Multi-touch Document Folding: Gesture Models, Fold Directions and Symmetries

Patrick Chiu, Chunyuan Liao, Francine Chen
FX Palo Alto Laboratory
3400 Hillview Avenue, Bldg. 4, Palo Alto, CA 94304, USA
{chiu, liao, chen}@fxpal.com

ABSTRACT

For document visualization, folding techniques provide a focus-plus-context approach with fairly high legibility on flat sections. To enable richer interaction, we explore the design space of multi-touch document folding. We discuss several design considerations for simple modeless gesturing and compatibility with standard Drag and Pinch gestures. We categorize gesture models along the characteristics of Symmetric/Asymmetric and Serial/Parallel, which yields three gesture models. We built a prototype document workspace application that integrates folding and standard gestures, and a system for testing the gesture models. A user study was conducted to compare the three models and to analyze the factors of fold direction, target symmetry, and target tolerance in user performance when folding a document to a specific shape. Our results indicate that all three factors were significant for task times, and parallelism was greater for symmetric targets.

Author Keywords

Multi-touch screens, gesture input, document visualization.

ACM Classification Keywords

H5.2. [Information interfaces and presentation]: User Interfaces - Input devices and strategies.

General Terms

Design, Experimentation, Human Factors.

INTRODUCTION

Multi-touch displays are becoming more widely used and appealing: in addition to the iconic Apple iPhone & iPad devices and the compelling Perceptive Pixel systems, other mobile phones, desktop PCs and large flat panels with multi-touch hardware have been productized by Google, HP, Smart Technologies, etc. Another trend is the increasing utilization of 3D graphics on personal computers and operating systems such as the Mac OS and Windows 7. These technologies are providing opportunities and challenges for the development of new applications and interaction techniques for multi-touch 3D graphical user interfaces.

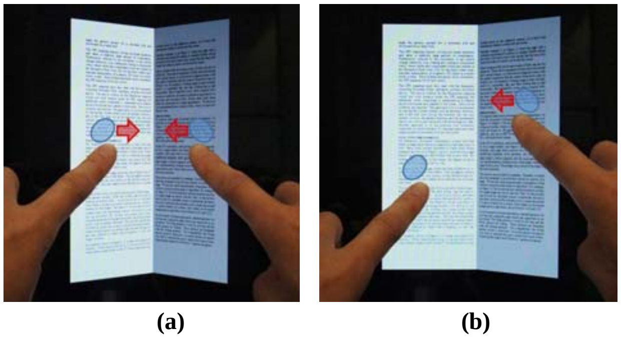


Figure 1. Examples of multi-touch document folding gestures: (a) Symmetric-Parallel, (b) Asymmetric-Serial.

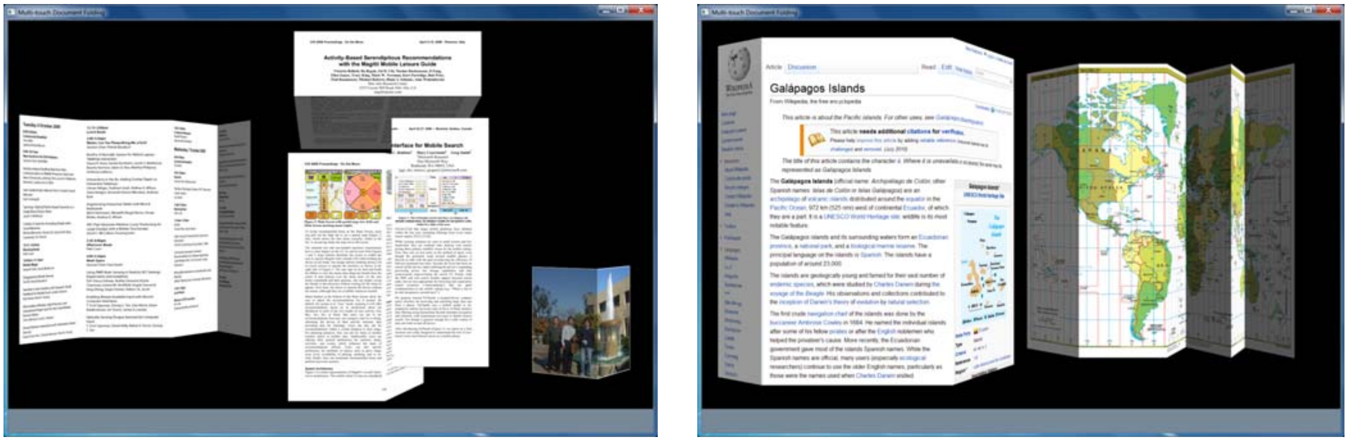
In this paper, we focus on the use of these technologies for the problem of interactive document visualization that enables viewing part of one or more document pages simultaneously, but in context. Toward this end, an effective principle of information visualization is that of focus-plus-context, where the content in a focus region is rendered at high fidelity with minimal distortion, and the content in a peripheral context region is rendered with a certain amount of distortion to make efficient use of screen space or computational resources.

A well-known example of a focus-plus-context technique is the fisheye view. However, this works poorly for documents due to legibility problems, as the fisheye view transforms straight lines into curves.

A better focus-plus-context approach for visualizing documents is to use folding. The idea is to render the objects or pages in 3D so that the focus regions are flat and parallel to the screen without distortion, and the context regions are folded off to the sides with perspective distortion. An early example is the Perspective Wall [16], which has a center wall facing the user and two walls angled off each side. A more recent visualization, Melange [9], uses a 3D space folding metaphor with multiple focus regions parallel to the screen and the rest of the space folded up.

Although these systems effectively address the visualization aspect, good techniques for interacting with folding

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.
Copyright 2011 ACM 978-1-4503-0267-8/11/05....$10.00.



(a) (b)

Figure 2. Screenshots of application with folded documents.

visualizations are still lacking. With the Perspective Wall [16] and Melange [9], a standard mouse and keyboard setup is used. Even going to a more direct style of interaction on a touch screen with a single input point would not be very natural—in the real world, folding a piece of paper is usually done with more than one finger and with two hands.

Our approach is to employ multi-touch screens, supporting more natural and direct interaction by allowing fingers from both hands to be used together. It is worth pointing out that for the folding task, multi-touch technology can be utilized to enable new forms of interaction that are simpler and modeless, and not just to perform tasks that can be done serially in a more efficient way (e.g. combinations of positioning-scaling and scrolling-selecting [3], selecting multiple targets [14]).

To frame the problem better, we will discuss several design considerations for simple gesture interfaces and for compatibility with standard multi-touch gestures. We then present two practical multi-touch gesture models for folding document page objects, along with a more general categorization based on the characteristics of Symmetric/Asymmetric roles of the hands and Serial/ Parallel use of the hands.

For exploring gesture interaction techniques, we have developed a prototype for folding document page objects in a 3D visualization. See Figure 1 and Figure 2. Useful types of folds for document visualization will be described including basic folds and accordion folds. We will also explain different ways of mapping the 2D gestures into folding actions in 3D space.

For understanding the performance of multi-touch gesture folding interactions, we have identified several factors and measures. Some of these, such as target tolerance and parallelism, have been studied in the context of bimanual interaction (e.g. [3], [11]) and the asymmetry between preferred[1] and non-preferred hands (e.g. [1], [10], [13]). Hence, we focus on those factors that are relatively less understood and pertain to folding, such as the folding direction and target symmetry.

A user study was conducted to compare the multi-touch gesture models given by our categorization, and to analyze the various factors involved. We found many of the factors to be significant. The study also suggests that both of our practical gesture models are viable, with tradeoffs that can make one more suitable than the other depending on the requirements of the specific application.

RELATED WORK

We have already mentioned Perspective Wall [16] and Melange [9] as systems that visualize information using folding techniques. Our interface also differs from these systems in the geometry used. While the Perspective Wall has a fixed geometric structure with changeable content rendered on it, our 3D page objects have a more dynamic geometry. And unlike Melange, our folding is performed on the objects and not on the space.

Another interface, Roller [21], uses an effect similar to folding for displaying information targeted for small displays on mobile devices. It has a trigonal prism with sections wrapped around it, and rolling and unrolling it allows the sections to be revealed or hidden.

Multi-touch interaction has been the subject of much research (e.g. see [4], [22]). It has been applied to origami folding, which has somewhat different goals than folding for document visualization. In origami, sections are often folded flat against each other and diagonal folds are common; whereas in document folding, sections need to be folded independently and not always completely. In the Origami Simulator system [5], combinations of touch points of two different sizes (finger and knuckle) correspond to origami functions which are not really suitable for document folding. For example, a 1-touch knuckle gesture results in an action that folds two sections completely together with animation.

[1] The *preferred hand* is also known as the *dominant hand*.

There are other interactive origami visualization systems at various stages of development, and the general computation problem is hard [15]. Most of these systems are not focused on supporting fluid interactions. For example, in Foldinator [19], performing a fold requires several operations with buttons and mouse strokes.

Other systems with related interactive folding techniques include the following: folding windows [2], paper flyer metaphors [7], Fold-and-Drop [8]. None of these support multi-touch. In contrast, LiquidText [20] is a multi-touch system that has a feature to collapse parts of the text via 2D interline compression. And the Microsoft Research physics-based system for interactions on surfaces [23] supports multi-touch operations for deforming and folding cloth-like meshes.

Since our gesture models are bimanual, the research in this area provides motivation and guidelines for designing interaction methods. For asymmetric methods with respect to the roles of the two hands, Guiard's Kinematic Chain theory [10] provides three principles that can be applied to our designs: the non-preferred hand helps delineate frames for the task, the preferred hand is more suitable for higher precision actions, and the non-preferred hand naturally starts before the preferred hand. For symmetric methods, a study on symmetric bimanual interaction [1] explores the visual integration of targets in a tracking task. In this paper, a different factor that we explore is the symmetry of targets; in particular whether these can be performed with greater parallelism and lower task times. Other relevant work in this well-studied area include: [3], [6], [11], [13], [14], [22].

GESTURE MODELS FOR FOLDING

In this section, we discuss some design considerations for gesture models, two practical models, and a categorization along two characteristics. We also describe several mappings of 2D gestures into 3D folding actions.

Design Considerations

For ease of use and to minimize gesturing errors, it is a good idea to use extremely simple gesture shapes such as those that have been successfully deployed on commercial products (e.g. iPhone/iPad and Windows 7).

To make it easier for novice users to learn, we avoid more sophisticated techniques involving complex gestures in which a single gesture path can be parsed into several gestures, such as detecting certain geometric features or other delimiters in the gesture (e.g. [12]). These complex gestures can be used for performing multiple selections and commands with a single gesture, but can be deprecated with multi-touch technology, which enables support of separate gesture paths using different fingers.

Modes should be avoided if possible, as they can be a source of interaction errors in software systems. For a gesture interface, having modes (e.g. one for translation and scaling, another for folding) would make the interaction less fluid. Furthermore, on larger screens, having to reach for buttons or menus to control the modes is awkward and interrupts the task.

Another important consideration is compatibility with standard gestures popularized in commercial products like the iPhone/iPad and Windows 7. Two important ones are the "Drag" and "Pinch" gestures. With the Drag gesture, using a single finger to drag on an object moves it along the drag direction.

With the Pinch gesture, touching an object with two fingers and moving them together or apart results in a scaling operation. We note that some implementations are more lax than the specification: moving just one finger (while holding the other finger still) is sufficient to perform a Pinch gesture. For our designs, however, we will be stricter and a gesture is interpreted as a Pinch gesture only if *both* fingers move together or apart. This will enable a richer supply of gestures.

There are other standard multi-touch gesture operations, but except for the Pinch gesture, these will not conflict with our gesture schemes for folding. We mention one other operation, which is part of the fundamental rotate-scale-translate operations. For translate and scale operations, we have already described the Drag and Pinch gestures. For the rotate operation, one common gesture in Windows 7 is to touch with two fingers and turn them simultaneously. Alternatively, the user can rotate by orbiting a finger around another stationary finger. The rotate operation is useful for tabletop displays where orientation matters, but it is not always needed for applications on vertical displays like desktop and wall displays.

To summarize our design considerations:

D1. Avoid complex gestures.
D2. Avoid modes.
D3. Compatibility with standard gestures.

Two Practical Gesture Models for Folding

Based on the above design considerations, we developed two practical multi-touch gesture models for folding. Depending on the requirements of the target application, one model and its variations may be more suitable than the other. Also, some variations may be quicker to perform than others or may be more compatible with existing standard gestures.

Symmetric-Parallel (SP)

In a simple multi-touch gesture model for folding, which we call *Symmetric-Parallel* (or *SP* for short), the user employs a finger from each hand to touch and select two sections of the document page object, and then drags the fingers to fold the sections about the crease between the two sections. This is illustrated in Figure 3.

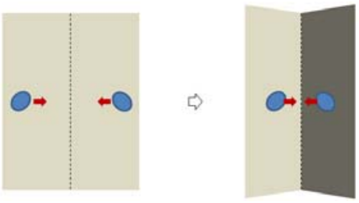


Figure 3. Symmetric-Parallel gesture model: A folding operation in which contact points can move simultaneously.

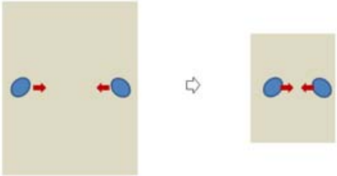


Figure 4. A standard Pinch gesture operation for scaling.

The operation is *symmetric* in the sense that both finger touch events are interpreted in the same way: touching a section selects it, and dragging a touched section folds it. Another way to think about it is that both hands are assigned identical roles [1]. It is *parallel* because both fingers can move simultaneously.

While conceptually simple, in practice this SP gesture model can be a bit difficult to use for novice users. We prototyped and informally tested this model on a few users and discovered that they had difficulties performing basic folding tasks with it. One task we tested is that given a target wireframe, the user had to fold two sections along a fixed crease of the document page object to match the target wireframe. Our observations were that users had difficulty moving the two sections simultaneously, or at least had to concentrate very hard. One reason could be that the users had trouble focusing on doing two things at once. For an expert user (e.g. project members who have spent many hours with the prototype), the SP model was not difficult to use. As we will see in the user study section below, users can be relatively fast but the perceived mental effort under this gesture model is greater.

Another design issue with the SP model is that it may violate our design considerations *D2* or *D3*, depending on how it is adopted. It is not compatible with the standard multi-touch Pinch gesture for scaling an object because the two contact points are moving simultaneously toward or away from each other. Compare Figure 3 and Figure 4. In order to employ the Symmetric model, extra mechanisms would be required such as providing modes or supporting more than two touch points—which further complicates the system and usability.

Thus, the SP model may be more appropriate for specialized applications like a dedicated origami application. A reasonable design for the origami application is to let the folding be the main interaction handled with gestures, and let the scaling be a secondary operation that can be handled with a slider widget.

Asymmetric-Serial (AS)

To address the deficiencies of the SP model, we developed a second gesture model that we call *Asymmetric-Serial* (or *AS*). It is motivated by how people often work with physical objects, and is consistent with Guiard's theory on asymmetric bimanual action discussed above. For example, when cutting an orange with a knife, the non-preferred hand holds the orange and does not move, while the preferred hand cuts the orange by moving the knife. For folding a document page object, our AS gesture model uses a finger from the non-preferred hand to touch a section and hold it fixed, followed by the use of a finger from the preferred hand to touch and drag a second section to perform a folding action. See Figure 5 to Figure 7.

With the AS gesture model, one hand is always held fixed while the other hand moves. In preliminary informal testing, this seemed easier for novice users to manage and requires less coordination than the SP gesture model.

A design advantage of the AS model is that it satisfies our design consideration *D3* since it is compatible with standard multi-touch gestures. It does not produce the same sequence of touch input events as the Pinch gesture.

A Categorization of Gesture Models

The two bimanual multi-touch gesture models described above can be viewed as belonging to a categorization along two sets of characteristics: {Symmetric, Asymmetric} and {Serial, Parallel}. See Table 1.

The Symmetric/Asymmetric characteristic refers to how the two hands function. The Serial/Parallel characteristic refers to whether simultaneous gesturing by both hands is enabled.

The categorization yields a third gesture model, *Symmetric-Serial* (*SS*). This model is not very practical because there is no reason to limit the user from moving both hands simultaneously since they function independently. Nevertheless, the SS model provides a good benchmark for comparing with the SP and AS models, as it differs from each along one characteristic.

The last case, *Asymmetric-Parallel*, is not possible because our definition of asymmetric is based on the non-preferred hand staying still while the preferred hand is moving.

We note that one can imagine using a single hand with two contact points to fold a document page object, but there are several drawbacks. While it is easy to move the fingers in a coupled manner like the Pinch gesture, it is much more difficult to move the fingers independently to manipulate two sections of a page to different positions, especially when trying to precisely position each section. Another problem is that sometimes the fingers need to be placed at locations that are far apart on the page object, which may be impossible to do with one hand (e.g. see Figure 8).

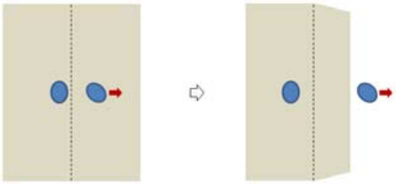


Figure 5. Asymmetric-Serial gesture model: A folding operation in which one contact point is held fixed, and another contact point is moved.

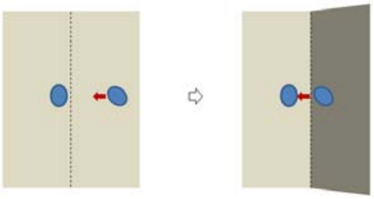


Figure 6. Another instance of Figure 5 where the folding is in the opposite direction.

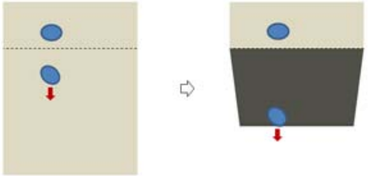


Figure 7. Another instance of Figure 5 where the folding is with respect to a horizontal crease.

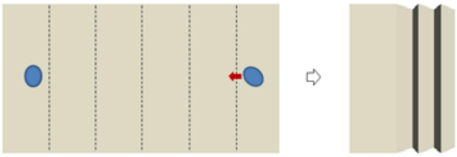


Figure 8. An accordion fold.

Mapping 2D Gestures into 3D Folding Actions

The direction and amount of folding, which is sampled every few milliseconds and dynamically rendered, are determined by the moving gesture paths. There are different ways to map the gesture path events, and we describe three possibilities:

- *Logical*: The amount of gesture movement perpendicular to the projection of the crease onto the screen is mapped to the fold angle. The fold direction is governed by a convention such as the "right hand rule" from vector geometry.
- *Directional*: The fold direction is based on whether the two fingers are moving toward each other or moving apart. The fold angle is based on the amount of movement.
- *Contact Point Constrained (CPC)*: The contact point of the finger on the screen corresponds to a point on a section via the projection of the section in 3D space onto the 2D screen, and this point on the section follows the movement of the finger so that it is always under the contact point.

The Logical mapping is easy for the user to understand; the finger motion is like moving a slider that controls the amount of rotation or bending. From the system standpoint, the Logical mapping is well-defined and simple to implement. Figure 5 and Figure 6 illustrate the Logical mapping, with the crease defining a fold axis vector that points upward. Dragging to the right rotates the section into the screen (a "Mountain" fold, Figure 5), and dragging to the left rotates out toward the screen (a "Valley" fold, Figure 6).

	Serial	Parallel
Symmetric	SS	SP
Asymmetric	AS	n/a

Table 1. A categorization of bimanual gesture models for folding.

For the user, the Directional and CPC might be more intuitive in some ways. With these two mappings, however, there is one problem that happens to occur at the most common configuration where the page object is completely flat and facing the user. If the user moves a finger in a straight line toward the other finger, the folding operation is not well-defined because there is an ambiguity: the section can be either a Mountain or Valley fold. Therefore, the gesture must encode more information to specify which way to fold. One solution for the Directional mapping is to angle the gesture upward for a Mountain fold and downward for a Valley fold.

The CPC mapping requires solving non-linear equations and takes a relatively large amount of computation. Furthermore, induced by the constraints, it can exhibit strange behaviors (e.g. rotating and scaling in unexpected ways); these subtle and complicated issues are studied in the Perceptive Pixel work [18] for the basic rotate-scale-translate manipulation of a generic 3D object in a multi-touch system. Since folding introduces more constraints, the CPC mapping will be less robust.

The CPC mapping must also deal with the frequently occurring Mountain-Valley ambiguity problem described above. The above solution for the Directional mapping would not work because under the CPC mapping, the additional vector component is interpreted as a rotation about a horizontal axis parallel to the screen. More clever solutions are required. One approach is to use momentum by having the system initially pick an arbitrary direction, and if the fold does not go the intended way, the user quickly reverses the gesture direction and the momentum carries the section pass the point with the ambiguous state. Another approach is to use pressure (not commonly supported on touch displays) by sensing a hard press and mapping that to a mountain fold into the screen.

FOLD TYPES AND EXAMPLES

For interactive document visualization, a basic fold that bends a page object about a single axis is the main type of fold. Basic folds include Mountain or Valley folds. Some examples are shown in Figure 2a above, where the three pages in the middle and the photo on the right are each folded with a basic fold. By folding the two page objects in the center, the figures in those pages can be compared side-by-side.

In Figure 2b, the Web page object has been folded twice, to achieve a Perspective Wall effect. The navigation section on the right and the side section on the left have been de-emphasized. This also makes more efficient use of screen real estate.

The accordion fold is a useful fold inspired by folded maps. Both the SP and AS gesture models can be used. An accordion fold is activated when there is at least one section between the two sections touched by the fingers. See Figure 8. As the user moves a contact point, the intervening sections will fold in or out like an accordion. By holding one contact point fixed (required under the AS gesture model), the fixed contact point's section remains stationary. This has the benefit that a folded up object can be easily selected and manipulated; otherwise, the object or its sections may become too narrow and thus difficult to select with a fingertip (i.e. the "fat finger" problem).

An example shown in Figure 2b is a map consisting of 6 sections. It has been folded twice: (1) an accordion fold with a fixed contact point on the 2nd section and a moving contact point on the rightmost section, (2) a basic fold bending the leftmost section into the screen.

Another example in Figure 2a is the schedule on the left with four sections that have been similarly folded with an accordion fold and a basic fold, except that the basic fold bends the leftmost section out toward the screen.

More complex origami style folds involving a series of folds can also be considered. These can be sequences of folds about horizontal and vertical creases, or a fold about a diagonal crease, with the added complexity that the creases themselves may be folded over. An example is folding a page in half vertically until it is completely flat, and then folding that in half horizontally. Due to legibility issues, these complex folds are not as useful for document visualization applications as the basic and accordion folds. We have not yet implemented these complex folds, which we leave for future work.

PROTOTYPES

To explore multi-touch document folding interactions, we built the document workspace application shown above in Figure 1 and Figure 2. We also built a special version for performing user studies, which is described further below. These were developed as 3D applications on the Windows 7 platform, and run on HP TouchSmart desktop and laptop/tablet PCs with multi-touch screens.

In the 3D visualization, each document page object appears to be a single artifact (see Figure 2). Each page object actually consists of a 3D model with an underlying triangular mesh geometry, and the document page image is texture mapped onto the mesh geometry. In our implementation, folding occurs at the edges where two page sections meet. These edges are a subset of the edges of the mesh geometry.

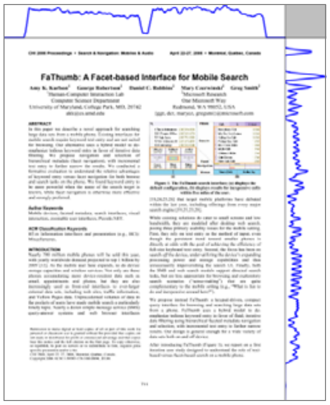


Figure 9. Projection profiles of a page image.

The sections are specified by metadata. Currently, we hand-label the metadata to describe the columns or sections of a page. We are also exploring automatic computation of the sections using document analysis algorithms [17]. For example, in Figure 9, the horizontal and vertical projection profiles of a page image are shown, and the "dips" in the profiles can be used as candidate crease locations. In particular, the deep dip in the vertical profile indicates a two column layout and the center of the dip is a good location for a crease. To interact with fold lines, one possible method is for the user to perform a Hold gesture to bring up a context menu to select and adjust, delete, or create a new fold line. The local minima ("dips") in the projection profile can help the user snap to a good location, mitigating the "fat finger" problem.

As a document workspace application, standard gestures for moving and scaling the page objects are supported in our prototype. The Drag gesture is used for moving, and the Pinch gesture for scaling. These gestures are integrated with the folding gestures. For compatibility, the AS gesture model is employed. For simplicity, the Logical mapping of gestures into folding actions is used. A custom gesture recognizer was built to interpret the user's multi-touch input events, bypassing the Windows 7 gesture recognizer.

This prototype enabled us to explore various designs for the gesture models, and designs for the mesh geometries with document page images. Working with the prototype had a great impact on the design process; for example, this is how we noticed that the SP gesture model may not be easy for some users and led to the development of the AS model.

USER STUDY

We conducted a controlled experiment to compare the three bimanual multi-touch gesture models in Table 1, and to learn about the various factors that affect the performance and the degree of parallelism of the two hands.

Experiment Task

To be general, we focus on the folding gestures without integration with others such as Drag and Pinch. We asked each participant to perform a basic folding task on a multi-touch screen, under different parameters and using each of the three gesture models in Table 1: SS, SP, AS.

We built a specific testing system, where there is only one object representing a blank sheet of paper with a vertical crease down the middle, along with a target wireframe (see Figure 10). The target wireframe represents a desired location with a specified tolerance, and is rendered as two wedges, one for each section of the object. At the beginning of a trial task, the sheet is flat and facing the user. The target wireframe is presented to the user, and the task is to fold the two sections so that they are placed inside the wedges.

When a section is inside a wireframe wedge, the wedge changes color (from blue to red). For the serial gesture models (SS and AS), the user always manipulates the section on the preferred-hand side first, and the other target wireframe wedge is grayed out until the first side is completed. All three gesture models were implemented along with the Logical gesture mapping.

The device employed for the user study was a HP TouchSmart tx2 tablet (convertible laptop). It has a multi-touch capacitive screen (12.1", 1280x800 resolution).

Hypotheses

We examine the factors that have significant impact on the user performance of folding and that have been less studied in the literature. In particular, we consider the gesture models, fold directions, target symmetry, as well as the more customary target tolerance.

First of all, the gesture models (SS, SP and AS) certainly play a key role in user performance. We hypothesize that SS is slower than SP because it does not have the time savings from simultaneous action, and SS is slower than AS because the preferred hand is faster despite having to move that hand from one section over to the other. For SP vs. AS, it is not clear which is faster because there is a tradeoff between simultaneous action and preferred hand dexterity.

Second, different fold directions, i.e. Mountain (Figure 11a) or Valley (Figure 11b), involve different interactions that may affect user performance. We did not consider other folding tasks in which one section bends into the screen and the other section out, as this is not a common type of fold. The Mountain fold may be more difficult than the Valley fold due to the 3D perspective projection. Since the Mountain target wireframes are "behind the screen" and farther away from the user, the widths of the rendered targets are smaller in screen coordinates. Another issue is the partial occlusion of the Mountain wireframe (see the left section in Figure 11a), but in a pilot study this was not observed to be a problem. On the other hand, the Valley fold has the potential of the fingers obstructing each other as they move.

Third, the symmetry of the targets is an interesting factor. Although two hands are used, if the participant can coordinate and mentally "couple" both hands so that they move in unison, the task time can be decreased through this kind of parallelism.

Finally, the target tolerances will affect the task times, which can be predicted by Fitts' law. Our aim is not to focus on or to verify Fitts' law, but we expect the results to be consistent with it.

To summarize our hypotheses:

H1. SS takes longer to perform than either SP or AS.

H2. For fold directions, the Valley fold takes less time than the Mountain fold.

H3. (a) Symmetric targets take less time than asymmetric targets. (b) Symmetric targets are performed with more parallelism than asymmetric targets.

H4. A small target tolerance takes more time than a large tolerance.

Independent Variables

To test the hypotheses, we chose the above four key factors as the independent variables and examined the impact of different factor values. For gesture models, all the three possible settings (SP, SS, AS) were tested.

For fold directions, participants were given an equal number of Mountain and Valley folding tasks for all four combinations of two different angles (30 and 60 degrees) on the left and right sides. The aggregate distance covered

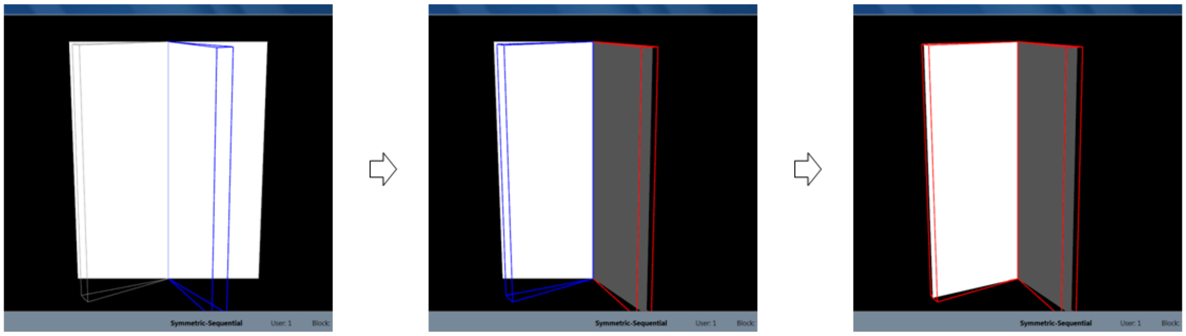


Figure 10. Folding on the user study testing application (the left and right sides of the screenshots cropped for space).

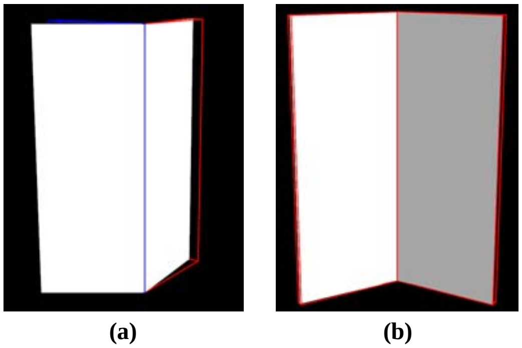


Figure 11. (a) An example of Mountain fold. (b) Symmetric target with small target size.

by each direction type is the same over the set of tasks (and also over subsets with the other factors' levels held fixed).

For target symmetry, half the tasks were symmetric targets and half were asymmetric. As with the fold direction, the aggregate distance covered by each task symmetry type is the same over the set of tasks.

We tested two target tolerances: small (wedge size of 4 degrees) and large (10 degrees). The size of the small target was tuned in a pilot study (with three participants) so that it was not too small to be frustrating.

Measurements

Our primary measurements were *task time*, *parallelism* and *perceived workload*. The task time is the duration from the first touch event on the screen to the last lift event of both fingers from the screen with the two document sections inside their target wireframes.

Parallelism indicates the degree to which both hands move in parallel. It can be computed using the Error-Reduction bimanual parallelism measure formulated in [1], which is defined as follows: First, for each hand's gesture path, the percentage of error reduction (%ER) per time step is defined as the magnitude of the movement towards the target divided by the movement required to reduce the error to zero. Then the amount of parallelism at each time step is the ratio of the two hand's %ER values, with the larger value in the denominator to normalize the result between 0 and 1. We then take the average, weighted by the time step lengths, to be the parallelism score for the task.

The perceived workload is measured with the standard NASA-TLX questionnaire. Users were asked to rate on a scale of 1 to 7, various types of workloads: mental effort, physical effort, temporal demand, performance, overall effort, frustration level.

Procedure

The study used a repeated measures design. The within-subjects factors were the gesture model types (SS, SP, AS), fold direction (Mountain, Valley), target symmetry (Symmetric, Asymmetric), and tolerance (Small, Large). The between-subjects factor was gender (Male, Female). Each participant used all the gesture models in one sitting, and the order was counterbalanced across the participants. For each gesture model, a participant performed one practiced block of trials (excluded from analysis), followed by 4 blocks. Each block of trials consisted of one instance for each of the 16 combinations of fold directions, target symmetries, and target tolerance. Users were instructed to perform the tasks as fast as they can. Within a block, the trials were presented in randomized order. Between blocks, participants were given a rest period of 10 seconds. The total time for a session was about 45 minutes. At the end of the session, users were given a questionnaire about the perceived workload, with space for writing down open-ended comments.

Results and Discussion

We recruited 12 unpaid participants in our laboratory, who were lab members and interns not involved in this project. All participants were familiar with touch screens, and one participant had not used a multi-touch screen before. All were right-handed, half were females.

The total number of (non-practice) trials for the study was 12 *participants* × 3 *conditions* × 4 *blocks* × 16 *trials* = 2,304. The data was pre-processed to remove outliers, defined as trials with a task time greater than 3 standard deviations away from the mean time of the trials with the same set of parameters. In total, 14 (0.6%) of the trials were removed for the following analysis. Unless otherwise noted, we used Greenhouse-Geisser correction to account for deviations from sphericity, and Bonferroni corrections for all post-hoc tests.

Task Time Performance

We first examined the learning effect by running a 2-way repeated measures ANOVA on the factors of gesture model and block, with task time as the dependent variable. The analysis indicates that block is not a main effect, and there is no *model* × *block* interaction. Therefore, the presentation order of the models in the experiment did not significantly affect the user performance; since there was no learning effect, we ignored the factor *block* for the following analyses.

We then ran a 4-way repeated measures ANOVA on the factors of gesture model, fold direction, target symmetry, and target tolerance, with task time as the dependent variable. We found gesture model (*model*) to be a main effect (Table 2). Post-hoc pairwise comparison showed that SS was significantly slower than SP ($p < 0.01$) and AS ($p < 0.05$), but there was no significance found between SP and AS (Figure 12-aggregated). This confirms hypothesis *H1*.

There was an interaction between gesture model and target tolerance (*model*×*tolerance*) (Table 2). Further examination reveals that with the small tolerance, model SS is significantly slower than SP and AS, with $p < 0.01$ for both; with the large tolerance, SS and AS are significantly slower

Factor	*F*-statistic	Significance	Partial η^2
model	$F_{2,22} = 9.99$	$p < 0.01$	0.48
direction	$F_{1,11} = 14.99$	$p < 0.01$	0.58
target symmetry	$F_{1,11} = 6.87$	$p < 0.05$	0.38
tolerance	$F_{1,11} = 202.44$	$p < 0.01$	0.95
Interacting Factors	***F*-statistic**	**Significance**	**Partial η^2**
model × *tolerance*	$F_{2,22} = 10.86$	$p < 0.01$	0.50
direction × *tolerance*	$F_{1,11} = 13.90$	$p < 0.01$	0.56
model × *direction* × *tolerance*	$F_{2,22} = 6.64$	$p < 0.01$	0.38

Table 2. Significance values of the factors.

than SP, with $p < 0.05$ for both (Figure 12). This suggests that AS allows users to achieve comparable performance to SP in fine-grained folding (i.e. small tolerance), but it is slower than SP in coarse-grained folding (i.e. large tolerance). On the other hand, SP is not compatible with the Pinch gesture. Therefore, the folding interface designer should take into account the targeted folding precision and possible integration with other gestures.

Fold direction (*direction*) was also observed to be a main effect (Table 2). Post-hoc pairwise comparison indicates that Mountain folding is significantly slower than Valley folding (Figure 13), and thus supports our hypothesis *H2*. We also observed a *direction* × *tolerance* interaction (Table 2). A further *t*-test shows that Mountain folding is significantly slower than Valley folding at the small tolerance ($t_{11} = 4.54$, $p < 0.01$), but there is no significant difference between the two directions at the large tolerance. This interaction may mainly be attributed to perspective projection: a target with a small tolerance appears smaller for the Mountain fold (than the Valley fold) as the target is farther away; the small tolerance is more sensitive to this difference.

Although we did not find a *model* × *direction* interaction, the analysis did reveal an interesting *model* × *direction* × *tolerance* 3-way interaction (Table 2): There exists a *model* × *direction* interaction with the small tolerance ($F_{2,22} = 4.03$, $p < 0.05$, partial $\eta^2 = 0.27$), while there is no such interaction with the large tolerance. This result further indicates that user performance is dependent on required folding accuracy.

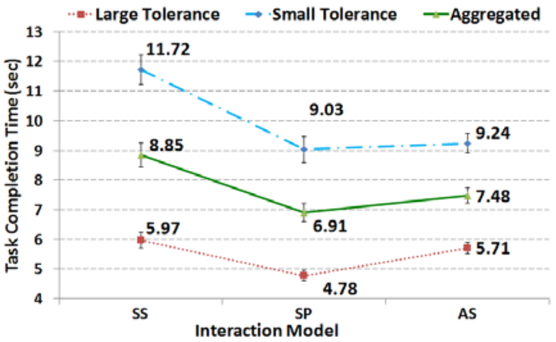


Figure 12. Mean Task Completion Time of each interaction model for large tolerance, small tolerance and aggregated over tolerances (bars show standard error).

The analysis revealed target symmetry to be another main effect (Table 2). The asymmetric targets took longer to perform than the symmetric targets (Figure 13). This result supports hypothesis *H3(a)* and suggests the lower cognitive load for symmetric hand movement. Application designers thus may prefer to use symmetric folding as much as possible.

Consistent with Fitts' law, target tolerance *(H4)* is a main effect (Table 2). The small tolerance took longer to perform than the large one (Figure 13).

There were no other 2-way, 3-way and 4-way interactions. We also analyzed task time with gender as a between-subject factor. Our data did not show gender to be a main effect ($F_{1,10} = 2.42$, $p = 0.15$).

Parallelism

The parallelism score only applies to the SP gesture model. Using only the data in this condition, we ran a *t*-test on the factor of target symmetry with parallelism as the dependent variable. The test shows that the asymmetric targets (M = 0.15, 95%CI = [0.12, 0.17]) yielded significantly less parallelism than the symmetric targets (M = 0.22, 95%CI = [0.18, 0.25]) with $t_{11} = -6.21$, $p < 0.01$. This supports hypothesis *H3(b)*.

This result indicates that the participants were able to move their fingers simultaneously toward the targets for a larger portion of the symmetric targets. Lower parallelism on asymmetric targets also indicates that the users were not able to coordinate the two fingers to move at *different* rates, proportional to the distances to the target so as to arrive at the targets at about the same time. This is quite difficult to do, especially for novice users.

Perceived Workload

To gather perceived workload data, users rated on a scale of 1 to 7 (1 is least effort) various types of workloads: *mental effort*, *physical effort*, *temporal demand*, *performance*, *overall effort*, *frustration level* (see Figure 14). A 1-way repeated measure ANOVA on the data found gesture model to be a main effect on mental effort ($F_{2,22} = 4.211$, p < 0.05). Post-hoc pairwise comparison with Bonferroni correction found that SP model (M = 4.92, 95%CI = [4.35, 5.48]) causes marginally more mental effort than AS (M = 3.75, 95%CI = [2.95, 4.55]) with $p = 0.068$, and it is similar to SS (M = 4.92, 95%CI = [4.26, 5.58]). There was no

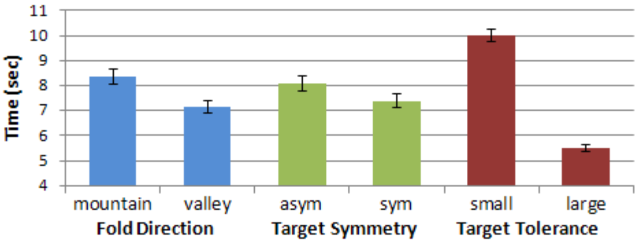


Figure 13. Mean Task Completion Time for three factors (bars show standard error).

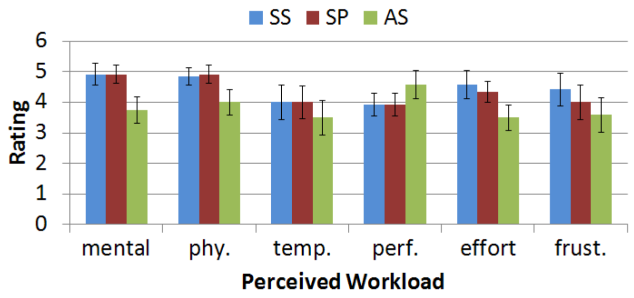


Figure 14. Means of perceived workload (bars show standard error).

significant difference between the other pairs.

Moreover, during the user study we did not observe any serious usability problems. As for the understandability, the participants readily understood the different models when they were explained and we did not observe any confusion between them. These qualitative results, combined with the previous quantitative analysis, suggest that the two practical models SP and AS are both viable, with some tradeoffs. The SP model is faster, but not significantly faster; while the AS model requires marginally less perceived mental effort. Comments from the participants reinforce this finding: "[SP] natural but inaccurate", "[SP] takes a lot of coordination and concentration", "[AS] more like operating physically".

One interesting finding from the SS model is a "wobbly finger" problem. Three participants commented that they had difficulty keeping the right finger still after they shifted their attention to moving the left finger. The problem was that with the small tolerance, when the user completed the left side, the right side occasionally drifted out of the target.

CONCLUSION & FUTURE WORK

We presented an exploration of document folding using multi-touch gestures. Our main contributions in this paper are identifying design considerations for simple folding gestures and providing a categorization of gesture models (Symmetric/Asymmetric, Serial/Parallel), developing two practical gesture models and a document workspace prototype, and conducting a user study which indicated that the factors (model, direction, target symmetry, tolerance) affect folding performance. Another finding was the higher parallelism for symmetric targets.

For future work, we plan to develop the aforementioned complex folds and to enable users to specify creases in document pages by gesturing and dynamically modifying the underlying mesh geometry. We also plan to explore the design of folding gestures with more than two contact points.

REFERENCES

1. Balakrishnan, R., Hinckley, K., Symmetric bimanual interaction. *Proc. CHI 2000*, pp. 33-40.
2. Beaudouin-Lafon, M. Novel interaction techniques for overlapping windows. *Proc. UIST 2001*, pp. 153-154.
3. Buxton, W. Myers, B. A study in two-handed input. *Proc. CHI 1986*, pp. 321-326.
4. Buxton, B. "Multi-touch systems that I have known and loved" (ver. Oct. 2009). http://www.billbuxton.com/multitouchOverview.html
5. Chang, S. H-H., Stuart, L. and Plimmer, B. Origami Simulator: A multi-touch experience. *CHI 2009 Extended Abstracts*, pp. 3889-3894.
6. Cutler, L.D., Fröhlich, B., Hanrahan, P. Two-handed direct manipulation on the Responsive Workbench. *Proc. I3D 1997*, pp. 107-114.
7. Denoue L., Nelson L., Churchill, E.F. A fast, interactive 3D paper-flier metaphor for digital bulletin boards. *Proc. UIST 2003*, pp. 169-172.
8. Dragicevic, P. Combining crossing-based and paper-based interaction paradigms for dragging and dropping between overlapping windows. *Proc. UIST 2004*, pp. 193-196.
9. Elmqvist, N., Henry, N., Riche, Y., Fekete, J.D. Melange: Space folding for multi-focus interaction. *Proc. CHI 2008*, pp. 173-176.
10. Guiard, Y., Asymmetric division of labor in human skilled bimanual action: the kinematic chain as a model. *J. of Motor Behavior*, 19 (4): 486-517, 1987.
11. Hinckley, K., Pausch, R., Proffitt, D., Patten, J., Kassell, N. Cooperative bimanual action. *Proc. CHI 1997*, pp. 27-34.
12. Hinckley, K., Baudisch, P., Ramos, G., Guimbretiere, F. Design and analysis of delimiters for selection-action pen gesture phrases in Scriboli. *Proc. CHI 2005*, pp. 451-460.
13. Kabbash, P., MacKenzie, I.S., Buxton, W. Human performance using computer input devices in the preferred and non-preferred hands. *Proc. CHI 1993*, pp. 474-481.
14. Kin, K., Agrawala, M., DeRose. T. Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. *Proc. Graphics Interface 2009*, pp. 119-124.
15. Lang, R.J. Computational origami: from flapping birds to space telescopes. *Proc. Symp. Computational Geometry 2009*, pp. 474-481.
16. Mackinlay, J.D., Robertson, G.G., Card, S.K. The Perspective Wall: Detail and context smoothly integrated. *Proc. CHI 1991*, pp. 173-176.
17. O'Gorman, L., Katsuri, R. *Document Image Analysis*. IEEE Computer Society Press, 1995.
18. Reisman, J.L., Davidson, P.L., Han, J.Y. A screen-space formulation for 2D and 3D direct manipulation. *Proc. UIST 2009*, pp. 69-78.
19. Szinger, J. The Foldinator modeler and document generator. *Proc. Origami 3 (2002)*, pp. 129-135.
20. Tashman, C. LiquidText: active reading through multitouch document manipulation. *CHI 2010 Doctoral Consortium*.
21. Wang, L., Sajeev, A.S.M. Roller interface for mobile device applications. *Proc. AUIC 2007*, pp. 7-13.
22. Westerman, W. *Hand Tracking, Finger Identification, and Chordic Manipulation on a Multi-Touch Surface*. PhD Dissertation (1999), U. of Delaware.
23. Wilson, A.D., Izadi, S., Hilliges, O., Garcia-Mendoza, A., Kirk, D. Bringing physics to the surface. *Proc. UIST 2008*, pp. 67-76.