *accept* state our system emits a well-formed intermediate representation of the query that matches a prewritten query (see below for intermediate representation). The advantages of a DFA-approach over a pattern-based approach is that it readily handles alternate wordings as well as articles of speech without pattern glut. For instance, in the pattern based approach, one has to have many patterns to represent a simple query such as "a person above an ottoman" because of the modifiers "a" and "an."

Additionally, the DFA-approach has a speed advantage over more complex approaches since it runs in linear-time to the length of input. This is a great result for the extensibility because it implies that extending the system to handle more query types, i.e. more states, will not negatively impact the running-time of the query parsing.

Intermediate Representation

One perennial challenge of natural language processing is communicating to the user how their query was understood. In most information-retrieval systems, this difficulty is elided in favor of merely letting users decide whether the results are suitable for their needs. Knowing that 3D scenes already have an implicit opaqueness (e.g. the

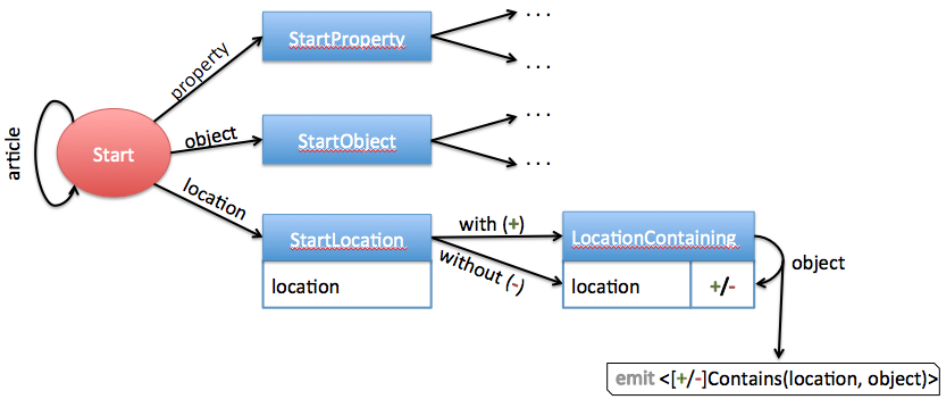

Figure 4.6: Illustrative subset of the query-parsing DFA

Query	Intermediate representation
office with a desk	+contains(office, desk)
room without many plants	-stats(room, many, plants)
scene that has a sofa above a rug	+relationship(scene, sofa, above, rug)

Figure 4.7: Sample intermediate representations of simple queries

default rendering of a scene place the viewpoint "outside" a residential scene without any view of the interior), we elected to explicitly communicate to users how their query was parsed through what we termed an *intermediate representation*.

Our goals for the intermediate representation were (1) readability at a glance and (2) succinctness. Our specific solution involved representing queries as ordered tuples: a tuple represents a single subquery, outside the tuple is both a sign (signifying inclusion/exclusion) and the query category, and inside the tuple is the associated data in query category-specific order. See Figure 4.7 for sample translations from simple queries to their corresponding intermediate representation.

When enabled, our system displays the intermediate representation of the query below the query box after a search is completed. Users can then examine it to judge whether their query translated as desired. If not, they can revise their query without the extra time spent examining results to determine whether the query was satisfying.

Furthermore, for users who wanted a more precise experience (i.e. researchers

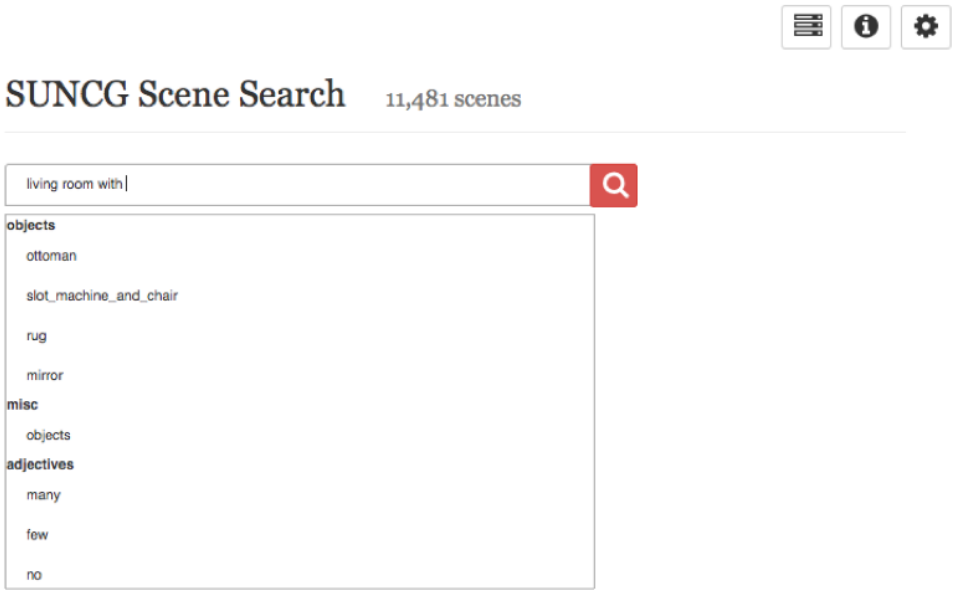

Figure 4.8: Screenshot of autosuggestions for a partial query

using the retrieval engine for daily tasks), we built in functionality to the NLP server so users could input their query using the intermediate representation instead of natural language.

Autosuggestions

Autosuggestion is a user-experience feature that extends naturally from the DFA approach of query-parsing.

By definition, each state s of a DFA has a transition function δ_s and given a valid token t, the machine will transition to some new state $\delta_s(t)$. Therefore, the domain δ_s is the set of valid tokens that would maintain query coherence.

In our system, we collected partial queries from users in real-time, simulated a pass through the DFA, and rendered the set of valid tokens as autosuggestions below the query box. The benefit of this query-specific assistive feature is that it informs novice users of potential types of queries as well as their valid syntax before submission. See Figure 4.8 for a screenshot of this feature in SUNCG Search.

Chapter 5

Evaluation

As this project seeks to provide generalized considerations for search engines built on 3D scenes and to describe the technical implementation of SUNCG Search, we provide multiple types of evaluation.

First, we present analysis of the generalized principals through comments about correctness. Then, we present analysis of speed per benchmarks of our proof-of-concept system SUNCG Search. Finally, we present a user study about SUNCG Search and its results to discuss the implications of design decisions.

5.1 Correctness

We specifically defined the query space available to a user as the traversal of a single relationship (either object-to-object or a containing relationship) or an informational query about a single node. With this limitation, it was feasible to handwrite each database query corresponding to a potential user query. For example, the query "small bedroom" corresponds to a MySQL stored procedure that sorts all bedrooms by size and returns those under a square-foot threshold. Since each query is handwritten, the stored procedure can then be tested and corrected before ever being called from the application. This is in direct contrast with a generative model of database interaction,

namely generating SQL on the fly from arbitrary user queries. The implicit benefit of a system proven correct by construction is the increased faith of users in the validity of results.

5.2 Speed

Despite the large volume of data being queried – recall, approximately 45,000 scenes – all queries complete in sub-600 milliseconds. This figure includes both the time required for natural language parsing and calling the stored procedure on the database. As explained in Section 4.3.3, the natural language parsing is, in practice, instantaneous due to the linear-time complexity of parsing in a DFA.

The bottleneck of the application is the modification and retrieval of images for the *scene snippets*. In fact, the average time to retrieve a single image through an HTTP request is 1.2 seconds, twice that of the combined time of parsing and database querying. An immediate speed-up would result if the images were moved to disk instead of being accessed from another server.

5.3 User Study

I carried out a user study to evaluate the comparative benefit of SUNCG Search's result visualization, natural language processing, and the expressiveness of the query space against a system without these features.

5.3.1 Hypotheses

H1: Users exploring the dataset with the improved visualizations would more accurately be able to ascertain facts about the dataset, i.e. distributions.

H2: Composing a query to accurately partition a subset will take less time with the

ability to query by relationships and statistics than without that ability.

5.3.2 Methods:

Participants

I recruited seven Princeton undergraduate students (5 female, 2 male) through a college listserv that has no engineering affiliation. No participants had previous experience or knowledge in the fields of 3D scene understanding or information retrieval. Users were only required to meet a single criterion, namely to *have used a search engine*.

Design

Each user completed two timed tasks with the goals of dataset exploration and subset identification respectively. Upon completion, users answered qualitative questions about the design of the system. Users performed each task twice, alternating between SUNCG Search and a control scene retrieval system[1] which removed query support for relational and statistical queries, query-specific visualization, and query-specific ranking.

Procedure

1. **Dataset Exploration** On their assigned system, users were given three minutes to *explore* the dataset, which was defined for them as understanding the *average* scene and locating outliers. At the conclusion of the three minutes, users were asked a slate of qualitative questions including "Wwat types of scenes were represented in the dataset?" and "what surprised you about the dataset?" Users were also asked to estimated the average number of rooms and objects per scene across the dataset. They then ranked the system from 1-5 where 1

[1] As of this writing available at the "Explore" tab of http://suncg.cs.princeton.edu

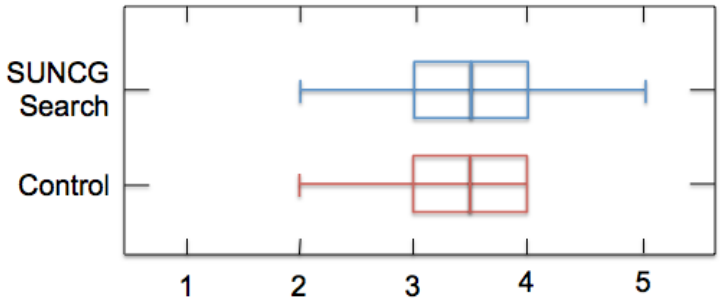

Figure 5.1: User favorability for the dataset exploration task.

was defined as "not at all suitable for this task" and 5 as "ideal tool for this task." Users then repeated the task on the other system.

2. **Subset Identification** On their assigned system, users were randomly assigned some subset (e.g. "movie theater," "office building," "realistic home") and asked to compose a query that returns a preponderance of scenes belonging to that subset. The iterative process of composing queries, searching, and analyzing the results was timed. Once the subset was successfully isolated, the time was stopped and recorded. Users were asked to rank the system from 1-5 where 1 was defined as "not at all suitable for this task" and 5 as "ideal tool for this task." Users then repeated the task on the other system.

5.3.3 Study Results

Dataset Exploration

- When asked to rank the suitability of each retrieval system for the task, users expressed higher satisfaction with SUNCG Search over the test. See Figure 5.1.

- On both systems, users performed poorly when asked to estimate distributions of the dataset. When using the control system, users accurately estimated the number of rooms per scene with only 4% error but then under-guessed both the

average number of objects and the average count of objects per room with 58% and 60.7% error respectively. When using SUNCG Search, users under-guessed both the average number of rooms per scene and objects per room with 45% and 71% error respectively but were able to ascertain the average count of objects per rooms with marginally more success than for the control system but still with 48% error. From this we can make two claims: (1) it is a uniformly difficult challenge when users are alloted only three minutes and not given priming (users were not informed of this task beforehand but instead relied on recall) and (2) SUNCG Search is better suited for the retrieval of extraordinary results (e.g. the smallest scenes) hiding the average rooms in its rank calculation. See Table 5.1 for a full report of user performance.

- Users were more readily able to answer the question "what surprised you about the dataset?" when using SUNCG Search. One user remarked that they found a scene made entirely out of carpets. Another user saw a scene shaped like the Nintendo character Mario. These anecdotal instances imply that odd scenes, meaning scenes with unexpected distributions, rise to the top of the ordered result list in SUNCG Search due to the weighting of the rank formula that rewards scenes increasingly for being of highest value for even a single subquery. For example, given a two-part query, a scene with the average value on both subqueries is assigned identical rank as a query that is most outlying on one query and only minimally fulfills the second.

- Users were more likely to remark that they only saw residential scenes when using the control system. Users of SUNCG Search reported seeing a variety of subtypes including hotel resorts, commercial gyms, office buildings, and residential homes.

	Ave. Rooms	% Error	Ave. Objects	% Error	Objects/Room	% Error
Actual	9		124		14	
Control	9.4	4%	52	58%	5.5	60.7 %
SUNCG Search	4.9	58 %	35	71 %	7.2	48 %

Table 5.1: Quantitative results for the dataset exploration task.

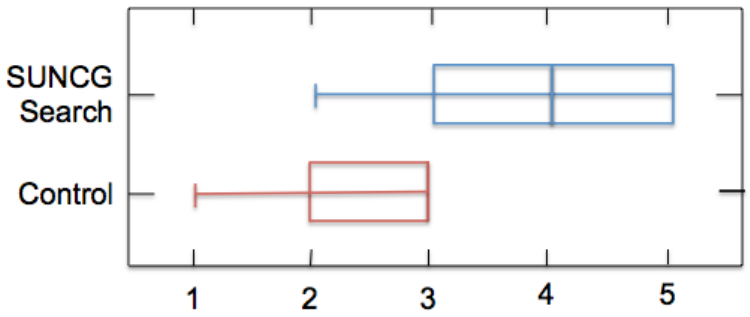

Figure 5.2: User favorability for the subset identification task.

Subset Identification

- Users completed the timed task 7.76 times faster when using SUNCG Search than when using the control. One feature users guessed was responsible for the increased performance was the addition of result ranking in SUNCG Search. Unlike the control, which implements only a naïve filtering system, SUNCG Search features a query-dependent ranking formula that outputs higher ranks for scenes that are extraordinary on one or more subqueries. Since most subsets are a minority of the dataset, scenes of that subset often have some abnormal object or spatial relationship distributions, bubble to the top of the result list, and are more easily discovered by users than "average" scenes.

- When asked to rank the suitability of each retrieval system for the task, users expressed higher satisfaction with SUNCG Search over the test. See Figure 5.2.

- The most effective user trial leveraged a relation-type query to describe functionality of a space. Tasked with identifying the subset "office buildings", the

user searched "person above desk chair". And after fifteen seconds of composing this single query, the resulting scenes were overwhelmingly "office buildings" which contained numerous instances of people sitting in desk chairs. By this example, we see the descriptive potential of spatial relationships when used as a proxy for functional descriptors. The user decided to describe their subset by the functions performed by people in those spaces instead of the containment associations one has with spaces. The success of this user emphasizes the importance of enabling users to search via their natural understanding of scenes as functional spaces and highlights the viability of our proposed methods of adding spatial relationship annotations to scene graphs in order to approach a more human-first method of scene retrieval.

- All users reported that the query-specific visualization feature gave them more confidence in their answer when proposing a successful query.

Note: The full transcript of the user study is included in Appendix A.

Chapter 6

Conclusion

The fields of computer vision and computer graphics are advancing daily as deep learning empowered by big data routinely produce solutions for historically challenging problems. We are glad to contribute to the entrance of 3D scene understanding into this brave new world, an entrance long delayed by the inherent complexity of 3D scene representations and fundamental lack of large datasets. Until now, a content-aware 3D scene retrieval system on large datasets had not been viable and consequentially neither had tasks that require specific scenes for learning. With this preliminary work, we produce efficient and expressive tooling for researchers working with large dataset as a means of understanding the limitations of their dataset and isolating subsets.

Ultimately, the project succeeded on two fronts: (1) constructing a proof-of-concept retrieval system for 3D scenes that outperforms all predecessors in the literature, outpacing the field both in expressibility and extensibility and (2) establishing best practice design decisions applicable for researchers seeking to build a 3D scene retrieval system for their own large, annotated 3D scene dataset.

Additionally, we are proud of many of the smaller contributions in this paper, specifically, our proposition that a relational database schema can be designed such

as to emulate a scene graph such that relationships can be represented as scene graph annotations. This innovation enables researchers to apply traditional scene graph processing algorithms to be applied even on this large scale. The implicit speed of this representation is an essential (though often overlooked) element in the adoption of a scene retrieval system. If users must wait for results, they will lose their sense of curiosity and be less likely to explore the query space, sticking to safe and uninteresting queries that are guaranteed to work.

6.1 Key Insight: Relationships

Beyond the introduction of considerations for a scene retrieval system, we are glad to introduce a robust method of geometrically identifying binary spatial relationships between pairs of objects and to present an application which effectively leverages their boolean nature. In the scene retrieval application, these relationships can be both efficiently stored and also naturally extended into natural language.

We assert that these spatial relationships enable 3D scene dataset evaluation on a large scale. One contemporary challenge in scene understanding is an inability to verify the *realness* of 3D scenes, i.e. the likelihood that such scenes resemble real world spaces. The current state-of-the-art (though workable due to the feasibility of manually filtering small datasets) is naïve as it verifies a closed set of requirements – e.g. rooms must have doors and rooms must have at least n objects. However, this approach is error-prone and only discards the most obvious of outliers. The reader can imagine all sorts of rooms that fulfill these criteria but still do not resemble real world spaces – e.g. a well-formed room with fourteen lamps and no other objects. Of course, another rule could be added to the set to filter such a scene, but the problem is clear: a researcher must be aware of an outlying scene to filter it.

With the introduction of this system of spatial relationships, it is possible to learn

the conditional probabilities that enable functional relationships. For example, given a manually verified subset of realistic scenes, a researcher can generate the probability that tables are *surrounded by* chairs and those chairs are *facing* the table. Then, on an unverified section of the subset, users can use these conditional probabilities to estimate whether a given scene makes functional sense. With this ability, researchers can swiftly and confidentially verify large datasets given some manually verified subset. Additionally, researchers can extend this knowledge to augment their datasets through generating scenes based on these conditional probabilities. With such an augmented dataset, it becomes viable to apply deep learning to 3D scene tasks.

Chapter 7

Future Work

From the outset, SUNCG Scene Search was a proof-of-concept designed with extensibility in mind. In order to achieve the holy grail of being both a general-use dataset exploration tool and a tool for fine-grained subset identification, there are a number of new features that would be much desired. Additionally, with the goal of the dataset exploration, which is a task completed by users not oriented to the dataset, there are a number of improvements to make to make the tool more user-friendly.

7.1 Decrease the Learning Curve

One challenge inherent to expressive systems with simple UIs is the difficulty of communicating the query space to the user. Indeed, over half of the subjects in the user study communicated frustration that they did not feel adequately informed of what types of queries the system could understand. Although the solution presented by our system – that is, context-aware autosuggestions – was found to be generally useful, certain situations flummoxed our word-at-a-time suggestion engine. For example, for more complicated constructions involving conjunction – e.g. "a living room with a sofa and a chair" versus "a room with a sofa and a room with a chair" – the autosuggestion system would require more than a one-token look-ahead knowledge to

distinguish between a compacted query (the first) and a two-part query (the second) when making suggestions.

Additionally, users attempted to use the system as if it were a robust, commercial-grade search engine, feeding it single-word qualitative queries. For example, one user, when prompted to craft a query that would return office building, merely searched "office." A future iteration of this project needs to take into account a users long history with search engines. By emulating the features of commercial search engines, a system can be intuitive even for a very novice user who is unfamiliar with the dataset. Fuzzy matching is key. A system should not expect a user to precisely guess at an object class name. More than one user during the study unsuccessfully attempted to search for "tree" while the system stubbornly waited for "plant."

Some further human computer interaction improvements include prefetching an estimated number of results for populating the autosuggestion box. For example, if a user has typed the partial query "kitchens with a," the autosuggestion box would render with the suggestions with the most results first, e.g. "knife" before "pool". This improvement would additionally aid a user's ability to estimate distributions in the dataset given that they understand the amount of filtering each additional subquery would add.

We acknowledge that a major limitation of the user study as performed is that our users had no previous experience with the system. Therefore, relying on the performance of these users to gauge the benefit of expressibility or relational queries when these users were unfamiliar with the range of the system potentially prevented us from collecting accurate data on the benefits since users did not necessarily not find those features in the course of their short time with the system. Still, we believe that our efforts in making SUNCG Search approachable are valuable as we understand that at least one part of our target audience – researchers deciding whether to adopt the dataset – will probably have limited investment in the system as they are pursuing the

dataset on a trial basis. Providing these users with an extensive tutorial or otherwise requiring them to uncover the limits of the system by trail and error would deter potential adoption on account of frustration.

7.2 Further Annotating the Scene Graph

We have consistently mentioned the concept of "annotating the scene graph." It would be possible to take this even one step further and add annotations to the object nodes. The key idea behind this suggestion is to make the entirety of the dataset searchable. SUNCG contains much more information than is currently indexed. For example, every object instance has properties unique to that instance – e.g. color, texture, and scale. Indexing this information makes no assumption about how a user would want to use the search engine, instead taking a laissez-faire approach and allowing users to perform any sort of query on any information on the scene graph.

Another potential improvement to the scene graph approach lies in a limitation of the current query space wherein searches (for non-relationship queries) are required to originate at a parent element. This requirement was an intrinsic design element given the limitations of this project and helped to formalize the query set that the system could handle. However, with more time (and, perhaps, a generative and robust NLP solution), it would be ideal to support queries that originate at any node. For example, supporting the simple search "office" by returning all the scenes with the roomtype of office. Additionally, expanding the query set to include more statistical parameters such as containment with a minimum (or maximum) number of matches would further expand the set of tasks users could complete using this retrieval system.

7.3 Greater Range of Query-Specific Visualizations

Ultimately, a better visualization of results will improve the user experience of the retrieval system. Currently, all the query-specific visualizations modify the appearance of objects without altering the view of the scene. There is exciting recent work being performed in the field of intelligent viewpoint selection (e.g. Genova et al. (2017)) that would be applicable to a query-dependent viewpoint solution. There is research to support the preference of users for canonical viewpoints of 3D scenes (Ehinger and Oliva (2011)). By intelligently selecting a viewpoint, users would more naturally be able to understand the scene.

Moreover, as suggested in the user study, users would prefer to be given more control over the ordering of results. Currently, the ranking of results in query-dependent and the resulting "search snippets" are displayed in descending order. When performing specific searches, rather than database exploration tasks, users wanted the option to display search results in both ascending order and in random order. On this second suggestion, this ability would improve the performance of users on the task of recounting the average distributions of the dataset.

Appendix A

User Study Documentation

A.1 Hypotheses:

H1: Users exploring the dataset with the improved visualizations would more accurately be able to ascertain facts about the dataset, i.e. distributions.

H2: Total time spent composing a query to accurately partition a subset will be lower with the ability to query by relationships and statistics than without that ability.

A.2 Orientation:

Hello. My name is Elizabeth Bradley, a senior in the Computer Science Department at Princeton University. Today, you will be taking part in a user study for my thesis research entitled: Toward Content-Aware Scene Retrieval using Natural Language. You will be interacting with a proof-of-concept search engine for 3D scenes. This search engine, hereafter S3, emphasizes expressibility and ease of use. You will be completing three tasks to test whether S3 achieves these goals. Before we begin, a disclaimer: this is a proof-of-concept prototype and might fail suddenly. I reserve the right to pause the clock to fix any errors not related to the design principles being tested. This includes but is not limited to restarting server and refreshing the page.

A.3 Tasks

A.3.1 Task 1: Dataset Exploration

- Spend 3 minutes with assigned scene retrieval system
- Answer:
  - What kinds of scenes are represented in the dataset? What are the most common types? What are some outliers you found?
  - What surprised you about the dataset?
  - Can you guess at the distribution of number of rooms? Number of objects?
  - Out of 5, what score would you give this system?
  - What would make this task easier?
- Repeat with the new system

A.3.2 Task 2: Subset Identification

- Find your assigned subset on your assigned scene retrieval system.
- Answer:
  - Out of 5, what score would you give this system?
  - What would make this task easier?
- Repeat with the new system

Bibliography

Planner5d. https://planner5d.com/.

The sims resource. https://thesimsresource.com/downloads/browse/category/sims3-lots/.

Abella, A. and Kender, J. R. (1993). Qualitatively describing objects using spatial prepositions. In *Qualitative Vision, 1993., Proceedings of IEEE Workshop on*, pages 33–38. IEEE.

Chang, A. X., Savva, M., and Manning, C. D. (2014). Learning spatial knowledge for text to 3d scene generation. In *EMNLP*, pages 2028–2038.

Cutrell, E. and Guan, Z. (2007). What are you looking for?: an eye-tracking study of information usage in web search. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 407–416. ACM.

Ehinger, K. A. and Oliva, A. (2011). Canonical views of scenes depend on the shape of the space.

Fisher, M., Ritchie, D., Savva, M., Funkhouser, T., and Hanrahan, P. (2012). Example-based synthesis of 3d object arrangements. *ACM Transactions on Graphics (TOG)*, 31(6):135.

Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., and

Jacobs, D. (2003). A search engine for 3d models. *ACM Transactions on Graphics (TOG)*, 22(1):83–105.

Gapp, K.-P. (1994). Basic meanings of spatial computation and evaluation in 3d space.

Genova, K., Savva, M., Chang, A. X., and Funkhouser, T. (2017). Learning where to look: Data-driven viewpoint set selection for 3d scenes. *arXiv preprint arXiv:1704.02393*.

Hearst, M. A. (1995). Tilebars: visualization of term distribution information in full text information access. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 59–66. ACM Press/Addison-Wesley Publishing Co.

Hu, R., van Kaick, O., Wu, B., Huang, H., Shamir, A., and Zhang, H. (2016). Learning how objects function via co-analysis of interactions. *ACM Transactions on Graphics (TOG)*, 35(4):47.

Hu, R., Zhu, C., van Kaick, O., Liu, L., Shamir, A., and Zhang, H. (2015). Interaction context (icon): Towards a geometric functionality descriptor. *ACM Transactions on Graphics (TOG)*, 34(4):83.

Johnson, J., Krishna, R., Stark, M., Li, L.-J., Shamma, D., Bernstein, M., and Fei-Fei, L. (2015). Image retrieval using scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3668–3678.

Koutsoudis, A., Stavroglou, K., Pavlidis, G., and Chamzas, C. (2012). 3dsse–a 3d scene search engine: Exploring 3d scenes using keywords. *Journal of Cultural Heritage*, 13(2):187–194.

O'Brien, M. and Keane, M. T. (2006). Modeling result–list searching in the world

wide web: The role of relevance topologies and trust bias. In *Proceedings of the 28th Annual Conference of the Cognitive Science Society*, pages 1–881.

Retz-Schmidt, G. (1988). A replai of soccer: Recognizing intentions in the domain of soccer games. In *ECAI*, pages 455–457.

Savva, M., Chang, A. X., and Agrawala, M. (2017). Scenesuggest: Context-driven 3d scene design. *arXiv preprint arXiv:1703.00061*.

Song, S., Yu, F., Zeng, A., Chang, A. X., Savva, M., and Funkhouser, T. (2017). Semantic scene completion from a single depth image. *IEEE Conference on Computer Vision and Pattern Recognition*.

Tombros, A. and Sanderson, M. (1998). Advantages of query biased summaries in information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 2–10. ACM.

Zhao, X., Wang, H., and Komura, T. (2014). Indexing 3d scenes using the interaction bisector surface. *ACM Transactions on Graphics (TOG)*, 33(3):22.