Supporting The Design and Fabrication of Physical Visualizations

Saiganesh Swaminathan[1]

Pierre Dragicevic[1]

Conglei Shi[2]

Lora Oehlberg[1]

Yvonne Jansen[1]

Jean-Daniel Fekete[1]

[1]Inria
firstname.lastname@inria.fr
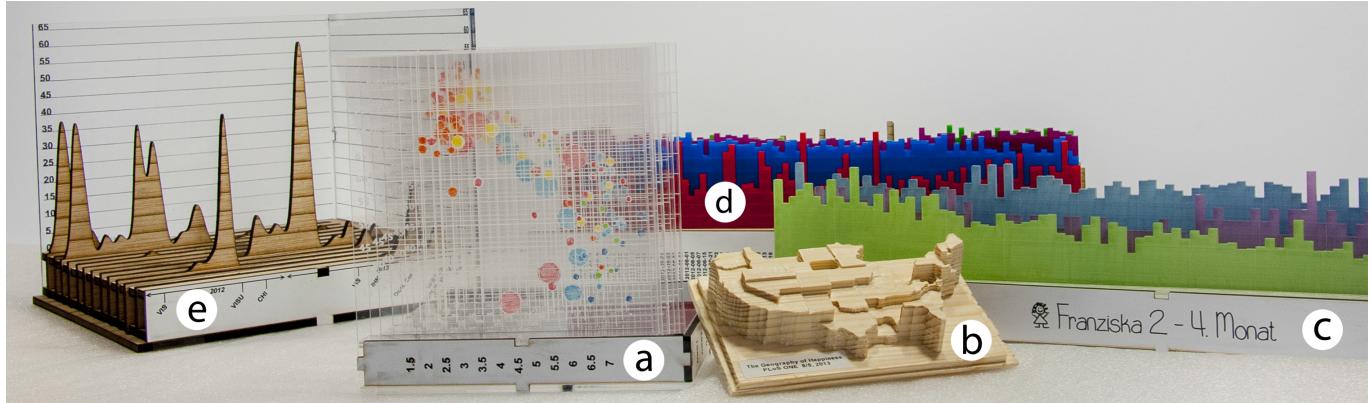
[2]HKUST
clshi@cse.ust.hk

Figure 1. Physical visualizations created with our fabrication tool, MakerVis: a) a layered scatterplot modeled after Rosling ; b) a prism map showing relative happiness in US states computed from Twitter sentiments; c), d), e) layered bar charts and line charts crafted by end users.

ABSTRACT

Physical visualizations come in increasingly diverse forms, and are used in domains including art and entertainment, business analytics, and scientific research. However, creating physical visualizations requires laborious craftsmanship and demands expertise in both data visualization and digital fabrication. We present three case studies that illustrate limitations of current visualization fabrication workflows. We then present MakerVis, a prototype tool that integrates the entire process of creating physical visualizations, from data filtering to physical fabrication. Design sessions with three end users demonstrate how tools such as MakerVis can dramatically lower the barriers to producing physical visualizations. Observations and interviews from these sessions highlighted future research areas, including customization support, using material properties to represent data variables, and allowing the reuse of physical data objects in new visualizations.

Author Keywords

Physical visualization; digital fabrication; infovis.

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User interfaces; D.2.2 Design Tools and Techniques: User interfaces

INTRODUCTION

Visualization systems beyond traditional desktop settings offer radically new ways of interacting with data, an emergent area of research [17, 12]. In particular, *physical visualizations* — visualizations that map data to physical matter instead of pixels — are an information medium that promises to be more compelling [9], more expressive [20], and in some cases more effective than on-screen setups [13]. A wide range of physical visualizations have already been crafted by artists, analysts and scientists, for various purposes ranging from infotainment to goal-oriented tasks [5].

However, creating physical visualizations remains a major bottleneck. While digital fabrication technologies are increasingly accessible [31], most software tools that drive fabrication machines focus on static object models, ignoring data-driven content. Conversely, a wealth of software tools exist for creating on-screen visualizations [29, 33], but do not support the creation of physical objects. In addition to the fundamental differences between physical objects and graphical content (e.g, physical objects are three-dimensional rather than flat, and are subject to physical forces such as gravity and friction), fabrication machines have unique constraints that differ from those of computer displays and 2D printers.

There is no integrated tool for building physical visualizations; it is a laborious and time-intensive process. It requires complex workflows, a heterogeneous set of tools, and expertise in both data visualization and fabrication. The lack of cross-domain tools ultimately limits a designer's abilities to create physical visualizations of larger-scale data, to reuse an existing physical design with new datasets, or to rapidly iterate through possible design variations. This paper addresses these problems through the following contributions:

- We present case studies and discuss users' current design and fabrication process for physical visualizations, and current tools' support during this process;
- We introduce *MakerVis*, a physical visualization design tool prototype that imports users' raw data tables and produces digital fabrication-ready design files (see Figure 1 for examples);
- We present an observational study of three end users who used MakerVis to create a physical visualization of personal data;
- Based on our observations, we identify key design requirements and research challenges for physical visualization design tools.

By developing tools for creating physical visualizations, our work bridges the gap between information visualization and digital fabrication. Not only does it offer visualization experts renewed access to physical media, but it also dramatically expands the application range of digital fabrication technologies. Our work also aims to facilitate the replication of research on effective physical visualization designs [7].

BACKGROUND & RELATED WORK
We draw on research literature from information and digital fabrication to inform our work. In the following section we address the importance of physical visualization, and briefly discuss related work from information visualization and digital fabrication that impact the creation of physical visualizations. We explain the problems that each area tries to address and review current tools that attempt to address them. We explain why the design and fabrication of physical visualizations creates unique problems that neither community is currently able to properly address.

Physical Visualizations
Roughly defined, a physical visualization maps data to physical form instead of pixels or ink [13]. People have externalized information in physical form since before the invention of writing, e.g., in the form of clay tokens [26]. Although flat surfaces – i.e., clay tablets, then paper and ink, and later computer screens – became a more convenient media for data visualization, physical visualizations are becoming more popular. One revealing trend is the explosion of data sculptures, i.e., aesthetic physical visualizations created by designers and artists to support informal reflection about data [20]. An increasing variety of data sculptures are being created, exhibited and shared, from small data jewelry (Figure 4) to large-scale museum installations (Figure 3).

In addition to artistic applications, physical visualizations are increasingly considered for their potential to support analytical tasks. In the mid-20th century, companies and research scientists already used physical visualizations for practical purposes [5]. More recently, Hans Rosling used physical visualizations to explain population growth and income inequalities, and General Motors started using 3D LEGO visualizations to track car failures [5]. Consistent with numerous psychology studies suggesting that physical object manipulation can enhance cognition [1], recent studies suggest that in some cases physical visualizations can be more effective than on-screen visualizations for communicating and analyzing data [9, 13]. Additionally, a variety of data exploration tasks can be carried out by simply rearranging modular physical visualizations [2, 12, 30].

Physical visualizations have the potential to dramatically enrich and complement paper and computer displays. They bridge the gap between artistic and pragmatic visualization [15], and between casual and traditional infovis [22]. Physical visualizations can not only be useful to analysts by faithfully conveying complex, three-dimensional data, but also by inheriting all of the practical and social advantages of everyday objects: they can be "touched, explored, carried, or even possessed" [20]. Moreover, they are constantly "on" and are therefore convenient as ambient displays [12].

Information Visualization: Problems and Tools
The goal of information visualization (or simply *visualization*) is to help humans make sense of data using vision and perception. It translates data into a visual form, mapping data attributes to visual attributes (e.g., shape or color). These primitive shapes are then combined into visual representations such as bar charts, line charts, scatterplots, or choropleth maps. To be optimally effective and support "external cognition" [3], this process is informed by experience and knowledge of the capabilities of the human visual system.

Historically, all data visualizations were hand-crafted. Two-dimensional visualizations were created with a pencil and a ruler on paper, while physical visualizations were made using hand tools [2, 13, 5]. The process of hand-crafting both types of visualizations is laborious and requires considerable expertise. In the last 30 years, researchers have developed many computer tools to simplify the creation of visualizations for screen and printed media (e.g., [29, 14, 10]).

Modern charting systems such as Tableau [29] now support many types of visual representations and offer GUIs to interactively tune visual parameters and polish the produced output. Popular statistical and numerical-computing tools offer charts that can be extensively customized through scripting, requiring programming skills. Support for full customization and creative exploration is still an active research topic [32]. Nonetheless, the process of generating effective visualizations on screen and paper is now accessible to a large audience with little training, and produces high-quality output. However, for physical visualizations, this is not yet the case.

This is partly due to a lack of an appropriate conceptual model. Modern tools rely on the visualization reference model introduced by Card et al. [3]. The model identifies a visualization "pipeline" with four main stages: *raw data*

is transformed into *processed data*, which in turn is mapped into an *abstract visual form*, then refined into a full visual specification, the *visual presentation*. Most of the research on the architecture of visualization toolkits and tools has focused on identifying how to split this conceptual pipeline into software components offering the appropriate level of functionality. This model has been only recently adapted to account for physical visualizations [12], where an additional stage, the *physical presentation*, captures the presentation medium.

Existing toolkits and systems are intended for flat physical presentations, providing essentially 2D visualizations (sometimes using projections from 3D models). The primitives managed by existing systems take for granted the unified raster display abstraction, which allows any image to be specified and rendered on any screen (CRT, LED) or 2D printing technology (laser, ink-jet) transparently. However, the raster model is not appropriate to produce general physical visualizations. There is no unified description of a 3D physical model that can be turned into an efficient set of instructions for a fabrication machine.

Additionally, visualization systems typically automate all pipeline transformations, supporting interaction by allowing users to alter the pipeline at different levels and have the screen dynamically updated [12]. However, in the case of physical visualizations, pipeline transformations may be semi-automated or entirely manual (e.g., physical assembly), and most interactions are deferred to the physical object (e.g, reordering data by rearranging parts) [12]. Specification of interactions for physical visualizations is still at an early stage: existing physical visualizations use ad-hoc or opportunistic interactions, far from any general operational model of physical interaction.

In summary, existing visualization systems cannot be easily adapted to support physical visualizations; neither their conceptual model nor their graphical model is appropriate for physical construction.

Digital Fabrication: Problems and Tools
In digital fabrication, digital *design files* are passed to computer-controlled manufacturing machines to create physical objects. Although many different machines exist, there are two main approaches to digital fabrication. Subtractive techniques (e.g., laser cutting, CNC milling) cut away or remove material. Additive techniques (e.g., 3D printing) build up material layer-by-layer. In addition, fabrication techniques can be either 2D or 3D: laser cutters operate on flat sheets of material, while CNC mills and 3D printers produce 3D solid objects. Materials and manufacturing techniques are tightly coupled. The choice of manufacturing technique may limit the types of materials available for use, and vice versa.

The primary challenge when fabricating objects is to ensure that they physically embody specific physical properties. Most properties must be negotiated by the designer on a case-by-case basis. However, necessary properties often include:

Manufacturability. A digitally-specified object can exist virtually while being impossible to physically manufacture. While several thousand objects models are freely available on the Internet, these models may or may not be at the right scale, or be tailored for a particular fabrication technique. Some software tools attempt to address this issue. For example, *Chopper* [18] partitions 3D digital models into parts that fit within the working volume constraints of a 3D printer. Research tools such as crdbrd [11] and commercial tools such as Autodesk *123DMake*[1] cut 3D models into 2D slices.

Assembly & Fit. Multi-part objects must be easily assembled – and sometimes disassembled – into a well-fitting object. Designers must choose from a wide range of attachment techniques [8], define design specifications (e.g., joint location, feature dimensions, clearance between parts) and iterate to identify the right design for assembly. To computationally facilitate design for assembly, Lau et al. [16] devised a tool that takes IKEA-style 3D models and applies formal grammars to identify the parts and connectors needed to build it.

Balance & Stability. Once fabricated and assembled, objects usually need a stable resting state. Professional CAD software (e.g., AutoDesk Inventor[2]) facilitate design iteration through physics simulation, while the tool *Make It Stand* specifically helps design physically stable 3D virtual objects by carving or deforming them [23].

Strength. Fabricated objects need to support their own weight and withstand human forces from normal use. Again, professional CAD software provides simulation and visualization tools for design iteration. Similarly, the *Sculpteo*[3] Web App for 3D printing offers a heatmap stress visualization on the 3D object preview, which is updated as the user changes the object's dimensions.

While professional CAD tools have extensive design and fabrication support, they require considerable training. Meanwhile, tools for general audiences focus on the fabrication of existing designs. Helping non-experts both design and fabricate is an active research topic. *SketchChair* [24] lets users sketch 2D profiles of chairs that are then extruded and transformed into design files for laser cutters or CNC mills. *Midas* [25] lets users design customized touch-sensors, producing both fabrication-ready design files and assembly instructions. Direct manipulation tools for digital fabrication enable users to concurrently design and fabricate [21].

One promising approach for lowering the barrier to design and fabrication is *parametric design*, i.e., letting users build specific classes of objects by specifying their parameters [27]. For example, *BoxMaker*[4] lets users specify the dimensions of a box and automatically generates 2D stencils with finger joints. Although it only creates boxes, infinite variations are possible. Similarly, our goal is to facilitate the design and fabrication of a particular class of objects –physical visualizations– without the necessity for domain-specific expertise in digital fabrication. As a result, MakerVis treats physical visualizations as parametric objects whose final shape is driven by both raw data and user preferences.

---

[1] http://www.123dapp.com/make

[2] http://www.autodesk.com/inventor

[3] http://www.sculpteo.com

[4] http://boxmaker.rahulbotics.com/

CASE STUDIES

To illustrate the difficulty of crafting physical visualizations, we discuss current practices through four real-world case studies taken from a curated list of data sculptures and physical visualizations [5]. All four designs were documented online and involved varying degrees of manual, automated and computational work.
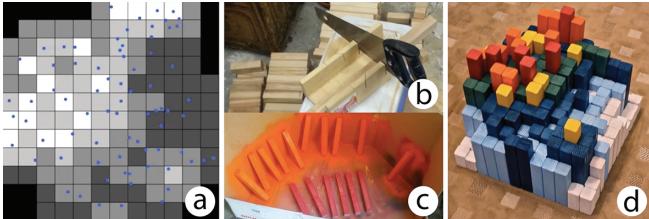
Manual Fabrication

Figure 2. Building a 3D heatmap. Image courtesy of Doug McCune.

Doug McCune constructed a 3D heatmap (Figure 2), showing the performance of elementary schools in San Francisco. With the help of a computer, McCune divided the entire city area into 12×12 squares and computed a weighted score for school performance per square (see Figure 2a). Score values were then converted in heights. McCune then cut 144 pieces of wood according to these heights using a hand saw, which he now does not recommend (Figure 2b). After spray-painting each bar according to its height (Figure 2c), he glued the bars to a base for the final design (Figure 2d).

Apart from the initial steps of data collection and transformation, this visualization was created entirely by hand. This fabrication approach is not only very labor intensive, but also less accurate than digital fabrication.

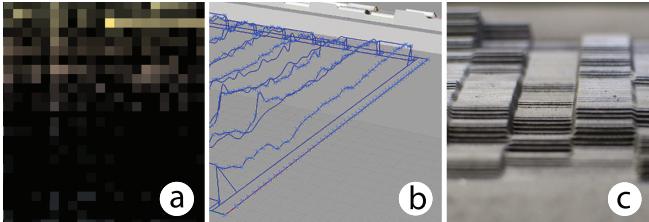### Semi-Automatic Fabrication, Standard Software

Figure 3. Building a data sculpture. Image courtesy of Moritz Stefaner.

The emoto data sculpture (Figure 3c) shows the emotional response to the London 2012 Olympic games based on Twitter traffic [28]. The visualization began as a heatmap visualization in Tableau (Figure 3a), and then was transformed into 3D shapes using Grasshopper (Figure 3b), a graphical algorithm editor for Rhino 3D. The model was fabricated using a CNC milling machine (Figure 3c).

Emoto is digitally created from start to finish, therefore ensuring a data-accurate physical visualization. However, it involved a sequence of different software tools to transform the original data set into fabrication-ready design files. This hinders iterative design; every early-stage change (e.g., filtering data) requires a manual re-execution of the entire process.

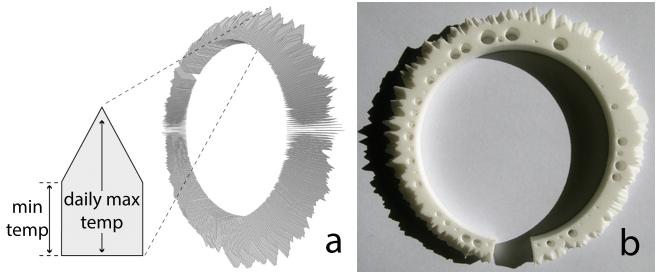### Semi-Automatic Fabrication, Custom Software

Figure 4. Building a data bracelet. Image courtesy of Mitchell Whitelaw.

Mitchell Whitelaw created a bracelet showing one year of temperature and rain fall data from Canberra (Figure 4b). He first used the Processing programming environment[5] to generate a 2D shape for each day (Figure 4a). His program arranged these shapes radially and connected them to form a continuous 3D object. He then used MeshLab[6] to remove redundant vertices and normalize all faces of his model to make it ready for 3D printing. Whitelaw finally used the 3D modeling software Blender[7] in order to subtract cylinders representing rainfall data (also generated by a Processing program) and to finalize the three-dimensional model (see Figure 4a). The model was exported and sent to the on-line 3D printing service Shapeways[8] (see Figure 4b).

This example shows how creating non-conventional visualizations may require writing custom software. On-screen visualizations also may require writing custom software, although several programming toolkits are available.

With this approach, designers with programming skills are able to precisely control and define the design. However, the 3D models created by Processing are still not immediately suitable for fabrication; converting to fabrication-ready files require additional steps. These steps must be explicitly repeated if the design needs changes or adjustments.

(Nearly) Fully-Automated Fabrication

Figure 5. Pendants created with meshu.io. Image courtesy of Meshu.

Meshu.io[9] is a Web service where users create jewelry from their travel data, without any prior knowledge of fabrication. Users either grant the service access to their existing social media data, or manually enter cities. A 2D mesh is built that

---

[5] `http://processing.org/`

[6] `http://meshlab.sourceforge.net`

[7] `http://www.blender.org`

[8] `http://www.shapeways.com`

[9] `http://meshu.io/`

spans the visited cities as represented on a map. Users then select the material and fabrication technique (3D printing or laser cutting). The object is remotely fabricated and mailed to the user, together with a map that decodes the visualization.

Unlike previous case studies, this approach requires no expertise on visualization or fabrication. Users only need to upload data and fine-tune parameters. On the other hand, they have very little control over the final design. The visualization is very specific, and the service only allows the user to re-orient the object and select the fabrication material. Also, while most of the process is automated (from the user's perspective), it takes several weeks to receive the object.

Summary
These case studies illustrate the need for tools that integrate the entire design workflow of physical visualizations, from data processing to final object fabrication. This integration is important not only for making data-accurate visualizations, but also for supporting fast iteration cycles.

MAKERVIS
In this section we discuss and motivate the features – workflow, visualizations and fabrication technologies– currently supported by MakerVis. We then review its user interface, and discuss its implementation.

Features Supported
MakerVis is an initial proof of concept prototype that offers a deeper understanding of users' needs and technical challenges in building design tools for physical visualizations.

Although MakerVis acknowledges all stages in the workflow, we chose to not focus on the earliest and latest stages of the workflow which address more general information visualization and digital fabrication problems. MakerVis only has a basic support for data transformation (i.e., filtering out data points); more elaborate data processing operations are delegated to specialized tools such as Tableau [29]. Concerning the later stages, the support for final customization (e.g., adding a personalized title to a visualization) is limited and primarily delegated to specialized vector graphics and 3D authoring tools. The use of standard formats (i.e., CSV as input and SVG or STL as output) greatly facilitates this process.

As the number of possible physical visualizations is overwhelming (see [5]), we chose to focus on pragmatic visualizations (such as Figures 2 and 3) instead of more artistic data sculptures (such as in Figures 4 and 5). A large family of such visualizations consist of layered (also called 2.5D [6]) visualizations, or stacked 2D visualizations. These are well-suited for displaying complex temporal datasets. MakerVis supports layered bar charts, layered line charts, layered scatterplots and prism maps.

All of the layered physical visualization designs are modular and rearrangeable: layers can be taken out from the base to be reordered or laid out separately. This makes these visualizations more interactive and allows users to perform a range of infovis tasks by direct physical manipulation [2, 30, 12].

Figure 1a shows a laser-cut physical layered scatterplot displaying fertility and life expectancy of the largest countries in the world across the last 60 years, as famously commented by Hans Rosling[10]. Figure 1b shows a CNC-milled prism map displaying the happiness levels of US states in 2011, extracted from Twitter sentiments[11]. The other physical visualizations from Figure 1 have been crafted by end users and will be discussed in MakerVis Design Session section.

MakerVis has extensive support for laser cutters and basic support for 3D printing and CNC milling (through an STL export function). Laser cutting is one of the most popular fabrication techniques in fab labs, in part due to its speed and efficiency, and its ability to cut a wide range of materials. As a 2D fabrication technique, it can be challenging to create 3D visualizations, but it is well-suited to layered visualizations.

Workflow and User Interface
MakerVis's user interface is shown Figure 6. The workflow involves six steps: 1) loading the data, 2) selecting a type of visualization, 3) mapping the data dimensions to visual variables, 4) setting the visualization's geometry, 5) setting the fabrication machine and parameters, 6) downloading the design file. Breaking down the workflow in different steps helps novices learn the software. However, users are not restricted to these steps and can return to a previous step at any point.

After loading a data file in the CSV format (*step 1*), the user can choose an initial visualization type (*step 2*). The *step 3* involves dragging data dimensions to visualization dimensions in a similar way to Tableau [29]. Bar charts and line charts have three dimensions: *Slices*, *X axis* and *Y axis*, while scatterplots offer *Radius* and *Color* as extra dimensions. In Figure 6, the user loaded data on electricity consumption of 10 countries across 10 years, and chose a layered bar chart where each slice is a country, and each country is an extruded 2D bar chart showing electricity consumption over time.

After all visualization dimensions are mapped, MakerVis immediately shows a 3D preview of the physical visualization, together with the stencils for the laser cutter (the stencils follow the design described by [13]). The 3D preview shows the properties of the selected material, including its thickness, color, and texture. In order to give a better idea of the actual size of the physical visualization, the 3D visualization sits on a 30×30 cm base that users can physically recreate, and the laser stencils are shown in 1:1 scale.

The 3D and laser stencil previews are both interactive. On the laser stencil preview, each tab contains a separate sheet to be laser-cut. The user can reposition individual pieces – which can help save material – and change the material selected for each piece. As the material changes, pieces are moved to a tab that corresponds to a sheet of the new material. All available materials are stored in human-readable files.

Step 4 consists of tuning the visualization's geometry. Geometry parameters include the size and spacing of elements (e.g., bars for layered bar charts) and font sizes. These parameters

---

[10] http://tinyurl.com/rosling
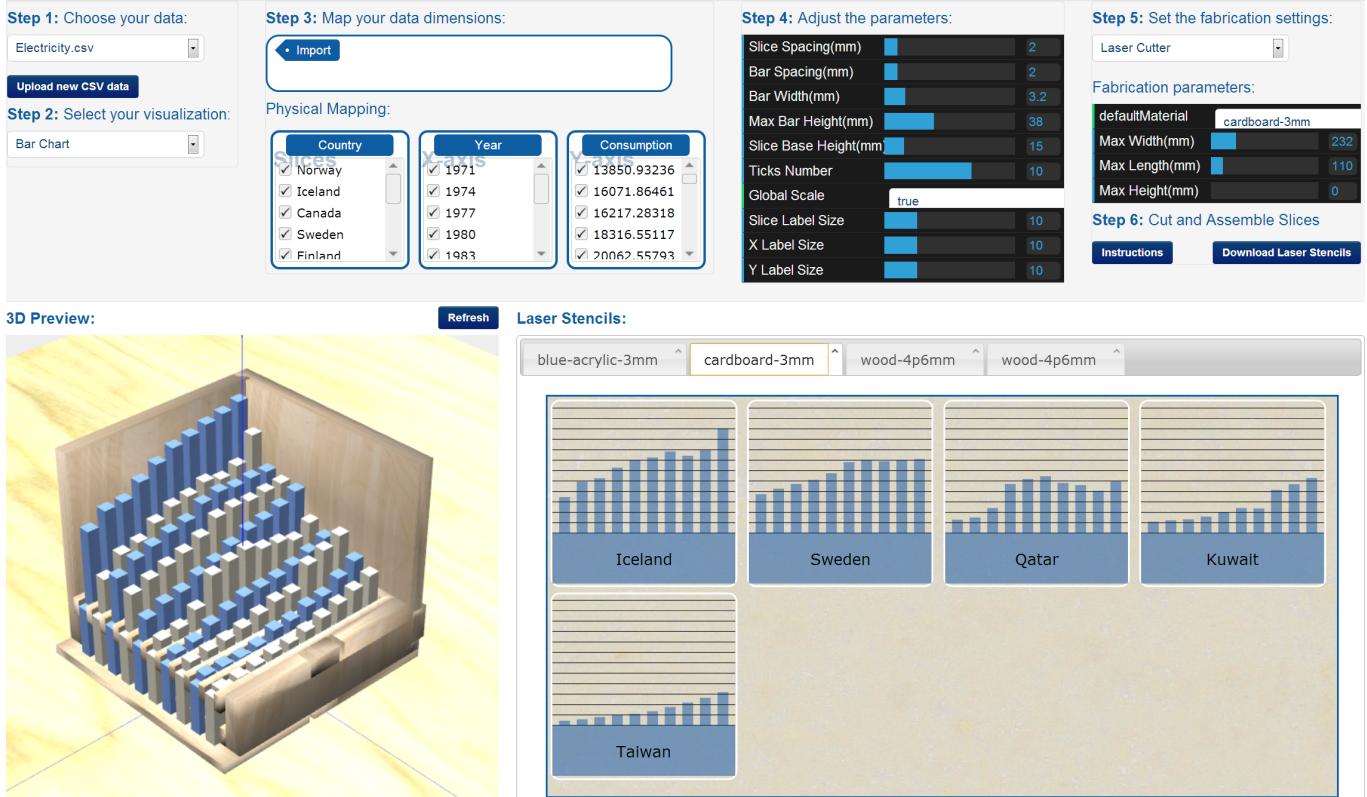[11] http://tinyurl.com/geohappiness

Figure 6. Screenshot of the MakerVis prototype. The control panel is on the top, from Step 1 (left) to Step 6 (right). The previews are shown below.

help adjust design features, aesthetic features and assembly or manufacturing features. Changes in any of these parameters are directly reflected on the two previews.

The user can change fabrication techniques or adjust parameters, such as default material and sheet size for laser cutters (*Step 5*). Once the user is satisfied with the design, she can finally download the design file (e.g., laser stencil, 3D model) and its assembly instructions (*Step 6*). Currently, assembly instructions are general guidelines for assembling laser-cut designs and customizing objects (e.g., using spray paint to change material color). Alternatively, the design file can be customized with separate software (e.g., CorelDraw for laser stencils, SketchUp for 3D models) before fabrication.

Implementation and Extensibility
MakerVis is a web application built on top of NodeJS, D3, JQuery and ThreeJS frameworks. It uses Java as a backend to process SVG boolean operations requests that are currently not supported by Web browsers. Supported input file formats are CSV and topoJSON, used for creating prism maps. Export formats are SVG for laser stencils and STL for 3D printers and CNC mills, both standard file types for fabrication.

Adding New Material
Laser cutter materials are specified in human-readable JSON files. Currently supported parameters are *name*, *thickness*, *color* and *texture*. Texture points to a bitmap file that is rendered in both the 3D preview and the laser stencil preview.

Users can therefore easily add new materials; in principle, MakerVis could interface with a local supply chain or a remote fabrication service that sends these JSON files.

Adding New Visualizations
MakerVis can be extended to support new layered visualizations. An abstract layered visualization class holds methods for i) generating the individual 2D visualization layers, ii) adding physical assembly features for laser cutters (e.g., joints), iii) specifying the way layers need to be geometrically assembled, iv) generating the parameter panels, v) generating a 3D model and vi) generating laser stencils.

Most of these methods do not need to be overriden. The default laser stencil generation method uses a layout manager that employs a simple bin packing algorithm to lay out visualization parts on each laser sheet. Parts are distributed across sheets based on their material. The default method for generating 3D models extrudes the parts, assembles them and merges them. The parts are assembled as specified by the assembly logic (step iii), above), which is an array of 3D translations and rotations that are applied to individual parts.

The 2D visualization that appears on each layer needs to be implemented, but D3 can assist in this process. The physical visualization's geometrical parameters also need to be implemented. The parameters are listed separately by way of a JSON file. Once these two steps are completed, users can immediately test their design with MakerVis.

Support for Fabrication Process and Limitations

Our goal is to explore the problem space for tools that support the creation of physical visualizations, and not to implement a exhaustive, fully-fledged tool. We therefore discuss the extent to which MakerVis currently supports fabrication, and suggest directions for future extensions.

As discussed in Background & Related Work, supporting fabrication involves helping users ensure that their design meets specific physical properties. In a parametric design tool such as MakerVis, three strategies can be considered:

S1. Provide *user feedback* through simulation and visualization, a strategy typically used by professional CAD tools,
S2. Ensure that the property is met *by design*, i.e., is satisfied for all possible parameter values,
S3. *Constrain* the ranges of possible parameter values so that the desired property is always met.

We recommend against **S3** because users should not be restricted in their explorations; the importance of a property is ultimately the designer's judgement. For example, balance may not be critical for a hand-held object or a data jewelry. MakerVis employs a mix of **S1** and **S2**. More specifically:

Manufacturability. All visualizations are compatible with 2D and 3D fabrication machines (**S2**). For laser cutting, MakerVis distributes parts across sheets (**S2**) and the preview reveals parts that are larger than sheets (**S1**). Although size constraints are not yet supported for 3D machines, their working areas could also be shown in the preview (**S1**).

Assembly Fit. MakerVis supports assembly fit by design (**S2**). All visualizations except the prism map include a sliding mechanism that lets users take layers apart and rearrange them. For parts that need to be glued, MakerVis uses finger joints whose size and location are automatically computed from the visualization's geometry and material thickness. These assembly features can be easily added to new visualization types. In the future, we envision a tool that supports a range of attachment designs (e.g., press-fits) [8] and lets users further explore the design space of physical interactions with visualizations [2, 12, 30].

Balance & Stability. Our current visualizations are stable by design (**S2**), as their weight is distributed on a flat, solid base. However, visualizations could be added in the future that may not be inherently balanced. In this case, users may benefit from seeing a physics simulation on the 3D preview (**S1**). Visualizations could also be post-processed to ensure balance (**S2**) as proposed by [23], although only carving – not deformation – can be used, since in our case geometry encodes data. Similarly, visualizations with floating parts such as 3D scatterplots could be post-processed into physically plausible objects by automatically adding support material (**S2**).

Strength. Although MakerVis generally produces robust visualizations, users are allowed to make arbitrarily thin features. The 3D preview (**S1**) can help users detect such issues, but solidity can be difficult to predict as it also depends on material. Future tools could assist users by showing a visualization on the 3D preview (**S1**), similar to that provided by Sculpteo.

MAKERVIS DESIGN SESSIONS

We organized design sessions to observe how users would use MakerVis in practice and to collect user feedback. We sent an announcement on two of our lab's mailing lists. To be eligible participants needed to have personal data that they wanted to visualize. Four people replied; we invited three of them to each participate in an afternoon-long design studio. No compensation was offered to participants beyond the final physical visualization, which they could keep.

User Profiles

Our users were one professor and two PhD students from our team who were not involved in this project. While they have both experience and interest in data and infovis, they had little or no training in fabrication. We summarize their demographic profiles and users' self-rated expertises on a 5-point Likert scale in the table below.

User	Dataset	Age	Sex	Infovis Expertise	Fabrication Expertise
1	Baby	35	F	4	2
2	Training	29	M	4	1
3	Projects	29	M	4	1

Procedure

The participant's task was to create a physical visualization, from loading the data to assembling the final object. To ensure the task was feasible given time constraints, we asked users to send us their data files in advance in order to clean and process the data into a suitable CSV file for MakerVis. As our prototype does not support elaborate data transformations, we also pre-computed aggregated datasets for each user. All three users sent time-series data that was compatible with our three layered-visualization designs.

Sessions were held in a research fab lab. We only offered laser cutter fabrication due to the long build times of 3D printers and CNC milling machines (1 hour for the laser cutter vs. about 10-20 hours for 3D printing or CNC milling). The material provided were 2-5 mm sheets of cardboard, plywood, clear acrylic, and four colors of opaque acrylic. The sheets of material were stored in a cabinet, while spray paint in 10 colors was visible on the workbench.

At the beginning of each session, we gave the user a short introduction to the concept of layered 3D visualization. We then asked the user to create a physical visualization using MakerVis. We provided no initial instruction for the tool, as the GUI already includes explicit steps. We provided help only when necessary. We encouraged users to talk aloud and offer verbal comments. After the session, we conducted semi-structured interviews. Interviews were recorded and design sessions were partially videotaped.

Timing

The times spent on each phase were similar across all three users. An entire design session lasted three to four hours. Users spent about one hour exploring MakerVis and converging on a design. They then spent about one hour customizing and decorating the laser stencils in an external vector editing

Figure 7. Photos from the design sessions: a) a user working on a layered bar chart design, b) a user extracting a visualization's structural parts from a laser cutter, c) a user assembling a physical bar chart.

application — suggested by the researcher as soon as the user asked for a decoration feature not provided by MakerVis. The laser cutting took 45 to 90 minutes depending on the material used. The finishing and assembly phase took 15 to 30 minutes depending on whether or not the user wanted to physically customize their visualizations (e.g., apply colors).

First User – Baby Data

The first user supplied us with data that she had collected on her baby daughter's activity during the first months of her life. This time-series data included sleeping hours, breastfeeding hours, and diaper changes across 3 months.

Initially, the user experimented with different visualizations and data mappings. She then focused on a layered bar chart and spent 20 minutes adjusting its geometry. However, she looked at other aggregated datasets in parallel by opening another instance of the tool. While tuning the geometry, she asked for calipers to measure her finger size, so that that slices of the final physical visualization could be easily removed.

The user asked to see materials, despite the textures displayed on the 3D preview. She was not satisfied with the range of colors provided in opaque acrylic (4 colors), so she decided to use clear acrylic and spray paint instead (10 colors).

While graphically editing the laser stencils generated by MakerVis, the user relabeled the time axis by week instead of by date, as the age of young babies is often described in weeks. She also replaced the default font with a more casual font that she downloaded from the Web, and added a title (see final design on Figure 1c – time labels are on the back face).

From the interview, the user seemed to enjoy the experience of making a physical visualization, even though she felt the process was a bit long. When asked if she got any new insights about the data, she reported being surprised about sleeping time not decreasing over time, but then remembered this was accurate. She also pointed at a peak in diaper changes and commented it was "crazy" but a "cool reminder".

When the user was asked how she could improve her physical visualization, she commented that data-specific additions were important, such as week units for babies. When asked whether the final visualization looked like what she had imag-

ined while using MakerVis, she said she "expected it to be a bit smaller", but was "finally happy with the size". Finally, when asked what she would do with the object she replied:

> "I will put it up in the living room and show to people. You can imagine the scenario: you are all sitting in the living room, you have visitors and take pieces out of it and everyone can look at a part of the visualization."

Second User – Training Data

The second user supplied us with his father's self-looged cycling activity data over the course of 8 months, with multiple performance measures per trip such as average heart rate, distance traveled, and average speed.

The user explored all three visualizations in parallel (bar charts, line charts and scatterplots) by opening several instances of MakerVis. He finally chose bar charts then went on to tune the geometry. At this point the user asked to see the actual material. The final result is shown in Figure 1d.

When asked if the final object matched his expectations, the user reported that "the size was too big, but actually I didn't have any idea how it would look". When asked about possible data insights, the user replied he was not too familiar with the data, and built this physical visualization as a gift to his father. When asked if he would use this type of tool again, the user said that he would do so only in a fab lab context because it is important to be able to play with the object physically.

Third User – Project Activities

This user supplied us with time-series data on commits in a versioning repository containing several research projects he was involved in over the first two years of his PhD thesis.

While exploring designs, the user wanted to aggregate his data by week instead of the monthly aggregation that we had prepared. One researcher produced the requested data but when the data was ready, the user had already converged to a design. Since MakerVis cannot save designs, we took a screenshot of the geometry parameters then replicated these settings on the new dataset.

After the design was finalized, the user was disappointed to realize that there was not enough plywood to build his visualization. He then manually rearranged the laser stencils

on the preview to fit the available material. Using the vector graphics editor, he also added conference deadlines on the time axis. After the parts were cut, he re-emphasized laser-engraved axis labels and lines by tracing with a pen. The final result is shown in Figure 1e.

When asked about his overall experience of building the visualization the user reported that "it was easy, but for the tweaking you needed an external editor", and that axis customization was important. When asked about missing features, he commented that the 1:1 scale on the laser stencil preview was useful, but a zoom feature would have reduced the need for scrolling. The user also reported under-estimating the size of the final visualization. When asked if he would use a fully-functional version of MakerVis in the future, he replied:

> "I am wondering for which kind of data I will use the tool. For standard data, no; but for personal data, yes. Or to give or show to people, especially if you can customize it, tweak everything, personalize it, really personalize the final visualization. It's for someone, so you want specific stuff."

DISCUSSION

Our user observations suggest that MakerVis helps create physical visualizations and has low entry requirements as a fabrication tool. Our interviews confirm that none of the users would have been willing to spend the necessary time for creating one without the tool. They were all satisfied with the results they obtained and could imagine creating more physical visualizations of personal data in the future. One of our users left stating "I feel like I accomplished something."

The three design sessions also identified shortcomings that were not salient earlier. We identified four major topics that demand further research:

Personalization. One of our users' main interests was personalization. Users want detailed control over all decorative aspects of the visualization they create, as they are already investing effort and resources into the creation of an object. Producing a finely-tuned design is more important than for on-screen visualizations, where users can always save the current state and improve it later on.

MakerVis only provides limited control over scale and axis design. All users therefore chose to use an external graphical editor. We identify two possible problems with this approach:

1. Any modifications done after exporting cannot be brought back into the tool. For example, should the user realize that she made a mistake when filtering her data, she would have to redo her modifications in the external tool.
2. Such interventions can easily break the design. Separate pieces need to fit together and be correctly aligned, e.g., for scales to make sense.

Future versions of MakerVis should allow users to hand-tune their designs within the tool, especially text-based decorations as used for scales, labels, and titles. MakerVis should also help users ensure that these modifications do not change the data displayed or violate a desired physical property (manufacturability, assembly fit, balance & stability, strength). However, a design tool should also allow to export designs for further customization with external tools. Users would do this, as always when hacking a system, on their own risk since such actions "void the warranty" that all the pieces will fit and the displayed data is correct.

Material. When crafting their physical visualizations, our users were confronted with both the constraints and expressive potential of physical material. MakerVis already allows users to assign different parts to different materials; we noticed that users sometimes used material color to encode categorical data. This suggests that future versions of MakerVis should explicitly support material properties as "visual" variables, such as texture, elasticity, transparence or shininess.

Preview. MakerVis renders approximate textures for available materials. With a wider variety of materials, more realistic rendering is needed. Despite the digital preview, all our users asked to see and to hold actual materials before finishing their design. Optical and haptic properties require elaborate equipment to emulate. Overall, it seems difficult if not impossible to give a preview of the real object. Our users also consistently underestimated the size of the final physical visualization. Therefore, effectively conveying physical scale, perhaps using anchoring [4], seems important.

Reuse. Some of our users were disappointed by MakerVis' inability to save and load designs. Such tools should let users share designs and reuse an existing design with new datasets. However, supporting reuse with very different datasets (either in terms of size or type) may be hard. Data can also change after a physical visualization has been created, making the object obsolete. Ideally, the tool should let users create physical visualizations that can be later updated with new data.

CONCLUSION

Despite the democratization of fabrication technologies, software tools are still needed to help users design and fabricate complex objects. In particular, building physical visualizations is still a laborious process, that requires expertise in both information visualization and digital fabrication.

We explored the possibilities behind a tool that integrates the entire physical visualization design workflow, from data filtering to physical fabrication. Using our prototype, users proficient in data visualization but not fabrication could easily create their own physical visualization. Their main request was for more opportunities for customization to highlight events with personal significance. In the future, a study involving users proficient in digital fabrication but not data visualization will yield a better understanding of user needs.

MakerVis will be released as an open source project to support ongoing research on physical visualizations, by facilitating the implementation of more visualization types and different fabrication and assembly techniques.

ACKNOWLEDGEMENTS

REFERENCES

1. Anderson, M. Embodied cognition: A field guide. *Artificial Intelligence 149*, 1 (2003), 91–130.

2. Bertin, J. Graphique et mathématique : généralisation du traitement graphique de l'information. *Annales. Économies, Sociétés, Civilisations.*, 1 (1969), 70–101.

3. Card, S. K., Mackinlay, J. D., and Shneiderman, B. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, Jan. 1999.

4. Chevalier, F., Vuillemot, R., and Gali, G. Using concrete scales: A practical framework for effective visual depiction of complex measures. *Visualization and Computer Graphics, IEEE Transactions on 19*, 12 (2013), 2426–2435.

5. Dragicevic, P., and Jansen, Y. List of physical visualizations. tinyurl.com/physvis, 2013. [Online; accessed 04-Sep-2013].

6. Dwyer, T. *Two-and-a-half-dimensional Visualisation of Relational Networks*. PhD thesis, School of Information Technologies, Faculty of Science, University of Sydney, 2005.

7. Gaines, B. R. Modeling and forecasting the information sciences. *Information Sciences 57* (1991), 3–22.

8. Genc, S., Messler Jr, R. W., and Gabriele, G. A. A systematic approach to integral snap-fit attachment design. *Research in Engineering Design 10*, 2 (1998), 84–93.

9. Gwilt, I., Yoxall, A., and Sano, K. Enhancing the understanding of statistical data through the creation of physical objects. In *Proc. ICDC2012* (2012), 117–126.

10. Heer, J., Card, S. K., and Landay, J. Prefuse: A toolkit for interactive information visualization. In *ACM Human Factors in Computing Systems (CHI)* (2005), 421–430.

11. Hildebrand, K., Bickel, B., and Alexa, M. crdbrd: Shape fabrication by sliding planar slices. *Comp. Graph. Forum 31*, 2pt3 (May 2012), 583–592.

12. Jansen, Y., and Dragicevic, P. An interaction model for visualizations beyond the desktop. *IEEE TVCG* (Nov. 2013). To appear.

13. Jansen, Y., Dragicevic, P., and Fekete, J.-D. Evaluating the efficiency of physical visualizations. In *Proc. of CHI 2013*, ACM (2013), 2593–2602.

14. Jean-Daniel Fekete. The InfoVis Toolkit. In *Proc. InfoVis 2004*, IEEE Press (October 2004), 167–174.

15. Kosara, R. Visualization criticism-the missing link between information visualization and art. In *Proc. of Information Visualization, IV'07*, IEEE (2007), 631–636.

16. Lau, M., Ohgawara, A., Mitani, J., and Igarashi, T. Converting 3d furniture models to fabricatable parts and connectors. In *Proc. of ACM SIGGRAPH 2011*, SIGGRAPH '11, ACM (2011), 85:1–85:6.

17. Lee, B., Isenberg, P., Riche, N., Carpendale, S., et al. Beyond mouse and keyboard: Expanding design considerations for information visualization interactions. *IEEE TVCG 18*, 12 (2012).

18. Luo, L., Baran, I., Rusinkiewicz, S., and Matusik, W. Chopper: partitioning models into 3d-printable parts. *ACM Trans. Graph. 31*, 6 (Nov. 2012), 129:1–129:9.

19. McCune, D. Physical maps–from digital to analog. tinyurl.com/physicalmaps. [accessed 2013-09-02].

20. Moere, A. Beyond the tyranny of the pixel: Exploring the physicality of information visualization. In *Proc. IV'08*, IEEE (2008), 469–474.

21. Mueller, S., Lopes, P., and Baudisch, P. Interactive construction: interactive fabrication of functional mechanical devices. In *Proc. of UIST 2012*, ACM (2012), 599–606.

22. Pousman, Z., Stasko, J., and Mateas, M. Casual information visualization: Depictions of data in everyday life. *IEEE TVCG 13*, 6 (2007).

23. Prévost, R., Whiting, E., Lefebvre, S., and Sorkine-Hornung, O. Make it stand: balancing shapes for 3d fabrication. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 81.

24. Saul, G., Lau, M., Mitani, J., and Igarashi, T. Sketchchair: an all-in-one chair design system for end users. In *Proc. TEI 2011*, ACM (2011), 73–80.

25. Savage, V., Zhang, X., and Hartmann, B. Midas: fabricating custom capacitive touch sensors to prototype interactive objects. In *Proc. of UIST 2012*, ACM (2012), 579–588.

26. Schmandt-Besserat, D. *How writing came about*. University of Texas Press, 1996.

27. Shah, J. J. *Parametric and feature-based CAD/CAM: concepts, techniques, and applications*. Wiley. com, 1995.

28. Stefaner, M., and Hemment, D. Emoto. tinyurl.com/emotovis, 2012. [accessed 2013-03-30].

29. Stolte, C., and Hanrahan, P. Polaris: A system for query, analysis and visualization of multi-dimensional relational databases. *IEEE TVCG 8* (2002), 52–65.

30. Stusak, S., Tabard, A., and Butz, A. Can physical visualizations support analytical tasks? Vis '2013 (poster) (Nov. 2013). In press.

31. Tanenbaum, J. G., Williams, A. M., Desjardins, A., and Tanenbaum, K. Democratizing technology: pleasure, utility and expressiveness in DIY and maker practice. In *Proc. CHI 2013*, ACM (2013), 2603–2612.

32. Victor, B. Drawing dynamic visualizations. vimeo.com/66085662, 2013. [Online; accessed 18-Sep-2013].

33. Viegas, F. B., Wattenberg, M., van Ham, F., Kriss, J., and McKeon, M. Manyeyes: a site for visualization at internet scale. *IEEE TVCG 13*, 6 (Nov. 2007), 1121–1128.

34. Whitelaw, M. Weather bracelet–3d printed data-jewelry. tinyurl.com/weather-bracelet. [accessed 2013-09-02].