Data Visualization as a Proving Ground for Graduate Studies

By Geoffrey Draper, and Aaron M. Curtis, *Brigham Young University–Hawaii*

We present a course that introduces undergraduate students to the concepts of graduate school, using data visualization as the topic of study. Students read seminal papers in the field of data visualization and implement many of its core algorithms. By the end of the course, students who may not have otherwise considered graduate studies as an option can make an informed decision about whether to attend graduate school or join the workforce upon graduation.

INTRODUCTION

The benefits of a diverse faculty are well recognized [18]. However, achieving this diversity requires a robust pipeline of non-traditional students and underrepresented ethnic groups [22] both attending graduate school and pursuing careers in academia. Undergraduate research is one vehicle for motivating students towards academic careers. Unfortunately, the number of undergraduate research positions is often limited both by funding and by faculty time constraints.

To address this concern, we present an undergraduate course that gives students an exposure to many of the basic concepts of graduate school, without necessarily requiring the faculty time commitment of one-on-one mentored research. Our course is patterned after the idea of a seminar or colloquium course typically taken during a student's first year of graduate studies. The topic of the course is data visualization, since that is a research area with which we are familiar—although in principle any advanced topic would suffice. Enrollment is limited to juniors and seniors majoring in Computer Science. In this course, we introduce students to many of the concepts and activities they will encounter in graduate school—such as reading academic papers, implementing previous work, proposing new projects, making presentations, and writing research reports. We tell our students that by the end of the course, they will say one of three things.

(1) *I want to attend graduate school and study data visualization. I am already familiar with the key papers, people, and techniques in the field, and am ready to hit the ground running!*

Or

(2) *I want to attend graduate school, but do not want to study data visualization. I enjoyed the experience of reading academic papers and making presentations to my peers; however, my research interests lie elsewhere.*

Or

(3) *I now know that I do not want to attend graduate school. I just want to find a job and start making money!*

Although we'd be thrilled if every student echoed sentiment #1, but one of these results is, to us, a successful outcome of the course. The students will have gained important research and critical thinking skills, which are of value in both industry and academia. Those who do go into graduate school will "know what they're getting into," and those who decide against graduate school will have made an informed decision, based on their own interests and career goals. It is far better for them to find out they don't like research before starting graduate school, rather than halfway through!

Topic-based seminars [1,19] and undergraduate courses in data visualization [3,13] are not new. However, to our knowledge, ours is the only one that combines the two: using data visualization to introduce undergraduates to the rigors of graduate-level study.

STUDENT DEMOGRAPHICS

This course began in 2013 as an attempt to boost awareness of graduate studies among our juniors and seniors, many of whom were the first in their extended family to get even a bachelor's degree. The format of the course, with its emphasis on reading and implementing seminal papers, and presenting oral reports, was decided early on. The specific topic of data visualization was chosen simply as a convenient and motivating vehicle to reify the ideals of the course. We have taught this course four times since 2013, with a total of 37 students. As a senior-level CS course at a small university, class sizes tend to be small, allowing for interesting discussions and individualized attention. At its lowest, enrollment was 4 students; the highest was 13 students. Class members came from a mix of nations across multiple geographical regions, as indicated in Table 1.

Table 1: Fewer than half of the students in our course were from North America.

Home Area	# students	Percentage
North America	14	37.8%
Asia	10	27.0%
Pacific Islands (including Hawaii)	9	24.3%
Latin America	2	5.4%
Other	2	5.4%

Unfortunately, we encountered the same gender disparity experienced across the academy [2]. Of the 37 students who have taken our course, only 6 were women. To put this number in context, however, our university produced only 59 CS graduates between 2013 and 2018. Of these, 7 were women. Thus, nearly two-thirds of our CS students (male and female) take this course.

CLASS FORMAT

Although not a "flipped classroom" [15] per se, the structure of class time is quite open-ended and driven by student needs. A typical class period begins with a short quiz based on the previous reading assignment, followed by a group discussion on the reading. To motivate participation, the instructor awards points to students who actively speak up in the discussions. Indeed, participation points are part of the overall grading rubric for the course. After discussing the reading, the instructor opens the class to questions about the current programming assignment. The nature of the questions asked lets the instructor gauge how the students are progressing on their readings and projects. Usually this fills up the remaining class time, but if any is left at the end, a few additional remarks may be given in the form of a traditional lecture.

READINGS

One of the main goals for the course is to guide students through the process of conducting a literature search [24] in the domain of data visualization. This topic was chosen, in part, to address the growing need for "data visualization literacy." [4]

By the end of the course, students will have implemented the fundamental algorithms and have read many of the seminal papers in data visualization; they also know the names of the most influential people in the field.

By the end of the course, students will have implemented the fundamental algorithms and have read many of the seminal papers in data visualization; they also know the names of the most influential people in the field. As a result, they should be able to begin contributing to a research group right away, with little ramp-up time.

In addition to Tufte's *The Visual Display of Quantitative Information* [23], students also read many academic papers in the field. The exact list of papers we study varies from semester to semester but is typically a proper superset of the following.

- *The Use of Faces to Represent Points in K-Dimensional Space Graphically* by Herman Chernoff [6]
- *Parallel Coordinates: a Tool for Visualizing Multi-Dimensional Geometry* by Alfred Inselberg and Bernard Dimsdale [10]
- *Tree-Maps: a Space-Filling Approach to the Visualization of Hierarchical Information Structures* by Brian Johnson and Ben Shneiderman [11]
- *The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations* by Ben Shneiderman [20]
- *The Hyperbolic Browser: A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies* by John Lamping and Ramana Rao [12]
- *Interactive Visualization of Serial Periodic Data* by John V. Carlis and Joseph A. Konstan [5]
- *ThemeRiver: Visualizing Theme Changes Over Time* by Susan Havre, Beth Hetzler and Lucy Nowell [8]
- *Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases* by Chris Stolte, Diane Tang and Pat Hanrahan [21]
- *Artistically Conveying Peripheral Information with the InfoCanvas* by Todd Miller and John Stasko [17]
- *A Visualization Paradigm for Network Intrusion Detection* by Yarden Livnat, Jim Agutter, Shaun Moon, Robert F. Erbacher and Stefano Foresti [14]
- *Interactive Visualization of Genealogical Graphs* by Michael J. McGuffin and Ravin Balakrishnan [16]
- *Baby Names, Visualization, and Social Data Analysis* by Martin Wattenberg [25]
- *ManyEyes: a Site for Visualization at Internet Scale* by Martin Wattenberg, Jesse Kriss and Matt McKeon [26]

In addition to these papers, students also read an assortment of more recent papers from visualization-related conference proceedings. Students also pick one or two extra papers to read individually, which they then share with the class via a formal oral presentation. This gives students an experience similar to what they would encounter in a first-year graduate seminar.

PROJECTS

In addition to reading, the students also have several in-depth programming assignments that require them to implement some of the classic techniques of data visualization, such as bar charts, line charts, scatterplots, parallel coordinates, and tree maps. All assignments are implemented in Java, using only primitive graphics APIs for drawing lines, rectangles, and text. We do not allow students to use visualization-specific frameworks; however, that may be an option for the future. Students thus obtain a deep understanding of the fundamental algorithms of visualization, having built them from the ground up.

PROJECT 1: SIMPLE GUI

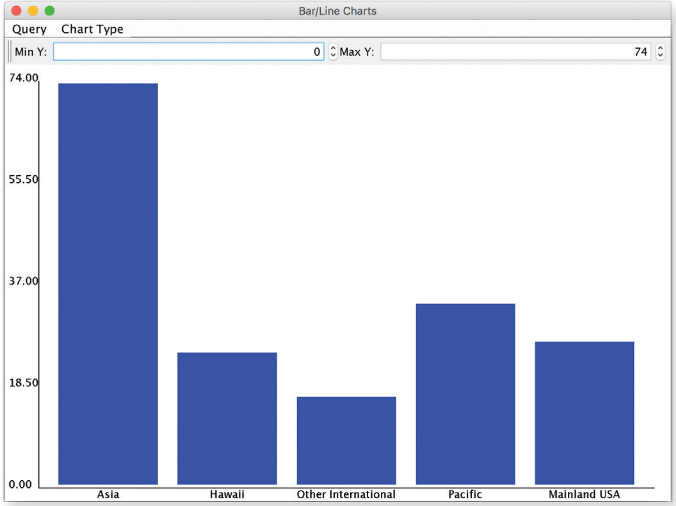
Since not all students may have built a full WIMP-style interface prior to taking this class, the first assignment requires them to build a program that displays a graphics window with multiple menus. Accessing the various menu items changes the graphics rendered in the window.

PROJECT 2: EMBEDDED SQL

Our students take a basic database course prior to their senior year and therefore have some exposure to SQL. But not all students will have invoked SQL statements from within a larger program, stored the results in a data structure, and displayed the results in a graphics window. In this assignment, we provide the students with a "pre-wrangled" data set (an SQL script that creates and populates a database table). The students load this data set onto their computers and write a Java program that connects to the database and runs a variety of queries on it. The queries are initiated at runtime via menu options.

PROJECT 3: BASIC 1D CHARTS

This assignment builds upon the previous two. First, the students get to practice their data-wrangling skills. Instead of an SQL script, we provide them a CSV file (not guaranteed to be bug-free) from which they must create a relational database. They next create a Java program that runs a variety of "select count"-style queries (initiated at runtime via menu options) and displays bar charts and/or line charts based on the results of those queries. The students only make use of a minimal Graphics API for rendering rectangles and line segments; the rest of the work—converting the raw data into graphics primitives, drawing and labeling the axis lines, resizing the chart as the window resizes—is all implemented manually (Figure 1).



Min Y: 0 Max Y: 74

Figure 1: Screenshot of completed Basic 1D Charts assignment. Students write the code to query the database and render the chart using only the minimal Java Graphics2D API. This chart shows the demographics of students in our department. Although our university is in the United States, most of our students are from Asia and the Pacific Rim.

PROJECT 4: 2D SCATTERPLOT

The students are provided with another CSV file, which they must convert to a database. They write a Java program that implements several bivariate queries (again, initiated by the user at runtime via the menus) and renders a scatterplot in the window. As with the previous assignment, the students use only a minimal Graphics API for drawing the axis lines and the individual dots. Additional details, such as converting the result set into a series of (*x*,*y*) coordinates and ensuring the chart scales smoothly as the user resizes the window, are implemented by the student. This assignment is also our vehicle for introducing the V*isual Information-Seeking Mantra* [20]. When the user first initiates a query, the student's program is to render the entire result set ("overview first") as a scatterplot. The user can then "zoom and filter" by dragging a selection-rectangle over a subset of the dots. The student's program must then rescale the chart to show only the dots that fall within the bounds specified by the rectangle. At any time, the user can also access "details on demand" by hovering over any dot in the chart.

PROJECT 5: PARALLEL COORDINATES

Parallel Coordinates [10] is a visualization metaphor that maps attributes in a data set to a set of equally spaced vertical axes in the display. While each axis is rendered at the same height, the scale of each axis is dependent on the minimum and maximum values for that attribute. Each entity in the data set is rendered as a sequence of connected line segments that intersect each axis at the y-position corresponding to the current value of that attribute.

In this assignment, students are provided with a CSV file of multidimensional data from which they must create a database. They then render a parallel coordinates visualization of the data in the traditional form (one axis per table column, one polyline per table row). This assignment gives students additional practice using the Visual Information-Seeking Mantra: in addition to a static rendering, they must also implement the ability for the user to highlight a single polyline by hovering over it (Figure 2). Students must also implement the ability to select a subset of polylines using selection-rectangles (Figures 3 and 4) reminiscent of the *TimeSearcher* user interface [9].

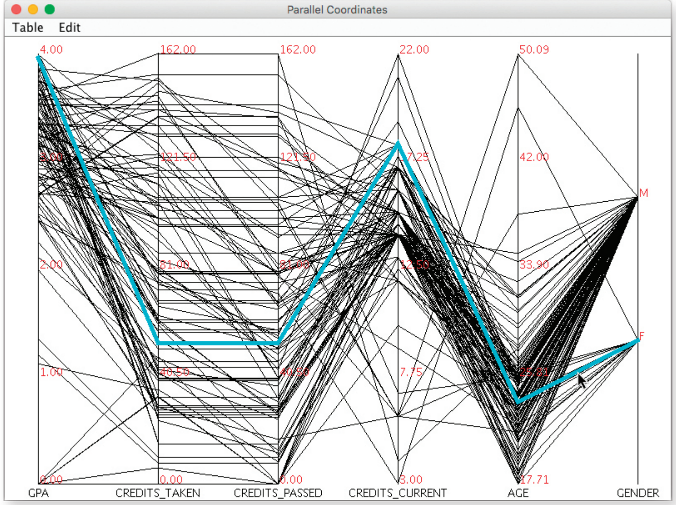


Figure 2: Screenshot of completed Parallel Coordinates assignment. Hovering over an individual polyline highlights it. This chart displays the GPA, credit hours (cumulative and current), age, and gender of students in our department. The highlighted polyline represents a 23-year-old female student with a 4.0 GPA who is fairly new to the program (low number of cumulative credits).

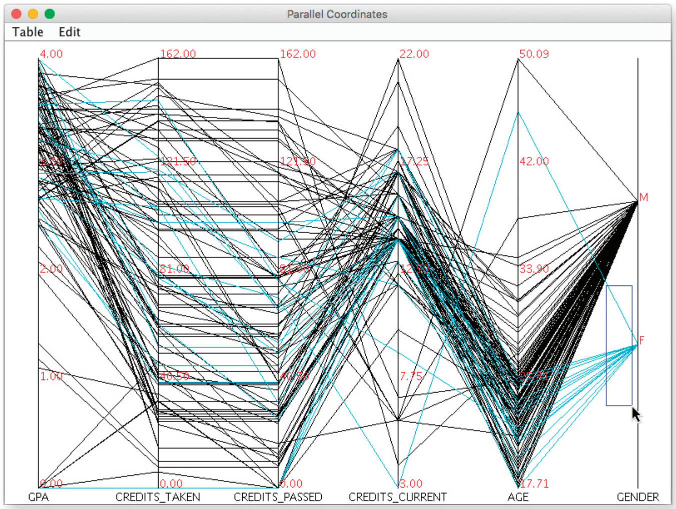


Figure 3: The user drags a "rubber-band" selection-rectangle near the bottom-right corner of the window, selecting a subset of polylines. In this case, the rubber-band is selecting all female students.

PROJECT 6: TREE MAP

A Tree Map [11] is a visualization metaphor for hierarchical tree structures, especially filesystems. Each individual rectangle represents a file, and each group of nested rectangles represents a directory or folder. The area of each rectangle is directly proportional to the size of the file relative to the root directory (the entire window).

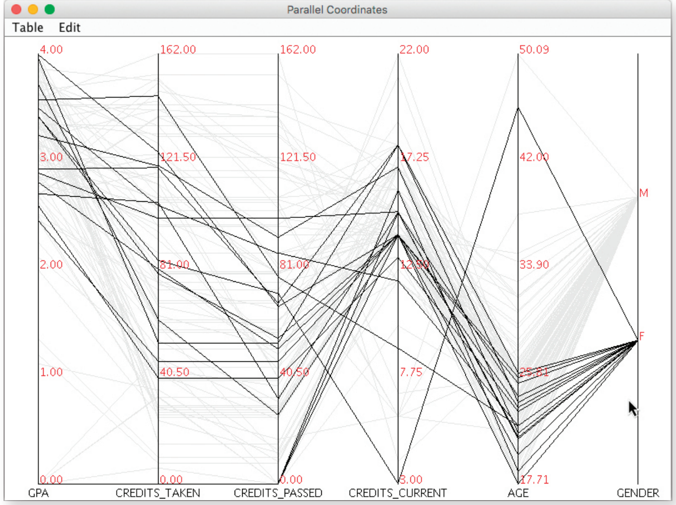


Figure 4: When the user releases the mouse, only those polylines that intersected the rectangle appear in the chart. The rest are faded.

For this assignment, students write a program that renders a tree map visualization of their local filesystem (Figure 5). Their program must include a menu option that allows the user to pick a new root folder for the tree map at runtime. The assignment also requires them to implement several color schemes for the visualization, for example, file type, file age, file permissions. As always, students can use only the standard Java line-drawing and file APIs; all the rest is coded by hand.

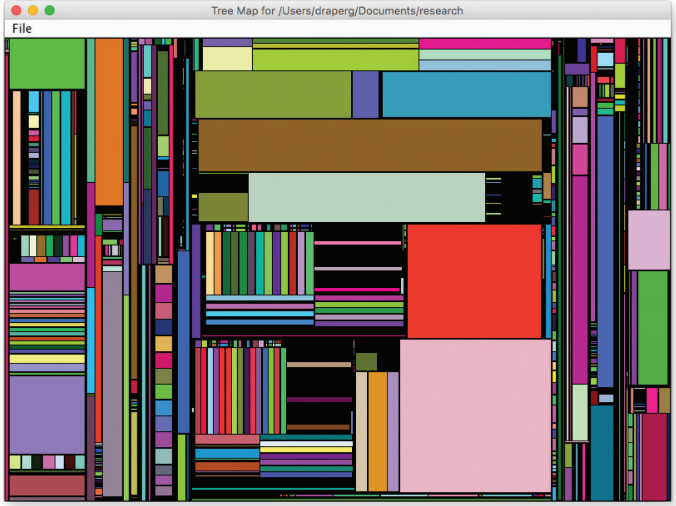


Figure 5: Screenshot of completed Tree Map assignment, visualizing a portion of the first author's filesystem.

FINAL PROJECT

For this assignment, students select a visualization scheme from one of the papers they have read and implement it. Alternatively, they can significantly extend one of the visualizations they already implemented. Less commonly, they may propose and implement their own completely new visualization scheme (Figure 6). This assignment is intended to give students a small taste of proposing and defending a thesis or dissertation. As such, they first write a formal proposal for their project. After completing the implementation, they write a short paper describing their visualization, and do an oral presentation to their peers.

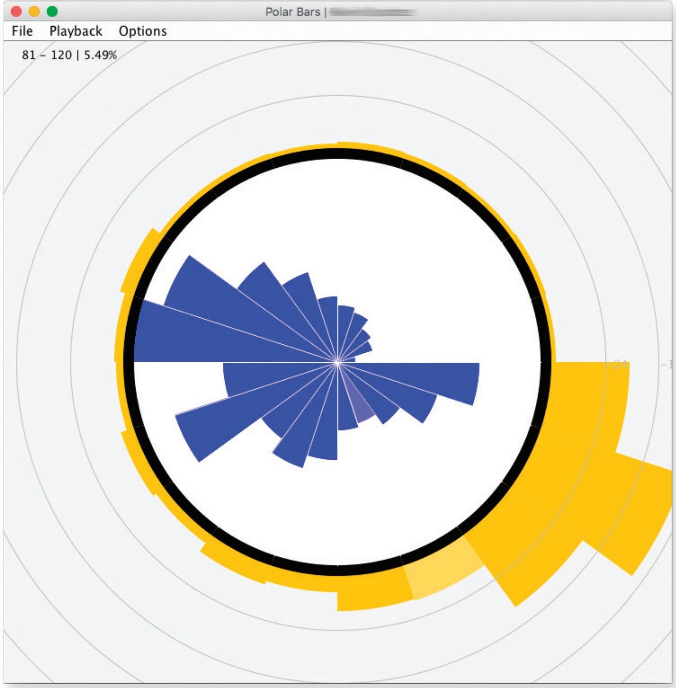


Figure 6: Screenshot of one student's final project, an animated visualization of the various audio frequencies in an MP3 file. As the song plays, the radii of the individual wedges increase and decrease based on the intensity of their respective frequencies. This visualization resembles the polar area chart, first created by Florence Nightingale in the 19th century [7].

OTHER TOPICS

In addition to the readings and programming assignments, other discussion topics in the course relate to the mechanics of graduate school, such as:

- How to select a thesis advisor
- How to fund your education via teaching and/or research assistantships
- The pros and cons of pursuing a master's degree versus a PhD
- Is it better to publish papers based on your dissertation before or after you finish graduate school?

STUDENT FEEDBACK

Quantitative feedback from students for the course has been quite positive. Table 2 shows students' "overall" course rating on a scale from 0 to 7 from the university's course evaluation questionnaire, for each semester we offered the course. The 2nd column shows how many students were enrolled that semester, while the 3rd and 4th columns show the average course rating in our course and the average class rating for the entire college that semester, respectively.

Table 2: Student ratings for our course versus the college-wide average.

Semester	# students	Course rating	College overall
Winter 2013	9	6.38	5.96
Fall 2014	13	6.22	6.09
Fall 2015	11	7	6.15
Spring 2017	4	6.67	5.99

In addition, we surveyed former students who took the course to see whether it influenced their decision to attend graduate school or not. Of the 37 students, 4 of them (10.8%) have attended or are currently attending graduate school, while 3 more (8.1%) tell us they are planning on attending graduate school in the near future. One student who recently attended graduate school told us this about the course:

> *I will say that it [did] a really good job of preparing me what to expect from a graduate level class. Read a white paper, discuss it, [dissect] it, and build some software to demonstrate its concepts. This is how almost all of my Georgia Tech Classes have been setup.*

Another student, who recently defended her master's thesis, said:

> *I think CS490R is a great class to prepare students for graduate school. The research paper reading process in that class mirrors what we do when we do research.*

The following two quotes are from students who have not yet attended graduate school, but who are planning on doing so shortly:

> *I feel that CS490R made me want to attend grad school. Prior to taking the class, I don't recall even considering the option, but the class gave me some interesting perspectives that have stuck with me in regards to continuing my education.*

> *[T]aking CS490R has definitely made [me] want to go to graduate school. ... I am saving up for tuition and taking time after work every day to study GRE, thanks to CS490R!*

Another student, who has not necessarily ruled out graduate school as an option, discovered through our course that his interests lie outside of data visualization:

> *I was more interested in seeing things work and move around then visualization and colors. If I were in grad school I wouldn't do it there.*

A student who was deterred from going to graduate school put it this way:

> *I think it was a good class, especially the one day where you laid out what a masters or a doctorate will do for you. For example, a doctorate is only really for if you want to become a professor or do research. ... Also reading those papers, I realized I did not really want to write one of those papers...*

Two students who did not go to graduate school nevertheless found aspects of the course useful in their respective employment:

> *I am working … as a algorithm engineer now, and I am doing exactly what I was doing in CS 490R - reading papers and trying to apply it in the project I am currently working on. … 490R definitely [helped] me get used to reading research papers and implement ideas into real projects.*
>
> *I remember taking the information visualization class and it did get me thinking about possibly getting a master's degree. Although my decision to go straight into working full time and not pursue a masters was made even before I took the class. … It helped me with my Java programming which I get to use in my job.*

These students' comments seem very much in line with our hoped-for outcomes of this course as outlined in the Introduction.

FUTURE WORK

Because our focus is preparing Computer Science students for grad school, we give several fairly heavy programming assignments requiring students to implement algorithms by hand, without the aid of visualization-specific APIs. However, we may relax this policy over the next few years. Commercial tools like Tableau, domain-specific languages like R or Processing, and high-level APIs like D3 are growing in popularity, and are increasingly at home in both pure visualization research and applied data science. Although we still plan to retain the course's focus on grad school preparation, in the future we may grant more flexibility in the way the programming assignments are implemented.

As another improvement, the next time we teach the course, we plan to survey the students about their plans for graduate school, both before and after the course. In this way, we can better measure whether students' attitudes about grad school change as a result of taking this class.

Finally, to familiarize students with the concept of peer review, we also plan to incorporate a system of peer critiques on programming assignments and oral presentations, specifically in the final project.

CONCLUSION

Having a diverse professoriate in the future requires that we attract and encourage more students—especially those who might not otherwise consider a career in academia—to pursue graduate studies. Although the topic of our course is data visualization, the motivating theme is to expose students to the daily routines of graduate school: academic reading and writing, implementing projects, and oral presentations. By the end of the course, students will have completed the equivalent of a first-year literature search in the domain of data visualization. Regardless of what students choose to do after they graduate, this course will help them make an informed choice about whether grad school is for them. ❖

References

1. Arras, R. J. and Motter, L. The senior seminar in computer science. *SIGCSE Bulletin*. 22, 4 (1990), 29-36.
2. Barr, V. Gender Diversity in Computing: Are We Making Any Progress? *Communications of the ACM*, 60, 4 (2017), 5.
3. Beyer, J., Strobelt, H., Oppermann, M., Deslauriers, L., and Pfister, H. Teaching Visualization for Large and Diverse Classes on Campus and Online. *Pedagogy of Data Visualization Workshop at IEEE VIS 2016* (October 2016).
4. Börner, K. Data Visualization Literacy. *Proceedings of the 27th ACM Conference on Hypertext and Social Media (HT '16)*. (ACM, New York, NY, USA, 2016), 1.
5. Carlis, J.V. and Konstan, J.A. Interactive visualization of serial periodic data. *UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology*. (ACM Press, New York, NY, USA, 1998), 29–38.
6. Chernoff, H. The Use of Faces to Represent Points in K-Dimensional Space Graphically. *Journal of the American Statistical Association*. 68, 342 (1973), 361-368.
7. Draper, G., Livnat, Y., and Riesenfeld, R. A Survey of Radial Methods for Information Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15, 5 (2009), 759–776.
8. Havre, S., Hetzler, B., and Nowell, L. ThemeRiver: Visualizing Theme Changes Over Time. *InfoVis 2000*, 115–123.
9. Hochheiser, H. and Shneiderman, B. Visual Specification of Queries for Finding Patterns in Time-Series Data. *Proceedings of Discovery Science 2001*, 441–446.
10. Inselberg, A. and Dimsdale, B. Parallel coordinates: a tool for visualizing multi-dimensional geometry. *IEEE Visualization '90*, 361–378.
11. Johnson, B. and Shneiderman, B. Tree-Maps: a space-filling approach to the visualization of hierarchical information structures. *VIS '91: Proceedings of the 2nd conference on Visualization '91*. (IEEE Computer Society Press, Los Alamitos, CA, USA,1991), 284–291.
12. Lamping, J. and Rao, R. The Hyperbolic Browser: A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. *Journal of Visual Languages and Computing* 7, 1 (1996), 33–55.
13. von Landesberger, T., Brodkorb, F., Schneider, P., and Ballweg, K. Tool for Teaching Visualization Techniques: Learning and Homework Assignments for Multivariate Data Visualization. *Pedagogy of Data Visualization Workshop at IEEE VIS 2016* (October 2016).
14. Livnat, Y., Agutter, J., Moon, S., Erbacher, R.F., and Foresti, S. A Visualization Paradigm for Network Intrusion Detection. *Proceedings of the 2005 IEEE Workshop on Information Assurance and Security*, 30–37.
15. Maher, M.L., Latulipe, C., Lipford, H., and Rorrer, A. Flipped Classroom Strategies for CS Education. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15)*. (ACM, New York, NY, USA, 2015), 218–223.
16. McGuffin, M.J. and Balakrishnan, R. Interactive visualization of genealogical graphs. *InfoVis 2005*, 16–23.
17. Miller, T. and Stasko, J. Artistically Conveying Peripheral Information with the InfoCanvas. *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '02)*. (ACM, New York, NY, USA, 2002), 43–50.
18. Moody, J. *Faculty Diversity: Removing the Barriers*. (Routledge, 2nd edition, 2011).
19. Rößling, G. 2008. Providing a Seminar++: innovation seminars. *Proceedings of the 13th annual conference on Innovation and technology in computer science education (ITiCSE '08)*. (ACM, New York, NY, USA, 2008), 312.
20. Shneiderman, B. The eyes have it: a task by data type taxonomy for information visualizations. *IEEE Symposium on Visual Languages*. (1996), 336–343.
21. Stolte, C., Tang, D., and Hanrahan, P. Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases. I*EEE Transactions on Visualization and Computer Graphics* 8, 1 (2002), 52–65.
22. Trowler, V. Negotiating Contestations and Chaotic Conceptions. *Higher Education Quarterly*, 69, 295-310.
23. Tufte, E. *The Visual Display of Quantitative Information*. 2nd ed. (Graphics Press, 2001).
24. University of Reading. *Literature Searching*; https://libguides.reading.ac.uk/literature-searching. Accessed 2018 August 7.
25. Wattenberg, M. Baby names, visualization, and social data analysis. *InfoVis 2005*, 1–7.
26. Wattenberg, M., Kriss, J., and McKeon, M. ManyEyes: a Site for Visualization at Internet Scale. *IEEE Transactions on Visualization and Computer Graphics*. 13, 6 (2007), 1121–1128.

Geoffrey M. Draper
Faculty of Math and Computing
Brigham Young University–Hawaii
55-220 Kulanui Street
Laie, HI USA
gmd2@byuh.edu

Aaron M. Curtis
Faculty of Math and Computing
Brigham Young University–Hawaii
55-220 Kulanui Street
Laie, HI USA
aaron.curtis@byuh.edu

DOI: 10.1145/3350745