
pART bench: a Hybrid Search Tool for Floor Plans in Architecture

Veronika Krauß
Ekaterina Fuchkina
Gabriela C. Molina León
Oana-Iuliana Popescu
Florian Echtler
Sven Bertel

Bauhaus-Universität Weimar
Weimar, Germany
firstname.lastname
@uni-weimar.de

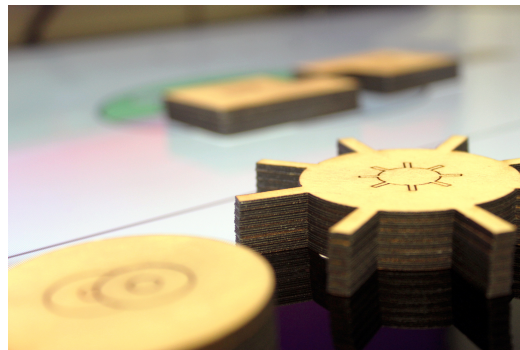


Figure 1: The set of tangibles for the tabletop prototype.

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.
Copyright is held by the owner/author(s).
ITS '15, November 15-18, 2015, Funchal, Portugal.
ACM 978-1-4503-3899-8/15/11.
<http://dx.doi.org/10.1145/2817721.2823491>

Abstract

Architectural databases often contain thousands of different floor plans which have either been collected from historical designs or, more recently, auto-generated by suitable algorithms. Searching for a floor plan that fits specific requirements in such a database involves setting a large number of parameters, such as lines of sight, lighting levels, room types and many more.

We present *pART bench*, a hybrid tabletop/tablet tool which allows the use of intuitive touch commands and tangible objects to quickly adjust search parameters, view resulting floor plans and iteratively refine the search. We report on a comprehensive requirements analysis with practising architects, on the design process, and describe our prototypical implementation of the system, both on a tablet and on a PixelSense tabletop device.

ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation (e.g. HCI)]:
User Interfaces

Author Keywords

tablet; touchscreen; tabletop; tangibles; architecture; search

Introduction

Architectural design processes often make ample use of existing designs, either for inspiration, exploring possible design variants or constraints, or allowing for a reuse of previously found design solutions. Traditionally, such information is accessed through large collections in books. Examples are Neufert's *Architects' Data* [9] or *Raumpilot* [4]. They are complemented by collections held as digital databases, such as *ar:searchbox* [6] or *Detail* [2]. More recently, it is not just information about historical designs that is being used, but also about designs which have been auto-generated by suitable algorithms [8]. Collections are typically very large and include large numbers of design parameters, such as relating to room types and sizes, lighting levels, spatial constraints of occupants' movements, etc.

Whereas learning to productively search in book-bound collections for existing designs fitting one's own design brief has traditionally been a part of architects' education, the usually very large number of designs held in more recent collections in databases requires specific types of interfaces for a productive use. To this end, we have designed *pART bench*, a hybrid tabletop/tablet tool which allows the use of intuitive touch commands and tangible objects to quickly adjust search parameters, view resulting floor plans and iteratively refine the search. The acronym *pART* stands for *parametric Architecture Retrieval Tool* while the *bench* extension is a reference to the concept of a *workbench* and expresses the project's aim to create a retrieval tool that frictionlessly fits into the established workflow of architects during design. In the following, we will report on a comprehensive requirements analysis with practicing architects, on the design process, and describe our prototypical implementation of the system on a PixelSense tabletop device and a tablet.

Related Work

One of the best-known examples of bringing tangible interaction into the architectural domain is *URP* by Underkoffler et al. [15]. *URP* provides tangible building models which can be repositioned on a tabletop surface to observe changing effects of sunlight and shadows over the course of a day; it is best characterized as a planning tool.

Most other research at the intersection of architecture and human-computer interaction also focuses on tools to support the planning process. Examples include *Urban Sketcher* by Sareika et al. [12], an augmented reality application for urban planning; the *Tangible 3D urban simulation table* by Salim et al. [11], an interactive tabletop system for wind simulation in urban settings; or *Illuminating Clay* by Piper et al. [10], a clay-based tangible landscaping simulation.

A different focus is set by *CubeExplorer* by Song et al. [13], a tangible interface to support the teaching of fundamental concepts of architecture — in this case, the concept of negative space or void.

When going beyond the specific application domain of architecture, the topic of enhancing general search tasks with tangible interfaces should not be overlooked. A recent example is *FacetStreams* by Jetter et al. [3], a hybrid touch/tangible interface which allows users to adjust a wide range of parameters relating to a product search using tangibles and to combine different search constraints through Boolean operations. A similar system, *CubeQuery* by Langner et al. [7], uses active tangibles based on Sifteo cubes to select search terms for a music collection. In a more general context, Ullmer et al. have investigated generic database search tasks with tangible input devices [14].

Initial Prototype

In this section, we will first outline some basic considerations that drove our design process before describing our investigation of architects' requirements for tools that support their floor plan retrieval from digital databases.

Design Concept

In order to offer a more efficient and interesting search tool for floor plans than atlas or architectural blogs, a proper and figural depiction of the retrieval parameters and the values setting was needed, such that the user would be able to directly understand the underlying metaphors. It was also crucial not to overload the interface with information, while still being sufficiently precise in the parametrization. During the early stages of design, we took the decision to employ pie menus [1] for the parametrization of queries. This was driven by the observation that they allow a quick interaction and are easy to understand.

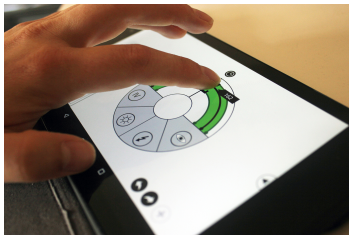


Figure 2: Setting of the preferred value for the square meter parameter on a Nexus tablet. The filled, green semicircle represents the admissible parameter range for floor plan search, while the black line running inside it represents the preferred parameter value.

A visualization with a high level of abstraction from the floor plan structure, especially from its geometric appearance, was desired. Our selected visualization excludes geometric elements that could resemble room or flat shapes (e.g., squares). This is also why pie menus with their circular shape seemed a good choice. Furthermore, the segments of a pie can directly represent set parameters and their values are represented through segment filling levels. When interviewed, architecture experts stated that they appreciated our approach when compared to querying by sketches: because of the pies' abstraction from a concrete geometry, the architects can focus on topology and other parameters. The concrete geometry would be visible in the retrieved floor plans that result from the search.

To guarantee a structured line of actions during the search and to provide a better overview, we introduced two different parametrization levels: a flat level, with only one pie for the

entire flat or floor plan, and a room level, on which users can create any number of pies, one per room in a flat, with individually set parameters. Existing pies, including their segments, can be copied to speed up the parameter setting. Connections between rooms are visualized by lines, which, again, can be parametrized. This representation was chosen because it resembles a graph and is easily understood.

Formative Evaluation

It was decided to first endeavour on further understanding the work flow and work processes of our target group. This included establishing architects' familiarity with various devices and tools, and their willingness to adapt to new technologies. Available test participants were allocated to one of two groups according to their architectural expertise: (1) students of architecture in Bachelor's and Master's degree programs, who were found to be more likely to query in a creative way, to explore and learn about possible design solutions; (2) professional researchers in architecture (either professors or advanced Ph.D. students, most of whom were also practicing architects, at least part-time), who were found to query more often task-based (i.e., directly for floor plans with specific parameters). The students were treated as non-experts with a higher affinity to new devices and novel ideas, whereas the researchers were treated as specialists and were expected to possibly give deeper insights into work processes.

Different methods such as questionnaires and interviews were used to collect data and review our preliminary assumptions about the design process of the architect and our design. Assumptions included benefits of non-geometry based querying with pies, how the users would interact with the tool and their expectations regarding retrieval tools. 14 architecture students completed the online questionnaire; five experts were invited to semi-structured interviews during

which their usual work flow as well as design ideas and first prototypes were discussed more detailed. The main topics were commonly used floor plan sources, frequency of use of touch devices, various ideas of how tangible interfaces might be used during floor plan retrieval, and, of course, details of the architects' work process.

Based on the results, we established that floor plan search is mostly used in the design phase of the work flow. However, such search is not performed very frequently, but rather selectively. The main resources are commonly known floor plan atlases and (more rarely) architectural blogs (because only few digital sources are available). Architects prefer working at desktop computers or on touch devices, as such hardware is already accessible; they were, however, very open to the idea of using tangibles for further interaction if their use would speed up the search or simply improve user experience. The most common expectation of the participants regarding the tangible use was that the tangibles should represent the search parameters (e.g., room size) in a metaphorical way. They also stated that the tangibles should improve the attractiveness and playfulness of the tool.

Based on these results, a paper prototype for touch devices was created, that evolved later on into a prototype implementation supporting touch and tangible interaction. Since the floor plan search is not done very often, it was necessary to create a self explanatory interface that does not require to learn and memorize interaction metaphors. In consequence, the visual scheme of the buttons was developed to signal possible interaction gestures. This design is consistent throughout the interface by associating similar interactions with similar designs. For example, every button that needs to be tapped is black, while every button that needs to be dragged is white. The prototype was, again, tested with groups of varying architectural expertise, from

students to experts; we also included a mobile interaction expert as a test person.

During the paper prototype test, the think-aloud technique was used. After the test, participants answered additional questions about their experience while interacting with the prototype. The tests revealed that the visual scheme was largely ignored by users. They applied their preconceived idea of gestures, regardless of the scheme. Nevertheless, the test results achieved by the paper prototype tests were satisfying.

pART bench

In this section, we will link the requirements derived from the formative evaluation with our prototypical implementation of *pART bench* for use with touch and tangibles.

Requirements & Implementation

Supporting a range of touch devices is important for developers and users. Therefore, we take into consideration tablets (starting from 7" screen size) as well as large screen devices, including tabletops. To allow the user to perform actions in zoomed-in mode in case the screen is too small, we introduce different zoom levels of interaction. On the other hand, large devices such as tabletops provide not only a larger workspace, but also the possibility to use tangibles. To support general touch-enabled devices, the interface shown in figure 2 was developed. By means of standard tap and drag commands, users can create new pies, assign values and permissible ranges for one or more parameters (currently including area, solar gain, exterior view, energy and room type) and create connections between room pies.

In order to take advantage of the size and the capacities afforded by a tabletop device, we included a supplementary tangible interaction mode that includes touch with new tangible gestures for a subset of the actions. Although both

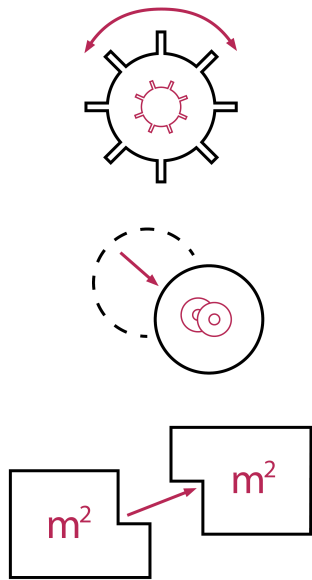


Figure 3: Schematic interactions type with tangibles: Turn the sun to set the solar gain. Use the pie stamp to create new pies. Move apart the area tangible to set the flat size.

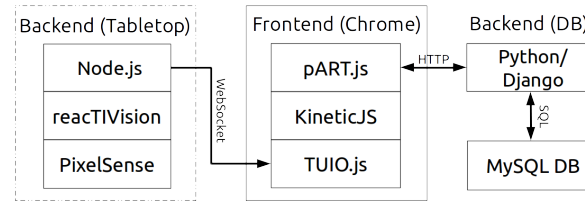


Figure 4: System architecture

versions provide the same functionalities, the main goal of this version is to provide an additional level of comfort, speed, novelty as well as fun during the work process. While the pie menus themselves still provide visual feedback of the parameter values, their creation and adjustment moved to the physical level as figural tangibles. Examples for these tangibles are the "pie stamp" for creating new pie menus, the sun symbol for solar gain and the dual square tangibles for setting the area (see also figure 3 and 1).

On the other hand, interaction should still be as simple and intuitive as possible and introduction or substitution of new methods should indeed provide interaction improvements. As tangibles exist in the real world, we enable architects to transfer their experience of working with physical tools onto the floor plan search process. For example, to set the area parameter, we use two tangibles that can be moved relatively to each other. Their L-shaped form implicitly creates a rectangle whose diagonal length is mapped to the area value. This resembles the real world action of people indicating size or area by moving their hands apart.

System Architecture

pART bench is designed as a web application to allow easy switching between different device classes. The frontend is structured around the Javascript frameworks PrototypeJS and KineticJS and currently runs on the Chrome browser

which is available on a multitude of platforms, while the backend is based on the Python framework Django, which connects to a MySQL database containing the actual floor plans (see figure 4).

This section of the overall system architecture is already sufficient for deploying *pART bench* on a wide range of devices such as tablets or laptops. However, to support tangible interaction on the interactive surface, we require a secondary software stack running on the PixelSense device. The raw video from the touch screen surface is processed by reactTIVision [5] to detect the common "Amoeba" fiducial markers. Data about marker IDs and positions is first sent via UDP in the widely used TUIO format, which is then adapted to a WebSocket transport by an intermediate layer based on node.js and finally received and interpreted by the TUIO.js library within the browser frontend. To support transparent handling of different input event types (touch/tangible), minor internal changes in KineticJS were required.

Conclusion & Outlook

We have presented *pART bench*, a hybrid touch/tangible tool to search databases of floor plans for architectural design. While our current setup is still in the prototype stage, we believe it is already capable of being used in a research context for exploratory search.

The pie model enables the user to search for specific floor plans with concrete geometries through an abstract graphical representation. The touch component provides search mobility between devices with varying sizes. The tangible component extends the touch interactions through physical, figural, and analogous representations of the most important query components.

We current plan further prototype evaluation with the group of architects who participated in the initial user study. We also

plan to investigate an extension of our setup using active tangibles such as Sifteo cubes, similar to CubeQuery. [7].

Acknowledgments

This work was in part funded by [BMBF](#) through grant T_209_M_SB. Thanks to Felicitas Höbelt, Aaron Solbach and Georg Erfurt as members of the original pART project, and to Anna Rack, Marie Bornemann, Adrian Kreskowski, Jannis Bossert, Jakob Wagner and Stefanie Wetzel for support in the lab and helpful discussions.

REFERENCES

1. J. Callahan, D. Hopkins, M. Weiser, and B. Shneiderman. 1988. An Empirical Comparison of Pie vs. Linear Menus, In Proc. CHI '88. ACM, New York, NY, USA, 95–100.
2. Institut für internationale Architektur-Dokumentation. 2015. DETAIL. (2015). <http://www.detail-online.com/>.
3. H.-C. Jetter, J. Gerken, M. Zöllner, H. Reiterer, and N. Milic-Frayling. 2011. Materializing the Query with Facet-streams: A Hybrid Surface for Collaborative Search on Tabletops, In Proc. CHI '11. ACM, New York, NY, USA, 3013–3022.
4. T. Jocher, S. Loch, A. Lederer, B. Pampe, M. Gasser, C. zur Brügge, M. Tvrtkovic, W. Stamm-Teske, K. Fischer, and T. Haag. 2010. *Raumpilot: 4 Bände*. Wüstenrot Stiftung.
5. M. Kaltenbrunner and R. Bencina. 2007. reactIVision: A Computer-vision Framework for Table-based Tangible Interaction, In Proc. TEI '07. ACM, New York, NY, USA, 69–74.
6. C. Langenhan, A. Seifert, A. Teichert, and F. Petzold. ar:searchbox - Knowledge Management for Architecture Students, In Proc. eCAADe '12. eCAADe, 639–645.
7. R. Langner, A. Augsburg, and R. Dachsel. 2014. CubeQuery: Tangible Interface for Creating and Manipulating Database Queries, In Proc. ITS '14. ACM, New York, NY, USA, 423–426.
8. P. Merrell, E. Schkufza, and V. Koltun. 2010. Computer-generated Residential Building Layouts, In Proc. SIGGRAPH ASIA '10. ACM, New York, NY, USA, Article 181, 12 pages.
9. E. Neufert, P. Neufert, and J. Kister. 2012. *Architects' Data*. John Wiley & Sons.
10. B. Piper, C. Ratti, and H. Ishii. 2002. Illuminating Clay: A 3-D Tangible Interface for Landscape Analysis, In Proc. CHI '02. ACM, New York, NY, USA, 355–362.
11. F. Salim. 2014. Tangible 3D Urban Simulation Table, In Proc. SIMAUD '14. Society for Computer Simulation International, San Diego, CA, USA, Article 23, 4 pages.
12. M. Sareika and D. Schmalstieg. 2007. Urban Sketcher: Mixed Reality on Site for Urban Planning and Architecture, In Proc. ISMAR '07. IEEE Computer Society, Washington, DC, USA, 1–4.
13. H. Song, F. Guimbretière, M. A. Ambrose, and C. Lostritto. 2007. Cubeexplorer: An Evaluation of Interaction Techniques in Architectural Education, In Proc. INTERACT'07. Springer-Verlag, Berlin, Heidelberg, 43–56.
14. B. Ullmer, H. Ishii, and R. J. K. Jacob. 2003. Tangible Query Interfaces: Physically Constrained Tokens for Manipulating Database Queries, In Proc. INTERACT '03. IOS Press, 279–286.
15. J. Underkoffler and H. Ishii. 1999. Urp: A Luminous-tangible Workbench for Urban Planning and Design, In Proc. CHI '99. ACM, New York, NY, USA, 386–393.