

ManyLists : Product Comparison Tool Using Spatial Layouts with Animated Transitions

Ran Liu, Tiffany Chao, Catherine Plaisant and Ben Shneiderman
{ranliu, tjyc, plaisant, ben}@cs.umd.edu
Computer Science Department
University of Maryland, College Park
College Park, Maryland, USA 20742

ABSTRACT

Product comparison is a common process consumers perform every day. However, current tools provide limited functions and poor visual designs. We present ManyLists, a product comparison tool that compares products' features using Spatial Layouts with Animated Transitions. In addition color-coding is used to highlight the best features among the products. A usability study with fourteen participants provided evidence that the visualization is easy to learn and the animated transitions are helpful. Finally we propose three guidelines for Spatial Layouts with Animated Transitions: break multi-step processes into small comprehensible steps, show animated transitions for each step, and use spatial layouts to indicate relationships among features.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design

Keywords: List Visualization, Product Comparison, Spatial Layout, Animation.

INTRODUCTION

Product comparison is a common task in our daily lives. Consumers frequently perform product comparisons to make purchase decisions, but the effort to make such decisions is nontrivial. With online shopping environments the difficulty of comparing products increases as the number of products rises and the number of product features available for review increases. Screen space is limited, and it is difficult to keep track of intermediate decisions made while comparing each attribute.

Product selection has been described as including these two steps [11]:

1. Screening: Consumers screen a large set of relevant products, perform quick evaluation on each product, eliminate

the bad choices and then select small number of promising alternatives.

2. Comparing: Consumers will then compare across the smaller number of products, evaluating the attributes in depth, and make a final purchase decision.

In the first step, consumers reduce a great number of choices by eliminating the extreme values such as high/low price or excessive/minimal features. Once they have narrowed down to a small number of alternatives, for example two to four. Then they shift from screening to comparing. Consumers perform in-depth evaluation and comparison among the candidates and make a final decision. For example, to choose a box of cereal, shoppers need to compare dozens of different cereals and choose among different flavors, brands and prices. In addition, they often want to compare the nutritional ingredients to make health-conscious decisions, and of course prices. Most e-commerce websites offer functionalities such as advanced search and recommendation to help users quickly eliminate choices, and lower the effort required for the screening stage, but little innovation can be seen in supporting the in-depth comparison across the small number of alternative products that may have 5 to 50 features. Our proposed interface aims to fill this gap.

Although product comparison is a task most people need to do frequently, we were able to find only a few research projects to facilitate it [2, 11, 20].

The most common approach is to use a static table, with one product per column, and usually one feature per row. A great number of online shopping websites provide such a mechanism to help consumers to compare and choose a product from a small numbers of alternatives. However the following aspects make comparison difficult:

1. It is difficult to find what is similar and what is different.
2. The list of features can be very long therefore existing designs require substantial scrolling.
3. Features in the table can only be compared if they are aligned in the same row.
4. There is rarely an indication of which product has the best value for a given feature (e.g. the lowest price, highest potassium or lowest sodium).

The human memory load required for the comparison is considerable: consumers need to remember which features they consider as more desirable (e.g. "is wall switch better than remote control?") which values they consider more preferable (e.g., "Is more potassium good?"), and which products have more of these features.

In this paper, we applied the design concept from our recent comparison tools: Twinlist. Twinlist was designed to assist clinicians in comparing medication lists to reconcile differences between two lists (e.g. comparing the list of drugs reported by a patient versus the list contained in the physician electronic health record for that patient, so that the current plan for the patient treatment could be updated) [6, 8]. Twinlist highlighted list similarities and differences using spatial layout and animation for only two lists. It focused on the task of merging the lists by selecting drugs to keep and drugs to be discontinued. Here we expand the concept of Spatial Layout with Animated Transition to be used in product comparison.

ManyLists and Twinlist are two different applications with distinct goals. While Twinlist is limited to two lists, ManyLists can compare more than two. Twinlist does not include a notion of "goodness" or "badness" when comparing drugs. ManyLists allows users customize the good/bad of the product features. Finally the goal of medication reconciliation is very different from product comparison (i.e. compiling a new combined list of medications, versus comparing all products to spot the most desirable ones).

Therefore, we present ManyLists, a visualization tool to facilitate the process of product comparison. It aims to meet the following three goals:

1. Support the comparison of at least four products, each with dozens of attributes
2. Provide an overview to help users spot products provide more desirable features with less effort.
3. Allow the customization of positive and negative features.

RELATED WORK

Some work focuses on the screening phase. For example InfoZoom (Focus) [20] is an interactive table viewer combining the focus+context technique [15] and a hierarchical outliner to visualize large feature sets. This approach can visualize a great number of products at once, and allows for rapid narrowing on subsets of interest, but it is not effective for in-depth comparison.

Another approach is to rank all products using the Analytic Hierarchy Process (AHP) for decision-making. AHP was developed to promote improved decision-making involving prioritization of potential alternate solutions through evaluation of a set of criteria elements. Once the hierarchical problem definition has been established, users need to weight all attributes (i.e. here product features) individually at each level, and decisions can be made based on the evaluation of these weights [19]. Visualization has been developed to display the hierarchy of ratings (e.g. using a treemap [2]). Multi-criteria decision analysis [25] deals with structuring and solv-

ing decision and planning problems involving multiple criteria. Users may convert the measures of criteria into a score and then rank all alternatives. Sacrifices will be made in at least one criterion due to constraints [26]. Unfortunately such designs require users to specify numerical weights for individual product feature, which consumers are generally unwilling to do.

Spatial layout is a major component of ManyLists. It is used to present the relationships among all product features. Similarity matrices are well-suited to visualizing similarities between one item and multiple others [10]. [11] used a comparison matrix (CM) approach to aid consumers in making in-depth comparisons. The Semantic Graph Visualizer developed by Andrews et al. [1] comparing graphs side-by-side, and reconciling multiple similar graphs into one. [9] combined parallel coordinates and traditional tag clouds to provide overviews of a document collection. Kang et al. applied a five-column spatial layout to visualize entity resolution in relational data [13]. Jigsaw [21] visualized connections, coordinated various views to encourage broad, dynamic document exploration.

Some previous work focusing on product opinion comparison is tangentially related to our work. Carenini et al. [4] described a multimedia interface for comparing digital cameras. Their interface used multiple histograms to show how opinions on each camera feature vary. DiG [5] is an interactive interface designed to compare multiple products based on product review and features. Unlike a classic parallel coordinates display, the interface only showed a few items. With this design, users can compare the first product and the last product.

Our decision to use animation was partly inspired by Chevalier et al. [7], in which animation and color are used to highlight the differences in document revisions histories. Another inspiration for this paper is the step animation used in SpaceTree [14], an interactive tree browser, that allows dynamic rescaling of branches of the tree to best fit the available screen space. Users can navigate the tree by clicking on nodes to open branches, navigate among siblings, ancestors and descendants and the tree is animated to its new layout in separate steps. The Les Misérables co-occurrence project¹ used rows and columns animation to transition from one state to another when users changed the visualization goal. Animation has the power to make complex transitions intelligible, [12, 16, 18] corroborated this. Bederson et al. [3] examined how animation improves users' ability to understand and reconstruct the information space, providing evidence for adding animated transitions in applications with fixed spatial data. [22] also suggested that step-by-step animation is more congruent to the way the mind understands and represents continuous organized action than continuous animation.

Although animation is attractive and engaging, studies have shown it is confusing. [24] stated static depictions of motion can be equally, if not more, effective than animation in some

¹<http://bost.ocks.org/mike/miserables/>

cases. Gapminder Trendalyzer² used animation in presentations to show trends in data in multiple dimensions. Robertson et al. [17] conducted usability studies to show animation can be confusing in trend animation. They also suggested presenters should ensure the data tells a "clean story" by avoid having too many data points, data points that reverse their tracks over time, or by having points that do not move in synchrony.

EXISTING COMMERCIAL PRODUCT COMPARISON TOOLS

Most of online shopping comparison pages format the features of competing products in a single static table, sometime with simple color coding to highlight similarities or differences.

Contrasting colors suggest whether the feature is desirable or not (Verizon Wireless³), or if the feature is supported by that particular product (Sprint⁴), while variant fonts differentiate distinct features from similar ones (BestBuy⁵, REI⁶). Many other types of websites also provide features for consumer product comparison. Many travel websites, such as Kayak⁷, allow users to compare hundreds of flights, hotels, and vacation packages. They provide control sliders for users to filter the options. But again, once users narrow down the choices, there is no effective design to facilitate in-depth comparisons. Consumer Reports⁸ defines quality metrics and visualizes them using Harvey Balls, but human raters are needed to produce homogeneous numerical ratings for all features.

Two comparison website designs are worth mentioning. Side-by-side camera comparison⁹ allows users to compare specifications of up to twenty cameras. The comparison layout is the table approach similar to other commercial sites but the site allows users to eliminate a product, or move a product to the left or right, directly engaging users into sorting products. Similar designs for comparison are used in Snap-sort.com, Carsort.com and Geekaphone.com¹⁰ for cameras, cars, and smart phones. Users can view the specifications and reviews for a product, and also compare the product with one of its rivals, using a unique comparison interface. In the comparison page, the website will calculate a score for each product from a database and suggest the winner. The advantages of each product are listed accordingly. The design also separated the differences and similarities between the two products.

OUR EARLY WORK

Our first prototype of Twinlist was presented in [8]. It is a list visualization tool designed to facilitate medication reconciliation. It progressively spreads the drugs onto a five column spatial layout and uses stepwise animation to guide users through the complex transformation. Color is used to highlight differences between list items. A pilot user study with

four participants highlighted problems with the design so a polished interface and refined animation was implemented and described in [6].

ManyLists DESIGN AND FEATURES

We now describe the ManyLists interface in more detail. Four examples of real comparison situations related to consumer products were selected to guide our design:

1. Vitamin supplements: here the features are lists of ingredients, available in a standard formats and measures. In almost every case having more of an ingredient is better.
2. Food products: they have nutritional ingredients in mostly standard formats. For some ingredients more is better (e.g. fiber, potassium) but for other ingredients less is better (e.g. cholesterol, sodium). For some ingredients user preferences may vary (e.g. while calories are usually considered bad in wealthy countries, sometime consumers want more calories, or more sugar).
3. Technical products such as laptops: they have many features to compare, but formats are not standardized. Sometimes different brands might have different format for specifications. Numerical values such as hard disk size or RAM are comparable, but other features described are hard to compare.

Each feature of the products is classified as being either the same for all products, unique to a single product, or similar across multiple products. Features that are identical (i.e. the same) in all products are grouped together and placed on the top of the screen. Features that are unique to a product are listed at the bottom of the screen, one column per product. Features that exist in two or more products are aligned to facilitate comparison, and grouped in the center of the screen. Color-coding is used to help users compare the products along those features. For each feature we use a light green color to show the best value among the products. To attract users' attention to only the better features, we also soften the appearance all the less preferable values.

While we believe this special layout can facilitate comparison, it is not a conventional layout of product features. In order to help users understand this layout ManyLists uses step by step animation to reorganize the features and their values. ManyLists starts off as standard multicolumn table view (Figure 1), where all the features of a product are put in the same column. The dataset we are using in the example is four different types of coffee. The nutritional ingredients are listed below each product. The final visualization layout is shown in Figure 3. Five animation steps guide users from the initial layout to the final layout (Figure 2). The animation consists of the following steps: 1. The identical features fade out from their initial location and move one by one to their location in the top "identical" section. 2. Unique features drop down to the bottom of their columns in the "unique" section. 3. The similar elements are aligned, remaining in the middle of the screen. 4. The empty rows are reclaimed to save screen space. 5. The legend for the color-coding is shown and all the best values are highlighted in light green, while the other elements are grayed out.

²Gapminder. <http://www.gapminder.org>

³<http://verizonwireless.com>

⁴<http://shop.sprint.com/>

⁵<http://www.bestbuy.com>

⁶<http://www.rei.com/>

⁷<http://www.kayak.com>

⁸<http://www.consumerreports.org>

⁹<http://www.dpreview.com/products/compare/cameras>

¹⁰<http://snapsort.com/>, <http://carsort.com/>, and <http://geekasort.com/>

Caramel Macchiato	Caffee Latte	Skinny Mocha	White Chocolate Mocha
Calories 190	Calories 130	Calories 140	Calories 420
Total Fat 1g 2%	Total Fat 0g 0%	Total Fat 1.5g 2%	Total Fat 13g 20%
Saturated Fat 1g 5%	Saturated Fat 0g 0%	Saturated Fat 1g 5%	Saturated Fat 9g 45%
Cholesterol 10mg 3%	Cholesterol 5mg 2%	Cholesterol 5mg 2%	Cholesterol 35mg 12%
Sodium 130mg 6%	Sodium 150mg 7%	Sodium 140mg 6%	Sodium 250mg 11%
Total Carbohydrate 35g 12%	Total Carbohydrate 19g 7%	Total Carbohydrate 18g 6%	Total Carbohydrate 64g 22%
Sugars 32g	Sugars 18g	Dietary Fiber 4g 16%	Sugars 60g
Protein 11g	Protein 13g	Sugars 15g	Protein 15g
Vitamin A 15%	Vitamin A 15%	Protein 14g	Vitamin A 15%
Calcium 35%	Calcium 45%	Vitamin A 15%	Vitamin C 2%
Caffeine 150mg	Caffeine 150mg	Calcium 40%	Calcium 50%
		Iron 45%	Caffeine 150mg
		Caffeine 150mg	

Figure 1: **Initial layout: All features in the product column.** Here four coffee drinks are compared. The feature lists of the four products are shown in four separate columns. Features are the nutritional ingredients in each drink and they are unsorted by default.

The diagram illustrates the ManyLists interface transitions. It shows four panels representing different stages of the layout optimization process. Orange arrows indicate the movement of features between panels.

- Top Left Panel:** Shows the initial state where identical features (Caffeine 150mg) are listed separately in each of the four product columns.
- Top Right Panel:** Shows the first transition where identical features are pushed up and merged into a single row at the top of the interface.
- Bottom Left Panel:** Shows the second transition where unique features (Dietary Fiber 4g 16%, Vitamin C 2%, Iron 45%) are moved to their respective columns at the bottom of the interface.
- Bottom Right Panel:** Shows the final optimized state where similar features are aligned across columns, unique features are at the bottom, and empty rows are compacted.

Figure 2: ManyLists animated transitions 1) **Push the identical features up.** All the identical elements among all four lists are merged together in the top row (e.g. Caffeine is the same in all four products so all four "caffeine 150mg" labels converge to the top to become a single label.) 2) **Drop the unique features below.** All the unique features are placed at the bottom in each column and a horizontal line separates them from the similar features. Practically all individual features slide down to their new place. For example, only the fourth product has some Vitamin C so the label "Vitamin C" slide to the bottom of the fourth column. 3) **Align the similar features across two or more products.** Features that appear in more than 2 products are aligned. A very light-colored horizontal striping background organizes the similar features, thereby facilitating comparison. 4) **Compact the display.** All the empty rows resulting from the earlier movements are removed, producing a more compact display.

ManyLists					compare lists	start over	show options
Identical features	Vitamin A 15%		Caffeine 150mg				
Similar features	Caramel Macchiato	Caffee Latte	Skinny Mocha	White Chocolate Mocha			
Preferred values	Calories 190	Calories 130	Calories 140	Calories 420			
	Total Fat 1g 2%	Total Fat 0g 0%	Total Fat 1.5g 2%	Total Fat 13g 20%			
	Saturated Fat 1g 5%	Saturated Fat 0g 0%	Saturated Fat 1g 5%	Saturated Fat 9g 45%			
	Cholesterol 10mg 3%	Cholesterol 5mg 2%	Cholesterol 5mg 2%	Cholesterol 35mg 12%			
	Sodium 130mg 6%	Sodium 150mg 7%	Sodium 140mg 6%	Sodium 250mg 11%			
	Total Carbohydrate 35g 12%	Total Carbohydrate 19g 7%	Total Carbohydrate 18g 6%	Total Carbohydrate 64g 22%			
	Sugars 32g	Sugars 18g	Sugars 15g	Sugars 60g			
	Protein 11g	Protein 13g	Protein 14g	Protein 15g			
	Calcium 35%	Calcium 45%	Calcium 40%	Calcium 50%			
	Unique features			Dietary Fiber 4g 16%	Vitamin C 2%		
			Iron 45%				

Figure 3: **Highlighting the best product for each feature.** ManyLists highlights the best value for each feature in light green.

Another example is the comparison is technical products (Figure 4). In this example, we compared four different digital cameras. The identical features are too many to be displayed in one row, ManyLists will detect the overflow of the features and put an additional row for the identical features. Our tool works fine for the numerical feature values, it suggests the best feature values among the four products. However, it is difficult to work with text features values. For example, the processors feature, the values are "Digic 4", "Venus Engine FHD" and "TruePic V". It is very hard to suggest a best value. In the future, we will need to explore a way to represent text feature values.

Customization

Since a small change in nutritional values can be considered as insignificant, we define a threshold of similarity to classify features as identical for all products, which reduces the number of features to compare. In ManyLists, users can adjust that threshold of similarity, e.g. they can specify that products that vary in calories count by less than 10% should be considered as identical. In the example in Figure 4, the screen size for the four digital cameras are: 3 inch, 3 inch, 3 inch, and 2.7 inch (the fourth feature in the first row of identical features region). Our threshold is set to be 10%, so in this case they are considered the same for all products. A range of "2.7-3 inch" is shown in the identical features on the top. ManyLists also supports users changing the default settings of goodness of the feature, e.g. users who seek to lose weight will favor products with fewer calories, while users who are trying to gain weight will prefer more calories. The option panel that support customized actions is showed in Figure 5. Here in the option panel, users can customize the visualization to suit their needs.

IMPLEMENTATION

ManyLists is a web app, written in HTML5, CSS and Javascript (jQuery library). Currently, ManyLists is not deployed to any

framework, and has no back-end. All the data processing and computation are done in the Javascript. The current version performs best in Google Chrome 14 and newer version.

Data

The data we are using is in preprocessed Json format. For each product, there is one Json array of all the features of that product. Each feature has a unique ID, a corresponding List ID, feature name, and a list of attributes. The ID for each feature is used as the identifier of the feature in future computing and rendering. For each attribute of the feature, there is a attribute name, value, unit and type associate with it. Attribute name is the name of that attribute, such as amount, daily value. Value and unit are the actual feature value and the unit of that value. The unit is optional since some features do not have a unit associated with them. Attribute type currently has two possible values: numeric and text. However in the future, we could expanded to support more types of attributes, such as categorical values and boolean values.

Architecture

ManyLists is use the Model-View-Controller (MVC) architectural pattern. The model handles all the data population and processing: parsing the Json data and calculating relationship between features. The controller manages the animation, handles user input.

The rendering of ManyLists is achieved by populating the background using an HTML5 Table. The product features are populated by Javascript using absolute positioning and layered on top of the background. Positions of the features in different animation stages are calculated in advance based on their relationships.

Animation

Our animation effects are implemented using the Effects provided by JQuery. The major part of the stepwise animation

ManyLists										compare lists	start over	show options		
Identical features	Body type	Compact	Sensor type	CCD	Articulated LCD	Fixed	Screen size	2.7–3inch	Touch screen	No	Live view	Yes	Built-in flash	Yes
	Self-timer	Yes	USB	USB 2.0	Battery	Battery Pack								
Similar features	Cannon	Panasonic			Olympus			FujiFilm						
Preferred values	Megapixels	10.0megapixels	Megapixels	10.1megapixels	Megapixels	10.0megapixels	Megapixels	12.2megapixels						
	Processor	Digic4	Processor	Venus Engine FHD	Processor	TruePic V								
	ISO	Auto, 80, 100, 125, 160, 200, 250, 3	ISO	Auto, Hi Auto (6400–12800), 80, 100	ISO	Auto (100 – 800), 100, 200, 400, 800	ISO	Auto, 64, 100, 200, 400, 800, 1600						
	White balance presets	7	White balance presets	5	White balance presets	6	White balance presets	6						
	Image stabilization	Optical	Image stabilization	No	Image stabilization	Sensor-shift	Image stabilization	No						
	Optical zoom	3.8x	Optical zoom	3.8x	Optical zoom	4x	Optical zoom	5x						
	Digital zoom	4x	Digital zoom	4.5x	Digital zoom	4x	Digital zoom	6.3x						
	Manual focus	Yes	Manual focus	Yes	Manual focus	Yes	Manual focus	No						
	Screen dots	461000	Screen dots	460000	Screen dots	614000	Screen dots	230000						
	Flash range	6.5m	Flash range	7.2m	Flash range	8.6m	Flash range	3.1m						
	Storage included	0MB	Storage included	40MB	Storage included	55MB	Storage included	13MB						
	Weight	195g	Weight	271g	Weight	275g	Weight	135g						
			Sensor photo detectors	10.1megap	Sensor photo detectors	10.1megap								
Unique features	Wireless	EyeFi			External flash	Yes			Waterproof	Yes			Shockproof	Yes
									Dustproof	Yes				

Figure 4: **Comparison of four digital cameras:** the identical features are too many to be displayed in one row. When this happens, all the other features slide down, the remaining identical features align in the second row on the top of the page.

ManyLists										compare lists	start over	show options		
Dataset:	Coffees	Jump to step:	separate	identical	unique	similar	compact	highlight						
Threshold:	0	%	Attributes	Calories	Total Fat	Saturated Fat	Cholesterol	Sodium	Total Carbohydrate	Sugars	Protein	Vitamin		
				A Calcium	Caffeine	Dietary Fiber	Iron	Vitamin C						

Figure 5: **Option panel.** ManyLists allows users to change datasets, replay the lists-matching animation one step at a time, change default threshold setting to consider the feature values that vary within a small range as identical features, and choose the desired/undesired features. The desired features are in green.

is done by using the animate() function with the help of delay(). Each animation takes 1.6 second, including a 0.4 second pause between two steps. We have experimented with various animation speed, from 1.2 second to 2 seconds, and 1.6 seems to be a satisfying speed that allows comprehension while not slowing users too much. In the future, we will provide options for users to adjust their preferred animation speed.

EVALUATION

We performed a user study with fourteen participants to determine if ManyLists was easy to learn, and to see if the animation helped users understand the novel spatial layout. We also wanted to observe what strategies users chose to perform product selection, and gather feedback and suggestions for further improvement. The user study used the coffee dataset shown in the Figure 1. It was chosen to be familiar to our users, yet moderately complex. We recruited fourteen partic-

ipants (six female and eight male) from various departments in University of Maryland. Thirteen of them were graduate students and one was an undergraduate. None of the participants were familiar with the ManyLists interface.

Procedure

Because our first goal was to validate the design of a consumer-oriented online tool, we did not provide any training. We described the general context (selecting a coffee drink) and encouraged participants to think aloud while performing the tasks. During the tests, we observed the participants' behavior and recorded their comments during and after the tasks.

Participants were asked to start by clicking on "compare lists", and to describe their understanding of the display. The goal for this task was to observe how users reacted to the animation, and how well they understood the layout. After they watched the animation, and described their initial understanding of the spatial layout and the animation, we asked

them to point to the unique features, identical features and similar features among the four products. This task is designed to see if the users really understood the spatial layout. Following this, the users were asked about the highlighted items, such as "Do you understand why the items are highlighted?", "Do you understand why some of the best options are the minimum values while some of them are the maximum of the four". The next task was to find their favorite coffee out of the four. This question was open-ended, to observe how users compare products, and if our visual design was found satisfying for them to select a product.

After the tests, users were debriefed with a series of questions regarding their overall impression of ManyLists, the features they liked/disliked, and whether they would choose to use ManyLists when they need to buy a product. We also asked users to make suggestions and rate the design for ManyLists.

Results

Eleven of fourteen of our participants were able to understand the animation and the new spatial layout after playing the lists-comparing animation only once. Three of them were confused by the animation at first, two of those three understood it after replaying it one more time. One of them needed us to replay the animation step-by-step, and needed an explanation of each step. Five of fourteen participants complained that the animation for identical features was too fast. One of them suggested we could blink or highlight all the identical features first, and then match them together, so they could have more time to respond to what was happening. Three of them were confused by the highlighted features at first, but after they carefully examined the first features and their values, they all understood the meaning of the green highlighting.

We observed some interesting patterns in their product comparison procedures: 1) All of the participants took into account the total number of best features. 2) Some of them applied more weight/importance to some of the features. For example, one of the participants was on diet, so she cared about calories and fat intake more than all the other features. 3) Some of the participants started with the two products with the most highlighted features, while others tended to start by eliminating choices (e.g. they eliminated the one with the highest calories). 4) Almost all participants first narrowed their choices down to two options. 5) Half of the participants also considered the second best values (i.e. not only the green best one).

During the debriefing session, typical comments included: "very cool", "I like the animation", "the color is very good, they are readable but they do stand out", "it is really useful", "it is good that it automatically takes out the identical and unique features", "it is very clear", and "I like that you suggest the best values for different features". Common suggestions included: 1) slow down the animation speed. 2) Add features to let users to set the importance for each feature. 3) Let users be able to eliminate products. Finally when we asked users to rate the overall impression of ManyLists on a 1-9 scale, where 1 is failing and 9 is exemplary, they gave an average rating of 7.7.

The user study results suggest that the majority of participants could learn ManyLists' visual design. They found the step-by-step animation to be helpful in understanding the differences between products. One participant needed help understanding the animation steps, especially the identical matching step. He reported that too many things are moving too fast in the beginning, and he did not notice the label for the identical features. Similarly, a few other participants also complained about the animation for identical features, but they were able to discover the meaning of the spatial layout by reading the labels. We believe that if we make the labels more prominent and have them appear first, users would comprehend the layout more reliably. A further improvement would be to wait for a second before the identical features begin moving, so as to inform users that these are identical features. Also, blinking the four identical features once or twice before moving them may draw users' attention to they perceive the process of identical features superimposing on each other. This will give the users more time to read the values of the features before they are overlaid. Another suggestion was to adjust the animation speed for the identical features: slow down the first group of identical features, and speed up for the remaining ones. Doing this will give users more time to understand the animation, but not too time consuming to be annoying.

DESIGN GUIDELINES

From our experiences, observations and the user study, we learned a few lessons for designing visualizations using Spatial Layouts with Animated Transitions. We generalized these lessons into the following three design guidelines for future developers.

(1) Break multi-step processes into small comprehensible steps.

This is a classic strategy for explaining a complex process, allowing users to absorb each step easily. Animations that are too fast or too complex make it difficult for users to understand, violating the Apprehension Principle [22]. In our early design stage for Twinlist, we did not separate animation into steps. We observed that users were often confused by the sudden changes of the spatial layout. By adding the animation, users often experienced less confusion and could learn the spatial layout faster. The use of step-by-step animation is similar to multi-step diagrams for the assembly of furniture. The goal is to provide users with enough guidance to understand complex spatial transitions.

(2) Show animated transitions for each step, which users can follow.

Animation is a method to draw users attention naturally, but done badly, it can be misleading and confusing. Tversky summarized three animation aphorisms: "Seeing isn't perceiving; perceiving isn't understanding; showing isn't explaining" [23]. To avoid these failures, designers should produce animations with a reasonably slow pace, and only a moderate number of items moving at once. In our early Twinlist prototype design, all the identical features moved together in the first stage of the step. One user was overwhelmed by this busy animation, so we only moved one

group of identical features at a time. This is crucial for beginners, but slow animation may irk frequent users. This requires designers to provide options for users to adjust animation speed.

(3) Use spatial layouts to indicate relationships among features.

From the five-column layout we used in the Twinlist project, and the three regions we presented in ManyLists, the spatial layouts represent the relationships among items. The five-column layout is straight forward, but in ManyLists, the spatial layout is more complex. First, the identical items are horizontally aligned on top, but both unique and similar items are vertically aligned. Moreover, the similar items also have correlations horizontally. To minimize the users' confusion, we used background shading and vertical and horizontal bars to help users understand the direction of the features' relationships.

DISCUSSION

Figures show an example comparison of 4 products. The design scales easily without modification to 6-10 products for consumers using larger screens and/or smaller font size. Such a number is adequate for the detailed comparison following the screening process. If space is an issue or more products need to be compared, traditional techniques can be used to compact the table (e.g. reducing column widths to the longest label length, abbreviating long labels). In addition sorting of the columns by their number of best (green) features will lead users to the more desirable products first for detailed comparison. Repeating the layout of the similar and unique features for multiple groups of 4 or 5 products may be investigated to replace the problematic horizontal scrolling by vertical scrolling, etc.

The purpose of the stepwise animation is to guide user through complex visualization step. It will help users understand the three region spatial layout. But after the users learned the visualization, they should be able to speed up the animation or simply switch off the animation.

CONCLUSIONS

Product comparison is common in our daily lives, but currently the existing product comparison tools use only simple visual designs without animation. We introduced a new visualization, ManyLists, to facilitate product comparison using Spatial Layouts with Animated Transitions. Our user study gave us encouragement that the ManyLists design is easy to learn and effective. We encourage future developers to continue building animated visualizations by generalizing the three design guidelines: 1) Break multi-step processes into small comprehensible steps. 2) Slow animated transitions for each step, which users can follow. And 3) Use spatial layouts to indicate relationships among features.

ACKNOWLEDGEMENTS

We would like to thank Dr. Barbara Tversky for her insights and valuable comments. Our thanks also go to Tak Yeon Lee and Jyothi Vinjumur for their valuable ideas and feedback.

REFERENCES

1. K. Andrews, M. Wohlfahrt, and G. Wurzinger. Visual graph comparison. In *Information Visualisation, 2009 13th International Conference*, pages 62–67, July 2009.
2. T. Asahi, D. Turo, and B. Shneiderman. Using treemaps to visualize the analytic hierarchy process. *Information Systems Research*, 6:357–375, 1995.
3. B.B. Bederson and A. Boltman. Does animation help users build mental maps of spatial information? In *Proceedings of the 1999 IEEE Symposium on Information Visualization, INFOVIS '99*, pages 28–, Washington, DC, USA, 1999. IEEE Computer Society.
4. G. Carenini and L. Rizoli. A multimedia interface for facilitating comparisons of opinions. In *Proceedings of the 14th international conference on Intelligent user interfaces, IUI '09*, pages 325–334, New York, NY, USA, 2009. ACM.
5. S. Carter, F. Chen, A. S. Muralidharan, and J. Pickens. Dig: a task-based approach to product search. In *Proceedings of the 16th international conference on Intelligent user interfaces, IUI '11*, pages 303–306, New York, NY, USA, 2011. ACM.
6. T. Chao. Visual techniques for medical reconciliation: spatial metaphor, animated explanation, and flexible decision-making. <http://www.cs.umd.edu/localphp/hcil/tech-reports-search.php?number=2011-33>, November 2011.
7. F. Chevalier, P. Dragicevic, A. Bezerianos, and J. Fekete. Using text animated transitions to support navigation in document histories. In *Proceedings of the 28th international conference on Human factors in computing systems, CHI '10*, pages 683–692, New York, NY, USA, 2010. ACM.
8. L. Claudino, S. Khamis, R. Liu, B. London, J. Pujara, C. Plaisant, and B. Shneiderman. Facilitating medication reconciliation with animation and spatial layout. *Proceedings of the Workshop on Interactive Healthcare Systems*, pages pp. 2–6, 2011.
9. C. Collins, F.B. Viegas, and M. Wattenberg. Parallel tag clouds to explore and analyze faceted text corpora. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, pages 91–98. IEEE, 2009.
10. B. Cornelissen and L. Moonen. Visualizing similarities in execution traces. In *Proceedings of the 3rd Workshop on Program Comprehension through Dynamic Analysis (PCODA)*, pages 6–10, 2007.
11. G. Häubl and V. Trifts. Consumer decision making in online shopping environments: The effects of interactive decision aids. *Marketing Science*, 19(1):pp. 4–21, 2000.

12. J. Heer and G. Robertson. Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1240–1247, November 2007.
13. H. Kang, L. Getoor, B. Shneiderman, M. Bilgic, and L. Licamele. Interactive entity resolution in relational data: A visual analytic tool and its evaluation. *Visualization and Computer Graphics, IEEE Transactions on*, 14(5):999–1014, sept.-oct. 2008.
14. C. Plaisant, J. Grosjean, and B.B. Bederson. Spacetree: supporting exploration in large node link tree, design evolution and empirical evaluation. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 57–64, 2002.
15. R. Rao and S. Card. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence, CHI '94*, pages 318–322, New York, NY, USA, 1994. ACM.
16. G. Robertson, K. Cameron, M. Czerwinski, and D. Robbins. Polyarchy visualization: visualizing multiple intersecting hierarchies. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves, CHI '02*, pages 423–430, New York, NY, USA, 2002. ACM.
17. 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, November 2008.
18. G. G. Robertson, J. D. Mackinlay, and S. K. Card. Cone trees: animated 3d visualizations of hierarchical information. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology, CHI '91*, pages 189–194, New York, NY, USA, 1991. ACM.
19. T. Saaty. How to make a decision: The analytic hierarchy process. *European Journal of Operational Research*, 48(1):9–26, 1990.
20. M. Spence, C. Beilken, and T. Berlage. Focus: the interactive table for product comparison and selection. In *Proceedings of the 9th annual ACM symposium on User interface software and technology, UIST '96*, pages 41–50, New York, NY, USA, 1996. ACM.
21. J. Stasko and R. Görg, C. Spence. Jigsaw: supporting investigative analysis through interactive visualization. *Information Visualization*, 7(2):118–132, 2008.
22. B. Tversky. Visualizing thought. *Topics in Cognitive Science*, 3(3):499–535, 2011.
23. B. Tversky, J. Heiser, S. C. Lozano, R. MacKenzie, and J. Morrison. *Enriching animations*. Cambridge University Press, Cambridge, 2007.
24. B. Tversky, J. Morrison, and M. Betrancourt. Animation: can it facilitate? *International Journal of Human-Computer Studies*, 57(4):247–262, 2002.
25. S. Zionts. Mcdm: If not a roman numeral, then what? *Interfaces*, 9(4):pp. 94–101, 1979.
26. S. Zionts and J. Wallenius. An interactive programming method for solving the multiple criteria problem. *Management Science*, 22(6):652–663, February 1974.