AN INTERACTIVE SYSTEM TO SUPPORT THE EXPLORATION AND MEASUREMENT OF ADAPTIVE 3D DESIGNS

A Thesis

by

CHEN LIANG

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Jeeeun Kim
Committee Members,	James Caverlee
	Jinsil Hwaryoung Seo
Head of Department,	Scott Schaefer

May 2021

Major Subject: Computer Science and Engineering

ABSTRACT

Thingiverse is one of the most popular 3D model repositories. As of October 2020, it hosted over 1.9 million 3D designs that are designed by semi-experts and enthusiasts and are shared across the fabrication community. Many of these designs are designed to interact with real-world objects and are customizable upon personal needs. However, it is challenging for novice users to discover such designs using textual information, comprehend what parameters mean to adjust for customization, where the measurement targets locate on target objects, and how to measure them correctly. These challenges may cause the design to be incorrectly adjusted and cannot function as expected, resulting in a failed print and thus requires the users to start over, which costs additional time and materials. To solve these problems, this work presents an interactive system that uses a graph-based structure to represent adaptive designs to improve the search and exploration, and then guides end-users through the measurement process using a set of modular measurement methods for primitive shapes. The proposed system assists users without expert skills to conveniently discover, adjust, and reuse adaptive designs that are shared online, which allows novice users to effectively and efficiently print adaptive designs. Eventually, the system can be generalized to other repositories, such as industrial parts repositories and medical model repositories, to provide an effective structure and interface for design discovery and adjustment.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Dr. Jeeeun Kim and Dr. James Caverlee of the Department of Computer Science and Engineering and Dr. Jinsil Hwaryoung Seo of the Department of Visualization. This work is also supported by Dr. Anhong Guo of the Department of Electrical Engineering and Computer Science at the University of Michigan.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

This work has not been supported by any funding sources.

NOMENCLATURE

NER	Named Entity Recognition
LiDAR	Light Detection and Ranging
RGB	Red Green Blue
HCI	Human-Computer Interaction
VR	Virtual Reality
AR	Augmented Reality
3D	Three Dimensional
CAD	Computer-Aided Design
IMU	Inertial Measurement Unit

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

1. INTRODUCTION

Personal fabrication has gained increasing interests among novice users in the past years [1]. Besides the fact that personal 3D printers have become accessible and affordable to a wide array of people, the growth of public 3D printing communities also made millions of 3D models accessible, which promotes the reuse of existing designs for 3D printing tasks. Based on the designs that were retrieved in October 2020, Thingiverse [1], which is one of the largest 3D printing communities [2], has hosted over 1.9 million designs for users to download, share, and discuss various 3D printing designs. Novice users without professional skills and expertise for 3D modeling tend to browse existing repositories for possible solutions to fabricate [3]. Existing tools, such as Thingiverse Customizer, could also help them to be able to customize designs to meet varying requirements. Built upon the script-based modeling format OpenSCAD, the Customizer creates a graphical user interface to allow easy adjustment of values, and then generates a new 3D printable file given inputs. This makes it even more convenient for adaptive (or augmented) fabrication, which is the process of printing artifacts that work with certain existing objects [4]. The designs that belong to this category are usually referred as adaptive designs, which usually involve some target objects (the real world objects that are being augmented), and the functionality or adaptation type of the current design. Figure 1.1 shows examples of adaptive designs, which are all interacting with some physical objects (target objects). By combining the customization feature with adaptive designs, the adaptive designs can fit other shapes of target objects as well by fine-tuning some key parameters of the original design. For example, changing the radius of the bottle holder (figure 1.1 left) can make it fit another size of the bottle, even though the design was initially designed for a specific size of the wine bottle. This makes it easier for users to adjust and reuse the existing designs based on their own requirements.

However, even with the help of customization tools and a large number of available models online, novice users still experience multiple barriers, ranging from effectively discover different

[1] https://www.thingiverse.com

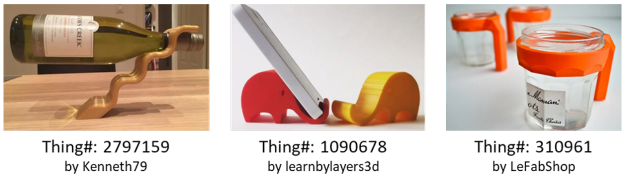


Figure 1.1: Examples of Adaptive Designs on Thingiverse.

types of designs, to correctly identify necessary physical information and be able to transfer the information to the digital space. This becomes an even more crucial issue for adaptive fabrication, as the current text-based search interface for discovering models is not accompanied by well-supported features to discover adaptive designs, such as discovering possible adaptation types and exploring target objects of the designs. Without prior knowledge of what are possible adaptations for a given object, one has to start by using the target object as the keyword to search for relevant designs and manually review all possible designs returned by the search query. As an example shown in figure 1.2, the search of ‘bottle’ returns over 500 pages of results, with 20 results per page. Even by limiting the search result to *customizable designs* which is common for adaptive designs only, there are still more than 180 results, which is not an efficient way for novices without clear intention of what to print to discover possible designs.

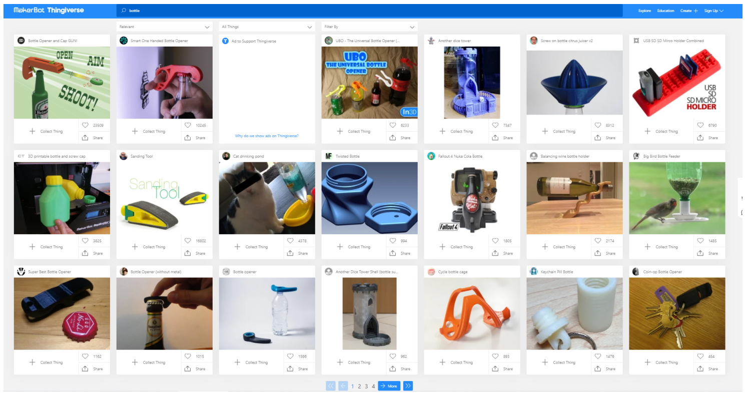


Figure 1.2: Example Thingiverse Search Result Page for the Search Query ‘bottle’.

Furthermore, the current method does not well-utilize the contextual information, such as the desired environment of where the object will be used, and possible objects that are used together with the proposed design. Having additional contextual information would assist the discovery process by showing more relevant designs to the users.

Besides the potential problems about discovering possible designs, measuring the required parameters for customization has also challenged novice users [5]. Since most of the adaptive designs require the measurement of the target object, incorrect measurement or errors in transition from physical values to the digital space may cause undesired prints, resulting in various kinds of 3D print failures such as not fitting to the target, falling off, incapability to fix, and more.

This work aims to provide an interactive system for novice users to discover, measure, adjust, and print adaptive designs. The project specifically focuses on creating an end-to-end pipeline that is intuitive to novices and is capable of reducing the required domain knowledge of the design process. The goal is to assist novice users to effectively go through the process of printing adaptive designs, which includes helping users effectively discover adaptive designs available in the design repository, and providing computer-aided measurement system to guide users to locate, measure, and transfer physical dimensions to the digital design files directly. Being provided as an interactive interface that can promote effective design discovery and adjustment, the underlying structure and cyber-physical measuring system can be applied into other open-source repositories as well.

This thesis covers three sections:

1. Proposal of the methods for retrieving adaptation-related and context-related information from existing designs as additional design metadata, and the corresponding data structure for the interactive system to facilitate the discovery of adaptive designs and the reuse of existing designs.

2. Introduction of the object measurement modules and computer vision based cyber-physical measurement methods.

3. An end-to-end pipeline to demonstrate the utilization of the proposed methods in an end-user oriented interactive system for design exploration and measurement.

2. RELATED WORK

2.1 Adaptive Designs, Customization, and Design Remix

Adaptive designs, which are 3D designs that are created to augment or extend the functionality of existing real-world objects[1], such as the examples shown in figure 1.1, have become one of the popular types of design on Thingiverse. While the original 3D model might be designed for a specific target object, the introduction of customization features on Thingiverse in 2013 [6] allows other users to fine-tune some preset parameters of a design in order to adjust the original model to fit different sizes of target objects. As an example, by changing the radius of the bottle holder, the holder shown in figure 1.1 left can then be used for other types of bottle with various shapes, which makes the original adaptive design much flexible to expand its usage to other target objects. This feature has been known to greatly attract users' interests, as the contributing authors on Thingiverse increased by almost 600% one year after the introduction of the customizer feature on Thingiverse [6].

The dramatic growth of 3D printing repositories that are accessible to individuals also attracts the interests of the HCI domain, as *Personal Fabrication* has also become a popular subfield. Researchers have investigated personal fabrication from various directions including software, hardware, data sciences, and how the users interact with the existing fabrication systems [7, 8, 2, 9]. Augmented Fabrication specifically focuses on the work which takes place when the design references the existing objects in real life [10]. The study of such augmented (or adaptive) designs and the process is particularly interesting not only because of its interactive nature [11, 12], but also the direct interaction with the **target physical object** (*ad rem*) and the **object's original context** (*in situ*).

With the growth of the maker movement [13], designers and users on Thingiverse also demonstrated this culture by making, remixing, and sharing designs on Thingiverse [6]. The introduction

[1] This paper will refer to real-world objects that interact with 3D printed adaptive designs as 'target objects' throughout this work.

of the Thingiverse Customizer makes this process even more convenient, where the customizable designs, together with the designs generated from these customizable designs, contributed over 45% of all designs on Thingiverse as of 2014 [6]. However, the rapid growth of remixed design from customization did not well contribute to the community and other remix activities, as most of these designs ended up being the 'dead end', with limited user activities based on those designs, which shows a great need to not merely contribute additional customized designs as individual document, but also to understand both these parametric models and the target reference models of these designs [6]. All these demonstrate the great demand of a system that is based on sufficient understanding of designs metadata in order to facilitate the effective organization and reuse of designs in 3D printing repositories.

2.2 Interactive Systems to Support Novice Users' Digital Fabrication

Many previous researches have proposed various techniques to assist novice users in the fabrication process. Some works aim to understand user intention from given input, and generate adaptive designs accordingly. For example, Reprise is a system that generating directly 3D printable adaptations for the target object based on user's high level intentions (such as desired type of action and adaptation) and can adjust some parameters of the generated designs with the sliders [7]. Makers' Marks demonstrates potentials of applying computer vision techniques to understand users' intentions and locate and approximate users' design requirements. It utilizes annotations with image markers and a sculpted object to help user demonstrate his/her intentions without requiring professional skills or knowledge. The sculpted object is then scanned, processed, and generated as a 3D model with functional artifacts that are ready to print [14]. While these approaches demonstrate possible applications that can take users' intention and convert them to the corresponding models, they are still comparatively limited, and they do not broadly utilize existing designs in public repositories.

Some other approaches utilize designs in public repositories to omit the modeling process. For example, Mix&Match [3] enables users to search from Thingiverse repositories via AR headset for them to connect virtual models with the physical world. Users are able to adjust models based on the physical context of placement and relative scale. However, the discovery process still remains challenging, as users still have to use virtual keyboard and browser to search desired model using text entry to query, which does not fundamentally solve the problem of effective discovery of customizable adaptive designs. Also, customizing the model relies on users to manually adjust model by performing basic operations such as scaling and boolean operations, which may not be ideal to fine tune the model accurately.

Other systems propose the idea of reusing the functionality of designs. For example, PARTs [15] provides abstraction of functional geometry, which promotes the abstraction and reuse of geometries and constrains for the supported adaptation type. This allows designers to *create* their designs from these preset adaptation types, with constrains checking whether the adaptation will correctly fit the target object or not. While this work provides a method for abstracting functionality to facilitate reuse, the system is primarily designed for modelers with some CAD software skills. The supported functionality also depends on the prepared functionality library, which is comparatively limited considering a variety of other possible types of adaptations available in the repositories.

This thesis instead focuses on abstracting the design information to promote the reuse of existing designs, which concentrates on discovering and reusing existing designs based on users' intention. Also, as the target users are novice users, the system will not approach the problem through CAD software which necessitates some amount of professional knowledge for discovery and adjustment process. In addition, this work provides an interface for users to reuse existing designs, including reuse designs for exactly the same purpose (e.g., use bottle holder for a bottle) and for similar purposes (e.g., use bottle holder for a can, due to similar shapes). This facilitates reuse of existing open-source repositories to make the proposed system versatile in its supported designs and adaptation types.

2.3 Challenges and Techniques for Design Search and Retrieval

Previous works have shown that novice users may face multiple challenges while discovering or using public 3D designs on Thingiverse [2, 4]. Since novice users tend to use online communities to find designs instead of learning 3D modeling [16], it is important that available design information and metadata should be organized in a clear and easy-to-understand manner to avoid potential confusions. The study on Thingiverse data shows that users often have questions related to designs' printability, functionality, and assembly [17], which may not always be explicitly included in the design descriptions. Thus, it has been advised that richer metadata, clarifications, and expert tips would be helpful to help users with design printing and customization [2]. Based on this finding, one of the motivations of this work is to retrieve and organize relevant metadata that are either explicitly included in the description or can be implicitly retrieved from existing data such as images. This promotes a more effective search and discovery process that is based on adaptation-related data fields and can ease the process of discovering possible adaptations for the given target objects.

The problem of search and information retrieval in large collections has been one of the research topics in information systems [18]. The current Thingiverse search interface is primarily based on textual query, with some search filters covering major design attributes, such as categories and community-related features (e.g., 'Things I've Liked'). With no or only few filters enabled, the current interface may return a long list of search results, which has been shown to be difficult to navigate, and users are likely to browse only the first page of the result [19]. This becomes a challenge for users to explore adaptive designs as well, as adaptation-related information, such as functionality (adaptation type) and target objects, are important to effectively locate the desired design, which unfortunately has not been incorporated into the current Thingiverse search interface or the metadata to assist search.

The previous work on faceted search proposed a characteristics (facets) based search method [20]. Facets are defined as a set of meaningful labels organized in a way to reflect the concepts relevant to a domain, and have been shown to be more helpful than merely providing a ranked list of result and can improve the search accuracy [21, 22]. This work utilizes the similar idea by applying named entity recognition to retrieve the target objects and adaptation type of the design as the facets, and use these facets to help novice users quickly understand what are possible types of prints before going through each individual design. In addition, this work also applies computer vision methods to retrieve descriptive information about the usage context (e.g., where the design is usually used) from the user uploaded images, to further sort the results to propose more relevant results to the users.

2.4 Object Measurement: Challenges and Possibilities

Another challenge that novice users may face, specifically in the augmented fabrication, is the obstacles to capture the information about the physical object (e.g., measuring target object), and be able to transfer that information to the digital realm [4, 5]. Users may encounter the questions of *what* to measure (which part needs to be measured for a given parameter) and *how* to measure (which instrument to use) [5]. This motivates us to incorporate computer vision based measurement method to solve the '*how*' question, and pre-programmed 'expert knowledge' to decide which part to measure to solve the '*what*' question of the measurement task. The measured value is then directly applied by the program to the source file to reduce unnecessary work for users to transfer information of physical object to the digital files.

To solve the problem of measurement, previous works have explored the possibilities of estimating distance from 2D image [23], or 2D with additional distance information along the z axis [24]. To further increase the accuracy, additional devices have also been utilized to perform the measurement. For example, LiDAR cameras have been used as the solution for measurement in logistics [25]. As newly manufactured mobile devices such as iPad Pro [2] have embedded time-of-flight sensors such as LiDAR, this expands the possibilities of performing more accurate measurement on mobile devices. This work combines computer vision methods with LiDAR readings to help the novice users locate and measure the required parameters for the customization to propose a potential solution to the challenges novice users may face in the course of adjusting and printing adaptive designs.

This work proposes the interactive system that is based on a processed and augmented dataset of Thingiverse designs, and uses computer vision based methods to perform the measurement.

[2] https://www.apple.com/ipad-pro/specs/

This is approached by investigating the *ad rem* and *in situ* problems by understanding the design itself, the target object, and the context where the object is used, and be able to perform abstraction of this information to promote reuse of designs for similar purposes. The proposed pipeline aims to connect all components of the process, from design discovery till the model adjustment, as a complete pipeline for the novices, in order to reduce unnecessary steps, skills, knowledge, and understandings that are required in the augmented fabrication.

3. METHODS

3.1 Design Discovery

Designs on Thingiverse contain a wide range of information, including design-related information (e.g., description, categories, files) and community activities (e.g., comments, likes, shares). These create the current structure of design metadata, and provide users with different methods to discover desired 3D designs. However, it is still challenging for users to discover adaptive designs, as the current metadata does not contain adaptation-related information, such as adaptation type and target object, that are important for adaptive design discovery. To improve the discovery of adaptive designs in Thingiverse repository, additional metadata about adaptation and context is retrieved on top of the existing metadata, including adaptation type, target objects, and descriptive tags of user uploaded image to understand the context of where the design is used. To better facilitate the adaptive design discovery, this thesis suggests an object-centric structure of storing additional design information, where the target objects and adaptation types are treated as individual nodes, and the adaptation relationships can then be represented as edges to connect objects and adaptation types together.

To start, customizable designs on Thingiverse are firstly retrieved using the Thingiverse RESTful API. Then the existing design metadata is augmented by incorporating recognized target objects, adaptation types, and customizable parameter information to the design metadata. In addition, Microsoft Computer Vision API is used to get descriptive tags of user-uploaded images. These images usually contain the demonstration of where and how the adaptive design is used. The idea is to utilize these descriptive tags to understand where the design is commonly used and what are usually used together with the given design, and suggest more relevant results for a given search query. At the end, the above information is organized in an object-centric way as mentioned above to convert the adaptation relationships to a graph that can be easily used by the proposed interactive system.

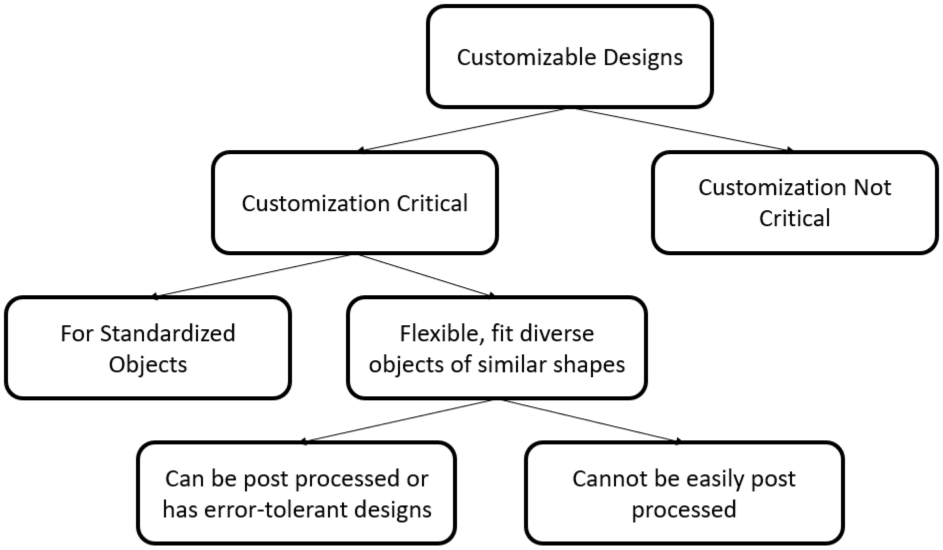

Customizable Designs
Customization Critical
Customization Not Critical
For Standardized Objects
Flexible, fit diverse objects of similar shapes
Can be post processed or has error-tolerant designs
Cannot be easily post processed

Figure 3.1: Categorization of Customizable Designs.

3.1.1 Retrieve Customizable Designs

OpenSCAD [1] is a script-based 3D modeling method. The OpenSCAD file of a customizable design contains necessary variables of the design, such as `bottleDiameter` for a bottle holder design (Thing#: 3057332), along with the code to perform modeling actions to generate a resulting 3D model. An OpenSCAD file is customizable only if it contains customizable parameters in the code. Such parameters need to be listed at the beginning of the code file and before any function calls or operations (e.g., calculations). Creators of the OpenSCAD file can specify the customizable parameters of the model, allowing other users to modify the value of these parameters and regenerate a customized version of the original design. Currently on Thingiverse, having at least one customizable OpenSCAD file (i.e., an OpenSCAD file with customizable parameters) is necessary for a design to be considered as a customizable design. Thus, to retrieve possible customizable designs on Thingiverse, I started by retrieving designs with OpenSCAD files on Thingiverse, and kept OpenSCAD files only if the file contains customizable parameters. For the scope of this work, only the designs with one OpenSCAD file are kept, as the designs having multiple OpenSCAD file

[1] `https://www.openscad.org/`

may require assembly after printing, which introduces additional post-processing required from the users.

Figure 3.1 represents various categories of customizable designs that are likely to be reused by other users, which can eventually guide users through different user experiences that they will need to take. For example, if the user wants to print a battery case (Thing#: 57281), then (s)he still needs to customize the design for different types of batteries, but there is no measurement required, as batteries are standard objects with unified dimensions for the same type of the battery. While in other cases, the target objects are not standard objects, and precise measurement is needed.

The designs are firstly identified as whether the customization is critical or not to the success of the print. For example, a customizable keychain where the only customizable parameter is the scale of the keychain belongs to the category of ‘customization not critical’ because the customization is not directly related to the functionality of the design. On the other hand, the customization for a pipe clip is critical as incorrect customizable values (e.g., incorrectly measured radius of the pipe) may result in the undesired 3D printed result (e.g., a clip slips off from the pipe when 3D printed).

Among designs that customization is critical, the designs are divided into two categories if its target objects are standardized objects (e.g., batteries) or non-standardized objects (e.g., cup). Standardized objects, such as batteries and iPhone X, may not require special measurement tricks from an individual end-user as these values can be found by searching and are more accurate than user-measured values. While for non-standardized objects such as a cup, the dimension can be very diverse, and thus a measurement of the target object is needed to get the necessary dimensions. Finally, the designs are classified if it contains error-tolerant designs, such as screw-tight pipe clip, or if post processing can be applied to effectively mitigate a large amount of error for an incorrectly adjusted design. In these cases, the design may still be modified after printing, even with the incorrect customization of the original design. This lowers the requirement of an accurate measurement during the customization process.

This thesis aims to provide solutions to the most complex case in the above categorization (figure 3.1, where customization is critical, target objects are non-standardized objects, and final print cannot be easily post processed). With the solution to this complex case, other cases mentioned before, such as designs for non-standardized objects, can also use the proposed pipeline for design discovery and adjustment. This reveals the challenges of understanding the target object, including object shape, parameters, and how to measure them, and achieving an acceptable accuracy for the measurement. A integrated pipeline is proposed to help the user go through the printing process of this case. Notice that the proposed pipeline is not limited to this category of designs. Other categories of customizable designs can also benefit from the proposed pipeline, and the pipeline can also be applied to the discovery and printing process of those designs.

3.1.2 Retrieve Adaptation-related Information

The current design metadata does not contain sufficient information about adaptation. Information such as target objects and adaptation type are not directly available from the given metadata, but are important for design discovery. For example, if the user does not have prior knowledge about what are possible adaptations to the target object, then it will be difficult to discover designs using current interface on Thingiverse, as there is no adaptation-related information to help users narrow down the search range. To expand the existing design metadata, I retrieved the target objects of the design, adaptation type, and customizable parameters for the target objects, as the additional adaptation-related information. This information will be used to organize the design information and help adaptive design discovery.

3.1.2.1 Retrieve Target Objects and Adaptation Types

When a designer posts a new design to be shared online, Thingiverse currently does not require a field to indicate the target objects and adaptation type of the given design. Designers often incorporate this information in the title. For example, for ‘Customizable Cable Cap’, ‘cable’ indicates the target object and ‘cap’ refers to the adaptation type. For ‘Clamp for telescopic bar’, ‘clamp’ indicates adaptation type and ‘bar’ indicates target object. Although many adaptive design titles usually have similar sentence structure, creating a rule-based parsing method to understand target objects and adaptation types is still not ideal, because the program also needs to correctly identify the part of speech tags of the word (e.g., ‘stand’ is a common adaptation type, but is usually used as verb), and also needs to avoid certain words (e.g., customizer, generator, maker) that appears to be adaptation type but are not in the context of Thingiverse community. To address this challenge, named entity recognition with the Spacy package [26] is utilized to detect target objects and adaptation type from the title. A NER model was trained on 1,000 labeled titles. On a test set of 100 samples, the performance of the model is summarized in table 3.1. The model performs well in detecting adaptation type, but not the target objects. This may be due to the diversity of brand/type/model of the target object, where the model may not detect the object name precisely, such as ‘C961 Handlebar Adapter’ (Thing#: 3182871) and ‘a8 y carriage for v groove bearings’ (Thing#: 3762588). Instead of trying to get the full name of the target object (which involves brand name, model number, etc.), the program focuses on the most general target object name, which is usually the last noun of the detected target object name, and use it as the name for a given target object. For example, given the target object of ‘Makerfarm Prusa i3 Printer’, it will recognize ‘printer’ as the target object to avoid keeping too many details of the target object. This allows the program to focus on more significant word in the target object, and can increase generality for the discovery process.

	Overall	Target Object	adaptation Type
Precision	0.725	0.562	0.929
Recall	0.707	0.595	0.825

Table 3.1: Evaluation of the Named Entity Recognition Model on 100 Test Samples.

3.1.2.2 Retrieve Customizable Parameters

In addition to the textual information of designs, I parsed the OpenSCAD code files to retrieve the customizable parameters, their variable names, variable comments, and parameter types. To limit the scope of this work, the search is limited to one of the following parameter types: radius, length, width, height, and diameter, since these are common parameters to present three primitive shapes: cylinder, cube, and sphere. Other parameters that require further interpretation, such as ‘offset’ and ‘spacing’, are not included in the parameter retrieval process. After these parameters are select, the program can then search for target object names, which were retrieved with NER before, in the variable names and its comments. If the target object name of the design has been found in the variable name or its comment, then the corresponding parameter will be mapped to that target object for this design. This allows the program to retrieve the corresponding customizable parameters for each of the detected target objects and keep parameter type, line position in code, and value range for future use. These will also be used for understanding target object primitive shape in Section 3.1.4.

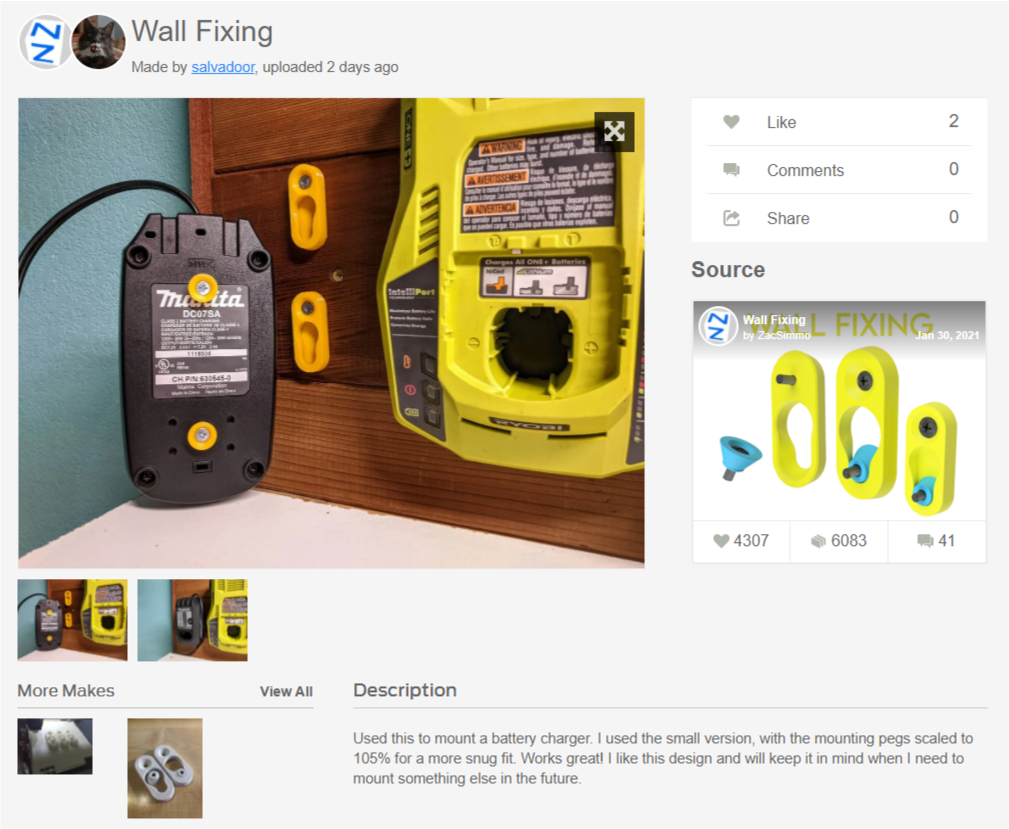
3.1.3 Retrieve Context-related Information

The context of where the design is used is helpful for users to discover the most relevant design. For example, given the user is in the kitchen, a bottle holder could be a more relevant suggestion than a bottle holder for bicycle. Often this is not obvious as it is not always mentioned explicitly by the designers in the design metadata such as title, tags, or summary description. Thus, to retrieve additional information about the context of the usage, I collected user uploaded images of the design and the ‘makes’ of the design. The makes are the posts made by users who printed the given design to show their own prints to list some comments and suggestions. These user uploaded images may serve as examples of where and how the design is used in real-world context, and can be used by the program to retrieve such contextual information to suggest more relevant designs.

I used Microsoft Azure Computer Vision [2] service to extract the tags that describe these images. Tags include the description of the environment, such as ‘indoor’, ‘kitchen’, as well as names of other objects that appear together in the given design. This gives additional information that describes where the design is used and what are usually used together with the design. Given the current user environment as image input (i.e., current user camera input), the program firstly extracts the descriptive tags of the current image. Whenever the user specifies a search or discovery

[2] https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/

query, the program can then sort the possible designs by comparing the similarity of the contextual information of each design and the current user environment, and this is done by calculating the similarity between the sets of descriptive tags of the current image input and each design using Jaccard similarity. Since many designs contain multiple user uploaded images, the frequency of tags among all images are counted and normalized between 0 and 1, and the similarity is then calculated as $\frac{\sum_i min(x_i,y_i)}{\sum_i max(x_i,y_i)}$, where x is a set of tags and their frequencies of the current user's camera input, and y is the similar set of a given design.



Wall Fixing

Made by salvadoor, uploaded 2 days ago

Used this to mount a battery charger. I used the small version, with the mounting pegs scaled to 105% for a more snug fit. Works great! I like this design and will keep it in mind when I need to mount something else in the future.

Figure 3.2: Example Page of a Make of the Design 'Wall Fixing'.

3.1.4 Object-centric Way of Organizing Information

Adaptive design discovery is usually object-centric. For example, users may want to discover possible adaptations for a specific target object, or similarly discover possible target objects for an adaptation type, which can also be considered as an adaptation object, such as holder, hanger, and extender. This object-centric nature indicates an alternative and more intuitive way of organizing adaptive design metadata, which is to store them into a graph. For example, as shown in figure 3.4 given four different designs: box holder, bottle holder, can holder, and can clip, a graph representation can be generated for these designs. The discovery of possible adaptation types for an object can be achieved by getting adaptation nodes connected to the given object node, and the search of possible adaptation between multiple objects can be converted as a graph search problem. Thus, to further improve the discovery and search process for adaptive designs, I used an object-oriented approach to store information as target objects and adaptation types instead of individual design document. This enables users to apply object-oriented design discovery and search for adaptive designs. The information for each object is stored as one node, which can be used to create connections between target objects, adaptations, and the designs, where the structures of nodes are shown in figure 3.3.

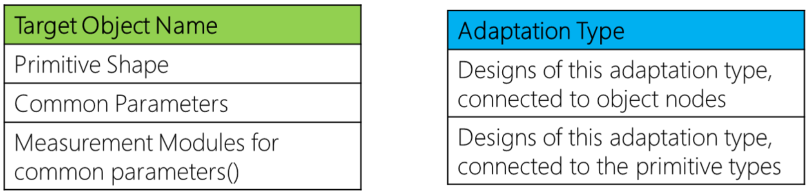

Target Object Name
Primitive Shape
Common Parameters
Measurement Modules for common parameters()
Adaptation Type
Designs of this adaptation type, connected to object nodes
Designs of this adaptation type, connected to the primitive types

Figure 3.3: Structure of Target Object Node (Left) and Adaptation Type Node (Right).

In order to achieve this, it requires to firstly retrieve the relevant information specifically for each target objects and adaptation type, and then be able to abstract them into a more general term, such as the general primitive shape of a target object, or the commonly supported type of primitive shapes for an adaptation type. For a target object node, I firstly retrieved common customizable parameter types among designs that involve the given target object. I counted the frequency of each parameter types and removed the parameters that only occasionally appear in a few designs by removing parameters with a frequency lower than 10% of the maximum parameter type frequency of each design. Based on common parameter types, I determined the primitive shape of the target object as cylinder, sphere, or cuboid. If most of detected parameter types of a given target object are radius or diameter, then the object is likely to be a sphere. Similarly, if common parameter types include both radius/diameter *and* height, then it is likely to be a cylinder. If most of parameters are length/height/width, then it is likely to be a cuboid. For cases where a dominant parameter cannot be easily detected, the program will not add primitive shape to the object to avoid potential confusions. The primitive shape of the target object can help the program to determine the appropriate measurement technique to measure the values.

Adaptations that commonly work on a type of primitive shape may be reused to other objects of the same primitive shape. An example could be reusing a bottle holder as the can holder, even though there is no existing model specifically designed as the can holder. While this may not always work as desired, it still provides possible solutions for the cases where no exact match can be found. The adaptation node stores designs that have the same adaptation types. In addition, it also stores the ‘abstract’ adaptation design, which abstracts the common target objects as primitive shapes. For example, given bottle/can/container holder and box holder, it creates new abstract designs named ‘cylinder holder’ and ‘cuboid holder’ using the existing design. Thus, the design can be reused to find and match target objects with the similar primitive shapes.

Based on the information prepared as mentioned above, the repository of adaptive designs can be represented using a graph structure. This structure provides the foundation to adaptive design discovery and search, which also assists interactive search methods. As mentioned in the system structure, given the detected objects from the camera view, user can now directly select a single object, or select multiple objects and perform a pinch gesture, to initiate a search, and the system can directly treat the selected objects as starting/ending node, and perform a graph search to find possible adaptations.

3.2 Measurement

To help users with the measurement process, I propose a modular approach to tackle a holistic measurement process. First, I begin with a commercial smartphone device using sensors and

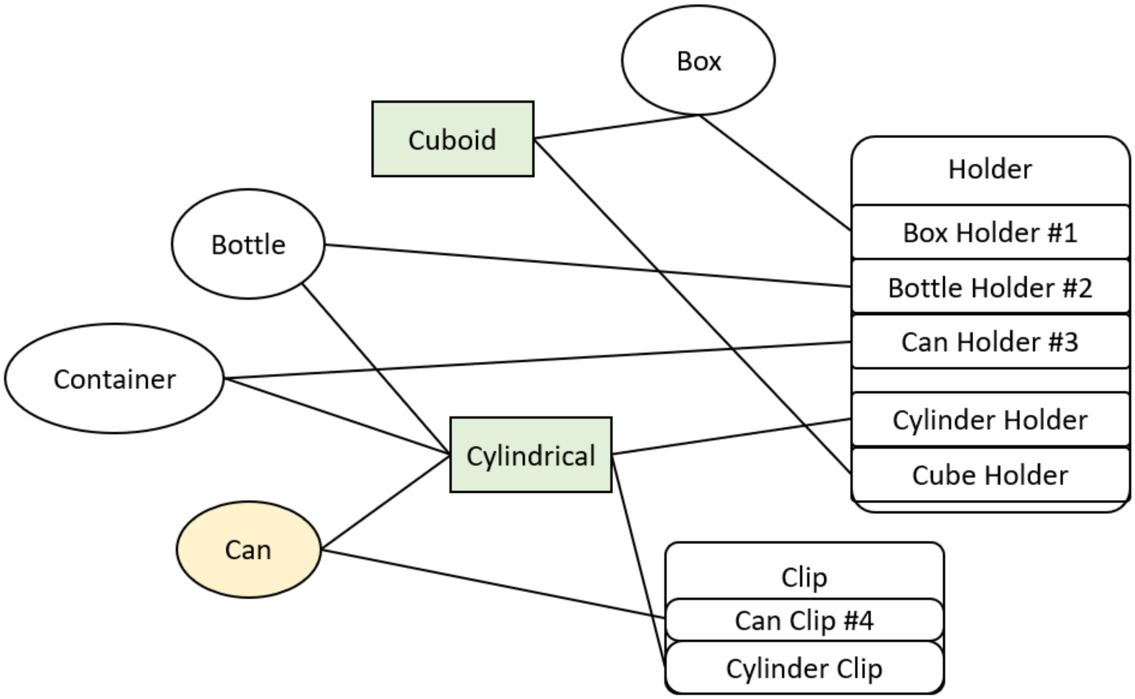


Figure 3.4: Example Graph Representation of Designs.

camera to solve the problem of 'how to measure', and use expert-coded measurement module to solve the problem of 'what to measure'. I implemented the measurement module with parameters of three primitive shapes, (1) the radius and height of the cylinder, length, width, and (2) the edges (width, height, length) of the cuboid, and (3) the radius of the sphere. With these basic building blocks, designers can build on top of these modules, such as building a more complicated measurement module for bottles based on the measurement modules for cylinders, and incorporate designers' expert knowledge on 'what to measure' to create a new module for another target object. This allows reuse of existing measurement knowledge (i.e., 'how to measure') to avoid understanding and performing the entire measurement task from scratch, and also shares the measurement knowledge among objects of similar shapes and measuring requirements.

The system needs to capture the RGB image and the corresponding depth image of the target object. As the foundation to more complicated shapes, I implemented the measurement modules for cylindrical, spherical, and cuboid primitive shapes. The program firstly detects the outline of the target object by detecting edges in the given image, then detects key points that are important for measurement for each parameter, and finally measure the distance by calculating the local 3D coordinates (relative to the camera) of the given key points and measuring the distance between them. These measurement modules can later be used by designers as the building blocks to measure more complicated shapes, by either inheriting and extending existing measurement modules, or by combining multiple modules together in sequence to form a new measurement module for more organic shapes. The details of the general pipeline for measuring the three primitive shapes are listed below:

3.2.1 Outline Detection

The outline of the shape is essential source of information to determine the key points and edges. Given the bounding box of a target object that needs to be measured, the program starts by applying canny edge detection [27] to detect edges inside the given bounding box, then keeps the outermost edge of the object as the outline. Inner edges of the object will be discarded to clear noisy edges and reduce potential impacts to the outline detection. Figure 3.5 shows the process and results of outline detection on each example object of three primitive shapes. The first column of the figure shows the results after canny edge detection, second column shows the result after the post processing, and the third column shows the outline applied on top of the original RGB image.

For cylindrical objects, an additional step is needed for post processing to remove the unnecessary part of the edge. In order to easily measure the radius, outline on the top and bottom are removed by calculating the changes of distance between the leftmost and rightmost pixels on the outline along the y axis. The motivation behind this is to only keep the side edges of the cylinder without the top and bottom edges, as shown in the center image in the first row of figure 3.5, so that the program can directly calculate the diameter by measuring the distance between these edges in 3D space, or getting the height of cylinder by measuring the length of either the left or the right edge. The program only keeps the outline when the change rate is below a threshold, which could be adjusted based on image resolution. In the current implementation, the RGB image from the device is resized to 640x480, and the threshold is hard coded as 2 pixels in the implementation.

This results in two outlines on the left and right side of the cylinder. As shown in figure 3.5, the edges for the tin can does not include the top and bottom part. When performing the measurement, program can directly calculate the distance between the left and right edges to get average radius.

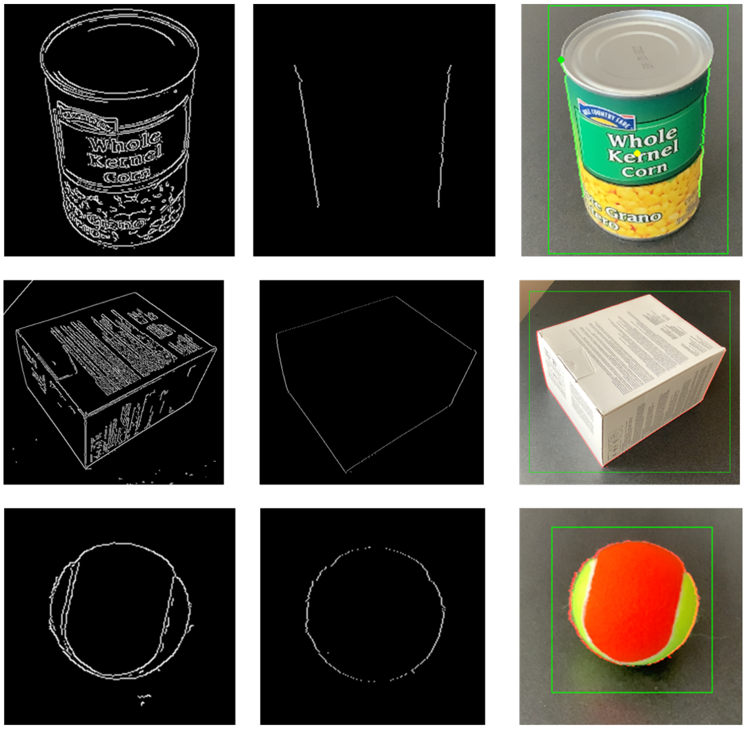


Figure 3.5: Outline Detection for the Three Primitive Shapes.

3.2.2 Key Point Detection

In order to perform the measurement, a set of key points needs to be selected from the given outline processed by outline detection above. The detection and selection of the key points depend on the primitive shape and parameters that needs to be measured accordingly. The demonstration of concepts is showed in figure 3.6.

1. Cylinder: Diameter and height are two parameters that can be detected from cylindrical shapes. Because the outline of cylindrical objects has been post-processed to contain the left and right edges only, the height is calculated from the distance between top to the bottom in

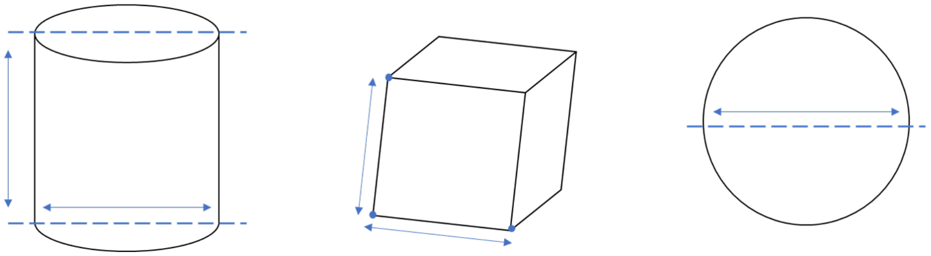


Figure 3.6: Demonstration of the Measurement Methods for the Primitive Shapes.

one of these edges. In this case, the key points will be the top and the bottom pixel of each outline. For diameter, the key point will be a pair of corresponding points (e.g., first point on both edges) on the left and right outline, and the diameter will be determined by the average distance between the outlines.

2. Cuboid: Key points of cuboid shapes are the corner points on the outline which can be detected by going along the outline, calculating the change of angle (change rate) between two adjacent points, and find the point with a significantly different change of angle compared to the previous pixel. In the implementation, as there may be missing pixels on the outline, the program checks the pixel in a small range (r=3 pixels) to find the next pixel in case the edge is disconnected. The resulting key points will be the vertices of the cuboid object in the current camera view. The key points will then be saved as pairs in order that they are discovered along the edge (e.g., (p1, p2), (p2, p3), (p3, p1)), and the distance between the points in each pair represents the length of an edge on the cuboid shape.

3. Sphere: Diameter is the only parameter that needs to be determined, and it can be easily found at the middle of the outline, as shown in figure 3.6. The key point will be the left and right most pixel on this line.

3.2.3 Distance Measurement Between Key Points

Given the key points of the parameters that need to be measured, the program locates the position of the points on the depth map, and calculates their local 3D coordinates using the camera intrinsic and the depth (distance) value of the corresponding pixel.

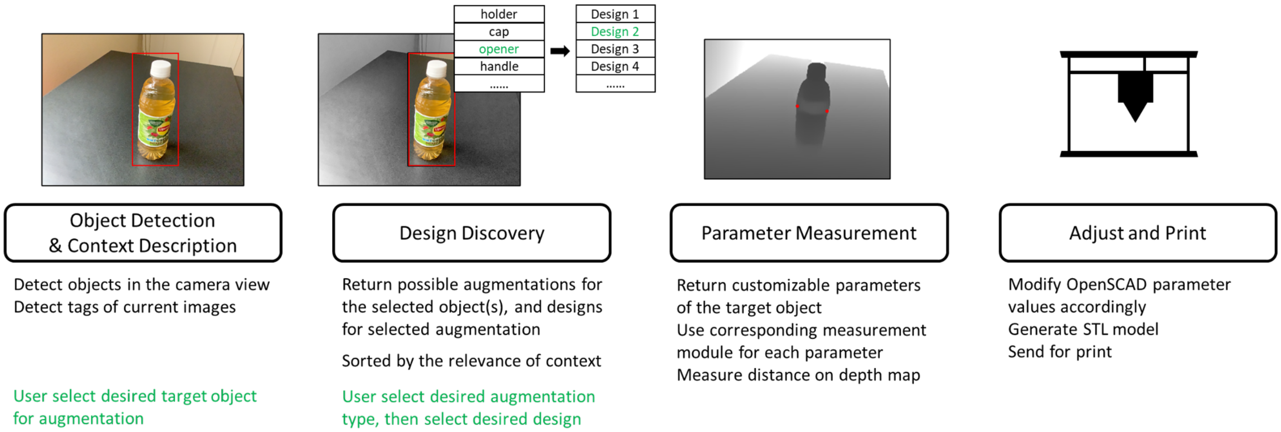
The current measurement module utilizes the depth map captured on the mobile device to calculate the distance between points. As mobile devices start to be equipped with LiDAR and stereo cameras, such as the iPad Pro released in 2020[3], the mobile devices are becoming increasingly capable to perform measurement tasks. With high accuracy on-board sensors, the program can correctly compute the required parameters without requiring additional tools, reference objects, or markers. This can simplify the measurement process for users and avoid challenges novice users may face for the measurement. When necessary, designers can incorporate additional measurement techniques in the existing pipeline to measure different types of target objects. As an example, the process to measure a bottle radius, which is extended from the measurement of cylinders, can reuse the existing measurement module for cylinder radius. The designer can specify additional locations to measure, including top, bottom, minimum and max radius, which could be used in different designs, such as bottle cover where the radius of the bottle cap (top) is important to be measured. This allows additional measuring techniques to be incorporated in the measuring system, and can facilitate the reuse of expert knowledge about measuring target objects of similar shapes.

3.3 Integrated System Pipeline

Figure 3.7 demonstrates the overall structure of the proposed system, which integrates the components implemented above. The interface is designed for mobile devices with stereo cameras or time-of-flight sensors. The step-by-step process is described as the following:

1. **Step 1. Obtaining Objects and Context Information:** Once the user launches the application, the interface starts with a camera view. The system firstly detects objects shown in the scene and their bounding boxes in the camera view using the Microsoft Computer Vision API, and gets a list of tags describing the current context, including location information (e.g., indoor/outdoor, kitchen) and other objects in the scene.

[3] `https://www.apple.com/ipad-pro/specs/`



Detect objects in the camera view
Detect tags of current images

User select desired target object for augmentation

Return possible augmentations for the selected object(s), and designs for selected augmentation

Sorted by the relevance of context

User select desired augmentation type, then select desired design

Return customizable parameters of the target object
Use corresponding measurement module for each parameter
Measure distance on depth map

Modify OpenSCAD parameter values accordingly
Generate STL model
Send for print

Figure 3.7: Proposed System Structure. Green Text Shows the User Input Given to the System.

2. **Step 2. Selecting Target Objects:** The user then selects his/her desired target object(s) to measure for adaptation design by either single tapping an object from the view, or selecting two objects and performing a pinch gesture to indicate two objects are involved in the desired adaptation. The program will then perform a search in the graph structure proposed in section 3.1.4. For a single object, the program returns the adaptation nodes connected to the given target object node; for two target objects, the program performs a graph search to find paths from one object node to the other that pass through one adaptation node, and returns these adaptation nodes.

3. **Step 3. Discovering Possible Adaptations:** The program displays possible types of adaptation (e.g., holder, cover) that are retrieved from available Thingiverse designs next to the chosen target objects. Once user selects one option, the program will further list designs of the selected adaptation types. The proposed designs are sorted by the similarity of context tags of designs and context tags of the current image view from the user's camera. This is done by calculating the Jaccard similarity between the set of tags for each design under the selected adaptation type and the set of tags of the current image view. User can explore possible designs, and select the desired one from the list.

4. **Step 4. Measuring Required Parameters:** Once design is selected, the program retrieves

the parameters that are needed to be measured from the database of parameters parsed from its original OpenSCAD code. The program calls corresponding measurement modules to measure the values for each required parameter. When necessary, the program will utilize device's IMU data to rotate the image to make sure the target object is vertically placed in the camera view, because the current vision-based measurement module assumes that the target object is placed vertically and the image is not rotated. The measured values will then be placed to the corresponding assignment code of variables in the OpenSCAD source code. The new 3D model with adjusted parameters is then generated to be ready for 3D printing.

4. A PROOF-OF-CONCEPT DEMO APPLICATION

In this section, I will introduce a demo application that utilizes the above pipeline. The application was developed using iPad Pro 11' that is equipped with a LiDAR sensor. I used the Apple ARKit [28] to retrieve the depth and 3D point information to be used for the measurement value computation.

Figure 4.1 and 4.2 show two user scenarios.

In the first scenario, the user wants to discover possible adaptive designs for a tea bottle. (S)he points the camera towards the target bottle. The system detects the bottle from the camera view, and highlights it with a red bounding box. The user taps the highlighted bottle, which triggers the system to pop up a list of possible adaptation types, such as opener and holder. The user then selects the *opener* as the desired adaptation type. The system lists possible bottle opener designs. The user now explores some of these designs, decides on one bottle opener design, and clicks the 'measure' button to proceed to the next step of the measurement. The program finds the required parameters (bottle cap diameter), measures the top diameter of the bottle, and embeds the value to the OpenSCAD code file of the selected design, and generates a 3D model for exporting accordingly.

In the second scenario, the user wants to discover possible adaptations designs that adapts multiple objects together. (S)he taps to select two bottles in the camera view, and performs a pinch gesture. The system recognizes this case to suggest the *connector* as a possible adaptation type among others. The user then selects the bottle connector design, and the program performs the measurement similar to the first scenario.

As a proof of concept, the preliminary application demonstrates possible use cases of the proposed interactive system, which requires novice users less steps to discover designs, measure objects, and adjust models than the conventional search and 3D printing process. However, due to the limited LiDAR sensor resolution on iPad Pro, the current measurement could not be as accurate as desired. The current resolution of depth map generated from the LiDAR sensor is 256 by 192,

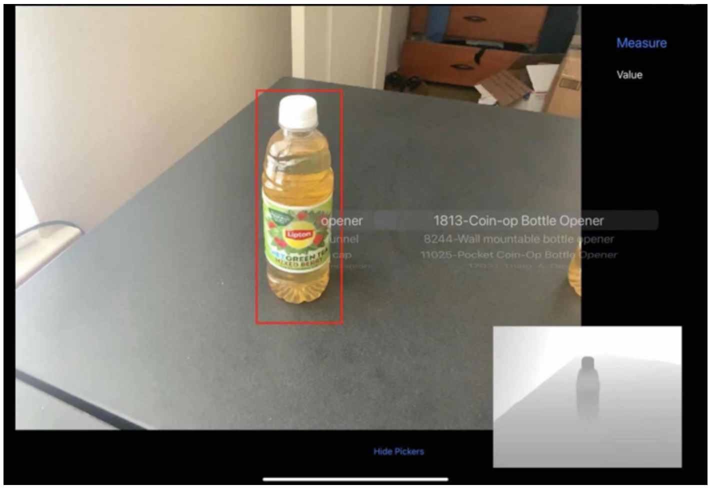

Measure
Value
opener
unnel
cap
1813-Coin-op Bottle Opener
8244-Wall mountable bottle opener
11025-Pocket Coin-Op Bottle Opener
Hide Pickers

Figure 4.1: Application Snapshot of Discovering Adaptive Designs for a Single Target Object. The Image in the Bottom Right Corner Shows the Depth Map Generated from the LiDAR Sensor Readings.

which may not be sufficient to precisely measure short parameters. The accuracy of measurement could be improved with the improvement of the on-board sensors. While other AR-based methods of measurement can be achieved with Apple ARKit, this implementation did not explore and compare those techniques, as the proposed pipeline could also be used with other mobile devices with stereo cameras that have no AR support on the software side.

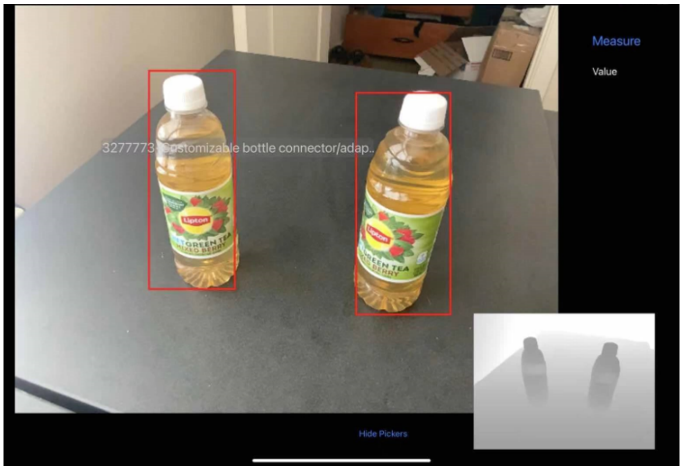


Figure 4.2: Application Snapshot of Discovering Adaptive Designs for Two Target Objects.

5. LIMITATION AND FUTURE WORK

This thesis proposes solutions for adaptive design discovery and measurement, and presents an interactive system to guide novice users through the process of printing adaptive designs.

While the proposed pipeline brings improvements to the existing solutions for design discovery and measurement, it still has several limitations and remaining technical challenges to be addressed in the future.

5.1 Detecting Target Object from Textual Information

As mentioned in table 3.1, the performance of detecting target object using named entity recognition from the title does not have a very high performance. While the current method can process titles consist of relatively clear and simple words, it cannot handle long title which includes many descriptive words, for example, ‘lm8uu ao x motor end side print version holder’ as in design 28776 on Thingiverse. In addition, it does not handle nested target objects. Designs such as ‘Camera tripod leg stick cap’ may introduce additional confusion to the entity recognition model to determine the most relevant object and the relationship between these possible target objects in the title. This could introduce one of the future directions of this work, which is to understand the sequence of target objects and be able to decompose large nested object names into smaller components. This could also help the measurement process, since the program will have a clearer focus on which part/component of the target object needs to measure, such as focusing on ‘lens’ in ‘Camera lens’, instead of measuring the whole object.

5.2 Computer Vision Based Measurement

For the measurement module, the current system requires a relatively clear background and good lighting condition to be able to correctly identify the object outline, and this is not always the case in real use cases. Thus, a measurement module that utilizes both RGB images and LiDAR readings might be more robust than the proposed method to locate target object outline, and can then be used in more diverse environment, such as environments with cluttered background.

Nonetheless, the proposed solution still demonstrates a great potential of utilizing computer vision techniques along with accurate time-of-flight sensors to detect, locate, and measure objects automatically. This would provide a more accurate result as high-resolution sensors become increasingly popular on mobile device, and be able to demonstrate its great potential to be applied to many other interactive systems as well.

5.3 Potentials to Be Used in Other Open Source Repositories

The current proposed system, pipeline, and additional metadata are primarily based on the adaptive designs on Thingiverse, which are more related to daily use objects, hobby projects, and artistic designs. However, the proposed system is not necessarily limited to 3D repositories. It also shows potentials in other (open source) repositories as an interface to understand, organize, and reuse existing designs, which could also be a potential direction for this work. For example, the proposed method could be used as a discovery and adjustment system for industrial components repositories, where each design (or part) has a clear definition of its functionality and a list of parameters of this part. The goal of this work is to propose a pipeline that can be used to assist the discovery and measurement (or adjustment) of designs in public repositories. Given a higher accuracy for each component in this pipeline, this could potentially be applied to other repositories as well.

6. CONCLUSION

This thesis proposes an interactive system for discovering and customizing adaptive 3D designs that are open-sourced for reuse. I discussed potential issues of the current search interface and document structure on Thingiverse, and listed possible challenges that novice users may encounter in the process of customizing adaptive designs. I then proposed an interactive pipeline that requires less user input and understanding of the design to guide them through a step-by-step procedure to discover desired designs and perform required measurement process. To support this pipeline, I discussed methods for retrieving additional adaptation-related and context-related information on top of the existing design metadata, and how they can be organized in an object-oriented graph structure to model the adaptation relationships between target objects and designs. After that, I introduced the modular measurement structure, and proposed an inheritance-based measurement strategy. I demonstrated the measurement methods for three most common primitive shapes that are found in Thingiverse repository of 3D designs, and how they can be expanded to objects with similar shapes but with additional measurement requirements. At the end, I showcased a proof-of-concept demo application for end users, demonstrating the potential of the proposed pipeline. I concluded with the current limitations of the system running on current devices, and showed possible applications of the proposed system for design discovery and adjustment in other open source repositories as well.

REFERENCES

[1] C. Mota, “The rise of personal fabrication,” in *Proceedings of the 8th ACM Conference on Creativity and Cognition*, Camp;C ’11, (New York, NY, USA), p. 279–288, Association for Computing Machinery, 2011.

[2] C. Alcock, N. Hudson, and P. K. Chilana, “Barriers to using, customizing, and printing 3d designs on thingiverse,” in *Proceedings of the 19th International Conference on Supporting Group Work*, GROUP ’16, (New York, NY, USA), p. 195–199, Association for Computing Machinery, 2016.

[3] E. Stemasov, T. Wagner, J. Gugenheimer, and E. Rukzio, “Mixamp;match: Towards omitting modelling through in-situ remixing of model repository artifacts in mixed reality,” in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI ’20, (New York, NY, USA), p. 1–12, Association for Computing Machinery, 2020.

[4] C. Mahapatra, J. K. Jensen, M. McQuaid, and D. Ashbrook, “Barriers to end-user designers of augmented fabrication,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI ’19, (New York, NY, USA), p. 1–15, Association for Computing Machinery, 2019.

[5] J. Kim, A. Guo, T. Yeh, S. E. Hudson, and J. Mankoff, “Understanding uncertainty in measurement and accommodating its impact in 3d modeling and printing,” in *Proceedings of the 2017 Conference on Designing Interactive Systems*, DIS ’17, (New York, NY, USA), p. 1067–1078, Association for Computing Machinery, 2017.

[6] L. Oehlberg, W. Willett, and W. E. Mackay, “Patterns of physical design remixing in online maker communities,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI ’15, (New York, NY, USA), p. 639–648, Association for Computing Machinery, 2015.

[7] X. A. Chen, J. Kim, J. Mankoff, T. Grossman, S. Coros, and S. E. Hudson, "Reprise: A design tool for specifying, generating, and customizing 3d printable adaptations on everyday objects," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, (New York, NY, USA), p. 29–39, Association for Computing Machinery, 2016.

[8] C. Weichel, M. Lau, D. Kim, N. Villar, and H. W. Gellersen, "Mixfab: A mixed-reality environment for personal fabrication," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, (New York, NY, USA), p. 3855–3864, Association for Computing Machinery, 2014.

[9] S. Mueller, "Interacting with personal fabrication devices," *it - Information Technology*, vol. 60, no. 2, pp. 113–117, 2018.

[10] D. Ashbrook, S. S. Guo, and A. Lambie, "Towards augmented fabrication: Combining fabricated and existing objects," in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, (New York, NY, USA), p. 1510–1518, Association for Computing Machinery, 2016.

[11] S. Mueller, P. Lopes, and P. Baudisch, "Interactive construction: Interactive fabrication of functional mechanical devices," in *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, (New York, NY, USA), p. 599–606, Association for Computing Machinery, 2012.

[12] K. D. Willis, C. Xu, K.-J. Wu, G. Levin, and M. D. Gross, "Interactive fabrication: New interfaces for digital fabrication," in *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '11, (New York, NY, USA), p. 69–72, Association for Computing Machinery, 2010.

[13] D. Dougherty, "The maker movement," *Innovations: Technology, Governance, Globalization*, vol. 7, no. 3, pp. 11–14, 2012.

[14] V. Savage, S. Follmer, J. Li, and B. Hartmann, “Makers’ marks: Physical markup for designing and fabricating functional objects,” in *Proceedings of the 28th Annual ACM Symposium on User Interface Software amp; Technology*, UIST ’15, (New York, NY, USA), p. 103–108, Association for Computing Machinery, 2015.

[15] M. Hofmann, G. Hann, S. E. Hudson, and J. Mankoff, “Greater than the sum of its parts: Expressing and reusing design intent in 3d models,” in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI ’18, (New York, NY, USA), p. 1–12, Association for Computing Machinery, 2018.

[16] E. Buehler, S. K. Kane, and A. Hurst, “Abc and 3d: Opportunities and obstacles to 3d printing in special education environments,” in *Proceedings of the 16th International ACM SIGACCESS Conference on Computers amp; Accessibility*, ASSETS ’14, (New York, NY, USA), p. 107–114, Association for Computing Machinery, 2014.

[17] T. Ludwig, O. Stickel, A. Boden, and V. Pipek, “Towards sociable technologies: An empirical study on designing appropriation infrastructures for 3d printing,” in *Proceedings of the 2014 Conference on Designing Interactive Systems*, DIS ’14, (New York, NY, USA), p. 835–844, Association for Computing Machinery, 2014.

[18] X. Niu, X. Fan, and T. Zhang, “Understanding faceted search from data science and human factor perspectives,” *ACM Trans. Inf. Syst.*, vol. 37, Jan. 2019.

[19] B. J. Jansen and U. Pooch, “A review of web searching studies and a framework for future research,” *Journal of the American Society for Information Science and Technology*, vol. 52, no. 3, pp. 235–246, 2001.

[20] S. Ranganathan, *Colon Classification: Basic Classification*. Madras library association publication series, Asia Publishing House, 1969.

[21] M. A. Hearst, “Clustering versus faceted categories for information exploration,” *Commun. ACM*, vol. 49, p. 59–61, Apr. 2006.

[22] X. Niu and B. Hemminger, “Analyzing the interaction patterns in a faceted search interface,” *Journal of the Association for Information Science and Technology*, vol. 66, no. 5, pp. 1030–1047, 2015.

[23] O. Kainz, F. Jakab, M. W. Horečný, and D. Cymbalák, “Estimating the object size from static 2d image,” in *2015 International Conference and Workshop on Computing and Communication (IEMCON)*, pp. 1–5, 2015.

[24] N. A. OTHMAN, M. U. SALUR, M. KARAKOSE, and I. AYDIN, “An embedded real-time object detection and measurement of its size,” in *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, pp. 1–4, 2018.

[25] “Lidar camera l515 - intel realsense depth and tracking cameras.” `https://www.intelrealsense.com/lidar-camera-l515/`. Accessed: 03/01/2021.

[26] “Spacy named entity recognition.” `https://spacy.io/usage/linguistic-features#named-entities`. Accessed: 03/01/2021.

[27] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[28] “Arkit - augmented reality - apple developer.” `https://developer.apple.com/augmented-reality/arkit/`. Accessed: 03/01/2021.