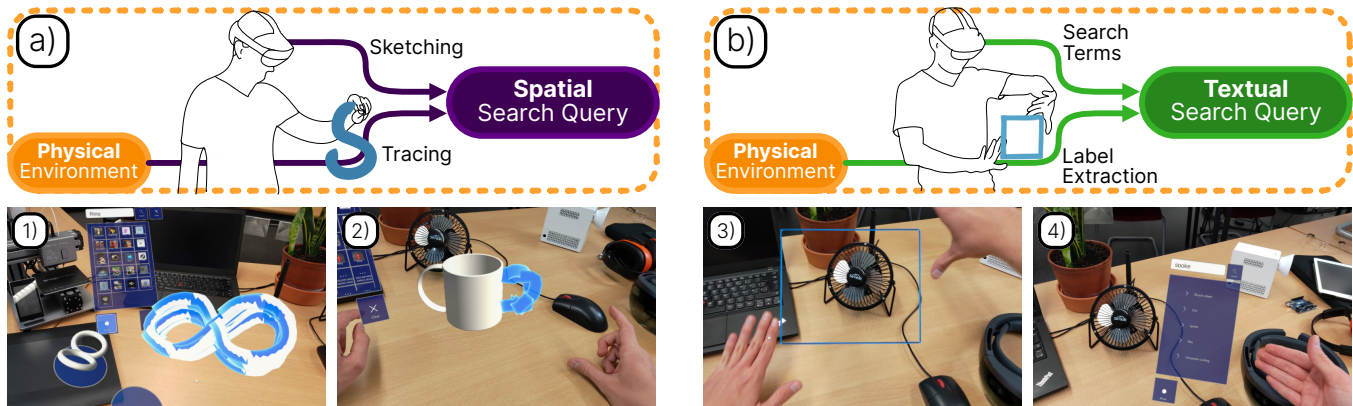


# ShapeFindAR: Exploring In-Situ Spatial Search for Physical Artifact Retrieval using Mixed Reality

Evgeny Stemasov  
Tobias Wagner  
evgeny.stemasov@uni-ulm.de  
tobias.wagner@uni-ulm.de  
Institute of Media Informatics,  
Ulm University  
Ulm, Germany

Jan Gugenheimer  
Télécom Paris – LTCI, Institut  
Polytechnique de Paris  
Paris, France  
jan.gugenheimer@telecom-paris.fr

Enrico Rukzio  
enrico.rukzio@uni-ulm.de  
Institute of Media Informatics,  
Ulm University  
Ulm, Germany



**Figure 1:** ShapeFindAR is a proof-of-concept implementation of in-situ spatial search. It enables users to search for future physical artifacts through spatial (a) or textual (b) queries, while coupling the search and previewing process to the users' physical context. Spatial search queries (a) rely on in-situ sketches, which may be drawn mid-air (1), be tracings of objects/features in the physical environment, or combine prior search results with sketched features (2). Textual search queries (b) rely on written or spoken terms known by the user. Alternatively, a user may use a photo-based label extraction to frame objects of interest (3) and receive labels to potentially use for searching (4).

## ABSTRACT

Personal fabrication is made more accessible through repositories like Thingiverse, as they replace modeling with retrieval. However, they require users to translate spatial requirements to keywords, which paints an incomplete picture of physical artifacts: proportions or morphology are non-trivially encoded through text only. We explore a vision of in-situ spatial search for (future) physical artifacts, and present ShapeFindAR, a mixed-reality tool to search for 3D models using in-situ sketches blended with textual queries. With ShapeFindAR, users search for geometry, and not necessarily precise labels, while coupling the search process to the physical environment (e.g., by sketching in-situ, extracting search terms from objects present, or tracing them). We developed ShapeFindAR for HoloLens 2, connected to a database of 3D-printable artifacts. We

specify in-situ spatial search, describe its advantages, and present walkthroughs using ShapeFindAR, which highlight novel ways for users to articulate their wishes, without requiring complex modeling tools or profound domain knowledge.

## CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; *Mixed / augmented reality*; • **Information systems** → Information retrieval.

## KEYWORDS

Personal Fabrication, Spatial Search, In-Situ Search, Mixed Reality, Physical Artifact Retrieval, Model Repositories, 3D-Printing

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CHI '22, April 29-May 5, 2022, New Orleans, LA, USA

© 2022 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-9157-3/22/04.

<https://doi.org/10.1145/3491102.3517682>

## ACM Reference Format:

Evgeny Stemasov, Tobias Wagner, Jan Gugenheimer, and Enrico Rukzio. 2022. ShapeFindAR: Exploring In-Situ Spatial Search for Physical Artifact Retrieval using Mixed Reality. In *CHI Conference on Human Factors in Computing Systems (CHI '22)*, April 29-May 5, 2022, New Orleans, LA, USA. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3491102.3517682>

## 1 INTRODUCTION

Personal Fabrication is a powerful opportunity for technology enthusiasts and consumers alike: It empowers them to design and fabricate artifacts that are unrivaled in their degree of personalization and precision. While technology enthusiasts may be compelled to invest time in the process of personal fabrication, consumers may generally care more about the result [21] and expect a low-friction workflow [52]. To be able to include the whole spectrum of users to benefit from the progress of personal fabrication, future creation tools are increasingly addressing the needs of non-experts and laypeople [14, 46, 55], which are often not able or not willing to express their needs with the precision an expert would. This likely applies to the process of *modeling* artifacts, but also to the process of *searching and retrieving* the correct ones [50]. Through more affordable and robust hardware (e.g., 3D printers, laser cutters) and improved software workflows (e.g., modeling tools, slicers), personal fabrication is now in reach for a wider user base than ever before.

However, most established workflows used to define artifacts to fabricate (e.g., 3D modeling  $\Rightarrow$  slicing  $\Rightarrow$  3D printing), require both learning and usage effort. Such challenges, in part, explain the popularity and importance of open, crowd-based model repositories, such as Thingiverse or MyMiniFactory [2, 13, 21]. Model repositories provide users ready-to-print artifacts, making the interaction with the repository more related to shopping and less like established notions of modeling or designing artifacts [52]. Model repositories are an appropriate alternative to modeling artifacts from the ground up, and will become even more viable as they grow in size and artifact diversity. However, model repositories require feasible ways to be searched, to be a viable asset for users. Text-based search was adopted from established domains (e.g., file search), but largely misses the spatial nature and physical context of searching for (future) physical artifacts. Similarly, query formulation and artifact previewing are disconnected from the physical context they are meant to be placed in later on. When facing a requirement or challenge, users then are either required to invest effort in modeling artifacts or are required to translate their requirements to search terms and subsequently refine them.

We propose ShapeFindAR, a mixed-reality tool to search model repositories for 3D printing in-situ. It combines two types of search queries: textual and spatial. Users may sketch coarse 3D shapes that are then used for geometry-based (i.e., spatial) searches. They may likewise enter known search terms for textual searches. To leverage the user's physical context, ShapeFindAR provides ways to support users in the task of defining and combining these two query types (Figure 1): users are able to trace features of the environment for spatial searches, or may retrieve labels for objects in the environment for textual searches. This bridges the disconnect between the users' physical environment and the search process. ShapeFindAR was developed as an application for the Microsoft HoloLens 2<sup>1</sup> and is connected to a custom database of 3D-printable artifacts. By leveraging additional, spatial modalities (i.e., sketches), ShapeFindAR allows users to omit the task of precisely defining search

terms for artifact retrieval. While some objects have clear and established terminology, more niche artifacts may require knowledge of the terminology used to describe them (i.e., a certain degree of domain knowledge). An example can be seen in Figure 1 a-1: a ring meant to be worn on two fingers simultaneously. Users may not necessarily know the term "*double ring*", but likely have an intuitive understanding of the *geometry* they desire. Hence, they would be able to sketch the shape in a coarse fashion and use this sketch, combined with imprecise search terms (e.g., "ring"), to find the desired artifact. Similarly, users may have an intuitive understanding how an elongated vase with handles looks like, but may not know the term "*amphora*" used to *precisely* denominate it. By searching for geometry, users may receive results that fulfill their intended (functional) needs, without limiting themselves to objects *originally* meant for their goal. By conducting the sketching procedure in-situ (i.e., at the location of the future artifact), users benefit from previewing and referencing real-world features (e.g., by tracing them). ShapeFindAR also enables users to get machine-generated labels and guesses after framing an object of interest (Figure 1 b). This enables them to discover potentially unknown terms. In turn, this helps finding similar objects, or objects meant to interact with the existing object. By providing multiple ways (i.e., *modalities*) to express and refine searches, ShapeFindAR aims to allow users to choose the path to a desired artifact that fits the task and the user best. Users are also encouraged to iteratively refine their queries. A fitting 3D model could be used as a base for a new spatial search query, by sketching new features onto it, as seen in Figure 1 a-2. With ShapeFindAR, we take the perspective of 3D printing as a domain for physical artifact acquisition. However, we believe that in-situ spatial search, and, by extension, ShapeFindAR, are applicable to the search for and acquisition of physical artifacts in general, regardless of the actual method to fabricate or acquire it.

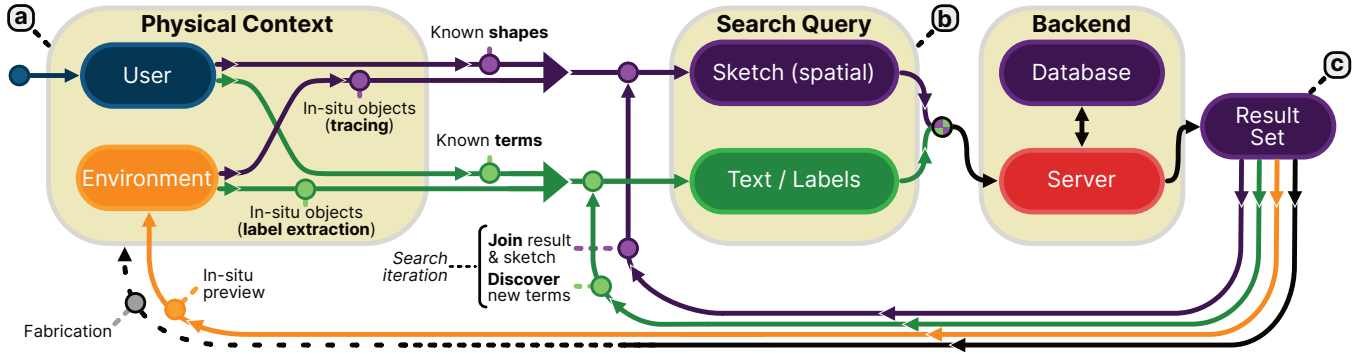
The contributions of this work are as follows:

- **The concept of in-situ spatial search for (physical) artifact retrieval.** It actively embeds both spatial input, textual input and the user's physical context in the search procedure.
- **Proof-of-concept implementation of ShapeFindAR,** a Mixed-Reality application enabling in-situ spatial search of a model repository for 3D printing.
- **Exploration of the concept** of in-situ spatial search, using walkthroughs enacted with the ShapeFindAR prototype. The walkthroughs highlight novel ways for users to formulate and refine queries for future physical artifacts.

## 2 CONCEPT: IN-SITU SPATIAL SEARCH

With this section, we want to elaborate on the concept behind ShapeFindAR. *In-situ* spatial search supports users with the *transfer of features in their physical context to search queries*. This currently happens through users' mental efforts, domain knowledge, and through textual queries only. In-situ *spatial* search also *embraces the spatial nature of physical artifacts*, not only for result-presentation, but, where beneficial, also during the search process. We argue that text-based search adds an unnecessary abstraction from a three-dimensional, spatial problem to a one-dimensional search term. Only in the final step, the retrieved artifact is transformed back to

<sup>1</sup><https://www.microsoft.com/en-us/hololens>, retrieved on 12.12.2021



**Figure 2: Conceptual process of in-situ spatial search, as demonstrated with ShapeFindAR.** The users reside in the same physical context in which a future artifact may be fabricated for (a). They may search the repository with 3D sketches or known words, composing a search query (b). As the search happens *in-situ*, users may also extract labels from the environment or trace existing objects. While refining their queries, users may learn new terms from the search results for textual searches and sketch new features onto the results. The query yields a result set (c) generated by the database server. Both query types (spatial, textual) can be combined, enabling multimodal search, while allowing users to be ambiguous in one or both modalities.

3D by presenting a 3D object (either a digital, or a physical one) to the user. This abstraction towards a 1D space (i.e., text) is helpful and powerful for expert users that know the (domain-)specific language and terminology. However, it creates an unnecessary burden for laypeople and novices who may be able to coarsely describe their 3D problem in-situ without necessarily knowing the appropriate and precise terms.

The **in-situ** aspect of in-situ spatial search supports the transfer of features in the user’s physical context to search queries. Physical objects may be traced with (spatial) sketches, scanned (if the headset is capable of doing so), or be photographed to extract labels. These approaches transfer and translate context features (e.g., geometry) to search queries. With in-situ spatial search, more abstract aspects of context could also be available to use and embed in queries. Context may mean the room a user is in (e.g., the kitchen), the task the user is attempting to solve (e.g., attach an object to another), the general design of the room (e.g., “modern”), or the color climate of the interior (e.g., earth tones to be matched by the future artifact). ShapeFindAR enables users to trace objects and apply photo-based label extraction. Apart from the use of the context’s features for search queries, it is likewise used for previewing artifacts in context. This enables the user to estimate the form and function of the future artifact, while also allowing interactions like positioning and scaling to evaluate and improve the artifact prior to fabrication [53] or acquisition [33] in general.

The **spatial** aspect of in-situ spatial search embraces the spatial nature of physical artifacts during the search process. By embedding geometry and sketches in the search procedure, geometry receives an emphasis over terminology. If systems sidestep text and labels, functions and functionality ascribed to objects by contributors to an artifact database (e.g., users on Thingiverse) can be ignored. The general topology of cups, pencil holders, vases, pots, and other cylindrical containers is comparable. However, this geometry of a bin is labeled differently by contributors, while all

can be used for “holding” tasks (e.g., holding pencils). Not all artifacts *suitable* for holding pencils can be found through the original label (e.g., “pencil” or “pen” holder). While geometry does not necessarily strictly *define* function, it often is tightly coupled to it, particularly for static objects. Furthermore, we argue that users are likely to have at least a coarse understanding of their desired geometry or shape. Approximating it through a spatial sketch to find existing artifacts (to use or remix) may be a feasible procedure to retrieve a fitting one. This avoids the enforced transfer of spatial requirements (3D) to *precise*, written keywords (1D), which is particularly relevant if a) fitting terminology is unknown or b) if the actual geometry of a future artifact is the most relevant aspect.

We developed ShapeFindAR as a proof-of-concept prototype embodying our vision of in-situ spatial search. With the means of personal fabrication, the process of artifact acquisition can be approached from distinct directions: 1) design (e.g., 3D modeling or sculpting an artifact from the ground up) or 2) retrieval (e.g., searching for an artifact on Thingiverse to fabricate). Both directions are equally viable to users focusing on the goal of “attaining an artifact”. Usually, the design or modeling approach exhibits the highest complexity, but also enables sufficiently proficient users to achieve *any* requirement they have [52]. However, designing artifacts requires knowledge and training, and therefore has a higher entry barrier. In contrast, retrieving an artifact generally requires knowledge of the correct terminology to pass to a search system. Alternatively, users have to rely on recommender systems, or invest time in elaborate iterations and refinement. When dealing with high-level concepts (e.g., “chair”, “vase”), terms are generally established and there is little to no mismatch between the users’ intentions and the understanding of the term by the artifact repository. The more specific and personal the requirements become, the more detailed and precise the search terms have to be to yield satisfying results. This makes the combination of multiple search modalities an appealing outlook.

ShapeFindAR provides two main ways for searching: spatial queries (i.e., through sketching) and textual queries. A diagram of the search process is depicted in figure 2. Users are supported in the task of embedding their physical context in the queries: by tracing existing features for a spatial query, or by using label extraction to generate suggested search terms for a textual query. This is crucial, as any artifact users retrieve is meant to be fabricated or acquired otherwise – ultimately interacting with the physical context [4]. In-situ search lowers the effort required to define searches and preview results in-context, as it omits *transfers* between a location of (future) use (e.g., the table a vase will rest on) and a location of searching/design (e.g., a computer). Both query types may also be created without involving the users’ physical context (e.g., by relying on known terminology or known geometry). This allows users to freely encode their existing wishes and knowledge, but demands a certain level of domain knowledge (for search terms) or at least a coarse understanding of the desired geometry. Through the use of in-situ spatial search procedures, a set of more specific advantages emerges for users: users may circumvent established terminology, ignore objects’ intended function and use a sketch’s scale as a filter or a target scale. In general, in-situ spatial search is meant to enable users to formulate and refine *ambiguous* search queries in *multiple modalities* to ideally retrieve finished, fitting designs from a model repository.

### 3 RELATED WORK

The concept of in-situ spatial search draws inspiration from different research directions. ShapeFindAR is meant to be a low-effort, novice-friendly tool for personal fabrication, which skips established steps such as 3D modeling. The development and usage of in-situ tools for design and fabrication is likewise a crucial direction related to ShapeFindAR. Similarly, novel methods for non-textual search and fabrication that do not necessarily follow established paradigms of (3D) modeling are fundamentally relevant to our work. Conceptually, the embedding and use of model repositories like MyMiniFactory or Thingiverse is a relevant direction, as it emphasizes the benefits of crowd-sourced artifacts (i.e., artifacts that have already been designed and specified).

#### 3.1 Novice-friendly Tools for Fabrication

Ongoing research efforts are being made to enable personal fabrication for a wider audience. This audience consists primarily of novices to design and fabrication (i.e., “consumers”). As modeling artifacts is a complex task, attempts were made to simplify this modeling process, by focusing on fewer primitives [6], 2D designs for 3D models [38], or sketch-based modeling [45]. Ballagas et al. explored voice input to generative models [5], Lee et al. applied gestures to the task of furniture customization [32]. CraftML by Yeh and Kim transferred 3D modeling to a declarative programming approach [64]. Apart from purely digital approaches, analog fabrication tools can be augmented and simplified, like Match-Sticks [59] or Turn-by-Wire [58] by Tian et al. To support novices, automation and generative designs were used [22, 35, 47], which infer the users’ design intents based on often coarse input, such as sketches [24, 25, 34, 45].

All aforementioned tools can be seen as approaches to reduce effort, learning, and overcome challenges in personal fabrication. With in-situ spatial search and ShapeFindAR, we similarly aim to address novices to personal fabrication and design. However, we focus on outsourced design artifacts (i.e., 3D-printable models from experienced users uploaded to Thingiverse or MyMiniFactory). This is a contrast to procedures inferring or generating geometry [35, 45] or requiring users to model artifacts by aggregating primitives [6, 64].

#### 3.2 In-situ Tools for Personal Fabrication

There is a disconnect between the location of artifact design and the artifact’s future location [4, 37]. This requires either assessment and transfer of requirements, or methods to compensate measurement errors [27], which is a challenge that is not exclusive to novices. In-situ methods for modeling and fabrication enable users to preview designs within their future context. This was demonstrated by Peng et al. [39], where design overlapped fabrication, or Yung et al. [66], where the target audience was children. In-situ tools relying on augmented or mixed reality also enable the augmentation and guidance of previously manual and analog fabrication approaches, such as 3D pen sculptures [65], carved models [17, 18], or linkages [23] by providing feedback during the process itself. They may also support remote collaboration and learning, as shown by Villanueva et al. [60]. Mixed reality approaches, such as the works by Weichel et al. [62] or Jeong et al. [23] enable situated design approaches, while other tools focus on fabrication [39, 69], or remixing [53]. The usage and embedding of real-world artifacts as counterparts to future fabricated artifacts is likewise an aspect of in-situ tools. It enables the design of fitting mounts [69], mechanisms actuating other objects [35], or previewing to-scale sketches in context [1].

Personal fabrication and any process that generates personal physical artifacts require grounding in the users’ physical context. This is achieved through situating design or retrieval processes in this particular context. Mixed reality is an outstanding opportunity to achieve this type of interaction. ShapeFindAR aims to provide a similar benefit, and, for instance, enables users to trace real-world artifacts to use these features for their searches. However, our focus lies on search and retrieval, instead of modeling or design.

#### 3.3 Alternative-modality Interfaces for Fabrication or Search

The combination of different input modalities enables users to bypass established metaphors for CAD (i.e., for the *design* of artifacts) or information retrieval (i.e., for the *search* for artifacts). This includes uses of sketches as a coarse input [31, 34, 45], but also gestures [10, 26, 32] or speech [5]. Alternatively, tangible manipulation may be employed for input and provide appropriate feedback [48, 57] not found in most industrial CAD tools. Such ways to express and define geometry may, depending on the user, reduce the effort to design geometry [51]. In contrast to the interfaces that focus on the *design and generation* of artifacts, retrieval interfaces focus on the search for finished designs or parts [20]. Fraser et al. presented ReMap, which embodies a prototype of multimodal search [15], an aspect we aimed to embrace with ShapeFindAR. On a more technical level, various approaches for the comparison of 3D



models are present in the literature. These include approaches to align point sets [3, 44], computation of object similarity [8, 28, 30], and associated retrieval methods based on geometry [56]. In particular, gesture-based systems, such as DataMiming by Holz and Wilson [20] or sketch-based systems, such as the ones introduced by Pu et al. [41] Enenhofer [12] or Eitz et al. [11] inspired ShapeFindAR. While DataMiming omitted visual feedback for the users and focused on gestures as input, it is conceptually similar to ShapeFindAR, as it uses a fairly natural way to describe artifacts, and offloads the precise definition or matching to other parties [20]. ShapeFindAR does not omit visual feedback but considers it to be a crucial part of the search process. It also benefits from context features (tracing, label extraction) and embraces multiple modalities to give users more freedom and ambiguity. With platforms like Thangs [40] offering 3D-model based search for 3D models, the notion of searching with shapes over labels is already in reach for consumers, albeit in an ex-situ fashion. Giunchi et al presented a combination of sketching and speech for the retrieval of chair models in VR [16]. While conceptually similar, our focus lies on users' physical environments, instead of virtual ones. This requires ways to transfer spatial requirements to searches.

Such novel user interfaces for search or design are an opportunity to enable widespread access to personalized artifacts, by lowering the skill floor while retaining high expressivity for more proficient users. Novel modalities are applicable both to searching for existing artifacts [20, 49] and generating new ones [5, 32, 63]. ShapeFindAR can be considered to be a comparable approach to artifact retrieval, with an emphasis on coarse inputs that are – ideally – transferred to high-fidelity artifacts from a model repository. ShapeFindAR similarly aims to enable and support iterative search procedures, where users leverage their physical context and retrieved artifacts to formulate and refine searches.

### 3.4 Usage of Model Repositories

Model repositories (e.g., Thingiverse) and their associated communities of makers are an important component of today's personal fabrication landscape. Repositories have been leveraged as a model source for remixing tools, such as Graftor by Roumen et al. [43], or Mix&Match by Stemasov et al. [53]. They have also been the subject of studies and investigations, generating novel frameworks like PARTs by Hofmann et al. [19] that aim to increase re-use of design effort. Likewise, systematic analyses of repositories like Thingiverse and their users unveiled challenges in terms of artifact use and customization by novices [2, 21]. Works by Flath et al. [13] or Kyriakou et al. [29] evaluated patterns and degrees of knowledge and design reuse. Domain-specific usage, such as the exchange of assistive technologies [7] is likewise a field benefiting from a combination of design sharing, re-use, and customization.

We agree with and embrace the notion that crowdsourced models or artifacts are crucial for personal fabrication. Notably, any storefront – on- and offline – provides users a similar experience, with varying degrees of potential personalization. When considering stores to be an alternative to personal fabrication (currently chosen by a majority of the population [52]), approaches relying on searching and customizing existing designs can be considered to be a viable path to take. We approach this aspect with an emphasis on

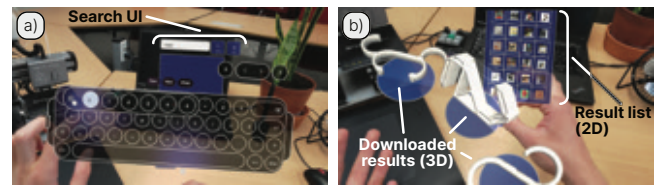
search and how it is applied to the task of finding future artifacts for one's own, personal, physical context.

## 4 PROOF OF CONCEPT IMPLEMENTATION

The following sections describe the development of ShapeFindAR as proof-of-concept implementation. The system consists of 3 components: the application for Microsoft's HoloLens 2, a server component connected to the database, and a data-gathering tool which was used to fill the database with models gathered from Thingiverse and MyMiniFactory. We report our approach and development process openly for replicability.

### 4.1 HoloLens 2 Application

We implemented the user-facing part of ShapeFindAR with Microsoft's HoloLens 2 using Unity<sup>2</sup> 2019.4.6f1. Version 2.4 of the Mixed Reality Toolkit (MRTK<sup>3</sup>) was used to implement the user interface and most of the interactions with it. Networking was implemented with the RestClient<sup>4</sup> library for Unity.



**Figure 3: The textual search interface, along with the keyboard for text input (a). After submitting the search query, a user is presented with a 2D list of results and thumbnails (b, right). Downloaded objects are rendered in 3D and are attached to the user's left hand (b, left).**

The search interface is initially attached to the user's right hand. If needed, the user can grab and position the panel statically in space (Fig. 3a). The search interface can be used to enter search terms by text or speech. After submitting a query, the interface loads a scrollable grid of results (Fig. 3b). The user can select results that appear promising, which are then downloaded in the background. As soon as the download and instantiation are completed, the model appears attached to the user's left hand, forming a palette where the user may gather up to 5 models to compare and evaluate (Fig. 3b). Each artifact loaded into the palette can be grabbed and positioned in the space around the user. The objects can then be repositioned, rotated, and scaled freely, which enables an in-situ preview of them.

The sketch search component can be activated by tapping on a button below the search interface. After activating the sketch mode, the users can draw in the space around them by pinching their thumb and index finger (Fig. 4a). The sketch can then be submitted as a query. To ensure appropriate results, sketch-based search currently requires the users to enter at least a coarse search term. This is mainly used to refine the search to receive faster results and can be removed in the future with either better (backend) hardware or a more optimized implementation of the matching algorithm. The

<sup>2</sup><https://unity.com/products/unity-platform>, retrieved on 16.12.2021

<sup>3</sup><https://microsoft.github.io/MixedRealityToolkit-Unity>, retrieved on 26.8.2021

<sup>4</sup><https://github.com/proyecto26/RestClient>, retrieved on 15.12.2021

users may also sketch onto downloaded and positioned meshes, to extend them and use them as a new query (Fig. 4b). This feature also enables users to position multiple models they retrieved from the repository and connect them through their sketch, which is then treated as a single spatial query by the server. Thereby, users do not necessarily have to sketch entire objects, but may start with an object and add features they deem missing from it (e.g., adding handles to a vase).

The image-based label retrieval extracts possible terms for textual queries based on an image captured by the headset. This component of ShapeFindAR is fundamentally a mapping from a 2D image to a search term (1D). It is enabled by connecting and holding the index fingers and thumbs of both hands for one second (dwell time) in the field of view of the device. This activates a "framing mode" (Fig. 5a). Afterwards, the users can drag their hands apart, to form a frame around the object they want to use as a query (Fig. 5a). The captured image is used as a query to the Google Cloud Vision API<sup>5</sup>, which retrieves suggested labels based on the image's contents.

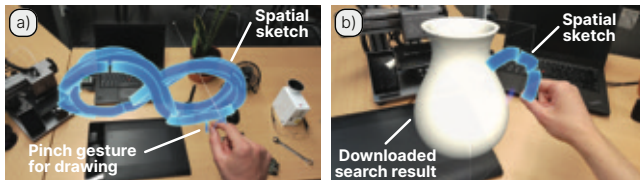
## 4.2 Data Collection

While Thingiverse and MyMiniFactory provide crucial infrastructure for makers, they are – by design – focused on established ways to be searched: textual queries along with sorting and filtering options. Notably, most filters focus on tags and categories (i.e., more abstract terminology). Sorting similarly focuses on non-spatial dimensions, like popularity or (textual) relevance. We chose Thingiverse and MyMiniFactory as data sources for ShapeFindAR. Thingiverse can be considered the de-facto standard for model repositories for 3D printing, while MyMiniFactory aims to curate the models offered on the platform. However, neither of these platforms offers ways for geometry-based searches<sup>6</sup>. Thangs<sup>7</sup> allows users to upload 3D models as queries, but offers no way to define them (either coarse sketches or precise models), while also functioning ex-situ only [40]. To comply with the platforms' API guidelines and to enable our novel functionalities without straining public infrastructure, we chose to duplicate a subset of their libraries. We chose high waiting times between all requests, to avoid putting unnecessary strain on the servers of Thingiverse and MyMiniFactory. This led to timespans of multiple days to gather batches of 500

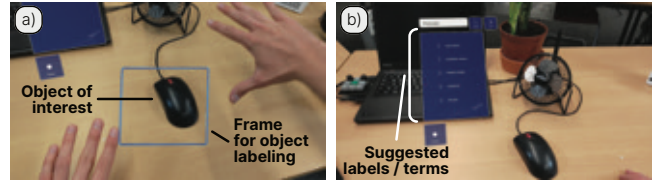
<sup>5</sup><https://cloud.google.com/vision/>, retrieved on 11.12.2021

<sup>6</sup>Implicitly, users may search for specific measurements or labels and hope to find results through textual search

<sup>7</sup><https://thangs.com/>, retrieved on 16.12.2021



**Figure 4: The sketching mode of ShapeFindAR. The users are free to sketch queries in the space around them with a pinch gesture (a). To iteratively refine spatial search queries, users may also sketch additions onto an object retrieved from the repository (b) and submit this combination as a new query.**



**Figure 5: The image-based label extraction of ShapeFindAR: The user activates a framing mode through a gesture, encloses the artifact to be classified in the frame, and confirms the selection via dwell-time (a). The "best guess" is pre-filled into the search field and additional suggested keywords are listed below (b).**

models. The keywords used were basic mechanical artifacts popular in 3D printing communities like hooks, and decorative artifacts like vases or figures. Furthermore, the most popular artifacts on each of the repositories were gathered. Data like the original source, the designer, the licenses used were all transferred to our database subset.

The entire dataset we gathered consists of 4118 objects in total. 2717 objects consist of one part only, while the remaining ones have 2 or more files (e.g., .stl) associated with them. 2356 objects out of 4118 provide mesh metadata suitable for our implementation of spatial search queries.

## 4.3 Postprocessing

Postprocessing of the metadata and the meshes themselves was necessary after each iteration of data gathering. The tool used for mesh handling was the trimesh library for Python by Dawson-Haggerty et al. [9]. Some meshes were corrupted, and were for instance missing correctly calculated normals or did not describe a volume. We did not apply any "opinionated" automated mesh repair methods<sup>8</sup>, due to the risk of corrupting the original geometry. The data sources Thingiverse and MyMiniFactory both employ the concepts of *categories* and *tags*. While tags are largely unmoderated and can be freely chosen by the users, available categories are specified by the repository administration. The categories used by Thingiverse and MyMiniFactory are comparable, but not identical. Categories were therefore grouped and matched into new categories used by our database.

To improve the text- and voice-based search, fields like the artifact name, description, or tags were subject to postprocessing with NLTK<sup>9</sup> (stemming, lemmatization, and stopword removal). To enable geometry-based searches, spatial metadata of the meshes was also established during this step. As detailed registrations and comparisons of meshes are computationally expensive, we implemented methods to pre-filter the result. We calculated two ratios based on the ordered dimensions of the bounding box to filter meshes based on their proportions. The voxel representation used for aligning and matching sketches to objects was computed at this step and serialized to the database, to avoid re-computing it on each search request.

<sup>8</sup>e.g., ones that fill holes larger than 1 triangle in size or remove self-intersecting or degenerate triangles, which are present in repositories like Thingiverse [68]

<sup>9</sup><https://www.nltk.org/>, retrieved on 1.12.2021

#### 4.4 Server Implementation

All server-side software was written in Python. The server component was written using flask<sup>10</sup>. The flask application provides a REST (representational state transfer) API for the HoloLens client: searches by text, searches by sketch, download of metadata, files, and thumbnails. For the database component, MongoDB<sup>11</sup> 4.4.0 was used, due to its flexible data model and its document-based approach. To accelerate queries, our database instance employs multiple indexes. Apart from indexing the ID fields, which accelerates retrieval of object details and files, a textual index and a 2D spatial index are used. This ideally accelerates searches submitted through their respective index. For queries ultimately converted to text, the database uses a text index<sup>12</sup>, based on the title, description, tag, and category fields. The fields are weighted based on their specificity (with categories being the least and names being the most specific). Searches passed to this index omit stopwords and employ stemming and lemmatization for the English language<sup>13</sup>. A second index configured for the database is a 2D spatial index<sup>14</sup>. We leverage the accelerated query time for 2D-data we derive from the meshes' object-aligned bounding box (OABB). The ratios between height/width and width/depth are used to pre-filter models based on their *proportions*.

Each search request is initially treated as a textual search, but gets by highly imprecise and general terms (e.g., "object"). This enables pre-filtering the results to amounts manageable by the server for computationally expensive operations and manageable by the user to get an appropriate overview. After having reduced the potential search result set, the server filters the results based on the previously introduced metric of OABB ratios. This set of results that match based on the OABB ratio is intersected with the set retrieved by the textual search. With this selection, it is now possible to calculate per-voxel overlap, which is the server's ranking/scoring function. The stored voxel representations are already size-normalized (i.e., their largest extent is set to 100 and the other extents are scaled proportionally to match). The incoming sketch is similarly normalized in scale. However, the original extents are retained, if the resulting model is meant to be scaled to match it. The sketch is then voxelized. The resolution of the voxelization is likewise defined by the model's largest dimension, as the voxelization pitch was chosen to yield 20 voxels across the largest dimension. The voxelized meshes are then used to run the ICP algorithm (iterative closest point [67]). The ICP procedure aligns the sketch to the repository model, as the latter usually has an orientation that follows the principal axes of inertia. The starting parameters for the ICP procedure are chosen based on the inertia of the mesh. The voxelized sketch is then rotated and moved with the calculated transformation. Due to the previously conducted size-normalization, no scaling is involved in this step. With the 2 models aligned and with maximal overlap, the amount (i.e., the number of voxels) of overlap is counted. Based on the resulting

overlap, we calculate multiple similarity metrics: normalized by the number of voxels in the sketch, normalized by the number of voxels in the repository model, and an average of these two normalized values. To accommodate for different volumes (e.g., sketches being entirely wrapped by the repository model), the average of the two normalized values is used as the core ranking for the search results.

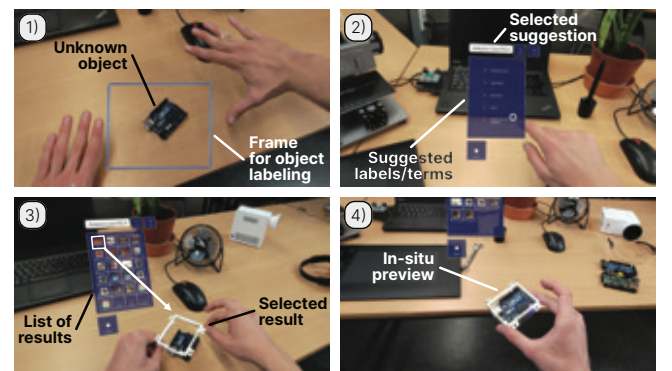
### 5 APPLICATION WALKTHROUGHS

The following scenarios were enacted with the ShapeFindAR prototype and present brief, self-contained examples of actual interactions with the system. The search results seen in the examples are based on our custom dataset and are therefore a *subset* of models available on Thingiverse or MyMiniFactory. However, due to a sizeable number of different objects of different categories, the user interacts with the dataset and the search results in a similar fashion to the potential interaction happening with a more extensive database. Furthermore, the sketch search is not ideally calculating similarities, but fails in specific edge cases (e.g., completely flat sketches, or when the user's sketch has entirely different proportions than any fitting object in the database). It does, however, generally retrieve a set of results where fitting or similar artifacts can be found among the first ones and iterated with. With the walkthroughs, we want to emphasize different ways in which in-situ spatial search enables users to ambiguously and iteratively approach physical artifact retrieval.

#### 5.1 Scenario 1: Bypassing Terminology

ShapeFindAR supports users by affording them a degree of *Term-Abstraction*: not having to know or choose the terminology of a domain. This is enabled through the use of photo-based label extraction (2D  $\rightarrow$  1D) and sketch-based search (3D  $\rightarrow$  3D).

The first example can be seen in Figure 6. A microcontroller (Arduino Uno), is in a user's vicinity. The name "Arduino" does not appear on the device, as it is made by an alternative manufacturer – however, this would be an ideal term to use in a search. Not knowing what it exactly is (i.e., what to search for specifically), the user may want to find a case to enclose or mount the exposed



**Figure 6: Term-Abstraction: Finding artifacts when terminology is unknown.** 1) Framing the object to be searched for; 2) using one of the classified labels; 3) search results and scaling of a selected one; 4) previewing the result in-situ.

<sup>10</sup><https://flask.palletsprojects.com/>, retrieved on 26.8.2021

<sup>11</sup><https://www.mongodb.com/>, retrieved on 22.8.2021

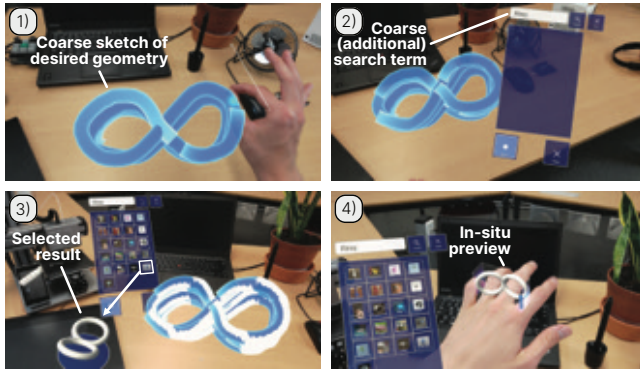
<sup>12</sup><https://docs.mongodb.com/manual/core/index-text/>, retrieved on 1.8.2021

<sup>13</sup>Notably, not all gathered artifacts are named and described in English. This is an additional argument in favor of non-textual queries.

<sup>14</sup><https://docs.mongodb.com/manual/core/2d/>, retrieved on 1.8.2021



microcontroller. He is able to use the photo-based label extraction component to frame the device (Fig. 6-1) and is suggested the term "Arduino" among others (Fig. 6-2). Executing this textual query yields a set of related models for 3D printing (Fig. 6-3). This includes cases, mounts, but also objects that require an Arduino to provide interactivity. The models can be downloaded and previewed in situ (Fig. 6-4), for the user to verify the appeal, and, to a degree, the fit of the model.



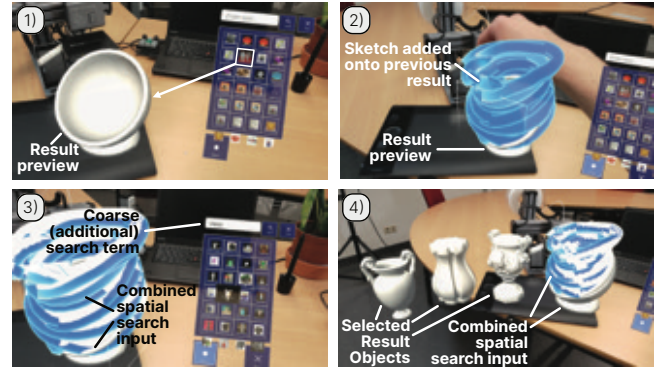
**Figure 7: Term-Abstraction: bypassing domain terminology of jewelry.** 1) sketching the coarse geometry of a double ring; 2) entering a coarse textual search term; 3) analyzing the list of (2D) results (input sketch and one fitting 3D result are visible); 4) previewing the (3D) object on the body.

Another example is depicted in Figure 7, where the user is interested in finding a double-ring<sup>15</sup> which is a piece of jewelry worn on two fingers, instead of one. Most users interested in such a construct likely have an intuitive idea of the *geometry* of the object: two similarly sized rings, possibly connected through a plate or other decorative element. However, they may not know the *ideal term* to find such objects in a repository. A user can then sketch this geometry (2 rings and, if needed, a connecting plate – Fig. 7-1), and use this in conjunction with a coarse search term (e.g., "ring", which would not retrieve the desired double ring as a high ranking result – Fig. 7-2). While actual double rings do not rank first in the ShapeFindAR search, they are among the first results presented to the user (Fig. 7-3). Lastly, the user may preview the ring in-situ, which may not guarantee a perfect fit, but at the very least indicates fit and aesthetics (Fig. 7-4).

## 5.2 Scenario 2: Iterative Refinement

We do not expect users to formulate "ideal" queries that lead to the desired results in one single step. Refinement in textual searches usually happens by browsing initial results and applying filters. Alternatively, users may discover new terms to search for by browsing through item titles or descriptions, thereby acquiring a degree of domain knowledge. ShapeFindAR enables both a textual and a spatial query refinement process.

An example can be seen in Figure 8. The user initially searches for bowls to use for decoration (Fig. 8-1). As she notices that she



**Figure 8: Iteration (spatial): Sketching onto retrieved artifacts to iterate on the (spatial) search query.** 1) the initial bowl retrieved from the repository; 2) Sketching onto it to elongate its shape; 3) submitting the sketch search with a new term; 4) a set of results for the combination of sketch and model.

would prefer ones that are generally taller, she can either take one of the pots retrieved from the repository and scale it in the Y direction, or she can enable the sketching mode and add onto the downloaded pot to enlarge it (Fig. 8-2). After having changed the pot's proportions this way, she may submit the combination of previous result and sketch, along with a coarse search term (Fig. 8-3) as a new query. Lastly, she may preview the results in-situ and apply other manipulations such as scaling to the (3D) results (Fig. 8-4). While the initial query is formulated without leveraging in-situ features, the query refinement may happen through referencing the physical context (e.g., comparing proportions of the results with existing decoration and altering it accordingly).

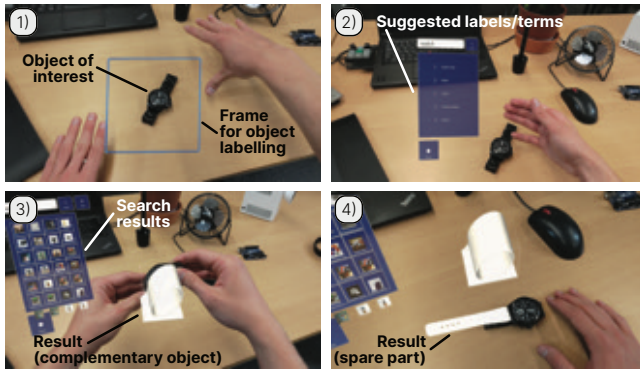
## 5.3 Scenario 3: Inspiration

Photo-based label extraction can be employed to provide inspiration and related items to the user. By applying this label extraction, users transfer aspects of their physical context to textual queries. This is similar to features found on platforms like Pinterest [61] or shopping interfaces like IKEA's smartphone app [54]. The items retrieved do not necessarily have to be the same artifact, but be similar to it, in terms of use-cases or domain. For a phone, the search may retrieve phone mounts or cases, which do not exhibit an identical geometry, but are meant to complement the geometry of the artifact.

Figure 9 depicts such a process. A user frames a watch (Fig. 9-1) and the resulting best guess is filled into the search field of ShapeFindAR (Fig. 9-2). Additional guesses are listed below and can likewise be used for a textual query (Fig. 9-2). While this is comparable to a coarse text search, it enables users to benefit from the machine's perception of the object or the scenery, ideally expanding the users' existing domain knowledge ad-hoc. This may include more precise terms (e.g., wristwatch instead of watch) or more general ones, in comparison to the user's wording. We consider this approach to be a feasible way to initiate a search with the goal of *browsing* a model repository. The results range from objects that are meant to *complement* the watch (e.g., a watch-stand – Fig.

<sup>15</sup> An example is Double Ring JVCR (22 mm x 2) by cepera3000 on Thingiverse, retrieved on 15.12.2021





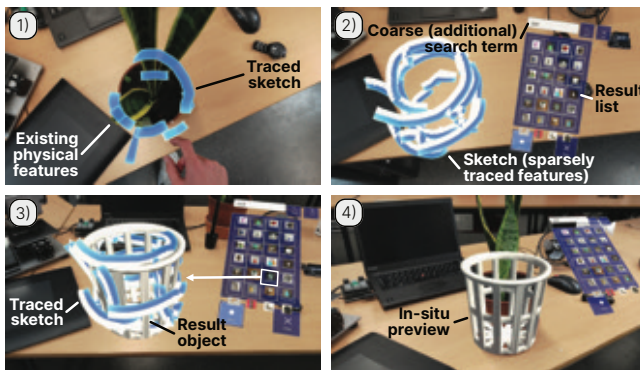
**Figure 9: Inspiration: Using the photo-based label extraction to receive inspiration for related items. 1) Framing a wristwatch as an object of interest; 2) choosing one of the suggested terms; 3) search results related, but not identical to the framed item; 4) in-situ previewing of the artifacts.**

9-3) or replacement parts for the watch (e.g., a watch strap – Fig. 9-4), all of which can be previewed in-situ.

#### 5.4 Scenario 4: In-situ Features

With ShapeFindAR, users may also trace features available in the vicinity. This enables users to transfer aspects of their physical context to spatial queries. The features can be either traced coarsely and serve as a basis for a sketch extending upon them, or be traced completely to search for (geometrically) similar artifacts.

A user may trace an existing flowerpot he wishes to replace or improve its design (e.g., by adding a decorative cachetop or planter to it). The ratio of the existing pot is important for this task, so the user tries to approximate the contours of the pot as closely as possible (Fig. 10-1). This does not necessarily leave a complete sketch of the pot. Afterwards, he fills some of the gaps of the outline to provide ShapeFindAR with volumetric features to work with (Fig.



**Figure 10: Tracing in-situ features for spatial searches. 1) coarsely tracing the coarse silhouette of the pot to be replaced; 2) removing the pot and completing the sketch; 3) submitting the query with a coarse search term, 2D result list is visible; 4) previewing in-situ.**

10-2). Along with the search term “planter”, the sketch search is submitted and yields decorative pots with comparable proportions to the sketch (and therefore to the original pot – Fig. 10-3). As the sketch was not entirely gap-free, a paper bin with sparse lines is among the first results (Fig. 10-3). Lastly, the user may move the 3D preview of the future artifact to cover the physical pot and evaluate aesthetics or apply modifications such as scaling (Fig. 10-4).

## 6 LIMITATIONS AND DISCUSSION

In this chapter, we want to discuss the conceptual and technical limitations of ShapeFindAR and the presented concept of in-situ spatial search.

### 6.1 Conceptual Limitations

Conceptually, it is important to consider the skill requirements and gaps associated with spatial sketching. Transferring partial requirements (e.g., tracing a physical object) to a satisfying sketch may prove hard for some users. We emphasize coarse approximation of shapes, but this may not suffice for complex queries, where for instance precise measurements are required. This is, to a degree, compensated by other modalities available to the users (i.e., text). However, this could potentially be overcome in the future with even better matching algorithms, which may consider more aspects of the environment. Alternatively, this may require ways to compensate for the lack of precision through automated corrections, or novel fabrication workflows [42, 55].

### 6.2 Technical Limitations

ShapeFindAR is, as a proof-of-concept, not without technical limitations. The ranking computed by the server is not impeccable, but under the candidates the system retrieves, fitting results can usually be found. The spatial search component is arguably slow, with one sketch-to-model comparison taking up to 2 seconds. This leads to request times of approximately up to 60 seconds. The more artifacts are considered to be feasible candidates, the longer the search may take. Accelerating the server would for instance require approaches such as shape histograms [28]. To scale ShapeFindAR to database sizes comparable to Thingiverse, far more optimizations and more aggressive filtering approaches are required. Unlike textual queries, sketch queries are far more individual and harder to cache or precompute. The ICP algorithm we use to align sketches and repository meshes, was originally meant to align point clouds originating from the same scene [67]. This is not the case with our combination, as a sketch merely approximates the target object. The approach of pre-filtering based on the OABB proved to be not ideal in some cases. If “flat” sketches are submitted (i.e., when the user merely traces an outline), similarly flat repository models are treated preferably. Automated extrusion would likely improve search performance and reduce the effort required from the users to fill volumes in their sketches. We argue that while there is room for improvement, coarse and ambiguous input, along with similarly coarse (but appropriate) results, may lead to more productive, creative explorations of the potential solutions to users’ requirements.

### 6.3 Opportunities of In-Situ Spatial Search

Regardless of the aforementioned limitations, we strongly believe that ShapeFindAR is a proof-of-concept that demonstrates the opportunities of in-situ spatial search for future physical artifacts. By abstracting away from terminology, users may express their desired artifacts in a modality that is arguably more natural and fitting: in 3 dimensions and in-situ, instead of ex-situ and converted to text and labels. In-situ spatial search enables **Term- and Function-Abstraction**. By ascribing terminology to the artifacts, they are prevented from appearing in searches not following this particular terminology (i.e., if novices to a domain are meant to express search terms). In-situ spatial search may also enable **scale-invariance** in a search process. With ShapeFindAR, we embrace the aspect that scale can be freely chosen by the user. This freedom enables scale-invariant searches. By normalizing all spatial input and output (while also focusing on proportions), ShapeFindAR sidesteps established conventions of size. Future implementations may also automatically scale objects to match the user's sketch, or use the sketch as a way to filter objects by size.

We argue that the "ideal search" for a physical artifact should happen in the most fitting (for the task) and easy (for the user) modality, and allow the combination of modalities to express partial requirements. There is no "one size fits all" approach to the search for physical artifacts, especially for personal use. A standardized artifact (e.g., an M3 screw) is easy to find, if the user knows the appropriate denomination (i.e., possesses the necessary domain knowledge). Likewise, *precise* measurements are easier to express through text (e.g., 4cm), compared to (spatial) sketches. As promising as a spatial-only search system may appear, it covers and excels at a *subset* of possible tasks. It also implicitly assumes that all users lack domain knowledge to express fitting search terms. If a user happens to possess the ability to express a precise search term in text, they may still achieve fitting results quickly. This requires systems to still offer textual means, which is what we did with ShapeFindAR, while also adding a transfer function from the physical context to textual queries (label extraction). However, proportions or geometry that are not standardized require other means to be encoded in a search, such as spatial sketches or 3D scans.

### 7 FUTURE WORK

We do not assume that 3D printing is the only domain ShapeFindAR may be applied to. Ideally, it may be a search and preview frontend to any database of physical artifacts. Databases that are domain-specific (e.g., furniture [36]) or general artifact databases (e.g., general shopping interfaces) could be addressed by ShapeFindAR and the concept of in-situ spatial search. With the capabilities of personal fabrication devices increasing, it may apply to any physical artifact in the (distant) future. Additionally, it is intriguing to explore users' strategies for in-situ spatial search in a task-oriented user study. Technical improvements to ShapeFindAR are possible in terms of processing speed and the addition of further input modalities. With devices like the HoloLens being outfitted with a depth sensor, more detailed interaction with the environment is a promising direction. For instance, one may scan existing artifacts and use them as search input (3D  $\rightarrow$  3D). This is fairly similar to the

label extraction approach (2D  $\rightarrow$  1D) present in ShapeFindAR, but abstracts even further from labels (*Term-Abstraction*) and emphasizes geometry instead (3D  $\rightarrow$  3D). This scan could also be altered with added sketch features or deformation. Silhouettes of artifacts are an intriguing aspect not fully considered in the ShapeFindAR prototype. Sketching outlines and applying basic operations like in revolving (for volumes with rotational symmetry), extrusion, or inflation (c.f., [22]) may be easier to execute than sketching the entire geometry (2D  $\rightarrow$  3D  $\rightarrow$  3D).

### 8 CONCLUSION

We presented the concept of in-situ spatial search and ShapeFindAR, a mixed reality search interface to a repository of 3D-printable models. The prototype system embeds multiple modalities for users to search for the physical artifact they desire: sketching shapes, photo-based search (i.e., label extraction), or textual search. The search itself, along with the object preview, happens in-situ, at the location of the future artifact, and happens iteratively, based on input gained from the search results and the physical environment. We specified this concept as in-situ spatial search, and described conceptual advantages, along with walkthroughs possible with ShapeFindAR. We consider this work to be a step towards lower-effort interfaces for personal fabrication, which may emerge more as sophisticated shopping interfaces [52], and less as simplified interfaces for (3D) modeling.

With this work, we argue for better ways for novices and even consumers (who are not involved in personal fabrication) to search and preview future physical artifacts. While users with intrinsic motivation for fabrication are willing to invest time in learning and designing, the majority of people can be considered "consumers" [52]. They are willing to benefit from unique, personal artifacts, but not willing to invest time in the process. For them, novel, low-effort, means for artifact *retrieval*, over artifact *modeling*, are required. The process of retrieval, in turn, should not enforce the use of a specific *language* to formulate queries and demand as few transfers between (future) artifact context and the search interface as possible.

### ACKNOWLEDGMENTS

We thank Thingiverse, MyMiniFactory, and their contributors for providing access to such vast amounts of diverse models. We furthermore thank our anonymous reviewers, who have supported us in improving the manuscript.

### REFERENCES

- [1] Harshit Agrawal, Udayan Umapathi, Robert Kovacs, Johannes Frohnhofen, Hsiang-Ting Chen, Stefanie Mueller, and Patrick Baudisch. 2015. Prototyper: Physically Sketching Room-Sized Objects at Actual Scale. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 427–436. <https://doi.org/10.1145/2807442.2807505>
- [2] Celena Alcock, Nathaniel Hudson, and Parmit K. Chilana. 2016. Barriers to Using, Customizing, and Printing 3D Designs on Thingiverse. In *Proceedings of the 19th International Conference on Supporting Group Work (GROUP '16)*. ACM, New York, NY, USA, 195–199. <https://doi.org/10.1145/2957276.2957301>
- [3] K. S. Arun, T. S. Huang, and S. D. Blostein. 1987. Least-Squares Fitting of Two 3-D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-9, 5 (Sept. 1987), 698–700. <https://doi.org/10.1109/TPAMI.1987.4767965>
- [4] Daniel Ashbrook, Shitao Stan Guo, and Alan Lambie. 2016. 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)*. ACM, New York, NY, USA, 1510–1518. <https://doi.org/10.1145/2851581.2892509>

- [5] Rafael Ballagas, Jishang Wei, Mithra Vankipuram, Zhiyuan Li, Keanu Spies, and Hiroshi Horii. 2019. Exploring Pervasive Making Using Generative Modeling and Speech Input. *IEEE Pervasive Computing* 18, 4 (Oct. 2019), 20–28. <https://doi.org/10.1109/MPRV.2019.2929130>
- [6] Patrick Baudisch, Arthur Silber, Yannis Kommanas, Milan Gruner, Ludwig Wall, Kevin Reuss, Lukas Heilman, Robert Kovacs, Daniel Rechlitz, and Thijs Roumen. 2019. Kyub: A 3D Editor for Modeling Sturdy Laser-Cut Objects. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, Glasgow, Scotland Uk, 1–12. <https://doi.org/10.1145/3290605.3300796>
- [7] Erin Buehler, Stacy Branham, Abdullah Ali, Jeremy J. Chang, Megan Kelly Hofmann, Amy Hurst, and Shaun K. Kane. 2015. Sharing Is Caring: Assistive Technology Designs on Thingiverse. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*. ACM Press, Seoul, Republic of Korea, 525–534. <https://doi.org/10.1145/2702123.2702525>
- [8] Benjamin Bustos, Daniel A. Keim, Dietmar Saupe, Tobias Schreck, and Dejan V. Vranić. 2005. Feature-Based Similarity Search in 3D Object Databases. *ACM Computing Surveys (CSUR)* 37, 4 (Dec. 2005), 345–387. <https://doi.org/10.1145/1118890.1118893>
- [9] Michael Dawson-Haggerty. 2020. Trimesh.
- [10] Kapil Dev, Nicolas Villar, and Manfred Lau. 2016. StandUp: Understanding Body-part and Gestural Preferences for First-person 3D Modeling. In *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering (Expresive '16)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 97–102.
- [11] Mathias Eitz, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. 2010. Sketch-Based 3D Shape Retrieval. In *ACM SIGGRAPH 2010 Talks (SIGGRAPH '10)*. Association for Computing Machinery, Los Angeles, California, 1. <https://doi.org/10.1145/1837026.1837033>
- [12] M.J. Enenhofer. 1996. Spatial-Query-by-Sketch. In *Proceedings 1996 IEEE Symposium on Visual Languages*. IEEE, New York, NY, USA, 60–67. <https://doi.org/10.1109/VL.1996.545269>
- [13] Christoph M. Flath, Sascha Friesike, Marco Wirth, and Frédéric Thiesse. 2017. Copy, Transform, Combine: Exploring the Remix as a Form of Innovation. *Journal of Information Technology* 32, 4 (Dec. 2017), 306–325. <https://doi.org/10.1057/s41265-017-0043-9>
- [14] Sean Follmer, David Carr, Emily Lovell, and Hiroshi Ishii. 2010. CopyCAD: Remixing Physical Objects with Copy and Paste from the Real World. In *Adjunct Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, New York, NY, USA, 381–382. <https://doi.org/10.1145/1866218.1866230>
- [15] C. Allie Fraser, Julia M. Markel, N. James Basa, Mira Dontcheva, and Scott Klemmer. 2020. ReMap: Lowering the Barrier to Help-Seeking with Multimodal Search. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. ACM, Virtual Event USA, 979–986. <https://doi.org/10.1145/3379337.3415592>
- [16] Daniele Giunchi, Alejandro Sztajman, Stuart James, and Anthony Steed. 2021. Mixing Modalities of 3D Sketching and Speech for Interactive Model Retrieval in Virtual Reality. In *ACM International Conference on Interactive Media Experiences*. ACM, Virtual Event USA, 144–155. <https://doi.org/10.1145/3452918.3458806>
- [17] Ammar Hattab and Gabriel Taubin. 2019. Interactive Fabrication of CSG Models with Assisted Carving. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '19)*. ACM, New York, NY, USA, 677–682. <https://doi.org/10.1145/3294109.3295644>
- [18] Ammar Hattab and Gabriel Taubin. 2019. Rough Carving of 3D Models with Spatial Augmented Reality. In *Proceedings of the ACM Symposium on Computational Fabrication (SCF '19)*. ACM, New York, NY, USA, 4:1–4:10. <https://doi.org/10.1145/3328939.3328998>
- [19] Megan Hofmann, Gabriella Hann, Scott E. Hudson, and Jennifer Mankoff. 2018. 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)*. ACM, New York, NY, USA, 301:1–301:12. <https://doi.org/10.1145/3173574.3173875>
- [20] Christian Holz and Andrew Wilson. 2011. Data Miming: Inferring Spatial Object Descriptions from Human Gesture. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 811–820. <https://doi.org/10.1145/1978942.1979060>
- [21] Nathaniel Hudson, Celena Alcock, and Parmit K. Chilana. 2016. Understanding Newcomers to 3D Printing: Motivations, Workflows, and Barriers of Casual Makers. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 384–396. <https://doi.org/10.1145/2858036.2858266>
- [22] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. 1999. Teddy: A Sketching Interface for 3D Freeform Design. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., USA, 409–416. <https://doi.org/10.1145/311535.311602>
- [23] Yunwoo Jeong, Han-Jong Kim, and Tek-Jin Nam. 2018. Mechanism Perfboard: An Augmented Reality Environment for Linkage Mechanism Design and Fabrication. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, 411. <https://doi.org/10.1145/3173574.3173985>
- [24] Gabe Johnson, Mark Gross, Ellen Yi-Luen Do, and Jason Hong. 2012. Sketch It, Make It: Sketching Precise Drawings for Laser Cutting. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems (CHI EA '12)*. ACM, New York, NY, USA, 1079–1082. <https://doi.org/10.1145/2212776.2212390>
- [25] Rubaiat Habib Kazi, Tovi Grossman, Hyunmin Cheong, Ali Hashemi, and George Fitzmaurice. 2017. DreamSketch: Early Stage 3D Design Explorations with Sketching and Generative Design. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. Association for Computing Machinery, New York, NY, USA, 401–414. <https://doi.org/10.1145/3126594.3126662>
- [26] Hyosun Kim, Georgia Albuquerque, Sven Havemann, and Dieter W. Fellner. 2005. Tangible 3D: Hand Gesture Interaction for Immersive 3D Modeling. In *Proceedings of the 11th Eurographics Conference on Virtual Environments (EGVE'05)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 191–199. [https://doi.org/10.2312/EGVE/IPT\\_EGVE2005/191-199](https://doi.org/10.2312/EGVE/IPT_EGVE2005/191-199)
- [27] Jeeun Kim, Anhong Guo, Tom Yeh, Scott E. Hudson, and Jennifer Mankoff. 2017. 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)*. ACM, New York, NY, USA, 1067–1078. <https://doi.org/10.1145/3064663.3064690>
- [28] Hans-Peter Kriegel, Peer Kröger, Zahi Mashaël, Martin Pfeifle, Marco Pötke, and Thomas Seidl. 2003. Effective Similarity Search on Voxelized CAD Object.. In *Proceedings of the Eighth International Conference on Database Systems for Advanced Applications*. 27. <https://doi.org/10.1109/DASFAA.2003.1192365>
- [29] Harris Kyriakou, Jeffrey Nickerson, and Gaurav Sabnis. 2017. Knowledge Reuse for Customization: Metamodels in an Open Design Community for 3D Printing. *MIS Quarterly* 41 (March 2017), 315–332. <https://doi.org/10.25300/MISQ/2017/41.1.17>
- [30] Roberto Lam and J. M. Hans du Buf. 2011. Using Mathematical Morphology for Similarity Search of 3D Objects. In *Pattern Recognition and Image Analysis (Lecture Notes in Computer Science)*, Jordi Vitrià, João Miguel Sanches, and Mario Hernández (Eds.). Springer, Berlin, Heidelberg, 411–419. [https://doi.org/10.1007/978-3-642-21257-4\\_51](https://doi.org/10.1007/978-3-642-21257-4_51)
- [31] Manfred Lau, Jun Mitani, and Takeo Igarashi. 2012. Sketching and Prototyping Personalised Objects: From Teapot Lids to Furniture to Jewellery. *National Conference on Rapid Design, Prototyping & Manufacture* (2012), 8.
- [32] Bokyoung Lee, Minjoo Cho, Joonhee Min, and Daniel Saakes. 2016. Posing and Acting As Input for Personalizing Furniture. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction (NordCHI '16)*. ACM, New York, NY, USA, 44:1–44:10. <https://doi.org/10.1145/2971485.2971487>
- [33] Bokyoung Lee, Gyeol Han, Jundong Park, and Daniel Saakes. 2017. Consumer to Creator: How Households Buy Furniture to Inform Design and Fabrication Interfaces. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 484–496. <https://doi.org/10.1145/3025453.3025666>
- [34] Changjian Li, Hao Pan, Adrien Bousseau, and Niloy J. Mitra. 2020. Sketch2CAD: Sequential CAD Modeling by Sketching in Context. *arXiv:2009.04927 [cs]* (Sept. 2020). [arXiv:2009.04927 \[cs\]](https://arxiv.org/abs/2009.04927)
- [35] Jiahao Li, Jeeun Kim, and Xiang 'Anthony' Chen. 2019. Robiot: A Design Tool for Actuating Everyday Objects with Automatically Generated 3D Printable Mechanisms. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New Orleans, LA, USA, 673–685. <https://doi.org/10.1145/3332165.3347894>
- [36] Joseph J. Lim, Hamed Pirsiavash, and Antonio Torralba. 2013. Parsing IKEA Objects: Fine Pose Estimation. In *Proceedings of the IEEE International Conference on Computer Vision*. 2992–2999.
- [37] Chandan Mahapatra, Jonas Kjeldmand Jensen, Michael McQuaid, and Daniel Ashbrook. 2019. Barriers to End-User Designers of Augmented Fabrication. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, 383:1–383:15. <https://doi.org/10.1145/3290605.3300613>
- [38] James McCrae, Nobuyuki Umetani, and Karan Singh. 2014. FlatFitFab: Interactive Modeling with Planar Sections. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 13–22. <https://doi.org/10.1145/2642918.2647388>
- [39] Huaishu Peng, Jimmy Briggs, Cheng-Yao Wang, Kevin Guo, Joseph Kider, Stefanie Mueller, Patrick Baudisch, and François Guimbretière. 2018. RoMA: Interactive Fabrication with Augmented Reality and a Robotic 3D Printer. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, 579:1–579:12. <https://doi.org/10.1145/3173574.3174153>
- [40] Physna Inc. 2021. Thangs - 3D Model Community. Search & Download Free 3D Models. Share 3D Models. <https://thangs.com/>. (Accessed 07.04.2021).
- [41] Jiantao Pu, Kuiyang Lou, and Karthik Ramani. 2005. A 2D Sketch-Based User Interface for 3D CAD Model Retrieval. *Computer-Aided Design and Applications* 2, 6 (Jan. 2005), 717–725. <https://doi.org/10.1080/16864360.2005.10738335>



- [42] Thijs Roumen, Jotaro Shigeyama, Julius Cosmo Romeo Rudolph, Felix Grzelka, and Patrick Baudisch. 2019. SpringFit: Joints and Mounts That Fabricate on Any Laser Cutter. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. ACM, New Orleans, LA, USA, 12. <https://doi.org/10.1145/3332165.3347930>
- [43] Thijs Jan Roumen, Willi Müller, and Patrick Baudisch. 2018. Grafter: Remixing 3D-Printed Machines. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, 63:1–63:12. <https://doi.org/10.1145/3173574.3173637>
- [44] S. Rusinkiewicz and M. Levoy. 2001. Efficient Variants of the ICP Algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*. IEEE Comput. Soc, Quebec City, Que., Canada, 145–152. <https://doi.org/10.1109/IM.2001.924423>
- [45] Greg Saul, Manfred Lau, Jun Mitani, and Takeo Igarashi. 2011. SketchChair: An All-in-one Chair Design System for End Users. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '11)*. ACM, New York, NY, USA, 73–80. <https://doi.org/10.1145/1935701.1935717>
- [46] Valkyrie Savage, Sean Follmer, Jingyi Li, and Björn Hartmann. 2015. Makers' Marks: Physical Markup for Designing and Fabricating Functional Objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 103–108. <https://doi.org/10.1145/2807442.2807508>
- [47] Maria Shugrina, Ariel Shamir, and Wojciech Matusik. 2015. Fab Forms: Customizable Objects for Fabrication with Validity and Geometry Caching. *ACM Transactions on Graphics* 34, 4 (July 2015), 100:1–100:12. <https://doi.org/10.1145/2766994>
- [48] Alexa F. Siu, Son Kim, Joshua A. Miele, and Sean Follmer. 2019. shapeCAD: An Accessible 3D Modelling Workflow for the Blind and Visually-Impaired Via 2.5D Shape Displays. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, Pittsburgh PA USA, 342–354. <https://doi.org/10.1145/3308561.3353782>
- [49] Pedro Sousa and Manuel J. Fonseca. 2010. Sketch-Based Retrieval of Drawings Using Spatial Proximity. *Journal of Visual Languages & Computing* 21, 2 (April 2010), 69–80. <https://doi.org/10.1016/j.jvlc.2009.12.001>
- [50] Evgeny Stemasov. 2021. Enabling Ubiquitous Personal Fabrication by Deconstructing Established Notions of Artifact Modeling. In *The Adjunct Publication of the 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21)*. Association for Computing Machinery, New York, NY, USA, 166–170. <https://doi.org/10.1145/3474349.3477589>
- [51] Evgeny Stemasov, Alexander Botner, Enrico Rukzio, and Jan Gugenheimer. 2022. Ephemeral Fabrication: Exploring a Ubiquitous Fabrication Scenario of Low-Effort, In-Situ Creation of Short-Lived Physical Artifacts. In *Sixteenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '22)*. Association for Computing Machinery, New York, NY, USA, 1–17. <https://doi.org/10.1145/3490149.3501331>
- [52] Evgeny Stemasov, Enrico Rukzio, and Jan Gugenheimer. 2021. The Road to Ubiquitous Personal Fabrication: Modeling-Free Instead of Increasingly Simple. *IEEE Pervasive Computing* 20, 1 (2021), 1–9. <https://doi.org/10.1109/MPRV.2020.3029650>
- [53] Evgeny Stemasov, Tobias Wagner, Jan Gugenheimer, and Enrico Rukzio. 2020. Mix&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)*. Association for Computing Machinery, Honolulu, HI, USA, 1–12. <https://doi.org/10.1145/3313831.3376839>
- [54] Liz Stinson. 2018. Ikea's AR App Now Lets You Search with Your Phone Camera. <https://www.curbed.com/2018/3/22/17149082/ikeas-place-app-update-ar-grokstyle-visual-search>. (Accessed 16.09.2020).
- [55] Lingyun Sun, Yue Yang, Yu Chen, Jiaji Li, Guanyun Wang, Ye Tao, and Lining Yao. 2020. ShrinkyKit: 3D Printing Shrinkable Adaptations for Everyday Objects. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA '20)*. Association for Computing Machinery, Honolulu, HI, USA, 1–7. <https://doi.org/10.1145/3334480.3383034>
- [56] Hans Tangelder and Remco Veltkamp. 2004. A Survey of Content Based 3D Shape Retrieval Methods. In *Multimedia Tools and Applications*, Vol. 39. 145–156. <https://doi.org/10.1109/SML.2004.1314502>
- [57] Paula Te. 2015. TADCAD: A Tangible and Gestural 3D Modeling & Printing Platform for Building Creativity. In *Proceedings of the 14th International Conference on Interaction Design and Children (IDC '15)*. ACM, New York, NY, USA, 406–409. <https://doi.org/10.1145/2771839.2771865>
- [58] Rundong Tian, Vedant Saran, Mareike Kritzler, Florian Michahelles, and Eric Paulos. 2019. Turn-by-Wire: Computationally Mediated Physical Fabrication. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New Orleans, LA, USA, 713–725. <https://doi.org/10.1145/3332165.3347918>
- [59] Rundong Tian, Sarah Sterman, Ethan Chiou, Jeremy Warner, and Eric Paulos. 2018. MatchSticks: Woodworking Through Improvisational Digital Fabrication. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, 149:1–149:12. <https://doi.org/10.1145/3173574.3173723>
- [60] Ana M Villanueva, Ziyi Liu, Zhengzhe Zhu, Xin Du, Joey Huang, Kylie A Peeper, and Karthik Ramani. 2021. RobotAR: An Augmented Reality Compatible Teleconsulting Robotics Toolkit for Augmented Makerspace Experiences. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–13. <https://doi.org/10.1145/3411764.3445726>
- [61] Kurt Wagner. 2017. Pinterest Will Now Let You Search for Products Using Any Image You Find Online — without Visiting Pinterest. <https://www.vox.com/2017/3/7/14837702/pinterest-chrome-extension-image-search>. (Accessed 12.09.2020).
- [62] Christian Weichel, Manfred Lau, David Kim, Nicolas Villar, and Hans W. Gellersen. 2014. MixFab: A Mixed-reality Environment for Personal Fabrication. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 3855–3864. <https://doi.org/10.1145/2556288.2557090>
- [63] Kun Xu, Kang Chen, Hongbo Fu, Wei-Lun Sun, and Shi-Min Hu. 2013. Sketch2Scene. *ACM Transactions on Graphics (TOG)* (July 2013).
- [64] Tom Yeh and Jeeun Kim. 2018. CraftML: 3D Modeling Is Web Programming. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, Montreal QC, Canada, 1–12. <https://doi.org/10.1145/3173574.3174101>
- [65] Ya-Ting Yue, Xiaolong Zhang, Yongliang Yang, Gang Ren, Yi-King Choi, and Wenping Wang. 2017. WireDraw: 3D Wire Sculpturing Guided with Mixed Reality. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3693–3704. <https://doi.org/10.1145/3025453.3025792>
- [66] Amanda K. Yung, Zhiyuan Li, and Daniel Ashbrook. 2018. Printy3D: In-situ Tangible Three-dimensional Design for Augmented Fabrication. In *Proceedings of the 17th ACM Conference on Interaction Design and Children (IDC '18)*. ACM, New York, NY, USA, 181–194. <https://doi.org/10.1145/3202185.3202751>
- [67] Zhengyou Zhang. 1994. Iterative Point Matching for Registration of Free-Form Curves and Surfaces. *International Journal of Computer Vision* 13, 2 (Oct. 1994), 119–152. <https://doi.org/10.1007/BF01427149>
- [68] Qingnan Zhou and Alec Jacobson. 2016. Thing10K: A Dataset of 10,000 3D-Printing Models. *arXiv:1605.04797 [cs]* (July 2016). [arXiv:1605.04797 \[cs\]](https://arxiv.org/abs/1605.04797)
- [69] Kening Zhu, Alexandru Dancu, and Shengdong (Shen) Zhao. 2016. FusePrint: A DIY 2.5D Printing Technique Embracing Everyday Artifacts. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems (DIS '16)*. ACM, New York, NY, USA, 146–157. <https://doi.org/10.1145/2901790.2901792>