2012

ANIMATED TRANSITION IN SIMILARITY-BASED TILED IMAGE LAYOUT

Huan Zhang
Michigan Technological University

ANIMATED TRANSITION IN SIMILARITY-BASED TILED IMAGE LAYOUT

By

Huan Zhang

A THESIS

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

(Computer Science)

MICHIGAN TECHNOLOGICAL UNIVERISTY

2012

This thesis, “Animated Transition in Similarity-Based Tiled Image Layout,” is hereby approved in partial fulfillment of the requirements for the Degree of MASTER OF SCIENCE IN COMPUTER SCIENCE.

Department of Computer Science

Signatures:

Thesis Advisor	
	Dr. Chaoli Wang
Committee Member	
	Dr. Scott A. Kuhl
Committee Member	
	Dr. Robert Nemiroff
Department Chair	
	Dr. Steven M. Carr
Date	

To my dear parents, my lovely sister

and

To myself

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Foremost, I owe my deepest gratitude and sincere respect to my advisor, Dr. Chaoli Wang for the continuous and valuable guidance and support of my M.S. study and research, for his patience, enthusiasm, motivation and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. It is to him that I dedicate this work.

I gratefully acknowledge Dr. Robert Nemiroff who maintains website APOD (The Astronomy Picture of the Day) with Jerry Bonnell at NASA and provides me with the research data. I also sincerely appreciate his helpful suggestions and comments for the revision of this thesis.

I thank all my committee members: Dr. Chaoli Wang, Dr. Scott Kuhl, and Dr. Robert Nemiroff, for their time and valuable input to this research and their suggestive comments for my future work.

I would like to thank my fellow labmates in Computer Science Graphics Lab, Jun Tao and Fang Ruan, for the stimulating discussions and selfless help. Also I offer my regards and blessings to all my friends who encouraged and supported me in the research and enriched my study and life in the United States.

Last but not least, I would like to thank my family: my father Xinqiu Zhang, my mother Xiaodie Ma and my younger sister Le Zhang, for giving me a warm and loving family, for always supporting me spiritually throughout my life.

ABSTRACT

Effective techniques for organizing and visualizing large image collections are in growing demand as visual search gets increasingly popular. iMap is a treemap representation for visualizing and navigating image search and clustering results based on the evaluation of image similarity using both visual and textual information. iMap not only makes effective use of available display area to arrange images but also maintains stable update when images are inserted or removed during the query. A key challenge of using iMap lies in the difficult to follow and track the changes when updating the image arrangement as the query image changes.

For many information visualization applications, showing the transition when interacting with the data is critically important as it can help users better perceive the changes and understand the underlying data. This work investigates the effectiveness of animated transition in a tiled image layout where the spiral arrangement of the images is based on their similarity. Three aspects of animated transition are considered, including animation steps, animation actions, and flying paths. Exploring and weighting the advantages and disadvantages of different methods for each aspect and in conjunction with the characteristics of the spiral image layout, we present an integrated solution, called AniMap, for animating the transition from an old layout to a new layout when a different image is selected as the query image. To smooth the animation and reduce the overlap among images during the transition, we explore different factors that might have an impact on the animation and propose our solution accordingly. We show the effectiveness of our animated transition solution by demonstrating experimental results and conducting a comparative user study.

Chapter 1

Introduction

With the booming of online image archiving and photo sharing technologies, efficient browsing and searching large image collections is becoming increasingly popular. Researchers have shown that arranging a set of thumbnail images according to their similarity does help viewers narrow down their search and identify images of interest (30-31). Recently, we have presented iMap which tiles images in a collection based on their visual and textual similarity (37). An example is shown in Figure 1.1 (a). Within a rectangular display area, iMap shows the query image at the center as the focus and arranges other images with decreasing similarity ranks as the context following a spiral pattern. By default, we display the focus image in a normal size and reduce the width and height by half for the successive layers. The user can adjust the number of repetition layers needed to keep the current width and height. Although effective, a key challenge of using iMap lies in the difficulty to follow and track the changes when updating the image arrangement as the query image changes. Figure 1.1 (a) and (b) show such an example. In this work, we specifically aim to help users comprehend the changes and answer questions such as “how many images previously displayed in the old layout do not show up in the new layout?” and “for an image that remains in the new layout, does its rank increase or decrease?”

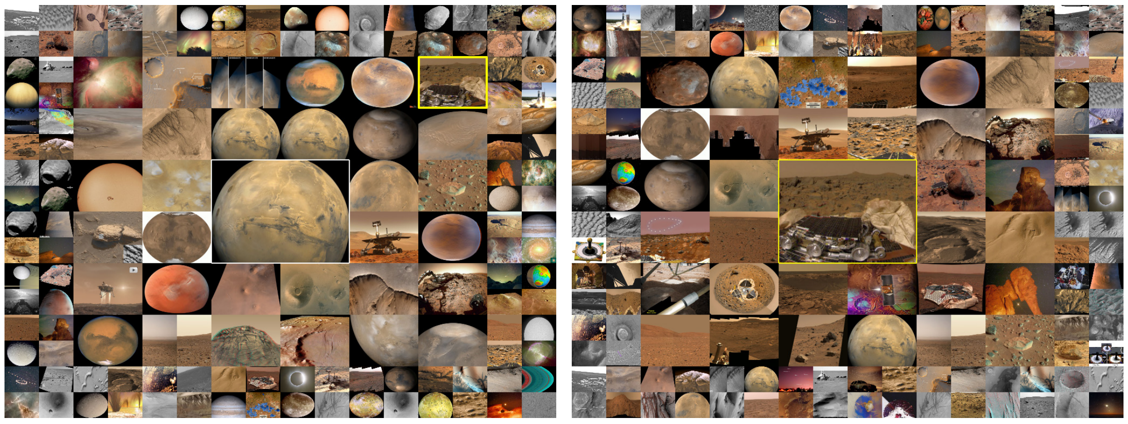

(a) Old layout (b) New layout

Figure 1.1: iMap query without animated transition
In the old layout (a), an image highlighted in yellow boundary is selected as the new query image. The new layout (b) after the selection is shown. The new query image switches to the center and other images around rearrange their positions.

Animation is a promising approach to facilitate the perception of changes when transforming from one layout to another. Previous research has found that animated transition can significantly improve graphical perception of changes between statistical data graphics (14), spatial relationships perception (2,5), and decision making (12). In dynamic graphic drawing, it was also suggested that tasks related to dynamic evolution might be better handled via animation compared to small multiples (8). However, others have also warned that animation could become problematic if not used appropriately. Compared to static depiction of trend, animation is least effective in trend analysis (29). Besides, animation shows no significant improvement in window navigation (10). Therefore, whether or not animation can facilitate image browsing and querying in our similarity-based tiled layout requires a careful and thorough study.

In our study, we explore three aspects, animation steps, animation actions, and flying paths, to design effective animated transition. In conjunction with the characteristics of the spiral image layout, we evaluate the advantages and disadvantages of different methods for each aspect and present an integrated solution. The new AniMap we develop is an enhanced version of iMap, introducing animated transition from an old layout to a new layout when a different image is selected as the query image. We argue that by breaking down a complex animated transition into several logical stages and making each stage simple enough to comprehend, we can significantly improve the understanding of image changes. By exploring different factors that might have an impact on the animation, we propose our solution for smoothing the animation and reducing the overlap among images during the transition.

To verify the effectiveness of our animated transition solution, we perform a user study to evaluate the three animation conditions: no-animation, one-step animation and multi-step animation (used in the AniMap). We are interested in understanding how effective these animation conditions are under various tasks ranging from locally structured tasks (e.g., the image's similarity rank change) to globally structured tasks (e.g., the overall number of image changes between the old and new layouts). Our results show that, in terms of tracking the changes between the old layout and the new layout, multi-step animation is significantly more accurate than one-step animation and no-animation, and is noticeably faster than one-step animation.

Chapter 2

Background

2.1. iMap[€]

iMap was our previous research accomplishment guided by Dr. Chaoli Wang and involved with the work from John P. Reese, Jun Tao, Dr. Robert Nemiroff and me, in which Dr. Robert Nemiroff provided us with the research data. iMap is an analysis and visualization framework that supports effective searching, browsing, and understanding of a large image collection.

Effective techniques for organizing and visualizing large image collections are in growing demand as visual search gets increasingly popular. An emerging trend is that images are now often tagged with names, keywords, hyperlinks and so on to improve the search and understanding. Striving for innovation on the organization and interaction aspects of image search rather than the search algorithm itself, iMap explores how to arrange images in a layout for better viewing and how to leverage the connection between image and text for better interaction. iMap analyzes image content and design measures to evaluate image similarity using both visual and textual information, arranges similar images close to each other and leverages a treemap representation to visualize image search and clustering results to facilitate the understanding. iMap strikes a good balance

[€] "The material contained in this section was previously filed as the *Technical Report CS-TR-12-01, Department of Computer Science, Michigan Technological University, 2012*.
Available from: *http://www.mtu.edu/cs/research/papers/pdfs/CS-TR-12-01.pdf*"

among simplicity, intuitiveness, and effectiveness and brings several benefits such as screen space utilization, occlusion minimization, and stable update.

2.1.1. Data Sets

All images in the iMap came from a website the Astronomy Picture of the Day (APOD) (26), a popular online astronomy archive, which was maintained by Dr. Robert Nemiroff and Jerry Bonnel at Michigan Technological University and NASA. Everyday APOD features a picture of our universe, along with a brief explanation written by a professional astronomer. Since its debut in June 1995, APOD has archived thousands of handpicked pictures, which makes it the largest collection of annotated astronomical images on the Internet. This makes it perfect for iMap to include textual information into similarity analysis and interaction design. iMap collected APOD webpages that contain meta-tagged keywords (since Sep. 26, 1997) till a cut-off date (Apr. 3, 2010), and videos were excluded from the collection. For images in the GIF format though, the first frame was extracted as the representative of the entire image sequence. The resulting data set consists of 4560 images with text information including keywords, hyperlinks, and explanations extracted from the HTML files.

2.1.2. Image Similarity Analysis

A visualization based on image similarity metric has been proved to be helpful. Rodden et al. (31) investigates whether it benefits users to have sets of thumbnails arranged according to their mutual similarity, so images that are alike are placed together. Their two experiments indicate that automatically arranging a set of thumbnail images according to their similarity does indeed seem to be useful when designers wish to narrow down their requirement to a particular subset.

Researchers in the information retrieval and image processing communities have proposed a large number of image similarity metrics, where content-based image analysis is at the heart of it. Its primary goal is to organize digital image archives by their visual content. Image features capture visual properties of an image, either globally or locally. To extract image features that help perform meaningful classification and retrieval, researchers have utilized key visual contributions such as color (30-31,33) texture (23,30-31), shape (4), and salient points (24). Advances have been made in both deriving new features and constructing signatures based on these features (9). For example, Yang et al. (46) extracted the high level semantic contents of images based on concept-sensitive image content analysis techniques to support effective image exploration. iMap leverages the color and spectrum information together with the grayscale version of images for similarity analysis.

2.1.3. Image Distance Measure

Measuring the similarity or distance between two images is central to any image searching or clustering tasks. Image themselves provide direct cues to visual comparison. Textual information associated with images, whenever available, gives additional hints for us to evaluate their similarity or difference. iMap compares images using their visual and textual information from multiple perspectives and defines several partial distances, including grayscale image distance, spectrum image distance, color histogram distance, keyword list distance and hyperlink list distance. The overall distance measure is a weighted summation of all the partial distances.

For visual distance, we consider three aspects of images, namely, grayscale image distance, spectrum image distance, and color histogram distance. To obtain textual features, meta-tagged keywords in the HTML header and hyperlinks in the explanation are extracted. With the overall image distance defined, we build a symmetric distance matrix recording the distance between any two images in the collection. During image search, the user selects a query image and all other images in the collection are ranked accordingly. The user can change the weights for partial distances to update the distance matrix and search results. For more details of image distance measurement, please refer to (37). Once image are ranked, image layout is important as it determines how image should be arranged for viewing. For iMap layout, we will discuss in later in the following Section 2.2.

2.2. Large Image Collection Layout

The most common way to organize a large collection of images is based on a two-dimensional grid of thumbnails, but it enforces a uniform thumbnail aspect ratio. Further, only parts of the dataset can be seen within line of sight when the image collection is excessively large. To overcome the disadvantages of traditional row and column layout, over the years, researchers have explored many other different layouts for large image collections, such as spreadsheet-like layouts (17,20), scatterplot layouts (18,45), spiral layouts (30,34), force-directed layouts(13,25), Voronoi diagram layouts (6), and some variants of treemap layouts (e.g., quantum treemaps and bubblemaps (3)).

2.2.1. Treemap Layout

Treemap is one kind of layouts to display hierarchical collections by using nested rectangles. It partitions two-dimensional rectangle into sub-rectangles that have an area proportional to a specified dimension on the data. This original “slice-and-dice” treemap layout generates rectangles of arbitrary aspect ratios. Over the years, researchers have been explored many variants of treemap layouts, and each treemap has its advantages and weaknesses. Squarified treemaps (7) create rectangles with smaller aspect ratios but give up on node ordering. Ordered treemaps (32) offer a good tradeoff among stable updates, order preserving, and low aspect ratios. Quantum treemaps developed for PhotoMesa (3) guarantee that the regions showing groups of photos have dimensions that are integer multiples of the dimensions of the photos (i.e., they must be sized to contain quantum, or indivisible contents). Spiral treemaps (35) place nodes along the spiral pattern which guarantees that neither the overall pattern nor the specific node ordering will change drastically as the data evolve over time.

2.2.2. Spiral Layout

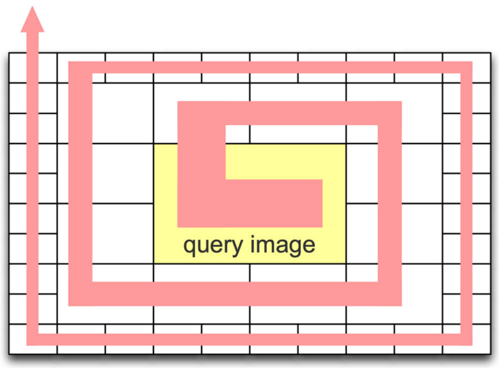
Spiral layout is to arrange images along with the spiral orbit. Torres et al. (34) proposed a spiral representation to explore image query results, which keeps the viewer’s focus on the query image as well as other images that are most similar to the query image. Mackinlay et al. (22) implemented a spiral calendar to make efficient use of screen space and also provided an animation to help users track changes. Helfman et al. (15) developed a two-iteration spiral layout algorithm to display large image collections. Images are first positioned in a spiral cluster using their image rating scheme and then each cluster is positioned in a larger spiral of clusters using a rating scheme based on the number of images in each cluster.

2.2.3. Image Layout Principle

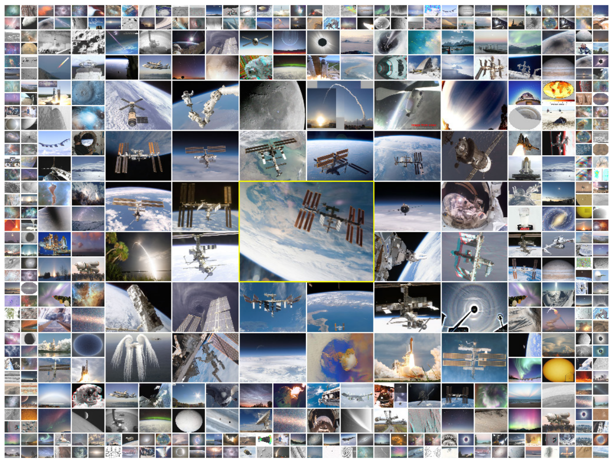
No matter what kind of layout pattern is used, for a large image collection, it is desirable to maintain a good visual overview while allowing flexible exploration and detailed examination. An appropriate image layout should fulfill the following criteria:

- *stable layout*: the layout should accommodate image ordering and maintain stable update when images are inserted or removed during the query;
- *screen utilization*: for efficiency, the layout should display as many images as users can comfortably view them;
- *occlusion minimization*: for effectiveness, images displayed should not occlude each other in the layout or their overlap should be minimized;

Therefore, based on image layout principles, for iMap, we propose a hybrid layout that combines the advantages of both quantum and spiral treemaps. Quantum treemap (3) is designed for displaying images or other objects of indivisible (quantum) size. The widths and heights of the rectangles in the quantum treemaps are integer multiples of a given elemental size, instead of arbitrary aspect ratios. It is this basic element size that is indivisible or cannot be made any smaller that led to the name of quantum treemaps. In our iMap, the use of quantum with a fixed aspect ratio simplifies the layout for images of different aspect ratios. Further, the adoption of spiral pattern maintains stable update when we insert or remove images. The 1D spiral also accommodates image ordering, such as the chronological order or rank order. Each node in iMap corresponds to a rectangle that displays an image thumbnail. The sizes of these rectangles can be determined by the importance of their images, such as search rank or hit count. The layout result of iMap is given in Figure 2.1.



(a) iMap spiral layout



(b) Layout example

Figure 2.1: iMap with the spiral layout
(a): iMap shows the query image at the center as the focus and arranges other images with decreasing similarity ranks as the context following a spiral pattern.
(b): Search result for the International Space Station query image. Results based on both visual and textual distances are ranked and arranged along the spiral circling out.

2.3. Animation*

Animation is the rapid display of a sequence of images used to convey the illusion of movement (39). Animation inherently reflects a change of some visual representation over time, hence is naturally used to facilitate perception of changes in some model over time. In the field of information visualization, animation has long been used to focusing on user interface for a variety of purposes (1). Hudson et al. (16) introduced toolkit support for animation, including techniques such as: simple motion-blur, “squash and stretch”, use of arcing trajectories, anticipation and follow through, and “slow-in/slow-out” transitions. The Information Visualizer (27) utilizes 2D and 3D animation with cognitive coprocessor to explore information and its structure.

* “The material contained in this chapter has been submitted to the *IEEE Information Visualization Conference 2012 (a special issue of IEEE Transaction on Visualization and Computer Graphics)*.”

Other researchers have concentrated on designing animations to facilitate human perception and transition understanding. The cone tree (28) and perspective wall (21) use 3D animations to help keep viewers oriented. Yee et al. (47) applies animated transition in dynamic graphs with a radial layout, yielding a smooth transition from one view to the next. The Name Voyager (38) stacked area chart visualization with animation when data is filtered, often including scale changes that involving animating gridlines and axis labels. DynaVis (14) is a visualization system supporting animated transitions between statistical data graphics backed by a shared data table. Two controlled experiments found significant advantages for animation across both syntactic and semantic tasks. Gapminder's Trendalyzer (11) uses an animated bubble chart to show trends over time, which can be effectively involved in both presentation and analysis or data exploration. Examples include marks movement to convey change over time, marks subdivision to indicate a drill-down effect, and animated transition from a stacked chart to a scatter plot.

2.3.1. Animation Principle

Given the promising effect and the potential pitfalls of animation, many researchers focus on the guidelines for crafting effective animation, such as staging, exaggeration, anticipation, squash-and-stretch, and slow-in slow-out timing.

Psychologists Tversky et al. (36) suggested two high-level principles for effective animation. One is *congruence principle*, which states that "the structure and content of the external representation should correspond to the desired structure and content of the internal representation." In our case, the external animation should reflect the internal similarity rank changes. The other is *apprehension principle*, which states that "the structure and content of the external representation should be readily and accurately perceived and comprehended." That is, our animation should be understandable from the viewer's perspective and the changes of image similarity ranking should be accurately perceived during the transition.

Our animation design should conform to these two principles, and furthermore, it should also abide by the characteristics of our iMap layout by allowing the user to focus on the centered query image and similar images retrieved for the display. As stated before, one of the problems of animation is its strong ability of distraction. So, our animation should avoid distraction to the greatest extent possible while maintaining the efficiency.

As shown in Figure 1.1, when a different query image is selected, the images displayed in the layout will need to be updated due to the changes of their similarity ranks with respect to the new query image. Some images in the old layout will not show up anymore, while other new images will be added. The key is that our animation should help the user focus more on remaining and newly-added images, instead of being distracted by the images that are removed. Besides, abiding by the congruence principle, our animation should also be easy to comprehend. Ideally, the viewers should clearly identify the similarity and difference between two successive query images and accurately understand the relationship between the old and new layouts.

Chapter 3[*]

Animation Factors

To explore the similarity change, user selects any query image to become the new focus. The new layout is displayed after recalculating the similarity rank of each image compared to the new focus. For iMap, while this is sufficient to show the final layout from the perspective of the new focus, simply switching to this new view can cause a highly disorienting rearrangement especially when the number of images displayed is large. To reduce this disorientation and improve visual comprehension, we use animation to perform a smooth transition.

In accordance with animation principles, we consider our animation from a global view and a local view. The global view focuses on the overall animation structure and refers to animation steps such as one-step and multi-step animations. The local view considers animation details, including animation actions such as fading and flying, and flying paths such as straight paths and curved paths.

3.1. Animation Steps

An animation usually consists of multiple sub-animations. To complete an animation, all steps could be finished simultaneously, or each successive step is accomplished before the next step starts.

[*] "The material contained in this chapter has been submitted to the *IEEE Information Visualization Conference 2012 (a special issue of IEEE Transaction on Visualization and Computer Graphics)*."

We propose two categories of animation steps: one-step and multi-step. For one-step animation, after a new query image is selected, all images are synchronously changed according to their similarity rank to the new focus as the query image changes. One-step animation is simple and only consists of one type of animation, such as fading, flying, or size change. While it is easy to implement and indeed informs the viewers that the change is happening, one-step animation only provides fairly limited clue for tracking images and the change of their similarity ranks.

To facilitate the tracking, we can break down the animation into multiple steps, i.e., staging the animation. Since multi-step animation involves in many different actions and it must elongate the time for finishing the layout changes, some principles should be considered to guide the design process. Lasseter (19) pointed out that it is important, when staging an action, that only one idea is seen by the audience at a time. If many actions are happening at once, the eyes do not know where to look at and the main idea of the action will be "upstaged" and overlooked. More importantly, the object of interest should distinguish itself from the rest of the scene. In a static scene, the eyes will be drawn to movement, while in a dynamic scene, the eyes will be attracted to something that is still. Since one is unable to perceive too many different actions at one time, every key action must be staged in the strongest and simplest way before moving to the next one. Based on these guidelines, only a few different actions should be played simultaneously in one step, and only those actions that will not distract and confuse the viewers can be issued together. Furthermore, the actions that are totally contrasted with each other will be performed separately, because it is very important for the viewers to look in the right place at the right time.

3.2. Animation Actions

Animation actions deal with how an image should be brought in or out of the current layout. In this work, we implement four representative animation actions: *fade-out/fade-in*, *fly-out/fly-in*, *size-out/size-in*, and *card-flip*. Here, “out” means image disappearance, and “in” means image appearance. The user is allowed to adjust the animation time and the number of animation steps for each category of action. The first three animation actions can be applied to both one-step and multi-step animations and mixed together (e.g., fade-out/fly-in). The card-flip only works for one-step animation due to its nature. The first three actions we discuss in this section apply to one-step animation. In multi-step animation, these three actions will be utilized as well. However, at the same time, the action taken by each image may be different, which we will discuss in detail in Chapter 4.

3.2.1. Fade-Out/Fade-In

This action is the most popular way to show the transition. In our scenario, all images shown in the old layout fadeout first and then all images in the new layout fade in. We use blending in OpenGL to achieve the effect of fading. Blending is used to combine the color of a given pixel that is about to be drawn with the pixel that is already on the screen. Besides R (Red), G (Green) and B (Blue), Alpha is the 4th color component, which can be used to combine colors. The alpha value means how opaque a material is. An alpha value of 0.0 would mean that the material is completely transparent. A value of 1.0 would be totally opaque. We compute the alpha value of each image at each fading step as

$$\alpha_i = s + i \times \frac{d-s}{n-1}, \tag{1}$$

where $i \in [0, n-1]$ is the ith fading step, n is the total number of fading steps. In fading out process, s = 1.0 and d = 0.0, image’s alpha value gradually decreases over the time,

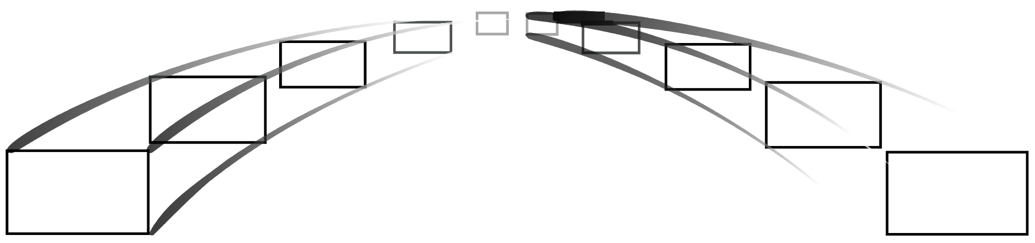


Figure 3.1: Fly-out and fly-in action illustration

which also means that the image is becoming more transparent. While in fading in process, $s = 0.0$ and $d = 1.0$, the image becomes more and more opaque over the time.

3.2.2. Fly-Out/Fly-In

For this action, images in the old layout fly out of the display area first and then images in the new layout fly in (see Figure 3.1). The center of an image's position before flying is the source p. For one-step animation, the "virtual" destination q of the image is located along the ray that connects the center of the entire display area c to p. Assuming the distance between c and p is d, we assign the distance from c to q as $d + \Delta d$, where Δd is some constant distance. For multi-step animation, the center of the image's new position after flying is the "actual" destination.

By flying each image from its source to destination, we can detect if the image's similarity rank increases or decreases. An image decreases its similarity rank if it flies from an inner layer to an outer layer and vice versa. As a matter of fact, one of the greatest challenges during flying process for us is to design an optimal way of flying to reduce image overlap. We will analyze this and present our solution in Section 3.3.

Another important factor of the fly-out/fly-in action is timing control, which gives meaning to movement – the speed of an action defines how well the viewer reads the idea behind the action. Proper timing is critical to make ideas readable and trackable. If too much time is spent on any of these, the viewer will be more likely to distract him/herself. If too little time is spent, the movement maybe finished before the viewer notice it, thus defeating its purpose. We assign the total timing control to the user, i.e., the user can adjust the total flying time from our control panel interface.

Besides total timing control, the spacing of the in-between steps between the extreme poses also plays an important role in the animation. We apply slow-in slow-out timing instead of straightforward linear timing for better motion perception. We compute the velocity at each flying step as

$$v_i = e^{-\left(\frac{i}{n-1} - 0.5\right)^2 \times s}, \tag{2}$$

where $i \in [0, n-1]$ is the *i*th flying step, *n* is the total number of flying steps, and *s* is a speed factor. Both *n* and *s* can be adjusted by the user. The formula is the form of Gaussian function (41), in which the graph of the Gaussian is a symmetric curve that dramatically falls off towards plus/minus infinity, i.e., normally distributed. When *s* is zero, there is a constant velocity of image flying. When *s* is set to a larger value, the difference between the speed at the middle of flying and the speed at both ends will be larger, i.e., the animation begins slowly, smoothly accelerates, and then decelerates in the end. Therefore, most of the movement occurs during the middle one-third of the given flying time, which makes the view of flying more natural and provides a good visual cue to help the user anticipate image movement.

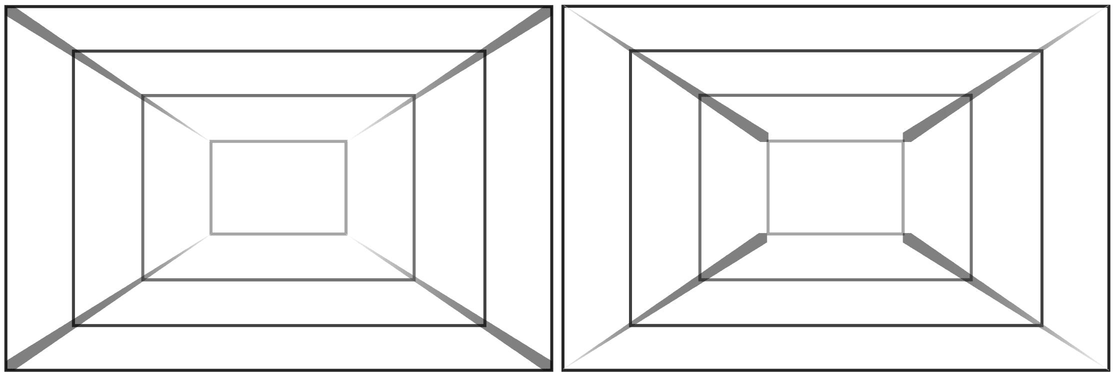

(a) Size-out action (b) Size-in action

Figure 3.2: Size-out and size-in action illustration

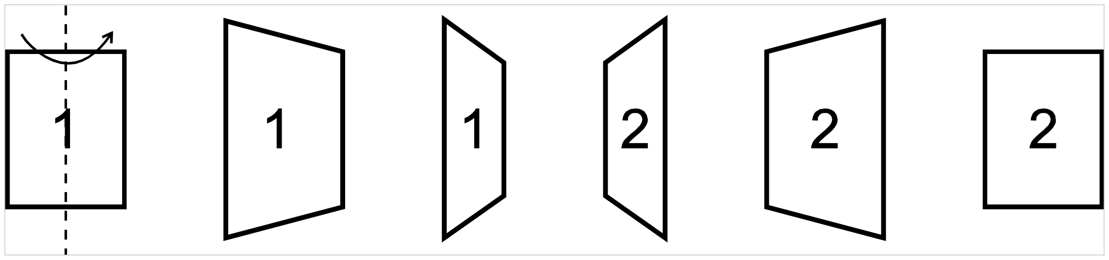


Figure 3.3: Card-flip action illustration

3.2.3. Size-Out/Size-In

For this action, we first gradually shrink the size of an image into a point, and then grow the size of the new image from a point to its target size (see Figure 3.2). The reason for us to take the size factor into consideration is mainly due to its role in the multi-step animation for overlap reduction, which we will discuss in Chapter 4.

3.2.4. Card-Flip

Unlike all previous three actions, the card-flip creates a 3D effect for the transition. Each image quad in the display area is now regarded as a slab where the front face is mapped with an image in the old layout. When a new focus is selected, images shown in the new layout are first mapped to the back faces of the slabs. Then in its local coordinate frame, we simultaneously rotate each slab along the *y* direction for 180° in an animated fashion, so that images on the front and back faces are flipped (see Figure 3.3).

3.2.5. Overlapping Actions

Under the condition that two actions cannot influence each other and make either of them unreadable, overlapping actions is helpful for conveying the main ideas of the transformation while maintaining the efficiency. Lasseter (19) stated that an action should never be brought to a complete stop before another action starts and the second action should overlap the first. In this way, the overlapping maintains a continuous flow among different phrases of actions.

We apply overlapping actions to both one-step and multi-step animations. For each animation action, we give a suggested overlapping degree for the user, but the user has the freedom to it from non-overlapping to fully-overlapping by moving the slider in the user interface. Overlapping value ranges from 0.0 to 1.0. The value of 0.0 represents non-overlapping, indicating that the next action will not be started if the previous action did not finish. The value of 1.0 means fully-overlapping, i.e. two consecutive actions happens at the same time. The overlapping value of 0.5 conveys that the next action will be started when the previous action finishes half of the movement. Of course, the prerequisite to apply action overlapping is that two actions cannot influence each other and make the other unreadable.

3.3. Flying Paths

For the fly-out/fly-in action, many images will fly from their sources to destinations at the same time, which easily leads to image overlapping and visual cluttering. Unlike overlapping actions which we leverage, overlapping images should be reduced as it would confuse the user and make it difficult to track the transition. To reduce image overlap, we consider curved paths with different radians, and clock-wise/counterclockwise flying directions. We first present a naïve solution that optimizes each flying path separately without considering the relationship among the paths. Then, we propose a greedy approach that determines the flying paths on an incremental basis. That is, for the current path to be determined, our goal is to reduce its overlap with all previously determined flying paths.

3.3.1. Measuring Image Overlapping

To quantitatively compare the percentage of image overlapping before and after path optimization, we propose an exact solution to measure image overlapping. In our implementation, each image is loaded as a texture and mapped to a quad for display. We compute the amount of quad overlap in terms of pixels and derive the average percentage of overlapping during the flying process as follows

$$o = \frac{\sum i \in N, j \in N, i < j \sum_{t=1}^{T} Q_{i,t} \cap Q_{j,t}}{T \times A}, \tag{3}$$

where $Q_{i,t}$ and $Q_{j,t}$ denote the positions of image quads Q_i and Q_j at flying step t and $\cap$ denotes the number of pixels overlapped by the two quads; N is the total number of images considered; T is the total number of flying steps; and A is the entire display area.

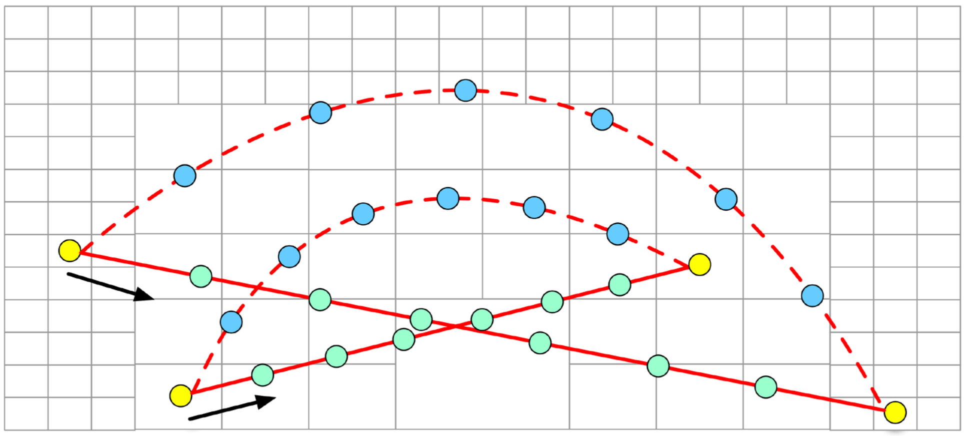


Figure 3.4: Staight vs. curved paths

Curved paths work better for the spiral layout and reduce the possibility of image overlap during the flying.

To actually show pixel overlap in an animated fashion, we draw all quads in red without texture with the same amount of transparency at the beginning (the background color is light blue). During the flying process, we blend together overlapped quads. At each flying step, the amount of opacity accumulated for each pixel is in proportion to the number of overlaps it has. This would produce qualitative results for intuitive comparison in addition to quantitative results.

3.3.2. Straight vs. Curved Paths

The simplest flying path between a source and its destination is to directly follow the straight line connecting them. This solution works well when the transition only involves a few images. As illustrated in Figure 3.4, moving along straight lines more likely leads to image crossover, especially around the center of the display area. Furthermore, in our spiral image layout, using a straight-line pattern would give the viewers a wrong impression. Since a new query image could be quite similar to the old one, in a typical transition, there could be many images staying on the same or adjacent layer (i.e., their rankings do not change dramatically). However, moving images only along straight lines could force these images to leave further away from its layer before turning back (shown in Figure 3.4).

Previously, Yee et al. (47) used the radial layout and linearly interpolated the polar coordinates of the nodes instead of their Cartesian coordinates, leading to a much smoother animation which dramatically reduces the cognitive effort to understand the animation. Similarly, our images are positioned along with the spiral orbit, so it makes more sense to make images move in curves rather than straight lines. In our spiral layout, we draw a curved path based on the following parameters: the two endpoints (source and destination) and radian (the higher the value, the more curvy the path). We derive the radian of the curved path as follows

$$r = \min\left\{r_0 \times d \times \left(1.0 - \frac{|\pi/4 - \theta'|}{\pi/4}\right), r_{\max}\right\}, \tag{4}$$

and

$$\theta' = \begin{cases} \pi - \theta, & \theta > \pi/2 \\ \theta, & \theta \leq \pi/2 \end{cases}, \tag{5}$$

where r_0 is a base radian which is a constant, d is the distance between the two endpoints, θ is the angle formed between the straight path connecting the two endpoints and the $+x$ axis, and $r_{\max}$ is the maximum radian allowed. The intuition is that when the distance is large and the angle is close to 45° or 135°, then the straight path is likely to cross the center of the display area. Therefore, the radian should be large so that the curved path could be effectively deviated from the center of the display area. On the other hand, r is bounded by $r_{\max}$, since a too large radian could force the flying path mostly out of the display area which is not desirable either. A naïve solution is to follow Equation 4 for each flying path separately and determine its radian accordingly.

Besides the two endpoints and radian, we also consider adding the flying direction to help the transition effect. Heer and Robertson (14) reported that the conventional clockwise order of radial graphs causes massive occlusion, and they resolved this issue by using the counter-clockwise ordering. In this work, we consider the flying direction for the following greedy optimization.

3.3.3. Greedy Optimization

We further present a greedy optimization method to reduce image overlapping. This greedy approach is a grid-based approximation, where each cell in the grid has the same size as the smallest remaining image displayed in the layout for flying.

We assume that all these images will be shrunk to this smallest size before flying. We create an occupancy buffer, in which a total of $n_c \times n_t$ counters are used to keep track of the number of paths in each cell for each time step, where n_c and n_t are the total numbers of cells and time steps, respectively. Assuming the number of existing paths in cell i at time step j is $n_{i,j}$, the occlusion among these existing paths in cell i at time step j is $n_{i,j} - 1$ if $n_{i,j} > 0$; otherwise the occlusion is zero. Therefore, the occlusion of a newly added path will be $\sum_{i,j} n_{i,j}$, where the newpath goes through cell i at time step j. Since the paths are more likely to occlude each other around the center of the display area, we can also calculate the occlusion value as a weighted sum $\sum_{i,j} w_i n_{i,j}$, where the closer a cell to the center, the higher the weight w_i.

Our approach starts from an empty set, and add the path which has the shortest Euclidean distance between the two end points at each step, since there is less space for the shorter paths to bypass the previous ones. For each path, we compute the occlusion values for several candidates, which include the straight line, clockwise and counter-clockwise curvy lines with radian as 0.5, 1.0, 1.5, 2.0 and 2.5, respectively. From these candidates, the one with the smallest value of occlusion will be selected. This process terminates when all paths are added.

Our results (see in Figure 5.2) show that the naïve solution using the curved path can successfully decrease overlapping and the greedy optimization can further reduce the average percentage of image overlap.

Chapter 4[*]

Multi-Step Animation

Decomposing the transition into several steps facilitates the viewers' understanding of the changes between the old layout and the new one. We propose several guidelines for designing our multi-step animation. First, the animation should be always directing the viewers' attention, leading them to look at where they should focus on at the right moment. Second, we should consider action overlapping for efficiency while avoiding issuing too many different kinds of actions at the same time for clarity. Third, the animation should be as simple as possible, conveying the main idea without distracting the viewers.

4.1. Stage Design

Following the above guidelines, we design our multi-step animation with three stages: *preparation*, *reorganization*, and *finalization*.

At the preparation stage, we swap the new query image with the old one. Meanwhile, we fade out those images that will not be shown in the new layout. Removing these images as early as possible helps the viewers pay more attention to the images in the new layout. We know that when staging an action, only one idea should be seen by the audience at a time. Nevertheless, performing swapping and fading actions simultaneously does not have a negative impact on our animation. In a still scene, the eye will be attracted to movement

[*] "The material contained in this chapter has been submitted to the *IEEE Information Visualization Conference 2012 (a special issue of IEEE Transaction on Visualization and Computer Graphics)*."

(19). Thus, the swapping of the old and new query images is the key action which takes the priority to attract viewers' attention, while the fading evolves gradually without distracting the viewers. After this stage, our new query image is located at the center of the display area. Other images remaining from the old layout stay put.

At the reorganization stage, we reorganize the positions of images remaining from the old layout. For these images, their positions in the new layout will likely change due to the change of their similarity ranking with respect to the new query image. The relative positions of remaining images are arranged according to their similarity rank to the old focus. By changing a new focus, each image's similarity rank will be recalculated and might be changed, as a result, their relative position on the screen should be altered correspondingly. Each image flies to their new position following the solutions proposed in Section 3.3.

After the reorganization stage, the remaining images are now located at their final positions in the new layout, but these images are only part of the images similar to the new focus. There are more images that are not shown yet. At the finalization stage, we bring in these newly-added images by applying one of the "in" processes discussed in Section 3.2, including fade-in, fly-in and size-in.

In summary, we decompose the animation into three stages to better assist the viewers in figuring out the changes of image layout. By breaking down a complex transition into several logical stages and making each stage simple enough to comprehend, we expect that our decomposition solution will significantly improve the understanding of image changes.

4.2. Further Consideration

Besides the three stages outline above, there are some other issues we need to consider for multi-step animation. We find that at the reorganization stage, even though we have already applied our greedy optimization scheme for flying paths, image overlapping and visual cluttering could still be rather severe especially with large number of remaining images. Therefore, we seek further improvement to reduce the overlap.

The first factor we consider is the size of images. In iMap, images in an inner layer could be larger than images in an outer layer. The larger the size of an image, the more likely it would overlap with other images and the larger proportion of overlap it would be throughout the flying process. To alleviate this problem, at the preparation stage, in addition to image swapping and fading, we also shrink all remaining images to the same size as the smallest remaining image in the layout. Note that even though these three different actions will be executed simultaneously, the fading and shrinking actions do not distract the viewers' attention on the swapping.

The second factor we consider is the number of flying images. At the reorganization stage, except for the new query image, all images remaining could fly to their new destinations, which easily leads to many crossovers when the number of images is large. Notice that for these remaining images, some of them will become less similar to the new focus and fly to the outer layer, while others will become more similar to the new focus and fly closer to the center. Therefore, we further split the flying process into two steps: flying images with decreasing ranks first and then flying images with increasing ranks. This further reduces image overlapping. To avoid distracting still images, we will fade still images to the background when other images are flying.

Combining size shrinking and fly splitting, we are able to largely reduce image overlapping and visual cluttering. The statistical results and Figure 5.3 demonstrates the effectiveness of reducing image overlap by adding one optimization strategy to the original solution.

4.3. AniMap– Put It All Together

Until now, we discussed different animation actions, analyzed the criteria to design an effective multi-step animation and proposed the solution to optimize it to be much smoother and less overlapping. Based on our analysis and the principles of animation, we present AniMap, a visualization framework that builds on top of iMap and supports animated transition.

AniMap utilizes multi-step animation, decomposing the animation into the preparation, reorganization, and finalization stages. At the first preparation stage, the new focus image swaps with the old focus, and at the same time, the images that will not be shown in the new layout will be faded out and all remaining images will be shrunk to the same size as the smallest remaining image in the layout. At the second reorganization stage, images with decreasing ranks will fly to their new destinations first, and then images with increasing ranks will fly. We leverages curved flying paths for overlap reduction and apply the slow-in slow-out timing technique for visual consistency. We also use the greedy path optimization method to further reduce the overlap and improve the readability. At the last finalization stage, all new images will fade in and at the same time, all these images will be enlarged to their target sizes. The effect of AniMap can be demonstrated by the snapshots in the Figure 5.4.

Chapter 5[*]

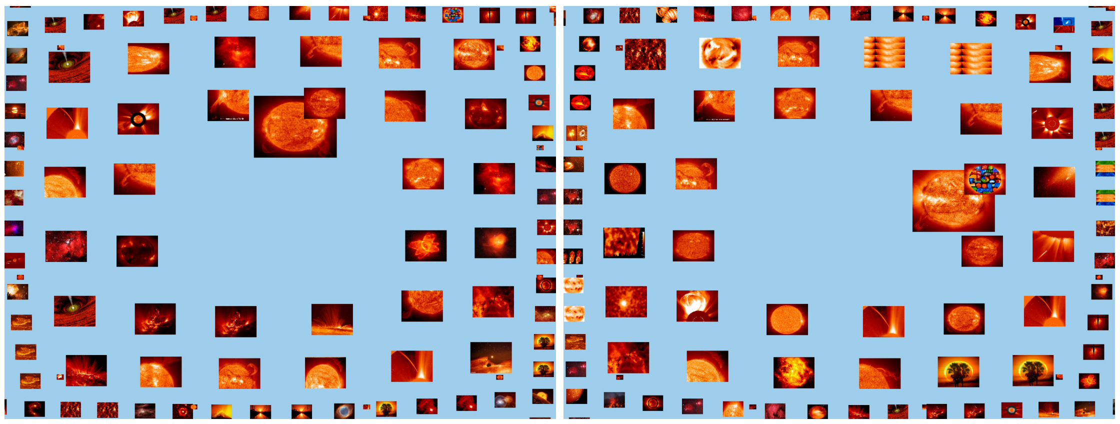
Results and Discussion

We experimented with our animated transition using the Astronomy Picture of the Day (APOD), the same data set as what we used in iMap. In the following, we present selected results of our approach. Due to the nature of animated transition, we strongly encourage readers to view the accompanying video clips (AniMap.mpg) for better appreciation.

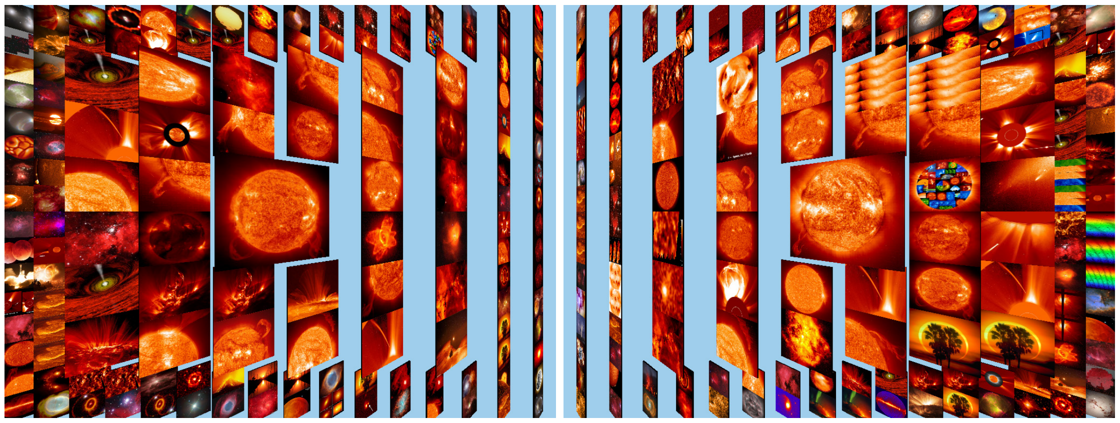
Figure 5.1 shows the snapshots of two animation actions, fly-out/fly-in and card-flip, for one-step animation. Figure 5.1(a) displays the snapshots of fly-out/fly-in actions. With fly-out and fly-in 70% action overlapping, images in the new layout fly in while images in the old layout fly out. Compared with straight paths, using curved paths makes the animated transition more attractive.

For the card-flip action (see Figure 5.1(b)), the flipping effect incurs image overlapping during the transition. However, since the images do not fly around, it is still easy to follow even though many images are being flipped simultaneously. The accompany video clips (AniMap.mpg) also include the results for the other two animation actions, fade-out/fade-in and size-out/size-in.

[*] "The material contained in this chapter has been submitted to the *IEEE Information Visualization Conference 2012 (a special issue of IEEE Transaction on Visualization and Computer Graphics)*."



(a) Fly-out/fly-in action with curved paths and 70% action overlapping

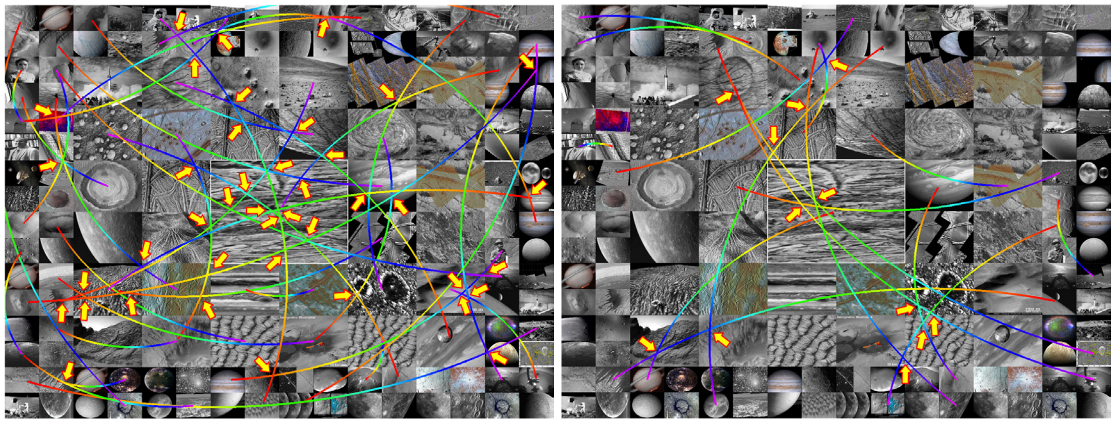


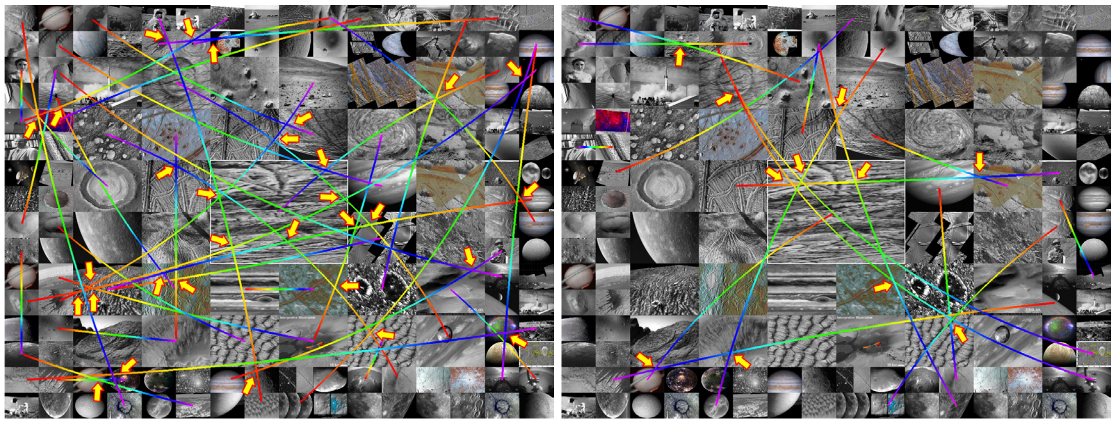
(b) Card-flip action

Figure 5.1: Snapshots of fly-out/fly-in and card-flip one-step animation

Figure 5.2 shows an example where we compare three different ways to determine the curved flying paths, including the flying paths with the same radians, with naïve improvement and with greedy optimization. Instead of displaying the actual flying process, we only draw each flying path from source to its destination. It's not difficult to find that there are many crossings among these paths. However, the crossing points do not exactly mean the existence of image overlapping at that point, because the points might be generated at different flying step. In order to correctly reflect image occlusion at the crossing points, we embed into the flying paths with rainbow colors. The color of the path gradually becomes from red to purple color (respectively corresponding to the beginning and the end of the flying), with the orange, yellow, green, cyan, and blue in between. As a result, if the colors at the crossing points between two paths are the same or closely similar, we can conclude that there occurs image overlapping.

We highlight the actual image overlaps during the flying with arrows. For the flying paths with the same radian, there are total of 53 overlaps during the flying process in this example, with 41 overlaps for images decreasing their ranks and 12 overlaps for images increasing their ranks. These two numbers drop down to 30 and 11 with the naïve improvement, and further down to 15 and 4 with the greedy optimization. It is clear that our greedy solution is able to well reduce image overlapping. This improvement has been consistently observed in many of our trials.


(a) Flying paths with the same radian (left: 41 overlaps, right: 12 overlaps)


(b) Flying paths with naïve improvement (left: 30 overlaps, right: 11 overlaps)

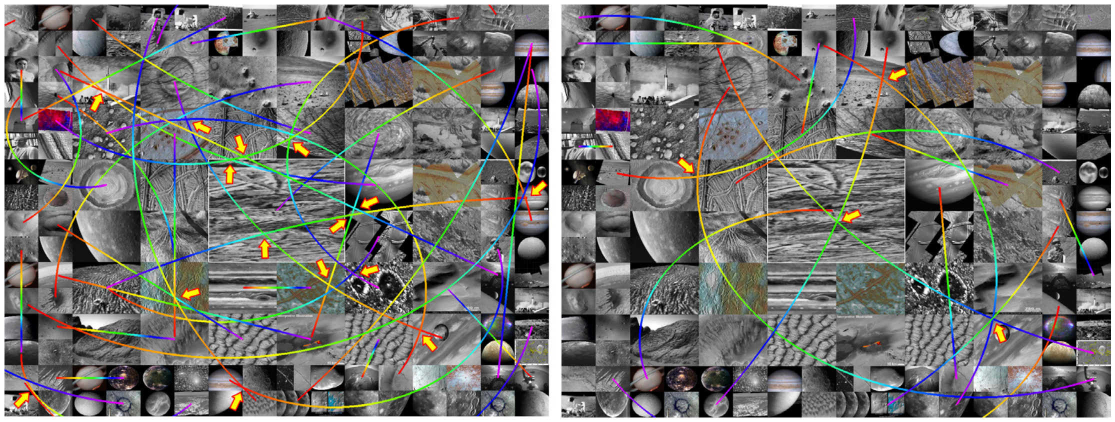

(c) Flying paths with greedy optimization (left: 15 overlaps, right: 4 overlaps)

Figure 5.2: Comparison of flying paths with three ways
Comparison of flying paths with the same radian (a), with the naïve improvement (b), and with the greedy optimization (c). Left column: flying paths for images with decreasing ranks. Right column: flying path for images with increasing ranks.

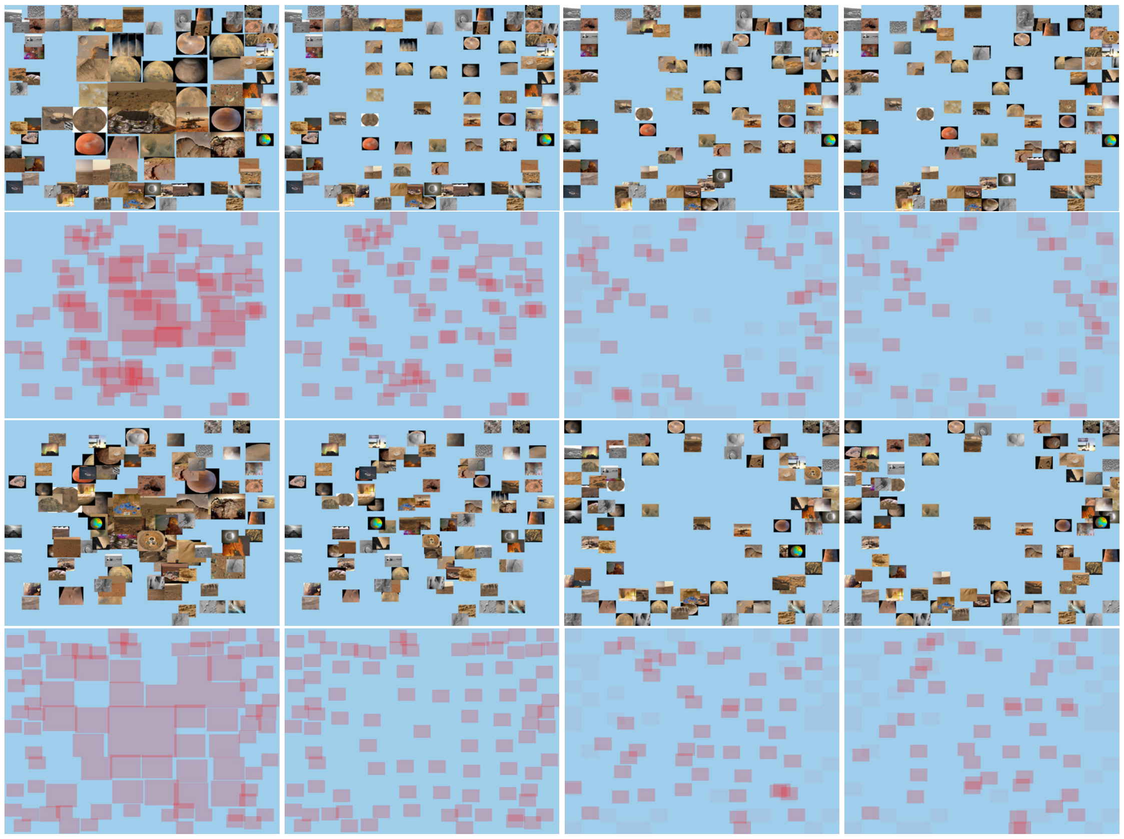

(a): original curved paths
(b): (a) + size shrinking
(c): (b) + fly splitting
(d): (c) + greedy optimization

Figure 5.3: Comparison of the flying solution

Comparison of the original flying solution with the addition of each new strategy at one time to reduce image overlap. The transition is for the two layouts shown in Figure 1.1. In (a) and (b), from top to bottom are selected at 20%, 40%, 60% and 80% of the entire flying process. In (c) and (d), the top/bottom two are selected at 33% and 66% of the flying process with flying images with decreasing/increasing ranks. The average percentages of overlapping for the entire flying process are 14.2%, 3.58%, 0.869%, and 0.375% for (a) to (d), respectively.

In Figure 5.3, we show the effectiveness of reducing image overlap by adding one strategy at a time to the original flying solution where curved flying paths are used. Shrinking the size of images contributes to the most percentage of average overlap reduction, from 14.2% to 3.58% for this example. Furthermore, when we split the flying process into two steps, the average percentage of overlap drops to 0.869%. Finally, we optimize the flying paths with our greedy strategy, further dragging down the average percentage of overlap to 0.375%. Side-by-side visual comparison also shows the gradual reduction in image overlap, especially for the images in the second and fourth rows where the bounding quads instead of the actual images are drawn. Note that such a trend in overlap reduction has been consistently observed and verified in many of our trials with varying numbers of images in the layout and different query images chosen.

Figure 5.4 shows snapshots of the AniMap. At the preparation stage, only the two focus images of the old and new layouts are swapping through flying. The rest of images shrink their sizes if they are larger than the smallest remaining image in the layout. Some of them also fade out because they do not appear in the new layout. Even though these three different actions are issued together, our attention is attracted to the swapping movement since the layout is mostly static (i.e., there are not many images flying around). At the end of this stage, only images that remain in the new layout stay. At the reorganization stage, images with decreasing ranks fly first, followed by images with increasing ranks. Therefore, comparing to Figure 5.4 (b), we see more images in (d) that are farther away from the display center. Similarly, comparing to Figure 5.4 (d), we see more images in (f) that are closer to the display center. Finally, at the finalization stage, new images are brought in to complete the transition.

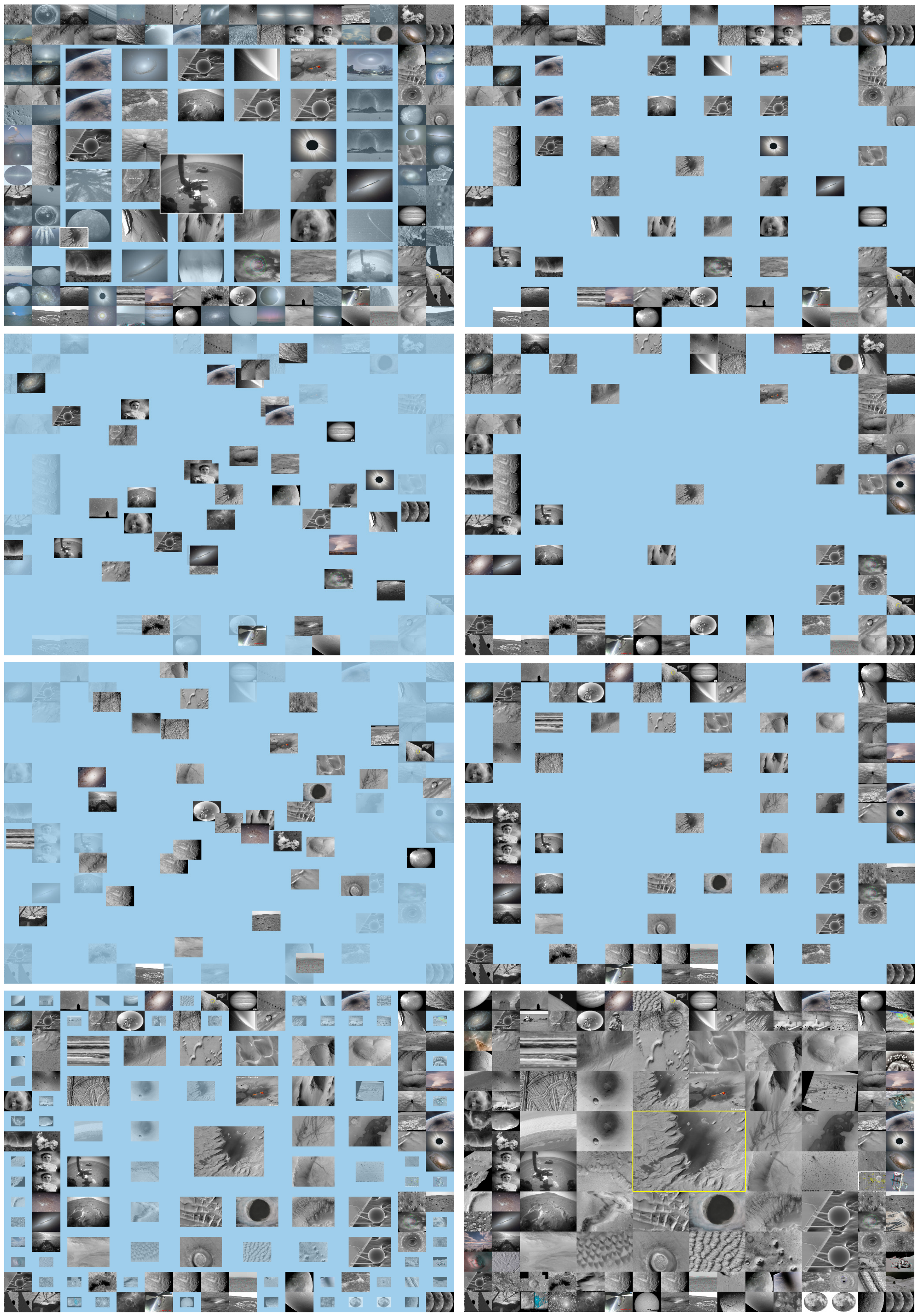


Figure 5.4: Snapshots of the AniMap

From top to down, left to right, the snapshots are named from (a) to (h) respectively. Stage 1: swapping the old and new focus images, fading out images that are not in the new layout and shrinking all remaining images (a). Stage 1 finishes (b). Stage 2: images with decreasing ranks fly (c) and the fly finishes (d). Images with increasing ranks fly (e) and the fly finishes (f). Stage 3: fading in new images while enlarging their sizes (g). The entire animation finishes (h).

Chapter 6[*]

User Study

Though instructed by design principles, crafting animated transition still involves many trade-offs. Empirical data are needed to evaluate the effectiveness of animated transitions. In this section, we present an experiment to gauge the effect of animated transition in similarity-based tiled image layout. We followed a 3 conditions (no-animation vs. one-step animation vs. multi-step animation) × 2 input image numbers (small vs. large) × 2 remaining image percentages (small vs. large) × 2 playback speeds (normal vs. slow) × 5 questions within-subject design.

Reviewed by the Office of Research Integrity and Compliance, this user study was approved on February 2nd, 2012 with approval number M0868. The project expiration date will be February 1st, 2013.

6.1. Overview

The experiment interface consists of a single window showing the image layout and its transition with one of the three animation conditions. A question and its answer options with radio buttons appear on the right-hand side of the window, which were visible for the duration of each task. The users selected their answer by clicking the corresponding radio button and then clicked the "Next" button to move on to the next question.

[*] "The material contained in this chapter has been submitted to the *IEEE Information Visualization Conference 2012 (a special issue of IEEE Transaction on Visualization and Computer Graphics)*."

All experiments were conducted in Computer Science Graphics Lab using four standard desktop PCs with the same configuration. The users were seated in front of 27 inch monitors with 1920×1080 screen resolution while the image layout occupied 1400×1050 pixels. They could sit in any fashion they found comfortable and were asked to answer a question for each task given.

Before the actual user study, we also conducted a pilot study on six users to identify the appropriate parameter values for the image number and percentage, and playback speed. After that, 24 new users were recruited to participate in the actual experiments. All of them are undergraduate and graduate students and each student was paid $10 in return. Each time, up to four users conducted the user study concurrently with one of our student researchers present.

The tasks were drawn from screen shots of our AniMap program running the APOD collection. Playing back the screen shots yielded the animation (around 15 frames per second for the normal playback speed). For the number of images displayed in the layout as the input, we considered two cases: small (49 images with four layers and one repetition level) and large (145 images with five layers and two repetition levels). For the percentage of images remaining from the old layout when a new query image is selected, we considered two cases: small ($\leq$ 30% of the input image number) and large ($\geq$ 40% of the input image number). We also took into account the playback speed. Our pilot study result showed that 10 seconds of animation for each task was comfortable, which we set it for the normal speed case. We doubled the playback time to 20 seconds for the slow speed case.

6.2. Interfaces

For no-animation, an old layout was first displayed in the window together with the question and answer options. There was no time limit for the user to read and understand the question. Once the user clicked the “Start” button, the user was allowed to use “Backward” and “Forward” buttons to see the old and new layouts back and forth within the time limit, which is the same as the animation time given. After that, no further interaction was allowed and the user must answer the question and click the “Next” button to move on to the next question.

For one-step animation, a repeated animation was shown. The animation started playing automatically after the user read the question and clicked the “Start” button. Within the time limit, the animation automatically repeated itself with an interval of two seconds between two playbacks. The users were instructed that they could select the answer and moved on to the next question even when the animation did not finish the playing.

For multi-step animation, the entire process of animation played only once. Similar to one-step animation, the user could select the answer and moved on to the next question before the animation finished. Except for the no-animation condition which allowed forward and backward interaction, no zooming, selection, or other forms of interaction on the animation were allowed for all three conditions.

6.3. Tasks

We designed the following five different questions in order to evaluate whether the users comprehend both the local and global layout structures under the three animation conditions or not. The users were asked to answer each question from multiple choices, making their best effort to answer correctly. Nevertheless, if there was no ground to make a choice, they were suggested to select "I don't know." as the answer.

Our first question considers the local evolution of image similarity rank from an old layout to a new layout. One image in the old layout was highlighted with a yellow boundary, and the user was asked:

Q1. *How does the similarity rank of the image with yellow boundary change (increase, decrease, or keep the same)?*

We chose this question because the essential difference between the old and new layouts was the changes of image similarity ranks. The ability to track such a change was one of the key criteria in our evaluation.

The second question explores image disappearance in the animated transition. Two images were highlighted in different colors, and the user was asked to select the image as the answer to:

Q2. *Given two images highlighted with yellow and green boundaries respectively, which image disappears in the new layout?*

We chose this question as image appearance/disappearance is one of the most basic results of the transition. This question is also locally structured.

In our scenario, images are arranged according to their degrees of similarity along a spiral pattern from inside out. The third question measures if the animation allows the user to detect the degree of change for image similarity ranks. Two images were highlighted in different colors, and the user was asked to select the image as the answer to:

Q3. *Given two images highlighted with yellow and green boundaries respectively, which image's similarity rank has a larger (or smaller) degree of change?*

In our fourth question, we test if the user is able to notice more global trends in the animated transition. Specifically, the question tests if an overall update in the number of images in the new layout is perceived. No images were highlighted and the user was asked:

Q4. *Given the total number of images in the old layout, please estimate the number of images that remain from the old layout.*

We selected this question because it is globally structured. Since the number of newly appeared images equals the total number of images minus the number of remaining images, we asked the user to estimate the number of remaining images so that they did not need to make an extra subtraction by themselves.

Finally, we tested whether the total number of images that increase or decrease their similarity ranks could be perceived or not. No images were highlighted and the user was asked:

Q5. *Given the number of remaining images, please estimate the number of images that increase (or decrease) their similarity ranks.*

This question is more of a global question as it requires the user to read both image updates and their similarity rank changes.

6.4. Experiment Design

The interface for each of the three conditions was demonstrated to the users before the test. They could ask questions, figure out the tasks, and see how the answers to the questions could be inferred.

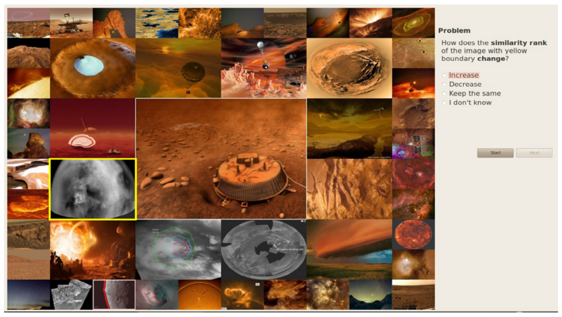
The experimental procedure required that the users answered all questions under one condition first, followed by all questions under the other condition. Therefore, any cognitive shift required to move from one interface to another only occurred twice. We counterbalanced between users by presenting six different orders of three conditions with four users following one order. However, within 40 tasks for each condition, the order of tasks was not counterbalanced, but rather given in the order from simple to complex so that the users could better prepare for the more difficult tasks. The user could take a short break between experimental conditions if preferred.

To help overcome the learning effect, each condition block was preceded by a practice block of five questions, selected randomly from the set of experimental tasks. The users were not aware that this initial block of five questions did not form part of the experimental data collection. Each user, therefore, completed a total of 135 tasks.

To avoid possible rushing toward the end of the tasks, we did not show the question number or the overall progress to the users. The time to read questions before the start of animation was not recorded. The completion time was recorded to include the animation time and the time to answer questions. The users were instructed to focus more on the accuracy rather than the completion time.

On average, it took one and an half hour for a user to complete all the tasks, which included the pre-experiment training, practice tasks, experimental tasks for all three animation conditions, and a post-experiment questionnaire.

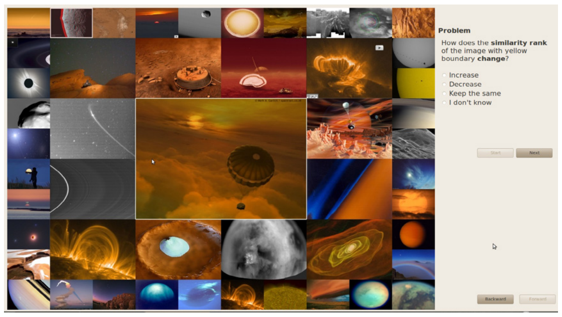
In Figure 6.1, we displayed three animation conditions and five questions user study test interfaces. The figure shows five questions interfaces with one of the three animation conditions. For the whole process of the user study, please refer to supplementary video (UserTest.avi).



Problem

How does the **similarity rank** of the image with yellow boundary **change**?

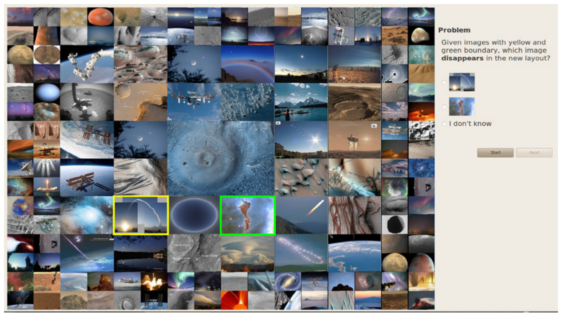
- Increase
- Decrease
- Keep the same
- I don't know



Problem

How does the **similarity rank** of the image with yellow boundary **change**?

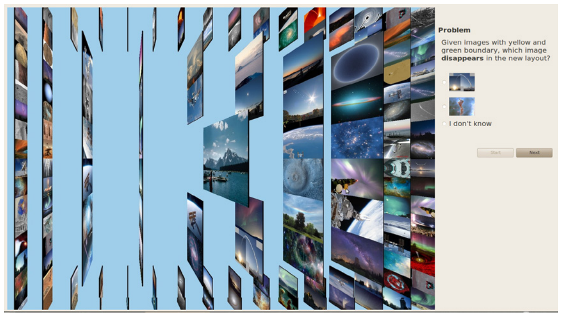
- Increase
- Decrease
- Keep the same
- I don't know



Problem

Given images with yellow and green boundary, which image **disappears** in the new layout?

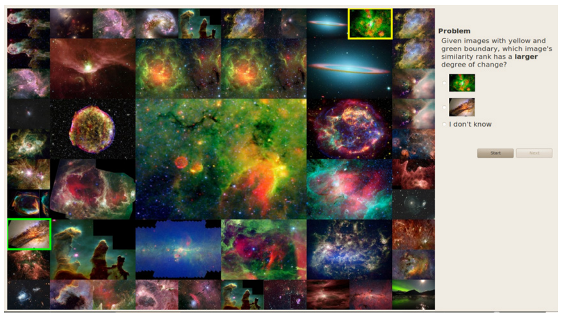
- I don't know



Problem

Given images with yellow and green boundary, which image **disappears** in the new layout?

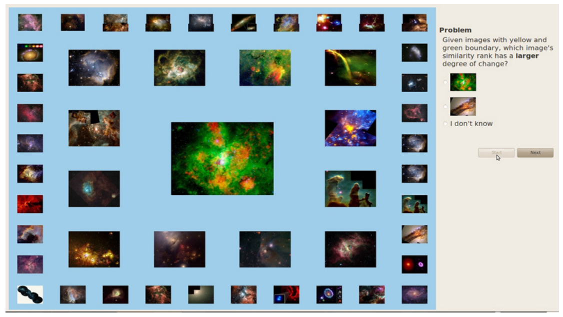
- I don't know



Problem

Given images with yellow and green boundary, which image's similarity rank has a **larger** degree of change?

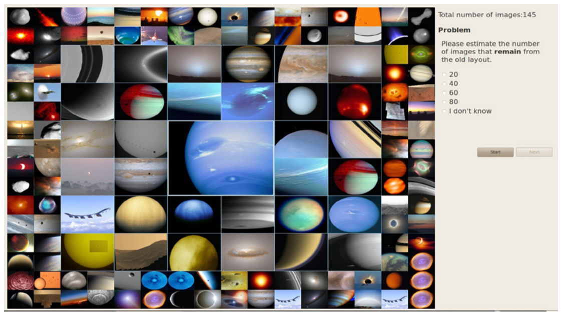
- I don't know



Problem

Given images with yellow and green boundary, which image's similarity rank has a **larger** degree of change?

- I don't know

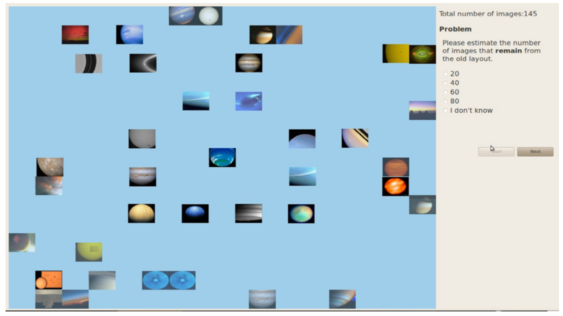


Total number of images:145

Problem

Please estimate the number of images that **remain** from the old layout.

- 20
- 40
- 60
- 80
- I don't know

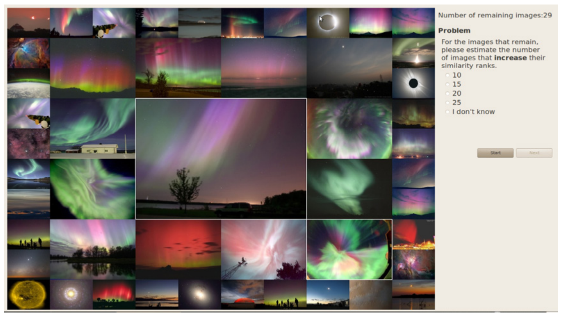


Total number of images:145

Problem

Please estimate the number of images that **remain** from the old layout.

- 20
- 40
- 60
- 80
- I don't know

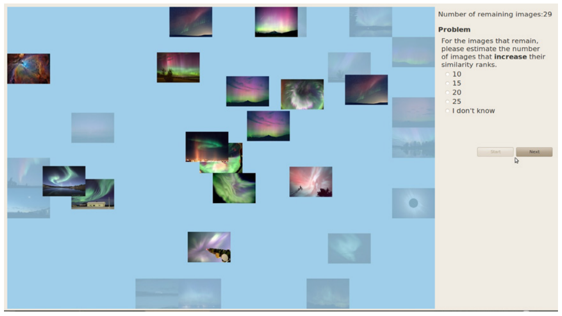


Number of remaining images:29

Problem

For the images that remain, please estimate the number of images that **increase** their similarity ranks.

- 10
- 15
- 20
- 25
- I don't know



Number of remaining images:29

Problem

For the images that remain, please estimate the number of images that **increase** their similarity ranks.

- 10
- 15
- 20
- 25
- I don't know

Figure 6.1: Snapshots of the user study test interface
From top to bottom, showing snapshots of Q1 with no-animation, Q2 with one-step animation, Q3 with one-step animation, Q4 with multi-step animation, and Q5 with multi-step animation. The left column shows the beginning of each question and the right column shows the middle process of the animation.

6.5. Results

Based on our experience in the pilot study, we anticipated that using multi-step animation would prove to be faster in completion time and more accurate than using no-animation and one-step animation, but we were not sure whether the input image number and playback speed would have a significant effect on the performance or not.

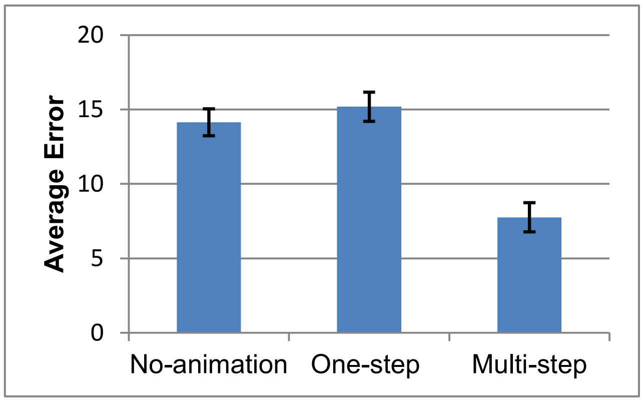
Since the data we collected do not form a normal distribution for most cases, instead of using one-way analysis of variance (ANOVA), we used a nonparametric Kruskal-Wallis (KW) one-way analysis of variance by ranks with a standard significance level $\alpha = 0.05$ to determine statistical significance between conditions. Kruskal-Wallis (42) test, a non-parametric method for testing whether samples originate from the same distribution, is majorly used for comparing more than two samples that are independent, or not related. When the Kruskal-Wallis test generates significant results, then it only indicates that at least one sample is different from the others, but it is unable to inform where the differences occur or how many differences actually occur. Different from its parametric equivalence ANOVA, KW analysis does not assume a normal distribution among the samples. Since out user study is designed to evaluate the difference among three animation conditions, i.e., three samples, and the data we collected is not normally distributed, KW test is an ideal method for us doing statistical data analysis.

We further applied a Bonferroni correction when we split the data to consider user performance on individual questions. Bonferroni correction (40) is a method for making multiple comparisons. The correction is based on the idea that if the significance level for the whole groups of *n* tests is (at most) α, then the significance level for each of the individual tests should be at α/*n*. In our analysis, the whole family of tests with three animation conditions is analyzed at a significance level 0.05, thus, in Bonferroni Correction, the significance level would drop down to 0.05/3, that is 0.0167.

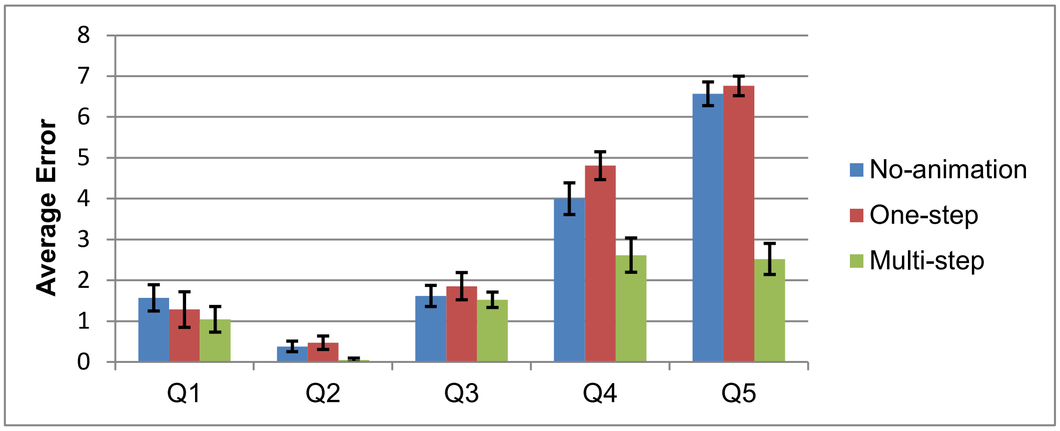
In statistics, correlation coefficient (or "*r*") majorly refers to Pearson product-moment correlation coefficients (46), or Pearson's *r*, which is a measure of the correlation between two variables *X* and *Y*, giving a value between +1 and -1 inclusive. The closer *r* is to +1 or -1, the more closely the two variables are related with each other. If the value of *r* is close to zero, there is no relationship between variables. If *r* is positive, it means that as one variable gets larger the other gets larger, while if *r* is negative, it means that as one variable gets larger, the other gets smaller. Using correlation coefficients to analyze user behaviors, we removed three users from further analysis due to their large negative coefficient values comparing against other users (see Appendix A). Therefore, the data collected from twenty-one users were used in the statistical analysis.

6.5.1. Accuracy

In statistical significance testing, the p-value (44) is a probability, with the value ranging from zero to one, to determine if a test statistic is at least as extreme as the one that was actually observed, assuming that the null hypothesis (e.g. there is no relationship between two measured conditions) is true. If the p-value is less than the significance level α, which is often 0.05 or 0.01, the result is said to be statistically significant.



(a) Mean values and standard errors of the average number of task errors overall



(b) The average number of task errors for each category of questions

Figure 6.2: Experiment results for accuracy

As shown in Figure 6.2 (a), the results for different animation conditions indicated a strong advantage for multi-step animation. KW analysis (see Appendix B) found a significant difference among the three animation conditions (H(2) = 24.69, $p \ll 0.0001$) and post-hoc analysis found that multi-step animation was significantly more accurate than no-animation (H(1) = 17.89, p = 0.0002) and one-step animation (H(1) = 21.01, $p \ll 0.0001$). The value of H is the result of KW analysis and the value of p is the p-value. No-animation was not significantly distinguishable from one-step animation.

The result of the average number of task errors for each category of questions is shown in Figure 6.2 (b). Post-hoc analysis using Bonferroni corrections showed that, for Q4 and Q5, multi-step animation was significantly more accurate than both no-animation (Q4: $H(1) = 5.84$, $p = 0.015$, Q5: $H(1) = 32.25$, $p \ll 0.0001$) and one-step animation (Q4: $H(1) = 14.92$, $p = 0.0001$, Q5: $H(1) = 34.53$, $p \ll 0.0001$). No significant difference was found in terms of accuracy for Q1, Q2, and Q3.

No significant difference was found under different input image numbers for all three animation conditions and all five categories of question except that for Q3, one-step animation was significantly more accurate with the small image number ($H(1) = 5.37$, $p = 0.0205$) but multi-step animation was significantly more accurate with the large image number ($H(1) = 10.30$, $p = 0.0013$).

Considering the playback speeds, KW analysis results were significant at the 0.01 level under the multi-step animation condition ($H(1) = 14.36$, $p = 0.0002$) for Q3, for which the average number of errors was significantly higher with the slow playback speed.

Figure 6.3 shows the average number of "I don't know." responses for the three animation conditions, where users were unwilling to make an estimate. A significant difference was found among the three animation conditions ($H(2) = 25.09$, $p \ll 0.0001$), with post-hoc tests showing that multi-step animation was the fewest in the number of "I don't know." responses. Considering task variants, "I don't know." responses were all from Q4 and Q5, i.e., questions on the global layout structure.

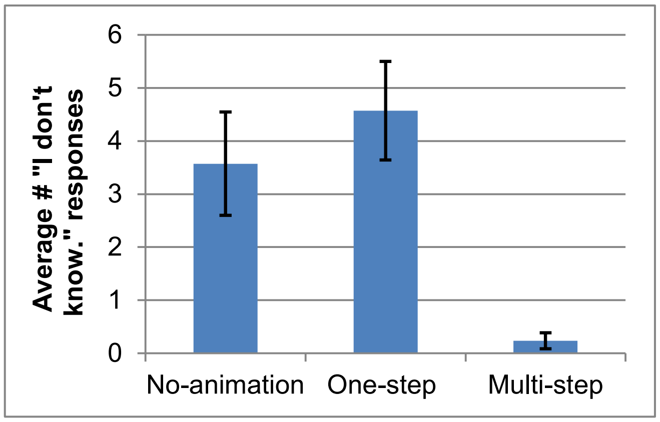
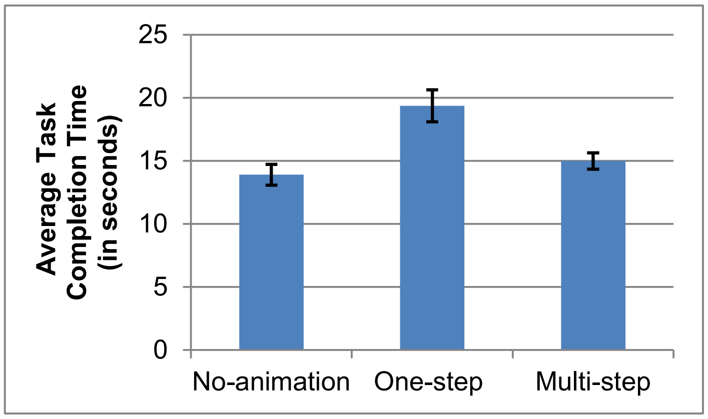


Figure 6.3: The average number of "I don't know." responses

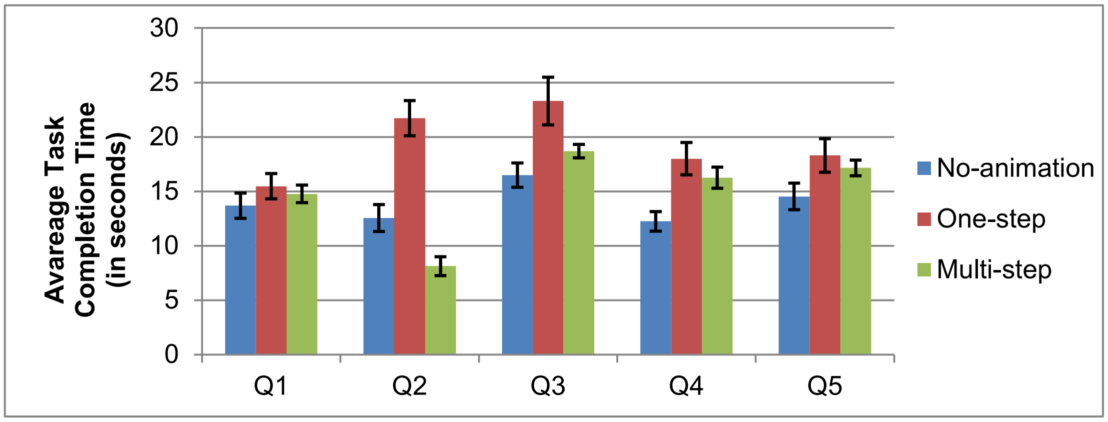
6.5.2. Task Completion Time

The time to complete a task was measured from when the user clicked the "Start" button to when the user clicked the "Next" button.

Figure 6.4 shows the average task completion time. We analyzed all tasks combined as well as tasks in each category of questions. In terms of task completion time, a significant difference was found among the three animation conditions (H(2) = 13.61, p = 0.0001) (see Appendix C), with post-hoc tests showing a significant interaction effect between multi-step animation and one-step animation (H(1) = 8.01, p = 0.0047). The mean time for completion for multi-step animation was 15.00 seconds compared to 19.37 seconds for one-step animation (about 29% faster). Multi-step animation and no-animation were not statistically distinguishable.



(a) Overall average task completion time



(b)The average task completion time for each category of question

Figure 6.4: Experiment results for task completion time

For all animation conditions, the input image numbers did not have a significant effect in terms of task completion time. Considering the playback speed, KW analysis results were significant at the 0.01 level for both one-step animation ($H(1) = 8.30$, $p = 0.0040$) and multi-step animation($H(1) = 18.08$, $p \ll 0.0001$). Decreasing the playback speed noticeably increased the task completion time.

Table 6.1
Average ratings for six statements for each animation condition.
*indicates significant differences (p < 0.01).

Statements	multi-step animation	one-step animation	no-animation
S1. It was easy to estimate the number of images remaining from the old layout.	4.43 *One, No	1.76	2.00
S2. It was easy to estimate the number of images that increase or decrease their similarity ranks.	4.38 *One, No	1.90	2.10
S3. It was easy to track the changes of image similarity ranking.	4.71 *One, No	2.43	2.43
S4. It was easy to detect the appearance or disappearance of images.	4.86 *One, No	3.14	3.24
S5. It was easy to estimate the degree of changes of image similarity ranking.	4.38 *One, No	2.48	2.81
S6. Overall, the solution was effective.	4.67 *One, No	2.38	2.52

6.5.3. Subjective Preferences

After the experiments, users completed a survey with six statements as listed in Table 6.1. Each were answered with a 5-point Likert scale (1 = strongly disagree, 5 = strongly agree). KW analysis was conducted on ratings for each animation condition (see Appendix D). Table 6.1 gives the pair-wise comparison results.

KW analysis results found a significant difference in rating among the three animation conditions ($H(2) = 39.32$, $p \ll 0.0001$), with post-hoc analysis showing that multi-step animation obtained significantly higher grades than no-animation($H(1) = 29.88$, $p \ll 0.0001$) and one-step animation ($H(1) = 32.16$, $p \ll 0.0001$). The users' subjective feeling that multi-step animation was much easier for all tasks was also consistent to the previous result that multi-step animation got the fewest "I don't know." responses, i.e., the users were more willing to estimate the answer under multi-step animation.

At the end of the survey, many users indicated in the open comments section that the movement of multi-step animation was understandable and clear. One user reported that *"It's very easy to track specific picture and a limited number of pictures."* Many users commented in a similar way that *"It's detectable and easy to track and relocate desired objects."* but some users also pointed out that *"It's much slower than no-animation." and "It could be more than perfect if the speed of animation could be controlled."*

For one-step animation, users complained that *"The shift between old layout and new layout totally confuses me and destroys my impression on the old layout if I have any."* and *"It was difficult to track changes and estimate the numbers."* Furthermore, some users pointed out that one-step animation was time consuming.

The users also expressed their preferences for no-animation because it was easy to focus and would not be interrupted by animations. Another important advantage of no-animation was the control of switching between the old and new layouts. However, in terms of estimating the numbers in Q4 and Q5, there was no clue for guessing under the no-animation condition.

6.5.4. Discussion

Overall, we have compelling evidence that using multi-step animation was significantly more accurate in answering questions than the other two conditions and was significantly faster in completion time than one-step animation.

In terms of accuracy, we found significant differences among the three animation conditions between questions on local and global structures. No significant difference was found for questions on local structures, which asked about the change of image similarity ranks and the disappearance of images. We discovered that it was not difficult to obtain the right answer as long as the users were able to identify highlighted images in the new layout. However, for global structured questions such as estimating the number of remaining images, multi-step animation was significantly more accurate than the other two conditions. One possible explanation could be that the different image sizes and limited comparison time added the difficultly to estimate under no-animation and one-step animation conditions. Some users might try their best to get an estimated answer, while many other users selected “I don’t know.” as the answer without much thinking or reasoning, increasing the errors in these two conditions. This was why the most number of “I don’t know.” answers came from Q4 and Q5.

In terms of task completion time, one-step animation was significantly slower than multi-step animation and no-animation, and no significant difference was found between multi-step animation and no-animation. Post-hoc analysis found a significant difference between multi-step animation and no-animation at the slow playback speed ($H(1) = 8.59$, $p = 0.0034$), but not at the normal playback speed. Besides, no significant difference was found in playback speed for the no-animation, but multi-step animation was significantly slower at the slow playback speed ($H(1) = 18.18$, $p \ll 0.0001$). Therefore, it is very likely that, using multi-step animation at the slow playback speed, more task completion time needed was simply due to the longer animation time itself. The users could not answer the questions without waiting for the animation playing to that specific step. For example, if the user were asked to estimate the number of images that increase their similarity ranks, they must wait until the second flying process begins at the reorganization stage.

Our results indicated that the input image number did not have a significant effect on the accuracy and task completion time. Only one exception need to be noticed, that is, multi-step animation was significantly more accurate with a large number of input images for Q3, which asked the users to track images with larger or smaller similarity rank changes. Our common sense might tell us that the users would be more likely to answer incorrectly given more input images, since the larger the size of the input images, the higher probability of more occlusion. However, in our scenario, the large number of image input consists of five layers with two repetition levels, the difference between each layer was itself much easier to identify, compared to the small number of image input with four layers and only one repetition. Thus, it might be easier for the participants to identify the degree of image similarity rank change.

No significant difference was found in terms of accuracy under different playback speeds except that, for Q3, multi-step animation generates significantly more errors at the slow playback speed. Similarly, we might argue that, given more time to do the task, the accurate should be higher. But in terms of animation, the result might be different. The longer the animation, the higher probability that the viewers forget what was happening before. In our case, given the slow playback speed, the users would be more likely to forget the positions of images in the old layout after the animation finished, leading to wrong answers to the questions on similarity rank change.

Chapter 7

Conclusion and Future Work

As the society generates increasing amounts of digital information, organizing and visualizing large-scale data sets becomes increasingly popular. Animated transition plays an important role in helping viewers grasp the changes of data, both locally and globally.

In this master thesis, we began by situating two basic animation principles that we should abide by, *congruence principle* and *apprehension principle*. We then presented four basic animation actions we applied in the one-step animation, including fade-out/fade-in, fly-out/fly-in, size-out/size-in and card-flip. Later, we explored how to decompose our animation into multiple animations, and how to optimize fly paths to make it smoother and generate less overlapping. We applied curved path instead of the straight-line path and we further proposed a greedy optimization to successfully reduce large amount of overlapping. Then we presented the AniMap, an animated transition solution specifically designed for similarity-based tiled image layouts. The image and video results demonstrate the effectiveness of AniMap in terms of improved understanding and increased engagement. At last, we have also conducted a user study to compare our multi-step animated transition against no-animation and one-step animation solutions. The results show that multi-step animation significantly improves the accuracy (especially for questions on global layout structure) and the overall ratings.

In the future, we would like to apply the general principles of animation to other information visualization applications where such a transition is not inherently given. Whenever applicable, besides position, size and opacity, other visual properties such as color, shape and orientation could also be incorporated into an effective animated transition design. Further, since the website APOD is largely used in education, we might consider to develop a Web version of APOD combined with the iMap and AniMap. iMap facilitates the image searching and browsing based on image similarity metrics, and AniMap allows the user to follow and track the changes with animated transition.

Bibliography

[1] Baecker R, Small I. Small. Animation at the interface. In: B. Laurel, editor. The Art of Human-Computer Interface Design. 1st ed. Boston (MA): Addison-Wesley Professional; 1990. p. 251-267.

[2] Bederson BB, Boltman A. Does animation help users build mental maps of spatial information? INFOVIS 1999. Proceedings of the 1999 IEEE Symposium on Information Visualization; 1999 Oct 24-29; San Francisco (CA). Washington (DC): IEEE Computer Society; c1999. p. 28-35.

[3] Bederson BB. PhotoMesa: A zoomable image browser using quantum treemaps and bubblemaps. UIST 2001. Proceedings of the 14th annual ACM Symposium on User Interface Software and Technology; 2001 Nov 11-14; Orlando (FL). New York (NY): ACM; c2001. p. 71-80.

[4] Bimbo AD, Pala P. Visual image retrieval by elastic matching of user sketches. Journal of IEEE Transactions on Pattern Analysis and Machine Intelligence. 1997;19(2):121-132.

[5] Bladh T, Carr DA, Kljun M. The effect of animated transitions on user navigation in 3D treemaps. INFOVIS 2005. Proceedings of the 2005 IEEE Symposium on Information Visualization; 2005 Oct 23-25; Minneapolis (MN). Washington (DC): IEEE Computer Society; c2005. p. 297-305.

[6] Brivio P, Tarini M, Cignoni P. Browsing large image datasets through Voronoi diagrams. Journal of IEEE Transactions on Visualization and Computer Graphics. 2010;16(6):1261-1270.

[7] Bruls M, Huizing K, Wijk JV. Squarified treemaps. VisSym 1999. Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization; 1999 May 26-28; Vienna (Austria). Goslar (Germany): Eurographics Association; 1999. p. 33-42.

[8] Curvehambault D, Purchase HC, Pinaud B. Animation, small multiples, and the effect of mental map preservation in dynamic graphs. Journal of IEEE Transactions on Visualization and Computer Graphics. 2011;17(4):539-552.

[9] Datta R, Joshi D, Li J, Wang JZ. Image retrieval: Ideas, influences, and trends of the new age. Journal of ACM Computing Surveys. 2008;40(2):Article 5.

[10] Donskoy M. Window navigation with and without animation: A comparison of scroll bars, zoom and fisheye view. CHI EA 1997. Proceedings of CHI Extended Abstracts on Human Factors in Computing Systems: Looking to the Future; 1997 Mar 22-27; Atlanta (GE). New York (NY): ACM; c1997. p. 279-280.

[11] Gapminder [Internet]. 2012 [updated 2012 Apr 18, cited 2012 Apr 19]. Available from: http://www.gapminder.org/

[12] Gonzales C. Does animation in user interfaces improve decision making? CHI 1996. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Common Ground; 1996 Apr 13-18; Vancouver (Canada). New York (NY): ACM; c1996. p. 27-34.

[13] Heer J, Card SK, Landay JA. Prefuse: A toolkit for interactive information visualization. CHI 2005. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems; 2005 Apr 2-7; Portland (OR). New York (NY): ACM; c2005. p. 421-430.

[14] Heer J, Robertson GG. Animated transition in statistical data graphics. Journal of IEEE Transactions on Visualization and Computer Graphics. 2007;13(6):1240-1247.

[15] Helfman JI, Hollan JD. Image representation for accessing and organizing web information. In: Beretta GB, Schettini R, editors. SPIE 2001. Proceedings of SPIE International Society for Optical Engineering Conference. Bellingham (WA): SPIE; c2001. p. 91-101.

[16] Hudson SE, Stasko JT. Animation support in a user interface toolkit: Flexible, robust, and reusable abstractions. UIST 1993. Proceedings of the 6th annual ACM Symposium on User Interface Software and Technology; 1993 Nov 3-5; Atlanta (GE). New York (NY): ACM; c1993. p. 57-67.

[17] Kandel S, Paepcke A, Theobald M, Molina HG, Abelson E. PhotoSpread: A spreadsheet for managing photos. CHI 2008. Proceedings of the 26th annual SIGCHI Conference on Human Factors in Computing Systems; 2008 Apr 5-10; Florence (Italy). New York (NY): ACM; c2008. p. 1749-1758.

[18] Kang H, Shneiderman B. Visualization methods for personal photo collections: Browsing and searching in the PhotoFinder. ICME 2000. IEEE International Conference on Multimedia and Expo; 2000 Jul 30-Aug 2; New York (NY). New York (NY): IEEE; 2000. p. 1539-1542.

[19] Lasseter J. Principles of traditional animation applied to 3D computer animation. SIGGRAPH 1987. Proceedings of the 14th annual Conference on Computer Graphics and Interactive Techniques; 1987 Jul 27-31; Anaheim (CA). New York (NY): ACM; c1987. p. 35-44.

[20] Levoy M. Spreadsheets for images. SIGGRAPH 1994. Proceedings of the 21st annual Conference on Computer Graphics and Interactive Techniques; 1994 Jul 24-29; Orlando (FL). New York (NY): ACM; c1994. p. 139-146.

[21] Mackinlay JD, Robertson GG, Card SK. The perspective wall: Detail and context smoothly integrated. CHI 1991. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Reaching through Technology; 1991 Apr 27-May 2; New Orleans (LA). New York (NY): ACM; c1991. p. 173-176.

[22] Mackinlay JD, Robertson GG, Deline R. Developing calendar visualizers for the information visualizer. UIST 1994. Proceedings of the 7th annual ACM Symposium on User Interface Software and Technology; 1994 Nov 2-4; Marina del Rey (CA). New York (NY): ACM; c1994. p. 109-118.

[23] Manjunath BS, Ma WY. Texture features for browsing and retrieval of image data. Journal of IEEE Transactions on Pattern Analysis and Machine Intelligence. 1996;18(8):837-842.

[24] Mikolajczyk K, Schmid C. Scale and affine invariant interest point detectors. International Journal of Computer Vision. 2004;60(1):63–86.

[25] Mueller C, Gregor D, Lumsdaine A. Distributed force-directed graph layout and visualization. EGPGV 2006. Proceedings of Eurographics Symposium on Parallel Graphics and Visualization; 2006 May 11-12; Braga (Portugal). Goslar (Germany): Eurographics Association; 2006. p. 83-90.

[26] Nemiroff RJ, Bonnell JT. Astronomy picture of the day: http://antwrp.gsfc.nasa.gov/apod/astropix.html. In Bulletin of the American Astronomical Society. 1995:1291.

[27] Robertson GG, Card SK, Mackinlay JD. The cognitive coprocessor architecture for interactive user interfaces. UIST 1989. Proceedings of the 2nd annual ACM SIGGRAPH Symposium on User Interface Software and Technology; 1989 Nov 13-15; Williamsburg (WV). New York (NY): ACM; c1989. p. 10-18.

[28] Robertson GG, Mackinlay JD, Card SK. Cone trees: Animated 3D visualizations of hierarchical information. CHI 1991. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Reaching through Technology; 1991 Apr 7-May 2; New Orleans (LA). New York (NY): ACM; c1991. p. 189-194.

[29] Robertson GG, Fernandez R, Fisher D, Lee B, Stasko JT. Effectiveness of animation in trend visualization. Journal of IEEE Transactions on Visualization and Computer Graphics. 2008;14(6):1325-1332.

[30] Rodden K, Basalaj W, Sinclair D, Wood K. Evaluating a visualisation of image similarity as a tool for image browsing. INFOVIS 1999. Proceedings of the 1999 IEEE Symposium on Information Visualization; 1999 Oct 24-29; San Francisco (CA). Washington (DC): IEEE Computer Society; c1999. p. 36-43.

[31] Rodden K, Basalaj W, Sinclair D, Wood K. Does organization by similarity assist image browsing? CHI 2001. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems; 2001 Mar 31-Apr 5; Seattle (WA). New York (NY): ACM; c2001. p. 190-197.

[32] Shneiderman B, Wattenberg M. Ordered treemap layouts. INFOVIS 2001. IEEE Symposium on Information Visualization; 2001 Oct 22-23; San Diego (CA). Washington (DC): IEEE Computer Society; c2001. p. 73-78.

[33] Swain M, Ballard B. Color indexing. International Journal of Computer Vision. 1991;7(1):11-32.

[34] Torres R, Silva C, Medeiros C, Rocha H. Visual structures for image browsing. cikm 2003. Proceedings of the 12th International Conference on Information and Knowledge Management; 2003 Nov 2-8; New Orleans (LA). New York (NY): ACM; c2003. p. 49-55.

[35] Tu Y, Shen HW. Visualizing changes of hierarchical data using treemaps. Journal of IEEE Transactions on Visualization and Computer Graphics. 2007;13(6):1286-1293.

[36] Tversky B, Morrison J, Betrancourt M. Animation: Can it facilitate? International Journal of Human-Computer Studies. 2002;57(4):247-262.

[37] Wang C, Reese JP, Zhang H, Tao J, Nemiroff RJ. iMap - a stable layout for navigating large image collections with embedded search. Technical Report CS-TR-12-01; Houghton (MI): Michigan Technological University, Department of Computer Science; 2012. Available from: http://www.mtu.edu/cs/research/papers/pdfs/CS-TR-12-01.pdf.

[38] Wattenberg M, Kriss J. Designing for Social Data Analysis. Journal of IEEE Transaction on Visualization and Computer Graphics. 2006;12(4):549-557.

[39] Wikipedia, the free encyclopedia [Internet]. Animation; 2012 [updated 2012 Apr 16, cited 2012 Apr 19]. Available from: http://en.wikipedia.org/wiki/Animation

[40] Wikipedia, the free encyclopedia [Internet]. Bonferroni Correction; 2012 [updated 2012 Jan 11, cited 2012 Apr 22]. Available from: http://en.wikipedia.org/wiki/Bonferroni_Correction

[41] Wikipedia, the free encyclopedia [Internet]. Gaussian function; 2012 [updated 2012 Apr 17, cited 2012 Apr 21]. Available from: http://en.wikipedia.org/wiki/Gussian_function

[42] Wikipedia, the free encyclopedia [Internet]. Krusal-Wallis one way analysis of variance; 2012 [updated 2012 March 21, cited 2012 Apr 22]. Available from: http://en.wikipedia.org/wiki/Krusal-Wallis

[43] Wikipedia, the free encyclopedia [Internet]. Pearson product-moment correlation coefficient; 2012 [updated 2012 Apr 16, cited 2012 Apr 22]. Available from: http://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient

[44] Wikipedia, the free encyclopedia [Internet]. P-value; 2012 [updated 2012 Apr 21, cited 2012 Apr 22]. Available from: http://en.wikipedia.org/wiki/P_value

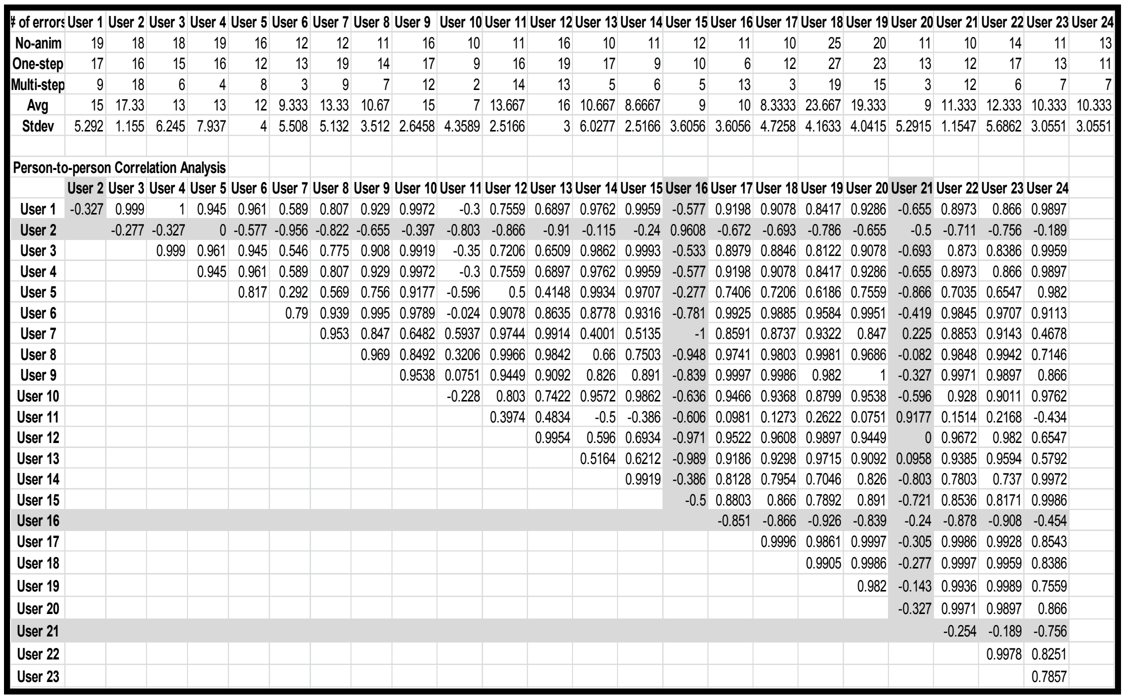
[45] Yamaoka S, Manovich L, Douglass J, Kuester F. Cultural analytics in large-scale visualization environments. Journal of IEEE Computer. 2011;44(12):39-48.

[46] Yang J, Fan J, Hubball D, Gao Y, Luo H, Ribarsky W, Ward M. Semantic image browser: Bridging information visualization with automated intelligent image analysis. VAST 2006. Proceedings of IEEE Symposium on Visual Analytics Science and Technology; 2006 Oct 31-Nov 2; Baltimore (MD). Washington (DC): IEEE Computer Society; c2006. p. 191-198.

[47] Yee K, Fisher D, Dhamija R, Hearst M. Animated exploration of dynamic graphs with radial layout. INVOFIS 2001. Proceedings of IEEE Symposium on Information Visualization; 2001 Oct 22-23; San Diego (CA). Washington (DC): IEEE Computer Society; c2001. p. 43-50.

Appendix A

User Study Result – Person-to-person Correlation Analysis



# of errors	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	User 12	User 13	User 14	User 15	User 16	User 17	User 18	User 19	User 20	User 21	User 22	User 23	User 24
No-anim	19	18	18	19	16	12	12	11	16	10	11	16	10	11	12	11	10	25	20	11	10	14	11	13
One-step	17	16	15	16	12	13	19	14	17	9	16	19	17	9	10	6	12	27	23	13	12	17	13	11
Multi-step	9	18	6	4	8	3	9	7	12	2	14	13	5	6	5	13	3	19	15	3	12	6	7	7
Avg	15	17.33	13	13	12	9.333	13.33	10.67	15	7	13.667	16	10.667	8.6667	9	10	8.3333	23.667	19.333	9	11.333	12.333	10.333	10.333
Stdev	5.292	1.155	6.245	7.937	4	5.508	5.132	3.512	2.6458	4.3589	2.5166	3	6.0277	2.5166	3.6056	3.6056	4.7258	4.1633	4.0415	5.2915	1.1547	5.6862	3.0551	3.0551

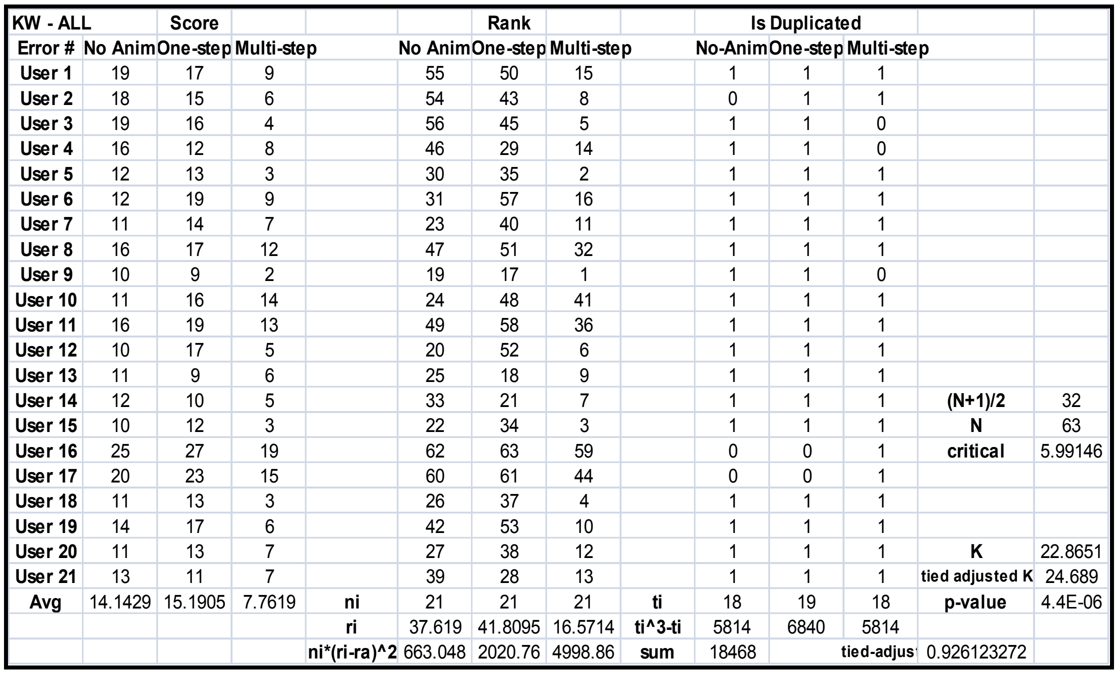
Person-to-person Correlation Analysis

	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	User 12	User 13	User 14	User 15	User 16	User 17	User 18	User 19	User 20	User 21	User 22	User 23	User 24
User 1	-0.327	0.999	1	0.945	0.961	0.589	0.807	0.929	0.9972	-0.3	0.7559	0.6897	0.9762	0.9959	-0.577	0.9198	0.9078	0.8417	0.9286	-0.655	0.8973	0.866	0.9897
User 2		-0.277	-0.327	0	-0.577	-0.956	-0.822	-0.655	-0.397	-0.803	-0.866	-0.91	-0.115	-0.24	0.9608	-0.672	-0.693	-0.786	-0.655	-0.5	-0.711	-0.756	-0.189
User 3			0.999	0.961	0.945	0.546	0.775	0.908	0.9919	-0.35	0.7206	0.6509	0.9862	0.9993	-0.533	0.8979	0.8846	0.8122	0.9078	-0.693	0.873	0.8386	0.9959
User 4				0.945	0.961	0.589	0.807	0.929	0.9972	-0.3	0.7559	0.6897	0.9762	0.9959	-0.577	0.9198	0.9078	0.8417	0.9286	-0.655	0.8973	0.866	0.9897
User 5					0.817	0.292	0.569	0.756	0.9177	-0.596	0.5	0.4148	0.9934	0.9707	-0.277	0.7406	0.7206	0.6186	0.7559	-0.866	0.7035	0.6547	0.982
User 6						0.79	0.939	0.995	0.9789	-0.024	0.9078	0.8635	0.8778	0.9316	-0.781	0.9925	0.9885	0.9584	0.9951	-0.419	0.9845	0.9707	0.9113
User 7							0.953	0.847	0.6482	0.5937	0.9744	0.9914	0.4001	0.5135	-1	0.8591	0.8737	0.9322	0.847	0.225	0.8853	0.9143	0.4678
User 8								0.969	0.8492	0.3206	0.9966	0.9842	0.66	0.7503	-0.948	0.9741	0.9803	0.9981	0.9686	-0.082	0.9848	0.9942	0.7146
User 9									0.9538	0.0751	0.9449	0.9092	0.826	0.891	-0.839	0.9997	0.9986	0.982	1	-0.327	0.9971	0.9897	0.866
User 10										-0.228	0.803	0.7422	0.9572	0.9862	-0.636	0.9466	0.9368	0.8799	0.9538	-0.596	0.928	0.9011	0.9762
User 11											0.3974	0.4834	-0.5	-0.386	-0.606	0.0981	0.1273	0.2622	0.0751	0.9177	0.1514	0.2168	-0.434
User 12												0.9954	0.596	0.6934	-0.971	0.9522	0.9608	0.9897	0.9449	0	0.9672	0.982	0.6547
User 13													0.5164	0.6212	-0.989	0.9186	0.9298	0.9715	0.9092	0.0958	0.9385	0.9594	0.5792
User 14														0.9919	-0.386	0.8128	0.7954	0.7046	0.826	-0.803	0.7803	0.737	0.9972
User 15															-0.5	0.8803	0.866	0.7892	0.891	-0.721	0.8536	0.8171	0.9986
User 16																-0.851	-0.866	-0.926	-0.839	-0.24	-0.878	-0.908	-0.454
User 17																	0.9996	0.9861	0.9997	-0.305	0.9986	0.9928	0.8543
User 18																		0.9905	0.9986	-0.277	0.9997	0.9959	0.8386
User 19																			0.982	-0.143	0.9936	0.9989	0.7559
User 20																				-0.327	0.9971	0.9897	0.866
User 21																					-0.254	-0.189	-0.756
User 22																						0.9978	0.8251
User 23																							0.7857

Using correlation coefficients to analyze user behaviors, user 2, user 16 and user 21 were removed from further analysis due to their large negative coefficient values comparing against other users.

Appendix B

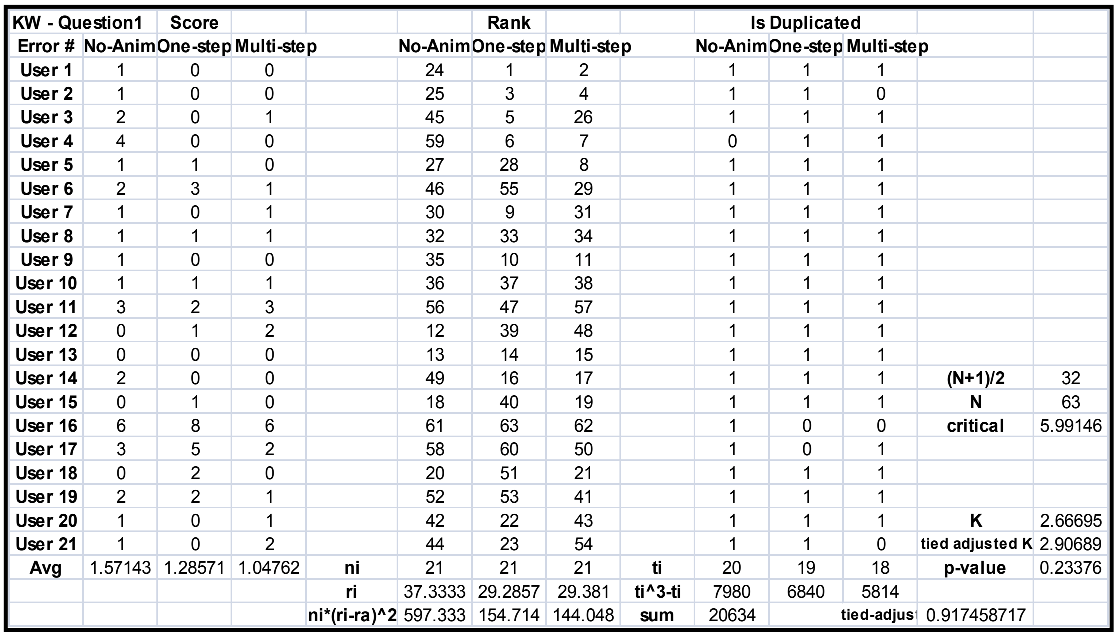
User Study Result – Accuracy KW analysis



KW - ALL		Score				Rank				Is Duplicated			
Error #	No Anim	One-step	Multi-step		No Anim	One-step	Multi-step		No-Anim	One-step	Multi-step		
User 1	19	17	9		55	50	15		1	1	1		
User 2	18	15	6		54	43	8		0	1	1		
User 3	19	16	4		56	45	5		1	1	0		
User 4	16	12	8		46	29	14		1	1	0		
User 5	12	13	3		30	35	2		1	1	1		
User 6	12	19	9		31	57	16		1	1	1		
User 7	11	14	7		23	40	11		1	1	1		
User 8	16	17	12		47	51	32		1	1	1		
User 9	10	9	2		19	17	1		1	1	0		
User 10	11	16	14		24	48	41		1	1	1		
User 11	16	19	13		49	58	36		1	1	1		
User 12	10	17	5		20	52	6		1	1	1		
User 13	11	9	6		25	18	9		1	1	1		
User 14	12	10	5		33	21	7		1	1	1	(N+1)/2	32
User 15	10	12	3		22	34	3		1	1	1	N	63
User 16	25	27	19		62	63	59		0	0	1	critical	5.99146
User 17	20	23	15		60	61	44		0	0	1		
User 18	11	13	3		26	37	4		1	1	1		
User 19	14	17	6		42	53	10		1	1	1		
User 20	11	13	7		27	38	12		1	1	1	K	22.8651
User 21	13	11	7		39	28	13		1	1	1	tied adjusted K	24.689
Avg	14.1429	15.1905	7.7619	ni	21	21	21	ti	18	19	18	p-value	4.4E-06
				ri	37.619	41.8095	16.5714	ti^3-ti	5814	6840	5814		
				ni*(ri-ra)^2	663.048	2020.76	4998.86	sum	18468		tied-adjus	0.926123272	

(a) Overall accuracy KW analysis

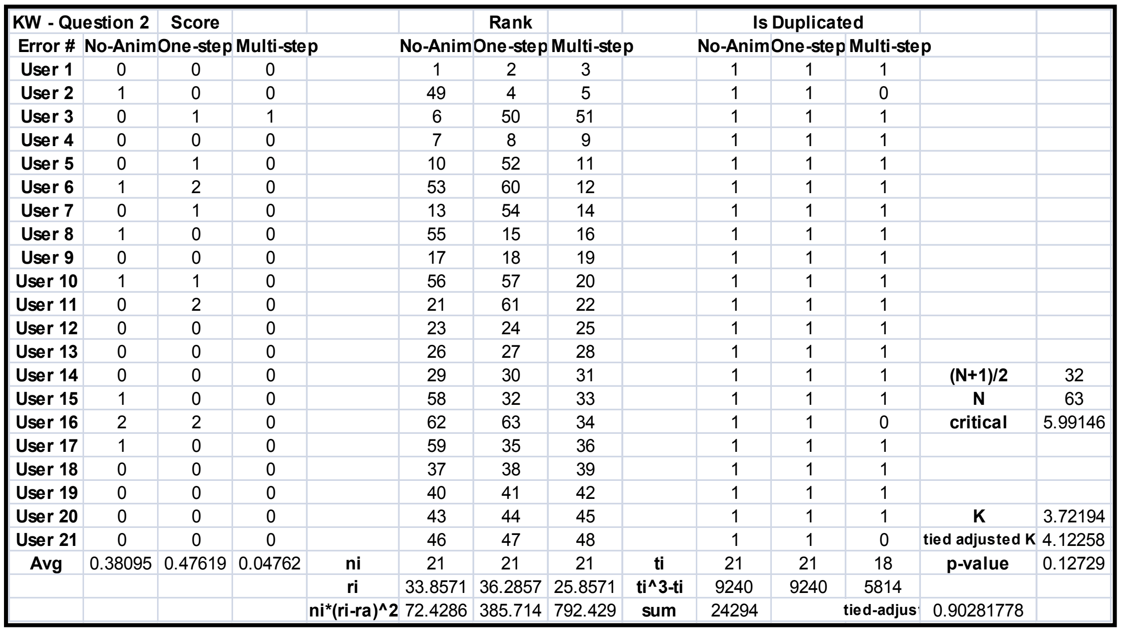
Significant difference was found among three animation conditions ($p \ll 0.001$)



KW - Question1	Score					Rank				Is Duplicated			
Error #	No-Anim	One-step	Multi-step		No-Anim	One-step	Multi-step		No-Anim	One-step	Multi-step		
User 1	1	0	0		24	1	2		1	1	1		
User 2	1	0	0		25	3	4		1	1	0		
User 3	2	0	1		45	5	26		1	1	1		
User 4	4	0	0		59	6	7		0	1	1		
User 5	1	1	0		27	28	8		1	1	1		
User 6	2	3	1		46	55	29		1	1	1		
User 7	1	0	1		30	9	31		1	1	1		
User 8	1	1	1		32	33	34		1	1	1		
User 9	1	0	0		35	10	11		1	1	1		
User 10	1	1	1		36	37	38		1	1	1		
User 11	3	2	3		56	47	57		1	1	1		
User 12	0	1	2		12	39	48		1	1	1		
User 13	0	0	0		13	14	15		1	1	1		
User 14	2	0	0		49	16	17		1	1	1	(N+1)/2	32
User 15	0	1	0		18	40	19		1	1	1	N	63
User 16	6	8	6		61	63	62		1	0	0	critical	5.99146
User 17	3	5	2		58	60	50		1	0	1		
User 18	0	2	0		20	51	21		1	1	1		
User 19	2	2	1		52	53	41		1	1	1		
User 20	1	0	1		42	22	43		1	1	1	K	2.66695
User 21	1	0	2		44	23	54		1	1	0	tied adjusted K	2.90689
Avg	1.57143	1.28571	1.04762	ni	21	21	21	ti	20	19	18	p-value	0.23376
				ri	37.3333	29.2857	29.381	ti^3-ti	7980	6840	5814		
				ni*(ri-ra)^2	597.333	154.714	144.048	sum	20634		tied-adjus	0.917458717	

(b) Question 1 accuracy KW analysis

No significant difference was found for question 1 (p = 0.234)



KW - Question 2	Score					Rank				Is Duplicated			
Error #	No-Anim	One-step	Multi-step		No-Anim	One-step	Multi-step		No-Anim	One-step	Multi-step		
User 1	0	0	0		1	2	3		1	1	1		
User 2	1	0	0		49	4	5		1	1	0		
User 3	0	1	1		6	50	51		1	1	1		
User 4	0	0	0		7	8	9		1	1	1		
User 5	0	1	0		10	52	11		1	1	1		
User 6	1	2	0		53	60	12		1	1	1		
User 7	0	1	0		13	54	14		1	1	1		
User 8	1	0	0		55	15	16		1	1	1		
User 9	0	0	0		17	18	19		1	1	1		
User 10	1	1	0		56	57	20		1	1	1		
User 11	0	2	0		21	61	22		1	1	1		
User 12	0	0	0		23	24	25		1	1	1		
User 13	0	0	0		26	27	28		1	1	1		
User 14	0	0	0		29	30	31		1	1	1	(N+1)/2	32
User 15	1	0	0		58	32	33		1	1	1	N	63
User 16	2	2	0		62	63	34		1	1	0	critical	5.99146
User 17	1	0	0		59	35	36		1	1	1		
User 18	0	0	0		37	38	39		1	1	1		
User 19	0	0	0		40	41	42		1	1	1		
User 20	0	0	0		43	44	45		1	1	1	K	3.72194
User 21	0	0	0		46	47	48		1	1	0	tied adjusted K	4.12258
Avg	0.38095	0.47619	0.04762	ni	21	21	21	ti	21	21	18	p-value	0.12729
				ri	33.8571	36.2857	25.8571	ti^3-ti	9240	9240	5814		
				ni*(ri-ra)^2	72.4286	385.714	792.429	sum	24294		tied-adjus	0.90281778	

(c) Question 2 accuracy KW analysis

No significant difference was found for question 2 (p = 0.127)

KW - Question 3	Score					Rank				Is Duplicated			
Error #	No-Anim	One-step	Multi-step		No-Anim	One-step	Multi-step		No-Anim	One-step	Multi-step		
User 1	2	1	2		34	10	35		1	1	1		
User 2	3	1	0		50	11	1		1	1	0		
User 3	2	1	0		36	12	2		1	1	1		
User 4	1	1	2		13	14	37		1	1	1		
User 5	1	1	2		15	16	38		1	1	1		
User 6	2	3	2		39	51	40		1	1	1		
User 7	1	3	2		17	52	41		1	1	1		
User 8	4	2	3		59	42	53		1	1	1		
User 9	0	1	1		3	18	19		1	1	1		
User 10	2	5	3		43	62	54		1	1	1		
User 11	2	1	1		44	20	21		1	1	1		
User 12	0	3	1		4	55	22		1	1	1		
User 13	0	0	1		5	6	23		1	1	1		
User 14	0	0	1		7	8	24		1	1	1	(N+1)/2	32
User 15	1	2	1		25	45	26		1	1	1	N	63
User 16	3	3	2		56	57	46		1	1	0	critical	5.99146
User 17	4	5	2		60	63	47		1	1	1		
User 18	2	0	1		48	9	27		1	1	1		
User 19	2	4	1		49	61	28		1	1	1		
User 20	1	1	1		29	30	31		1	1	1	K	0.00368
User 21	1	1	3		32	33	58		1	1	0	tied adjusted K	0.00408
Avg	1.61905	1.85714	1.52381	ni	21	21	21	ti	21	21	18	p-value	0.99796
				ri	31.8095	32.1429	32.0476	ti^3-ti	9240	9240	5814		
				ni*(ri-ra)^2	0.7619	0.42857	0.04762	sum	24294		tied-adjus	0.90281778	

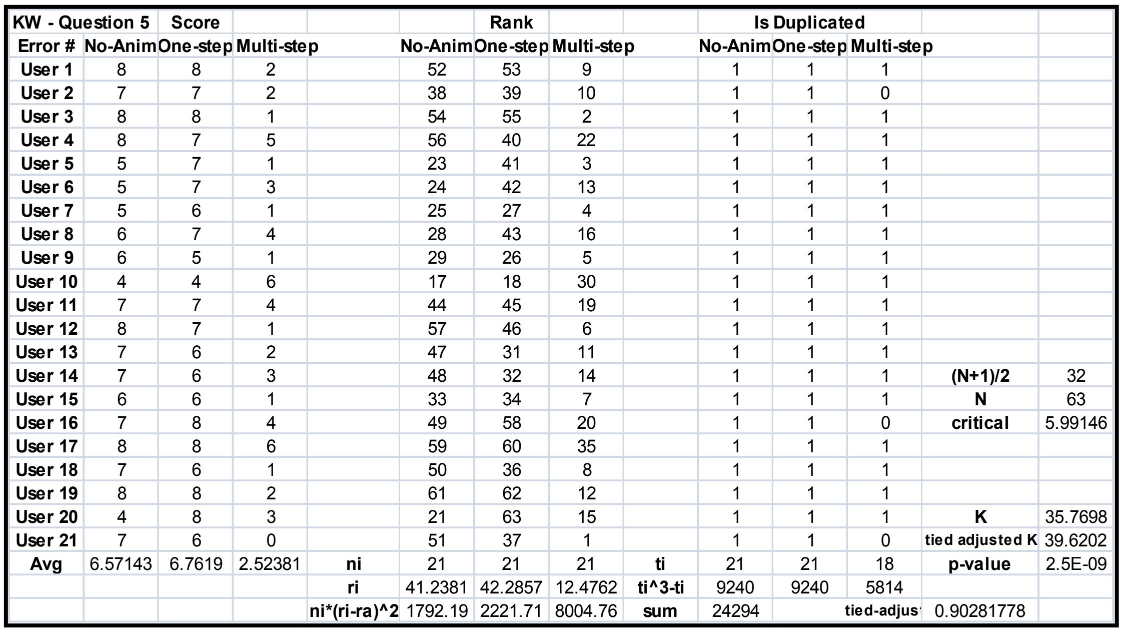
(d) Question 3 accuracy KW analysis

No significant difference was found for question 3 (p = 0.998)

KW - Question 4	Score					Rank				Is Duplicated			
Error #	No-Anim	One-step	Multi-step		No-Anim	One-step	Multi-step		No-Anim	One-step	Multi-step		
User 1	8	8	5		62	63	44		1	1	1		
User 2	6	7	4		52	56	29		1	1	0		
User 3	7	6	1		57	53	3		1	1	1		
User 4	3	4	1		17	30	4		1	1	1		
User 5	5	3	0		45	18	1		1	1	1		
User 6	2	4	3		9	31	19		1	1	1		
User 7	4	4	3		32	33	20		1	1	1		
User 8	4	7	4		34	58	35		1	1	1		
User 9	3	3	0		21	22	2		1	1	1		
User 10	3	5	4		23	46	36		1	1	1		
User 11	4	7	5		37	59	47		1	1	1		
User 12	2	6	1		10	54	5		1	1	1		
User 13	4	3	3		38	24	25		1	1	1		
User 14	3	4	1		26	39	6		1	1	1	(N+1)/2	32
User 15	2	3	1		11	27	7		1	1	1	N	63
User 16	7	6	7		60	55	61		1	1	0	critical	5.99146
User 17	4	5	5		40	48	49		1	1	1		
User 18	2	5	1		12	50	8		1	1	1		
User 19	2	3	2		13	28	14		1	1	1		
User 20	5	4	2		51	41	15		1	1	1	K	13.3095
User 21	4	4	2		42	43	16		1	1	0	tied adjusted K	14.7422
Avg	4	4.80952	2.61905	ni	21	21	21	ti	21	21	18	p-value	0.00063
				ri	32.9524	41.8095	21.2381	ti^3-ti	9240	9240	5814		
				ni*(ri-ra)^2	19.0476	2020.76	2432.19	sum	24294		tied-adjus	0.90281778	

(e) Question 4 accuracy KW analysis

Significant difference was found for question 4 (p = 0.0006)



KW - Question 5	Score					Rank				Is Duplicated			
Error #	No-Anim	One-step	Multi-step		No-Anim	One-step	Multi-step		No-Anim	One-step	Multi-step		
User 1	8	8	2		52	53	9		1	1	1		
User 2	7	7	2		38	39	10		1	1	0		
User 3	8	8	1		54	55	2		1	1	1		
User 4	8	7	5		56	40	22		1	1	1		
User 5	5	7	1		23	41	3		1	1	1		
User 6	5	7	3		24	42	13		1	1	1		
User 7	5	6	1		25	27	4		1	1	1		
User 8	6	7	4		28	43	16		1	1	1		
User 9	6	5	1		29	26	5		1	1	1		
User 10	4	4	6		17	18	30		1	1	1		
User 11	7	7	4		44	45	19		1	1	1		
User 12	8	7	1		57	46	6		1	1	1		
User 13	7	6	2		47	31	11		1	1	1		
User 14	7	6	3		48	32	14		1	1	1	(N+1)/2	32
User 15	6	6	1		33	34	7		1	1	1	N	63
User 16	7	8	4		49	58	20		1	1	0	critical	5.99146
User 17	8	8	6		59	60	35		1	1	1		
User 18	7	6	1		50	36	8		1	1	1		
User 19	8	8	2		61	62	12		1	1	1		
User 20	4	8	3		21	63	15		1	1	1	K	35.7698
User 21	7	6	0		51	37	1		1	1	0	tied adjusted K	39.6202
Avg	6.57143	6.7619	2.52381	ni	21	21	21	ti	21	21	18	p-value	2.5E-09
				ri	41.2381	42.2857	12.4762	ti^3-ti	9240	9240	5814		
				ni*(ri-ra)^2	1792.19	2221.71	8004.76	sum	24294		tied-adjus	0.90281778	

(f) Question 5 accuracy KW analysis

Significant difference was found for question 5 ($p \ll 0.001$)

Appendix C

User Study Result – Overall Task Completion Time

KW Analysis

		Score				Rank				Is Duplicated			
Avg time	**No Anim**	**One-step**	**Multi-step**		**No Anim**	**One-step**	**Multi-step**		**No Anim**	**One-step**	**Multi-step**		
User 1	12.1896	18.339	11.5896		11	48	10		0	0	0		
User 2	17.3615	25.3175	15.07488		44	61	35		0	0	0		
User 3	23.5701	32.49413	19.5817		58	63	51		0	0	0		
User 4	13.637	20.74785	13.50555		22	54	20		0	0	0		
User 5	15.8811	16.20873	14.8755		38	39	31		0	0	0		
User 6	8.78058	12.5252	13.42858		1	13	19		0	0	0		
User 7	14.9902	21.62745	14.9252		33	55	32		0	0	0		
User 8	14.621	18.26778	17.72855		30	47	45		0	0	0		
User 9	9.85123	15.35648	13.62388		4	36	21		0	0	0		
User 10	8.8888	10.88895	14.4988		2	6	29		0	0	0		
User 11	9.1775	11.23508	14.22815		3	9	26		0	0	0		
User 12	13.2418	17.21053	10.76108		18	43	5		0	0	0		
User 13	14.1971	24.62695	18.66753		25	60	49		0	0	0		
User 14	13.0484	23.7976	14.077		17	59	24		0	0	0	**(N+1)/2**	32
User 15	14.3138	19.06308	10.97163		28	50	7		0	0	0	**N**	63
User 16	12.9686	16.70225	22.57343		16	41	57		0	0	0	**critical**	5.9915
User 17	14.2731	19.62125	18.19545		27	52	46		0	0	0		
User 18	21.8984	31.95368	16.2816		56	62	40		0	0	0		
User 19	12.5575	13.9685	15.0486		14	23	34		0	0	0		
User 20	15.436	19.70028	12.33468		37	53	12		0	0	0	**K**	13.61
User 21	11.0302	16.84463	12.93473		8	42	15		0	0	0	**tied adjusted K**	13.61
Avg	13.9006	19.35699	14.99553	**ni**	21	21	21	**ti**	0	0	0	**p-value**	0.0011
				ri	23.4286	43.61905	28.95238	**ti^3-ti**	0	0	0		
				ni*(ri-ra)^2	1542.86	2835.048	195.0476	**sum**	0		**tied-adjust**	1	

Overall task completion time KW analysis

Significant difference was found among three animation conditions (p = 0.0011)

Appendix D

User Study Result – User Preference KW Analysis

		Score				Rank				Is Duplicated			
Avg grade	**No-Anim**	**One-step**	**Multi-step**		**No-Anim**	**One-step**	**Multi-step**		**No-Anim**	**One-step**	**Multi-step**		
User 1	3	2.5	5		30	22	56		1	1	1		
User 2	1.5	1	5		9	1	57		1	1	1		
User 3	1.5	1.5	4.666667		10	11	49		1	1	1		
User 4	4.66667	2.833333	3.833333		50	28	41		1	1	1		
User 5	2.66667	2.5	5		25	23	58		1	1	1		
User 6	3.5	2.5	4.333333		36	24	44		1	1	1		
User 7	2.66667	2.333333	4.5		26	19	47		1	1	1		
User 8	1.33333	1.5	4.833333		6	12	53		1	1	1		
User 9	2.33333	2.833333	5		20	29	59		1	1	1		
User 10	3.83333	4.5	5		42	48	60		1	1	1		
User 11	2	2	3.5		15	16	37		1	1	1		
User 12	3	3.166667	5		31	32	61		1	1	1		
User 13	2	1	5		17	2	62		1	1	1		
User 14	2	1.666667	4.833333		18	13	54		1	1	1	**(N+1)/2**	32
User 15	3.5	3.166667	5		38	33	63		1	1	1	**N**	63
User 16	4	4.333333	4.666667		43	45	51		0	1	1	**critical**	5.9915
User 17	1.16667	1.666667	3.666667		5	14	40		0	1	0		
User 18	3.5	3.333333	4.333333		39	34	46		1	1	1		
User 19	1	1	3.333333		3	4	35		1	1	1		
User 20	1.33333	1.333333	4.666667		7	8	52		1	1	1	**K**	35.538
User 21	2.33333	2.666667	4.833333		21	27	55		1	1	1	**tied adjusted K**	39.322
Avg	2.51587	2.349206	4.571429	**ni**	21	21	21	**ti**	19	21	20	**p-value**	3E-09
				ri	23.381	21.19048	51.42857	**ti^3-ti**	6840	9240	7980		
				ni*(ri-ra)^2	1560.05	2453.762	7926.857	**sum**	24060		**tied-adjust**	0.90375384	

User preference KW analysis

Significant difference was found among three animation conditions ($p \ll 0.001$)