Small Multiples Sequence-Controlled Transfer Function Editor for Designers

Meredith Young-Ng*
UC Davis

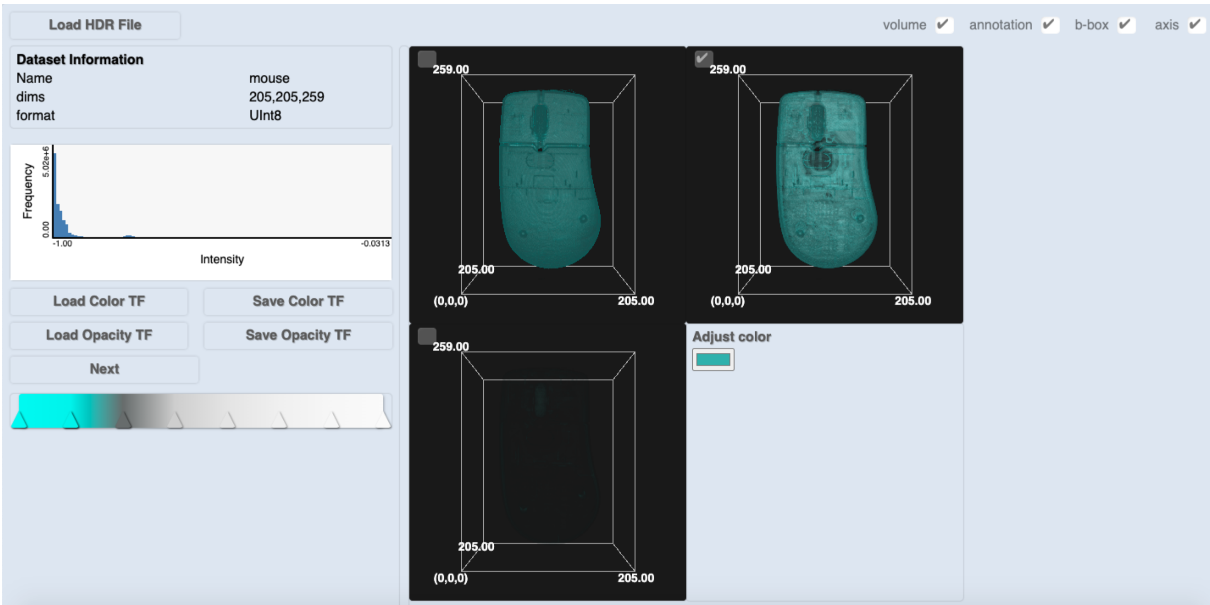

Figure 1: User interface for our small multiples sequence-controlled transfer function editor, currently displaying a small multiple containing three images of different opacity values for one sub-volume of a computer mouse CT scan dataset.

Abstract

Standard volume visualization systems focus on promoting more advanced features, complicating the transfer function (TF) interface. This makes it difficult for new users such as designers, who are unfamiliar with transfer functions and 3D graphics, to understand how these systems work. We present a volume visualization system with a small multiples sequence-controlled transfer function editor that attempts to improve learnability and facilitate rapid prototyping, particularly idea generation. Based on uniform partitioned sub-volumes, we display a small multiple (three images with different opacities) and a color picker for each sub-volume for user selection that control the opacity and color TFs, respectively. We provide an exploratory user study comparing our system to the standard separated graph-based opacity and color TF editor (as implemented in A3 and A4). Even though both designers preferred the standard TF editor, they had a considerably faster task completion time with our system. We also present a set of design recommendations for improving transfer function user interfaces' learnability.

Index Terms: Human-centered computing—Visualization—Visualization design and evaluation methods

1 Introduction

Many artists and designers work primarily with 2D graphics. When first learning how to visualize and interact with 3D graphics, however, they can be easily overwhelmed due to the steep learning curve of many 3D modeling and computer-aided design (CAD) software systems such as Rhino [9] and Maya [1]. While other modeling software systems such as Tinkercad [2] simplify the user interface, they primarily focus on modifying basic primitives. Even though all of these 3D modeling and CAD software systems can easily import traditional 3D meshes and objects, they lack support for volume visualization data. This often discourages new users from making 3D visualizations with existing volume data to share their prototyping ideas.

Existing volume visualization systems often display the color and opacity transfer functions (TFs) separately, using control points on a graph. While designers are generally familiar with some aspects (mainly the usage of color gradients to select a color and adjusting opacity to change the transparency of an image), they are unfamiliar with the usage of control points and graphs to control a visualization display. This hinders designers from being able to quickly integrate volume visualization data into their workflow, which usually involves a rapid prototyping design process [14]. Being able to quickly view and share 3D designs via 3D visualization tools is critical to early-stage development, particularly during the idea generation phase.

We built a volume visualization system that simplifies the transfer function editor that 1) attempts to make 3D visualization tools easier to learn for new users, particularly designers (e.g., users without previous experience with transfer functions and graph-controlled interfaces), improving learnability; and 2) faciliates rapid prototyping in the design process. After partitioning the data into sub-volumes, the system shows a small multiples sequence (three images) containing different opacities for each sub-volume. The user selects their desired image, which directly controls the opacity TF. A separate

*e-mail: mjyoungng@ucdavis.edu

standard color wheel selection enables the user to change the color of the corresponding sub-volume, and thus control the color TF.

We performed an exploratory user study with 2 design professionals comparing our system to the standard separated graph-based opacity and color TF editor (as implemented in A3 and A4). While both designers preferred the additional precision of the standard TF editor when trying to match a pre-rendered reference image, they were able to finish their visualization much faster when using our system. Based on their feedback, we have implemented a few modifications to our system and also present a set of design recommendations for improving the learnability of transfer function editors.

2 Related Work

2.1 Design Galleries

Existing transfer function editors often remove the graph components by instead showing a range of images, as in the design galleries method [8]. When showing a range of options, however, users may be affected by the central tendency bias [12]. Some work has focused on improving these editors' learnability for users of all ages in a museum gallery context [5], but not specifically for more visually experienced users. Furthermore, Jönsson et al. display image gallery thumbnails of the entire data domain, instead of enabling the user to focus on one subdivision at a time [5]. In contrast, we iterate through each subdivision, showing three images at a time, each with a different opacity. The user can also modify the color for each subdivision during the corresponding iteration.

2.2 Small Multiples

Small multiples have been used for visual data exploration, particularly to broaden the user's search space [15]. They have also been effective for information visualization analysis [13]. Prior work incorporating multiple small multiples shows the entire 3D view side-by-side [3] and in immersive spaces (mixed reality) [4, 7]. Johnson et al. use small multiples with different data domain subdivisions, but focus on simulation data and tailor their system to engineers and researchers [4]. Our system, however, shows a sequence of small multiples (one per subdivision) and focuses on a different context–learnability for non-scientific designers.

3 Method

Our approach starts by uniformly partitioning the data into n sub-volumes. For each sub-volume, the system will then display 3 images (each with a different opacity) of that sub-volume for the user to choose from. The user can also modify the color of the displayed sub-volume using a standard color wheel picker. The selected opacity image and color for each sub-volume directly control the corresponding TFs, and the system displays the final 3D view upon user selection for each sub-volume. Essentially, we show a sequence of small multiples for each sub-volume, with the user determining the optimal opacity and color choices for each sub-volume. Ultimately, this approach sacrifices some of the finer precision in the standard TF editors for greater simplicity.

3.1 Implementation

Our implementation uses WebGL, building upon the A3/A4 system. It runs on a 2015 Macbook Air with an i5 CPU, 8 GB memory, and Intel HD Graphics 6000 GPU.

3.1.1 Uniform Partitioning

Our simplified A3/A4 implementation (see Section 4.1) has a default of $n = 8$ uniformly spaced control points for both the opacity and color TF. Our small multiples sequence TF editor builds upon the existing TF framework with some modifications: it assumes that when the ”Next” button is first clicked, there is the same initial number of opacity and color control points, and that each color control point has the same horizontal value as its opacity control point counterpart. Our system renders the uniform partitioning based on the initial opacity control point positions, as it saves them the first time the ”Next” button is clicked. Based on the horizontal position of these control points, the system develops $n-1 = 7$ sub-volumes (the data values between each of the control points). An individual sub-volume is displayed by modifying the opacity TF to filter out the rest of the data.

3.1.2 Small Multiples Selection: Opacity TF

To render each sub-volume, start by removing all the current opacity control points from the graph. For each iteration, let x_0 and x_1 be the two horizontal positions of the iteration's adjacent control points, and let y be the opacity value of an image in the iteration's small multiple. We update the opacity TF to use the following 4 control points during each iteration of the sequence: $(x_0, 0)$, (x_0, y), (x_1, y), and $(x_1, 0)$. To render the other two small multiple images, we adjust the y value of the control points. This also enables us to reuse the same opacity buffer for rendering each of the images. A checkbox on the upper left hand corner of each of the three opacity images indicates the user selection. When the ”Next” button is clicked again, we store the selected y value and move onto rendering the next iteration. When the last iteration of the sequence is finished, we reset the view to the 3D view and show the updated opacity TF.

3.1.3 Color Selection: Color TF

Rendering the color picker uses a similar approach, but is instead 1D: based on the initial control point positions, we create two new control points for each sub-volume at positions $x_0 + \eta$ and $x_1 - \eta$, with η currently set to $1\mathrm{e}-11$. The color picker uses the HTML input color picker (the same as our standard TF editor implementation), and is displayed in the bottom right quadrant next to the three opacity images. Changing the color of the displayed sub-volume re-renders the three opacity images. The first time the sequence is finished, we remove the initial color control points to update the overall color TF. Subsequent iterations simply modify the corresponding pair of control points for a particular sub-volume.

3.2 User Instructions

Most of the interface works similarly to the A3/A4 implementation (3D view interaction, loading in data, loading/saving color and opacity TF). The main difference is our small multiples sequence TF editor. To start the small multiples sequence, click the ”Next” button beneath the ”Load Opacity TF” button. Three images of the first sub-volume, each with a different opacity, will replace the 3D view. Click the checkbox in the upper left hand corner to select the desired opacity image. Click the displayed color under the ”Adjust color” label to display the color picker pop up window and modify the color for the displayed sub-volume. Click ”Next” to confirm your choices and move onto the next sub-volume. Once the user has finished selecting choices for all sub-volumes, the 3D view will reappear with the updated opacity and color TFs.

4 Evaluation

We evaluated our system by running an exploratory user study (within-subjects) comparing our small multiples sequence-controlled TF editor to the standard TF editor implemented in A3/A4. Both systems and our two datasets (”mouse” and ”backpack”) are shown in randomized order with the task of having the user match a pre-rendered reference image as quickly as possible. Each participant used each system and each dataset once (for a total of two tasks).

4.1 Standard TF Editor

We simplified the standard TF editor used for comparison from our own A3/A4 implementation, opting to always use the raycasting, lighting, and pre-integration mode. These checkboxes were removed

Figure 2: Pre-rendered reference image of computer mouse.

from the user interface, in addition to the cutting planes in the 3D view and along the side. Finally, we integrated the color and opacity TF loading and saving functionality from Tyson's version.

4.2 Datasets

Two industrial CT scan datasets of real-life objects were used in our user study, in randomized order. These consisted of a computer mouse and its inner components (205 x 205 x 209) [11] and a backpack with items inside it (downsampled to 256 x 256 x 239) [6].

4.3 Participants

Our participants were two experienced designers without any experience with transfer functions or volume visualization tools. Designer A's background is in fashion design (materials), and Designer B's background is in textiles and jewelry. Both have limited experience with 3D design software, but are experienced with 2D design software (e.g., Adobe Photoshop and Illustrator).

4.4 Task and Procedure

The pre-rendered reference images for the designers to match are shown in Figure 2 and 3 for the mouse and backpack, respectively. Before the designer started each task, we provided a short demo of the system they will use, as well as the pre-rendered reference image for them to replicate. The designer was allowed to freely ask questions during each demo. The designers were encouraged to think aloud while performing each task.

We tracked the completion time for each task. After completing both tasks, the designers completed a short survey and provided their feedback for each system.

5 Results

Our system had mixed results. Both designers preferred the standard visualization TF editor, as it produced image results that most closely matched the reference image and had a consistent 3D view to refer to (see Figures 4–7). While our small multiples sequence TF editor wasn't the preferred option, as it lacked the higher precision offered in the standard visualization TF editor, it had a considerably

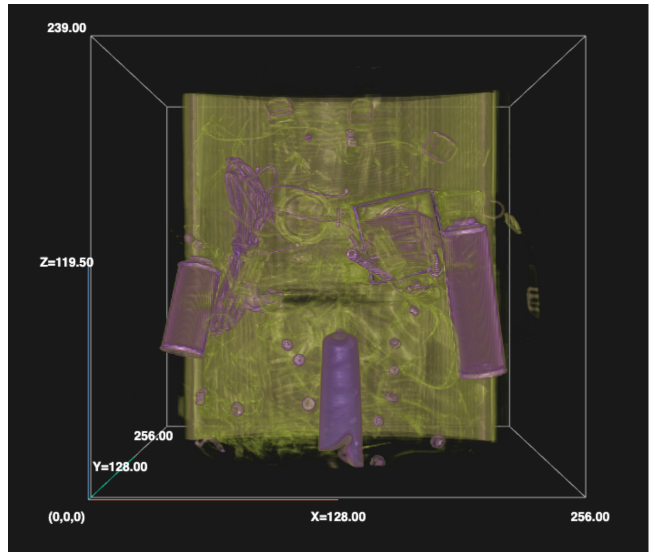

Figure 3: Pre-rendered reference image of backpack.

Designer	Task	System	Dataset	Time
A	1	standard	mouse	11:34:87
A	2	small multiples	backpack	8:45:72
B	1	small multiples	mouse	11:15:24
B	2	standard	backpack	22:21:37

Table 1: Task completion time for each designer.

faster task completion time (Table 1). Ultimately, we were able to determine improvements for our small multiples sequence TF editor and develop recommendations for improving learnability when designing transfer function editors.

To evaluate learnability and cognitive workload, we used the NASA-TLX questionnaire with a 5-point Likert scale [10]. See the average results for each system in Table 2 and Table 3. Note that while the performance quality of the small multiples sequence TF editor has a slightly lower average score than the standard TF editor, it has a lower average for both temporal demand and effort–Designer A noted that they almost gave up on the task when using the standard TF editor.

When asked about potential applications for our system, the designers noted their process of elimination approach, as it felt more playful and experiential, like a game. Our implementation of our small multiples TF editor, as shown to them, did not seem immediately helpful as a tool for rapid prototyping iterations due to its lack of finer precision; instead, it functioned as a more exploratory tool of the volume data for idea generation, another component of the design process. They noted that the additional precision of the standard TF editor made it feel more work-focused.

Overall, the designers were generally confused about how each

Scale	Score
Mental Demand	5
Physical Demand	1
Temporal Demand	3
Performance	3
Effort	4.5
Frustration	3

Table 2: Average NASA-TLX results using a 5-point Likert scale for the standard TF editor.

Figure 4: Designer A's re-creation of the computer mouse using the standard TF editor.

Figure 5: Designer B's re-creation of the computer mouse using the small multiples sequence TF editor.

Scale	Score
Mental Demand	5
Physical Demand	1
Temporal Demand	2.5
Performance	2
Effort	3.25
Frustration	3

Table 3: Average NASA-TLX results using a 5-point Likert scale for the small multiples sequence TF editor.

system worked despite familiarity with visual color and opacity components. For the standard TF editor, they encountered significant difficulties when trying to figure out how to filter out the noisy outer parts of the data via the opacity TF editor, as they were unused to a 2D graph approach. When using the small multiples sequence TF editor, they were confused about what each sub-volume was showing and how it related to the overall 3D view, since they lacked a current 3D view when making their selections. Both noted that it would be helpful to have a system of defined layers for the 3D data, similar to the 2D approach in Adobe Photoshop.

5.1 Design Recommendations for Learnability

Based on our observations and subsequent discussions with the two designers, we developed the following design recommendations for improving transfer function editors' learnability.

1. **Be careful when using color in the design of the opacity TF editor.** Although the opacity and color TF editors operate independently in the standard TF editor system, the designers frequently got confused since the gradient was shown in the region underneath the opacity TF line function, and tried to change the color by moving the opacity control points.

2. **Use 1D ranges (instead of 2D graphs) whenever possible.** 1D ranges, like the color TF in the standard TF editor, are somewhat similar to range sliders–it is clear what variable is being changed. When using 2D graphs, it isn't usually as apparent to new users what each dimension does. The designers struggled to figure out what moving opacity control points along the horizontal axis did (change what data values were affected).

3. **Make the relationship between a sub-volume and the overall 3D volume as clear as possible.** Our current implementation uses a uniform partitioning scheme by data value to create the different sub-volumes. New users, particularly those unfamiliar with 3D visualization systems, expect to see partitioning either by camera depth or by object component instead (when showing a group of objects, such as the items inside the backpack).

4. **Precision with opacity and color is important to visually focused users (e.g., designers).** Visually focused users, such as designers, place a lot of importance on being able to recreate specific image references (particularly with color schemes). This requires higher precision, which usually involves a more complicated user interface. Such high precision may not be necessary in other contexts (e.g., an exploratory visualization interface for children).

5.2 Suggested Improvements

Suggested user interface modifications include saving the existing checkbox options for the opacity TF images when iterating through the small multiples sequence again, as well a previous button to be able to go back and modify their previous choices. Hiding the opacity TF editor during the iterations would also be helpful, as it is somewhat misleading (it only displays the currently displayed opacity TF for the last of the three opacity images). Our current

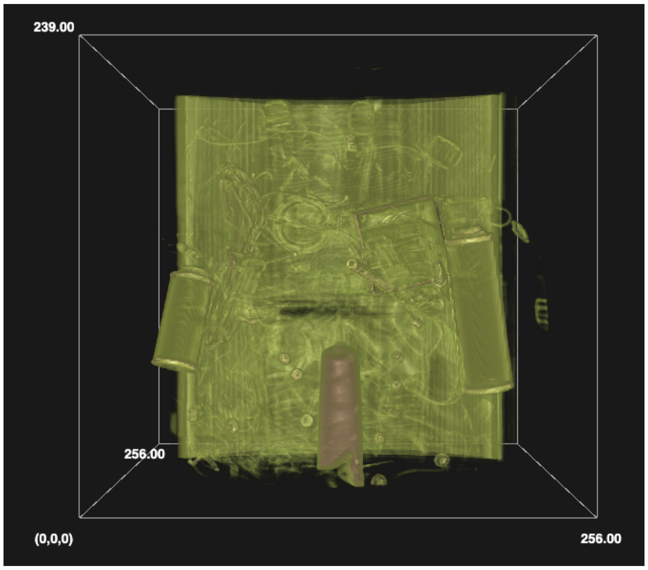

Figure 6: Designer A's re-creation of the backpack using the small multiples sequence TF editor.

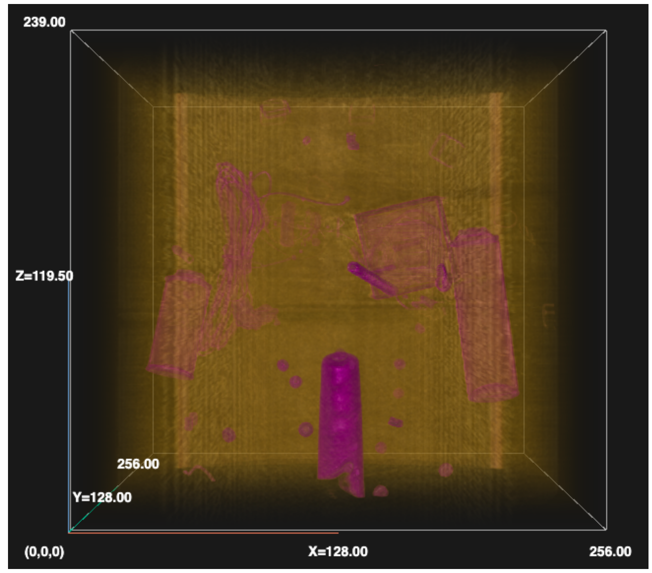

Figure 7: Designer B's re-creation of the backpack using the standard TF editor.

implementation has been modified to show the previous checkbox selections from the last iteration and also hide the opacity TF editor during the iterations.

6 Limitations and Future Work

Since we uniformly partitioned the volume data according to its data values, there were some sparse regions and some much more dense regions, resulting in lower precision for this implementation. Using an adaptive partitioning approach or regional partitioning (e.g., subdividing the data based on identified individual components such as the mouse wheel for the computer mouse) could help the user develop a clearer mental model for how each sub-volume relates to the overall 3D view when iterating through the small multiples sequence. This would also help improve the precision of the small multiples sequence TF editor, as the sub-volumes displayed would focus more on denser regions of the data.

In addition, incorporating a current 3D view of the data when iterating through the small multiples sequence would help users better understand the relationship between the currently displayed sub-volume and the complete volume. This could be shown in the quadrant where the color adjusting currently takes place, with the color adjustment moved to its own horizontal row at the top of the small multiples display.

We could test the effectiveness of these proposed modifications by running a more extensive user study with a greater number of participants. Based on the designers' feedback about our system's potential applications, we could test our system with a different task focused on learnability that requires less precision, such as 3D volume data exploration for children.

Acknowledgments

The authors wish to thank Tyson for his help during the quarter, especially with feedback and support on this project.

References

[1] Autodesk. Maya software, 2021. `https://www.autodesk.com/products/maya/overview`.

[2] Autodesk. Tinkercad, 2021. `https://www.tinkercad.com/`.

[3] J. Chen, A. S. Forsberg, S. M. Swartz, and D. H. Laidlaw. Interactive multiple scale small multiples. In *IEEE Visualization*. Citeseer, 2007.

[4] S. Johnson, D. Orban, H. B. Runesha, L. Meng, B. Juhnke, A. Erdman, F. Samsel, and D. F. Keefe. Bento box: An interactive and zoomable small multiples technique for visualizing 4d simulation ensembles in virtual reality. *Frontiers in Robotics and AI*, 6:61, 2019.

[5] D. Jönsson, M. Falk, and A. Ynnerman. Intuitive exploration of volumetric data using dynamic galleries. *IEEE Transactions on Visualization and Computer graphics*, 22(1):896–905, 2015.

[6] K. Kreeger. Backpack scan. `https://klacansky.com/open-scivis-datasets/`.

[7] J. Liu, A. Prouzeau, B. Ens, and T. Dwyer. Design and evaluation of interactive small multiples data visualisation in immersive spaces. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 588–597. IEEE, 2020.

[8] J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, et al. Design galleries: A general approach to setting parameters for computer graphics and animation. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 389–400, 1997.

[9] R. McNeel and Associates. Rhino - rhinoceros 3d, 2021. `https://www.rhino3d.com/`.

[10] NASA. Nasa tlx: Task load index, 2020. `https://humansystems.arc.nasa.gov/groups/tlx/`.

[11] T. Neuroth. Ecs 277 datasets. `http://vis.cs.ucdavis.edu/ECS277/data/`.

[12] M. Olkkonen, P. F. McCarthy, and S. R. Allred. The central tendency bias in color perception: Effects of internal and external noise. *Journal of vision*, 14(11):5–5, 2014.

[13] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko. Effectiveness of animation in trend visualization. *IEEE transactions on visualization and computer graphics*, 14(6):1325–1332, 2008.

[14] S. D. Tripp and B. Bichelmeyer. Rapid prototyping: An alternative instructional design strategy. *Educational technology research and development*, 38(1):31–44, 1990.

[15] S. van den Elzen and J. J. van Wijk. Small multiples, large singles: A new approach for visual data exploration. In *Computer Graphics Forum*, vol. 32, pp. 191–200. Wiley Online Library, 2013.