

# Testing the Robustness and Performance of Spatially Consistent Interfaces

Joey Scarr<sup>†</sup>, Andy Cockburn<sup>†</sup>, Carl Gutwin<sup>‡</sup>, Sylvain Malacria<sup>\*</sup>

<sup>†</sup>Computer Science

University of Canterbury  
Christchurch, New Zealand

{joey, andy}@cosc.canterbury.ac.nz

<sup>‡</sup>Computer Science

University of Saskatchewan  
Saskatoon, Canada

gutwin@cs.usask.ca

<sup>\*</sup>Computer Science

University of Canterbury  
Christchurch, New Zealand

sylvain@malacria.fr

## ABSTRACT

Relative spatial consistency – that is, the stable arrangement of objects in a 2D presentation – provides several benefits for interactive interfaces. Spatial consistency allows users to develop memory of object locations, reducing the time needed for visual search, and because spatial memory is long lasting and has a large capacity these performance benefits are enduring and scalable. This suggests that spatial consistency could be used as a fundamental principle for the design of interfaces. However, there are many display situations where the standard presentation is altered in some way: e.g., a window is moved to a new location, scaled, or rotated on a mobile or tabletop display. It is not known whether the benefits of spatial organization are robust to these common kinds of view transformation. To assess these effects, we tested user performance with a spatial interface that had been transformed in several ways, including different degrees of translation, rotation, scaling, and perspective change. We found that performance was not strongly affected by the changes, except in the case of large rotations. To demonstrate the value of spatial consistency over existing mechanisms for dealing with view changes, we compared user performance with a spatially-stable presentation (using scaling) with that of a ‘reflowing’ presentation (widely used in current interfaces). This study showed that spatial stability with scaling dramatically outperforms reflowing. This research provides new evidence of spatial consistency’s value in interface design: it is robust to the view transformations that occur in typical environments, and it provides substantial performance advantages over traditional methods.

## Author Keywords

Expertise; spatial memory; revisitation.

## ACM Classification Keywords

H.5.2 [User Interfaces]: Interaction Styles.

## INTRODUCTION

Spatial memory is a valuable capability in the design of user interfaces. It allows people to locate items quickly [14,

25, 28], it has a large capacity [19], and it is long-lasting [10]. Several research systems have recognized the importance of spatial memory and have used it as a fundamental principle of interface organization (e.g., CommandMaps [25], ListMaps [14], or the Data Mountain [10, 24]), and related features are appearing in commercial applications such as the ‘Hotbox’ in Autodesk’s Maya.

Users can build up spatial memory of an interface when the display is spatially consistent – that is, when the locations of the objects in a view are stable over time. Spatial consistency is determined relative to a particular frame of reference (Figure 1), and in typical interfaces, the windows and displays of desktop and mobile systems provide the spatial frame of reference for the graphical objects they contain. With experience, users learn the arrangement of objects within the frame of reference, and can anticipate the location of targets, such as the Windows ‘Start’ icon in the bottom-left display corner, or the ‘Close’ icon at the top right of a window.

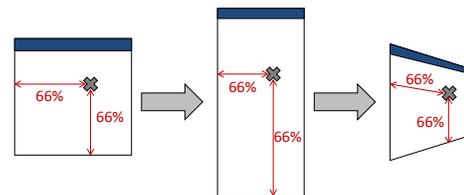


Figure 1: Spatial consistency keeps items proportionately stable with respect to the window bounds.

These examples show that spatial organizations are already a part of some interfaces (e.g., window managers) – but the organization of objects *within* windows, such as icons in file browsers, do not always use spatially consistent designs. For example, when the user changes the size of a window, or rotates a tablet computer from landscape to portrait mode, many systems *reflow* the items in the frame of reference – that is, they re-arrange items to fit the new aspect ratio of the window or display. This fills the window, but breaks spatial consistency – and items can be more difficult to find as a result. If spatial consistency was maintained in this situation, the primary arrangement of items (e.g., the portrait home screen) would be scaled to fit the altered frame of reference (e.g., the landscape window).

Although spatial consistency has been shown to provide enduring and fast retrieval of large data sets [10, 24], there

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2013, April 27–May 2, 2013, Paris, France.

Copyright © 2013 ACM 978-1-4503-1899-0/13/04...\$15.00.

are still issues that need to be understood to extend its utility as a design principle. In particular, designers need to understand its robustness to the transformations that commonly occur during interaction. These transformations can cause substantial changes to the appearance of objects from the user's perspective, such as scaling to accommodate an aspect-ratio change, translating a window across the display, or a rotated view on a tabletop display.

To understand the degree to which the fast performance offered by spatially consistent interfaces is robust to the types of transformations that commonly occur during interaction, we performed an experiment in which participants learned the locations of a series of targets within a window frame of reference. We then measured the effects on selection time of five transformations: *translation*, e.g., when a window is moved to another location on the screen; *rotation*, which is common in surface-based computing; *scaling* and *stretching*, such as when a window is resized; and *perspective distortion*, which occurs when viewing any kind of display from an oblique angle. Results show that many of these transformations had a low impact on performance: in particular, the effects of *scaling* and *stretching* were minimal, especially at low magnitudes.

To test how the spatial approach compares with traditional mechanisms for dealing with view changes, we carried out a second study with a realistic task and setting. Using a simulation of the Windows 7 control panel, we compared a spatially stable layout (where the presentation is scaled when the window size changes) to the 'reflow' layout currently used in Windows 7. Participants were initially trained with a stable presentation of the items. We then randomly resized the window in two ways, and asked participants to find and select target items. Results showed that the spatially consistent layout (using scaling) was dramatically faster than reflowing.

The results from these studies show that the performance enabled by spatially consistent displays is robust to the typical window/display transformations that occur in visual workspaces, and suggest that spatial consistency should be more often used as a fundamental UI organization principle.

## RELATED WORK

### Spatial Memory

Extensive prior literature in psychology and HCI has investigated human memory of object locations [1-2, 23], including mobile navigation through 3D environments and static memory for objects in 2D scenes (our interest). People learn item locations as a natural side-effect of interacting with them [11], and the rate at which locations are learnt follows a power law of practice [21]. There is also evidence that location memory is improved when effort is required to locate that object [6, 13].

Spatial memory is powerful: it allows fast decision-based retrieval, rather than comparatively slow visual search [17-

18]; it is enduring, with users able to quickly retrieve items months after creating spatial organizations [10]; and it has a large capacity, shown both by empirical studies (e.g., [19]) and by people's abilities in recalling hundreds of locations and routes needed to operate in everyday environments.

In HCI research, several interfaces have made use of spatial memory in order to explore or improve performance. For example, Robertson *et al.*'s Data Mountain allowed users to arrange thumbnails of web pages in a spatial environment [10, 24], and results showed that item retrieval was significantly faster than in a standard bookmarking system. More recently, the benefits of spatial memory have been exploited in window switching interfaces [29], as well as in list revisitation [14] and command selection [25].

### Mental spatial-transformation abilities

It is well known that people have a strong ability to recognize familiar forms, even when sizes or orientations are different [20], and various researchers have examined people's ability to deal with specific visual transformations in 2D and 3D. The most common instance of this is mental rotation, which is frequently used to measure differences in spatial ability (e.g., [9, 27]). Scaling was investigated by Bundesen and Larsen [5], who showed that the recognition time for 2D shapes at differing scales was a linear function of size ratio. Bryant and Tversky [3] investigated different methods of conveying 3D information, and showed that simple depth cues such as size and converging lines led people to easily interpret 3D scenes; however, there is also evidence to suggest that viewing familiar objects from unfamiliar viewpoints reduces recognition efficiency [4]. To our knowledge, no research exists on the transformation problem for UIs: that is, people's performance in locating familiar items in a transformed frame of reference.

### SPATIAL CONSISTENCY AS A DESIGN GUIDELINE

'Be consistent' is a fundamental rule of HCI, featuring in many design guidelines (e.g. [12, 22, 26]). However, this principle is abstract, and it does not prescribe which design elements should be held consistent. Hansen's 1971 interface guidelines [15] include a recommendation to support 'display inertia', meaning that "the size and layout of the display do not change drastically" (p.529). Hansen's objective in this guideline was to optimize user execution of operations by allowing users to make rapid decisions (modeled by Hick-Hyman choice reaction time [17-18]).

However, Hansen's argument that the *size* of the display should not change drastically is inconsistent with current interface designs, where users have freedom to resize and reorient windows. In this paper we investigate methods to achieve display inertia that are robust to commonly occurring size and layout manipulations.

We propose the design principle '*maintain relative spatial consistency within the frame of reference*' as a foundation of interface organization. The *frame of reference* will normally be provided by the display edge or by the borders

on a particular window, but it can also be perceived by Gestalt proximity [30]: for example, a grid of items with no visible border can still be seen to have a frame of reference. By ‘*relative spatial consistency*’, we mean that the arrangement of items within the frame of reference should remain proportionately stable with respect to the bounds of the frame. For example, if an item is the closest item to the top right corner of a frame before transformation, it should be similarly positioned after transformation as well. Figure 1 illustrates *relative spatial consistency* as the frame undergoes stretch and perspective transformations.

**Common transformations to the frame of reference**

Frames of reference in UIs commonly undergo (or are viewed in such a way that they are perceived to undergo) five forms of visual transformation (Figure 2). The thick ‘top’ edges of the frames in Figure 2 represent the standard orientation of the frame (i.e., which way is ‘up’).

1. *Translation* occurs frequently in desktop computing, when windows are moved to different screen locations.
2. *Scaling* also occurs frequently in desktop computing, when windows are resized by the user.
3. *Stretching (changing aspect ratio)*. Similar to scaling, stretching occurs when windows are resized in one dimension. This also occurs on mobile devices when an interface is reoriented to landscape or portrait mode.
4. *Rotation* is common in surface-based computing (e.g. digital tables or shared use of tablets), where displays or windows can be turned to face another person. It also occurs on mobile devices when an interface has not been programmed to adapt to device rotation (e.g., the Apple iPhone home screen, when viewed in landscape mode).
5. *Perspective distortion* occurs when viewing any kind of display from an oblique angle, as is common on shared wall or tabletop displays.

**Relative spatial consistency after transformations**

The previous section discussed transformations to the frame of reference itself. Relative spatial consistency, however, concerns the location of content *inside* the frame of reference after transformation.

When the frame of reference changes, UI designers can choose how the interface adapts to the new bounds. Translation, rotation and perspective transformations normally do not require any particular adaptation or response from the user interface – the window moves (with translation) or the user changes their viewing orientation (with rotation and perspective). However, an interface response is necessary when the user scales or stretches the frame of reference. ‘Reflowing’ the content is a common design strategy (e.g., the grid view in the Windows file explorer), as is item elision (e.g., the Office 2007 Ribbon moves items into hierarchies as the window gets smaller). As demonstrated by our Study 2, designers could also choose to maintain the original arrangement of items within the frame, and scale the entire grid when a stretch occurs.

The grid lines in each window in Figure 2 depict how ‘canonical’ relative spatial consistency can be maintained during the different transformations. Other approaches (e.g., 2D scaling in response to 1D stretching) can be achieved by combining these primitives. The final row of the figure shows transformation matrices for each effect.

**EXPERIMENT ONE: INTERFACE TRANSFORMATIONS**

It seems reasonable that the fast interaction enabled by spatial consistency will be robust to at least some of the transformations described above and shown in Figure 2. For example, users are unlikely to have difficulty locating items in a window after translating it. However, the time taken to adapt and respond to these transformations is less clear –

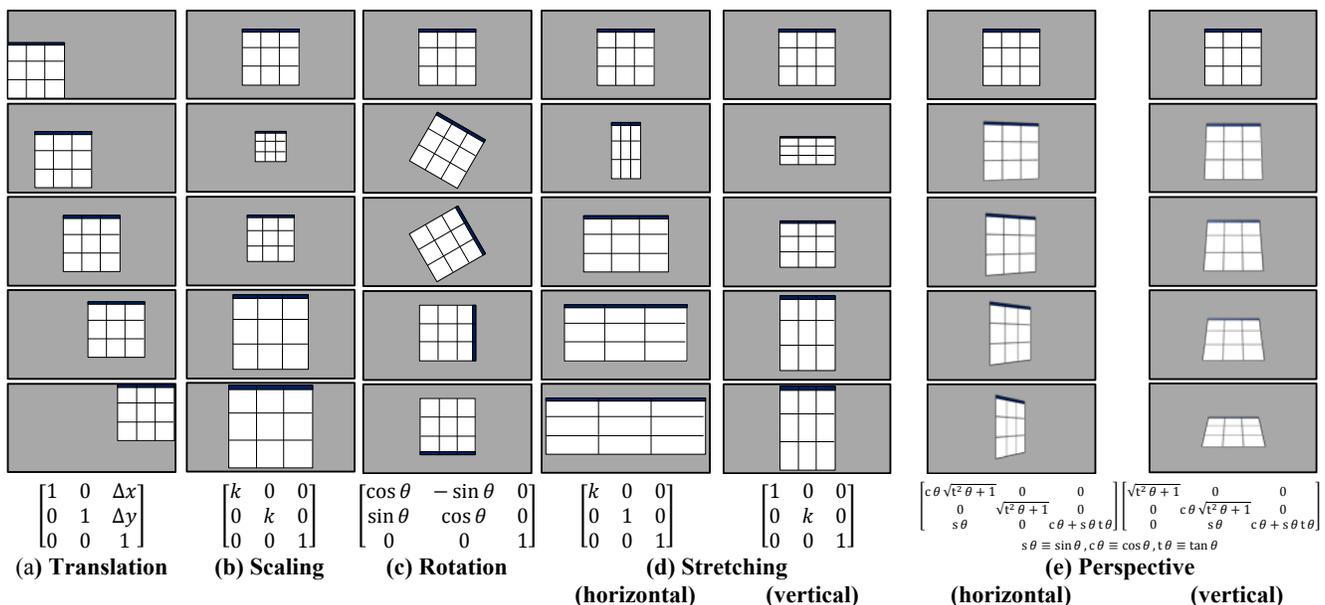


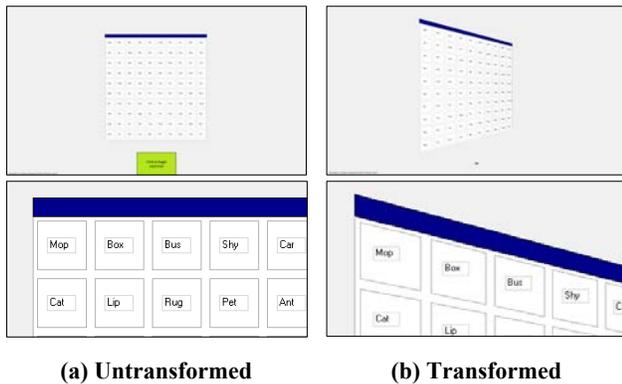
Figure 2. The transformations used in Experiment One. The top row shows the untransformed window used for training.

while it is likely that users can reorient their spatial understanding, there may be time costs in doing so.

We therefore performed an exploratory study to determine the additional time needed to acquire items following different types and magnitudes of spatially consistent transformation. The method involved repeatedly selecting the same four items in a spatially consistent layout, while the grid underwent the transformations shown in Figure 2. The time taken to select a target after each transformation involves perceptual and cognitive processes of reorienting to the display and deciding about item location (which we call *reorientation time*), as well as the mechanical time to point to the target. Our interest lies in reorientation time, so to extract the variable effects of target relocation caused by the transformations, we analyzed each participant’s Fitts’ Law pointing characteristics, and used them to subtract pointing time from the total selection time for each item. (Note that study 2 considers overall performance including pointing time; our interest here is the reorientation phase.)

### Tasks, stimuli, and instructions

Tasks involved a sequence of selections from a 10×10 grid of textual items (Figure 3a) that was either transformed or untransformed. All text items were common English three-letter words, which were used (rather than images or variable word sizes) to reduce confounds from visual pop-out. Also, to avoid effects of reading distorted text, text labels were not transformed with the interface (Figure 3b). An exception was made for rotation because pilot testing showed that participants use text orientation as a primary cue to establishing the frame of reference.



**Figure 3. Overview and close-up of the interface used in Study 1, before and after a horizontal perspective transformation.**

Tasks began by showing an untransformed window (Figure 3a). Participants clicked a button to reveal the target item and display the transformed window (Figure 3b). Task timing began with the button click, and stopped when the target was selected; this was achieved by clicking in a visible hitbox surrounding the text label, which was a constant size in all conditions. After selection, the display returned to the untransformed window. Subjects were asked to make selections “as quickly and accurately as possible”.

### Transformations and magnitudes

The study tested seven transformations: translation, scaling, rotation, horizontal stretching, vertical stretching, horizontal perspective, and vertical perspective (columns in Figure 2). Each transformation was tested at five levels of magnitude (rows in Figure 2). A summary is shown in Table 1.

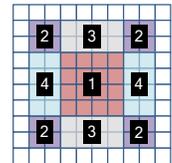
|                |  | Magnitude   |         |         |                 |
|----------------|--|-------------|---------|---------|-----------------|
|                |  | 1           | 2       | 3       | 4               |
| Transformation | Translation ( $\Delta x, \Delta y$ ), px | 290,100     | 580,200 | 870,300 | <b>1160,400</b> |
|                | Scaling                                  | <b>×0.5</b> | ×0.75   | ×1.25   | ×1.5            |
|                | Rotation                                 | 30°         | 60°     | 90°     | <b>180°</b>     |
|                | Stretching-x                             | <b>×0.5</b> | ×1.5    | ×2      | ×2.5            |
|                | Stretching-y                             | <b>×0.5</b> | ×0.75   | ×1.25   | ×1.5            |
|                | Perspective-x                            | 15°         | 30°     | 45°     | <b>60°</b>      |
|                | Perspective-y                            | 15°         | 30°     | 45°     | <b>60°</b>      |

**Table 1. Experiment One transformations and magnitudes. Bold items denote the level deemed most extreme.**

### Procedure

Each participant initially performed a bi-directional Fitts’ calibration task, consisting of 144 selections across 7 indices of difficulty. They then completed four blocks of trials with each of the seven transformations (Table 1). Order of transformation was counterbalanced using a Latin square. All four blocks were completed with one transformation before advancing to the next. The blocks comprised: *familiarization*, *training*, *recall*, and *learning*, always in that order. The *familiarization* block (data discarded) acquainted participants with the transformation, and consisted of ten trials (two for each magnitude), using different target items to the main experiment.

The *training* block consisted of 20 trials in the untransformed interface. The *training*, *recall*, and *learning* blocks used the same four target items throughout the experiment. To reduce potential confounds stemming from specific item locations, each participant had a unique set of target locations, with each item randomly selected from one of the four regions shown alongside. No adjacent locations were allowed.



The *recall* block was used to examine selection times immediately after transformation. It consisted of 20 selections: one each for the four target items at each of the five magnitudes of transformation, in random order. The un-transformed interface was displayed between trials, and became transformed once the participant initiated the trial.

Finally, the *learning* block was included to examine participants’ ability to re-learn item locations after the interface had been transformed. We used the most extreme form of each transformation (see Table 1), and participants selected each target five times (random order), without the untransformed window being presented between selections (i.e., the extreme view was continually shown).

In summary, each participant performed 1960 trials:

7 transformations × 4 blocks

familiarization: 10 selections (data discarded)

training: 20 selections }  
 recall: 20 selections } same 4 targets throughout expt.  
 learning: 20 selections }

### Participants and Apparatus

There were 14 participants; 7 male, 7 female, aged 19-42 (mean 26.9). The experiment was performed on a Windows 7 PC with a 1920×1200 monitor. Participants performed Experiments 1 then 2 in a single one-hour session.

### Design

The study compares the time needed to reorient to a transformed display to the time for the non-transformed view. *Reorientation time* ( $T_r$ ) is calculated by subtracting pointing time ( $T_p$ ) from total selection time ( $T$ ). Pointing time is calculated using each participant's individually calibrated Fitts' Law function, so  $T_r = T - T_p$ . For each transformation type, two pairwise measures are used to characterize the size of the effect of each transformation magnitude in comparison to the non-transformed condition: the statistical *effect size* using Cohen's  $d$ , which provides a sample-size independent estimate of effect size (Cohen [8] states that .2 is a small effect, .5 is medium, and .8 large); and the percentage increase in reorientation time. *Reorientation time* is analyzed using a 7×5 repeated measures ANOVA with within-subjects factors *transformation* {translation, scaling, rotation, stretchingX, stretchingY, perspectiveX, perspectiveY} and *magnitude level* {0, 1, 2, 3, 4}.

### Results

#### Fitts' calibration

Linear regression showed strong Fitts' models for 13 participants ( $R^2 \geq 0.95$ ), and one slightly weaker at  $R^2 = 0.89$ . The mean pointing time predicted by the models varied little between the transformation types and their magnitudes: the overall mean was 948ms (s.d. 25), ranging from 926ms in the 180° rotation condition to 1063ms in the maximum translation condition.

#### Time to find items during training

The experiment focuses on the additional time required for users to reorient to *known* spatially consistent displays when they undergo various forms of visual transformation. One relevant data point that helps understand the scale of reorientation cost (to determine whether the time increases are large or small) is the time taken to find the items when the user has no spatial knowledge regarding item placement. We therefore analyzed the mean time to select items for the first time in the *training* block, which occurred after familiarization but before any spatial learning. The mean selection time was 15470ms, of which 14493ms can be attributed to visual search (once predicted pointing time is subtracted). By the fifth repetition during training, the mean decision time (selection time minus pointing time) had reduced to 811ms, which can be attributed to spatial memory supporting much faster selections.

#### Reorientation time after transformation in recall blocks

The primary results concerning reorientation times are presented here. Mean reorientation times across all levels for each transformation, as well as the mean calculated Fitts' Law pointing times, are shown in Figure 4a – the dashed horizontal line shows the mean reorientation time for the non-transformed condition. The lower segment of each bar shows reorientation time, and the upper segment shows calculated pointing time. The two numbers in each bar show Cohen's  $d$  effect size compared to the baseline and the percentage increase from the baseline. Figure 4a suggests that most of the transformations (other than rotation) had a relatively small impact on reorientation time – within 388ms of the baseline, which is only 2.7% of the visual search time reported above. As expected, ANOVA (error trials removed) showed significant main effects of *transformation* ( $F_{6,78} = 19.1, p < .001$ ) and *magnitude* ( $F_{4,52} = 8.5, p < .001$ ), and an interaction ( $F_{24,312} = 4.8, p < .001$ ).

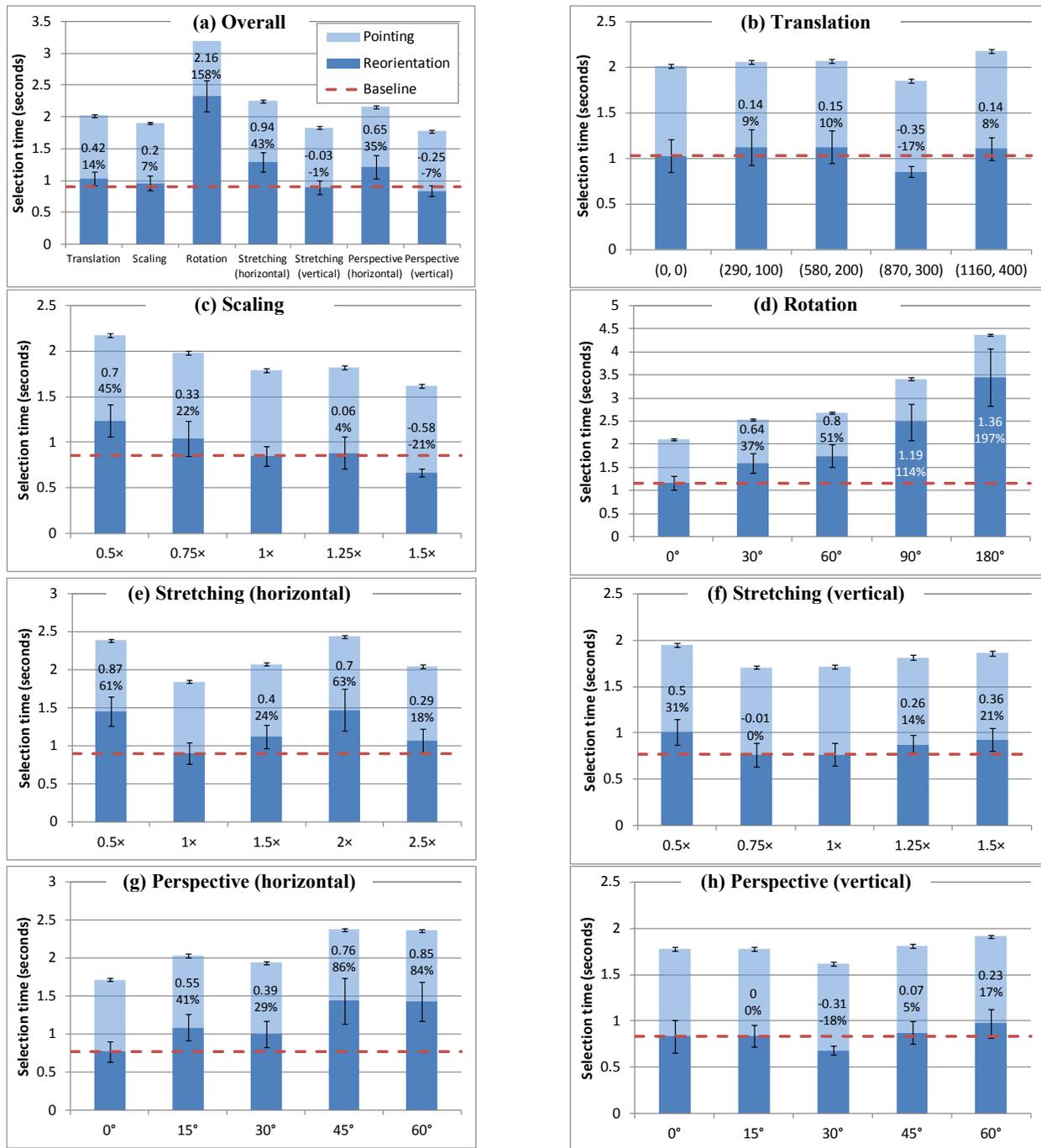
Our analysis shows that adapting to transformed displays caused a reliable increase in reorientation time, but that this increase is small compared to the visual search time needed when the item's location is unknown.

Figure 4a shows reorientation time averaged across all transformation magnitudes (except the no-transformation level). To gain further insight into the effects of each transformation magnitude, we separately compared each transformation magnitude with the no-transformation magnitude. The results are summarized in Figure 4b-h for each transformation, which include Cohen's  $d$  and percentage differences. Note that the baseline data is extracted from the no-transformation level within each transformation type. The key findings are as follows.

*Translation* caused small absolute increases in reorientation time (< 100ms), regardless of magnitude (Figure 4b). This finding is unsurprising given users' extensive experience in adapting to windows placed in different display regions. The reduced time at the (870, 300) translation level is attributed to participants having already moved their mouse closer to this translation (which was near the centre of the screen) causing a reduction in actual pointing time, and hence an under-estimation in calculated reorientation time.

*Scaling* (Figure 4c) had little effect on reorientation time at 0.75×, 1.25×, and 1.75× levels (increases of 190ms, 35ms, and -180ms). There was a larger effect at the extreme 0.5× level (387ms). This may have been influenced by our scaling method, which kept text size constant (to maintain legibility and pointing time) regardless of scale level. Text labels were thus very close to one another at small scales.

*Stretching* (Figure 4e,f), like scaling, showed relatively small absolute time increases for most levels (<250ms for all but 0.5× and 2× x-stretching, which exceeded 550ms). The higher time for 0.5× x-stretching can be explained in the same way as scaling above; the 2× result is reasonable



**Figure 4. Reorientation time and Fitts' Law pointing times for each transformation type. The baseline value (red dashed line) is the mean reorientation time of participants in the un-transformed condition.**

(and the lower time of the higher 2.5x level may be due to the proximity of the screen edge assisting reorientation).

*Rotation* had much larger effects on reorientation time (Figure 4d), with absolute mean time increases from 434ms (30°) to 2284ms (180°). Some of this time will be incurred by reading rotated text, but we suspect that most of it can be attributed to mental processes of reorienting to the rotated frame of reference. This is supported by prior work from Cooper [9], which showed that the time taken to interpret

rotated pattern stimuli increases linearly with rotation angle. Linear regression of our reorientation time data with degree of rotation supports Cooper's finding ( $R^2 = 0.98$ ).

Horizontal *perspective* changes (Figure 4g), like rotation, resulted in an approximately linear increase in reorientation time across angle ( $R^2 = 0.84$ ). However, the absolute value of the increase (compared to the baseline) was much smaller than rotation (ranging from 226 to 665ms). Vertical perspective changes (Figure 4h) had a much smaller effect

on reorientation time, ranging from 2ms at 15° to 139ms at 60° (and a negative effect of 152ms at 30°). One possible explanation is that this type of perspective is common in everyday life (e.g., reading on a flat table); regardless, subjects were quickly able to reorient to the transformation.

**Learning**

During the learning block, participants selected the target items five times each in a random order from a maximally transformed window. Reorientation time data (selection time minus pointing time) is analyzed using a 7x5 ANOVA for factors *transformation* and selection *repetition*. There was a significant effect of *transformation* ( $F_{6,78} = 5.1, p < .001$ ), largely due to the slow performance of rotation (1593ms) with all other transformations within 921±164ms. There was also a significant effect of *repetition* ( $F_{4,52} = 5.9, p < .005$ ), with mean reorientation times quickly improving from 1318ms in the first selection to a minimum of 879ms in the third (within 24ms of the time with untransformed windows in the training block). Participants’ performance with stable transformed windows quickly matched that of untransformed views. There was no interaction between *transformation* and *repetition* ( $F_{24,312} = 1.1, p = .35$ ), giving no evidence that any transformation type is harder to learn.

**Discussion**

To summarize, we analyzed how quickly users can reorient their expectation for the location of known targets when spatially consistent displays undergo likely transformations (translation, rotation, scaling, stretching, and perspective). Results showed that users can quickly adapt to all forms of transformation (much more quickly than the time needed to find unknown items in the display). Adapting to rotations was much slower than the other transformations (at 180°, 20x that of translation). We also replicated results showing that rotation reaction times are a linear function of angle.

These results provide a new human-factors characterization of performance with common display transformations; in addition, the study provides design insights that we deploy in the next study. In particular, the fast reorientation times in response to scaling and stretching suggest that users will be much faster when a spatially consistent approach is used to deal with transformation, than with approaches that rearrange items to fill the transformed window.

**EXPERIMENT TWO: SCALING VS. REFLOW**

Experiment 1 demonstrated that people are able to quickly select familiar items after a spatially consistent display is

transformed. Our second experiment tests the application of this finding in a realistic interface.

Many commercial interfaces, such as toolbars and file browsers, use a ‘reflow’ algorithm to rearrange items when the window dimensions change (Figure 5c). However, when items are rearranged in this manner, people lose their spatial knowledge of the interface, potentially slowing retrieval. We therefore compared a reflow-based layout strategy to two different spatially consistent designs.

**Interface Layout Designs**

We considered three designs for adapting the layout of a simple icon view (e.g., a file browser) to window size.

*Scaling.* This layout scales a grid of icons to fit the window bounds. Note that when the window’s aspect ratio is changed, spatial consistency is maintained relative to the perceived bounds of the item grid, rather than the window edges (Figure 5a).

*Scrolling.* This layout maintains spatial consistency to the original frame of reference, using scrolling to allow viewport translation over the icon grid (Figure 5b). The location of items is predictable as an absolute displacement from the information space’s origin, but the interface does not maintain relative spatial consistency with respect to the new frame of reference. Scrolling requires more interface manipulations to select targets than the other conditions.

*Reflow.* This is the standard layout strategy employed in contemporary file browsers: when the window changes size, icons are rearranged to fill the window, in reading order (Figure 5c). Reflowing makes efficient use of display space, but requires scrolling when icons do not fit the view.

**Procedure**

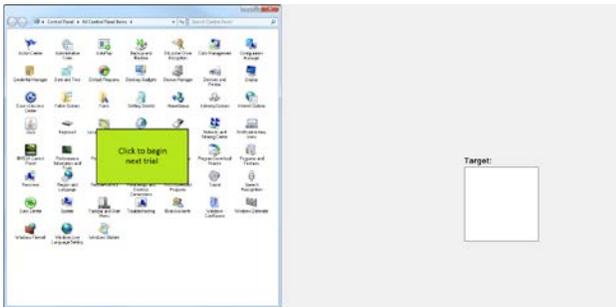
The experimental task consisted of a sequence of selections from a file-browser-like interface, populated with items from the Windows 7 control panel (Figure 6). Participants clicked a button to begin each trial, triggering the display of a stimulus in a sidebar. Selecting the target item completed the task and redisplayed the “Click to begin” button.

Two blocks (*training* and *recall*) were completed with each of the three layouts (*scaling*, *scrolling*, and *reflow*). The *training* block consisted of six repetitions of each of six target items, using a *square* window size with a content area of 700x700 pixels. In the *recall* block, the window bounds were varied on every trial to be either *square*, *wide*



Figure 5. The three alternative icon layout strategies in a wide window configuration.

(917×401), or *tall* (401×917). The *tall* configuration extended to the vertical height of the monitor, and was just wide enough for all of the items to fit into the *reflow* window without scrolling. The *wide* configuration was the transposition of *tall*. When window configuration changed, items were arranged according to the layout strategy (*scaling*, *scrolling* or *reflow*). With *square*, the three layout strategies were equivalent. Figure 5 shows the effect of each layout strategy on a *wide* configuration. For *scaling*, *wide* scaled the icon grid by 0.7×, and *tall* by 0.57×.



**Figure 6.** The system used in Experiment Two. Targets were displayed on the right, and participants selected the target items from the interface on the left.

Target items and window configuration sequences were different for each participant in each condition, and targets were selected such that no two target items were in the same row or column. The row and column constraint was used (without subject knowledge) to give an approximately uniform spatial distribution of items in the *scrolling* condition (to control the number of items that required scrolling). Each participant therefore performed 162 trials:

3 layout strategies × 2 blocks  
 training: 36 selections (data discarded)  
 recall: 18 selections

Participants completed NASA-TLX [16] worksheets and responded to visual appeal questions after each layout. They ranked the layouts for preference, speed and error rate at the end of the experiment.

### Participants and Apparatus

15 participants were recruited for the study, with 14 completing it directly after Experiment One. Experiments One and Two used the same hardware and setup.

### Design and Hypotheses

The experiment was designed as a 3×3 RM-ANOVA for factors *layout* {*scaling*, *scrolling*, *reflow*} and *configuration* {*square*, *wide*, *tall*}, with selection time as the dependent variable. *Layout* was counterbalanced using a Latin square.

Our primary hypotheses were as follows:

**H1:** *Scaling* will be faster than *scrolling* and *reflow*. *Scaling* keeps items spatially consistent, unlike *reflow*, and requires no extra user action, unlike *scrolling*.

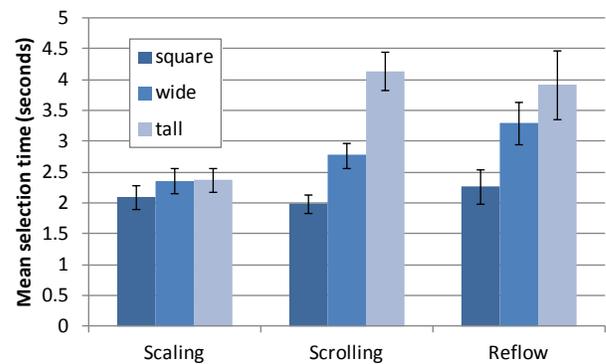
**H2:** *Scaling* will be subjectively preferred by participants.

## Results

Error rates were low in all conditions: 1.5% for *scaling*, 2.6% for *scrolling*, and 1.9% for *reflow*. Trials including incorrect selections were excluded from the analysis; this did not affect the significance of our results. For significant ANOVA effects, we include partial eta-squared ( $\eta^2$ ) as a measure of effect size (where .01 is a small effect size, .06 medium, and .14 large [7]).

### Selection Times

Mean selection times were fastest with *scaling* (2.27s, s.d. 0.76), followed by *scrolling* (2.96s, s.d. 1.26) and *reflow* (3.158s, s.d. 1.69), giving a significant main effect of *layout*:  $F_{2,28} = 7.3$ ,  $p = .003$ ,  $\eta^2 = .34$ . With the *scaling* layout, mean selection times following *wide* and *tall* view transformations increased by 262ms and by 277ms over the time taken with the *square* view used for training. These small increases contrast with the substantial increases of 1039ms and 1653ms with the *reflow* layout. Posthoc Bonferroni-adjusted pairwise comparisons ( $\alpha = 0.05$ ) showed that *scaling* was significantly faster than both *scrolling* and *reflow*, but there was no difference between *scrolling* and *reflow*. We therefore accept **H1**.



**Figure 7.** Mean selection times in Experiment 2.

As expected, there was a significant effect of *configuration* ( $F_{2,28} = 24.8$ ,  $p < .001$ ,  $\eta^2 = .64$ ) with *square* (2.11s, s.d. 0.83) faster than *wide* (2.81s, s.d. 1.05) and *tall* (3.47s, s.d. 1.65). More importantly, there was a *layout* × *configuration* interaction ( $F_{4,56} = 5.98$ ,  $p < .001$ ,  $\eta^2 = .30$ ), as shown in Figure 7: *scaling* performed similarly to *scrolling* and *reflow* in the *square* configuration, but was faster in the *wide* and *tall* configurations.

26% of *scrolling* trials in the *wide* configuration required the user to scroll the viewport, with 18% for *tall* and 0% for *square*. Figure 7 shows that scrolling was slowest in the *tall* condition, which required horizontal scrolling.

### Subjective Responses

TLX questionnaire responses showed no significant differences, perhaps due to low statistical power; further study in this area is needed. We therefore fail to find support for **H2**.

## GENERAL DISCUSSION

Experiment 1 showed that users can quickly adapt to many forms of view transformation, allowing rapid selections when items remain spatially stable within the frame of reference. Experiment 2 used this finding to compare user performance between the contemporary ‘reflow’ strategy (which reduces spatial stability) and a scaling layout that maintains spatial stability. Results confirmed that selection times following view transformations were much faster with the scaling layout than with the reflowing layout.

The following subsections discuss the implications of these findings, as well as the limits of their applicability.

### Generalizing the results of Experiment Two

Experiment 1 examined human performance factors in response to view transformations, requiring an experimental method that was substantially abstracted away from day to day interaction contexts. Experiment 2, in contrast, focused on a specific interaction context (contemporary icon layouts in file browsers), but in doing so necessarily introduced potential confounds, including icon visual saliency, dataset ordering and size, and specific transformation settings.

Experiment 2 used the actual icons from the MS Windows control panel. We did this to assist external validity, while aware of the differing visual salience across icons – for example, the colorful ‘Default Programs’ icon is more likely to pop out than the small, grey ‘keyboard’ icon. The experimental design mitigated these effects by randomizing the target set for each participant and layout.

The method also used an alphabetical ordering of icons in each view, again to maintain consistency with the current Microsoft layout. This arrangement is very likely to have assisted users in identifying target locations after reflowing. However, logical or predictable icon arrangements are challenging to determine in many contexts, and it is likely that the benefits of scaling would be more pronounced if predictable ordering was unavailable.

The size of the dataset was also determined by the typical size of the control panel window. The sizes of the tall and wide windows were selected to maximally utilize space in the reflowing condition without scrolling – i.e., the sizes were biased to aid reflowing. If the windows had been any smaller, the reflowing condition would have required scrolling. There are, however, interesting questions for the scaling condition around the relationship between performance and scale factor. Experiment 1 suggested that performance deteriorates as views are transformed further from 1× views, and there are likely trade-offs between the costs of reducing scale and the costs of increased scrolling. We intend to conduct further work in this area.

Finally, subjective responses in Experiment 2 showed no significant differences between conditions. Participants were neither strongly in favor nor strongly opposed to the scaling view, but we do not know how their opinions would change if, for example, more extreme scale factors were

used, or if the reflowing condition had required scrolling. Again, further work is needed, but it currently appears that scaling allows much faster performance following transformations than does reflowing, and that this benefit comes without the costs of negative subjective reaction.

### Applications of spatial consistency

The primary design implication of our results is that spatial consistency should be a *fundamental* consideration in the design of interfaces and information displays. In many cases, designing to maintain spatial consistency is a relatively simple matter – for example, on mobile devices that allow landscape and portrait view modes (switched by accelerometer input) interface design should favor relative spatial consistency of items, rather than seeking ways to rearrange interface components to exploit the variable display space in the different layouts.

Designing for spatial consistency also allows new styles of interaction, such as that demonstrated by the Data Mountain [10] or CommandMaps [25]. Another interesting possibility lies in creating new interface toolkits and APIs that are more robust to variable display requirements. Built-in scaling functions to accommodate different window sizes and/or display resolutions would greatly facilitate the implementation of interfaces that are spatially robust, rather than resorting to the current methods of reflowing, rearranging, and elision.

Finally, there are several potential questions that designers may have about a fundamentally spatial approach to interface design. We address some of these issues here, but as the discussion above suggests, spatial consistency is not an all-or-nothing principle: the idea can easily be used to improve existing interfaces in small ways, as well as to create new ways of accessing information.

*What happens with changing window content?* Experiment Two studied an icon set that changes slowly if at all (i.e., control panel icons). In windows where content changes more quickly (e.g., additions or deletions of icons), how well does the spatial approach work? There are four reasons why a spatially consistent presentation can work well, even with changing content. First, in many windows, items change slowly, and a person’s spatial memory can often keep up with the changes [28]. Second, users could be in charge of placing new items in a display (as with a phone’s home screen, or in systems like the Data Mountain [24]); in these cases, the act of placing the items can help to overcome the difficulties caused by changing content. Third, ordering by addition (i.e., new icons are added at the bottom of the display) would lead to stable arrangements that allow the development of spatial memory. Fourth, a spatial organization could be used as one of several views presented by an interface: in situations where content changes slowly, the user would gain the benefits of developing spatial memory; in situations where items change frequently, the user could switch to an alphabetic arrangement (or a list view).

*What happens with large data sets and small window sizes?* When there are many items in an interface, scaling the entire icon set to fit the window bounds may be impractical. Furthermore, pointing can be difficult at very small scales. In these scenarios we recommend a hybrid scaling/scrolling strategy, where the grid is scaled according to the width of the window and a vertical scrollbar allows users to access off-screen items. When the window width becomes too small to feasibly scale items, scaling ends and a horizontal scrollbar can be added. We note that as sets grow, the problems of the 'reflow' strategy also increase (i.e., items near the end of the list will be even further displaced from their original locations). Further work in this area is needed.

*What happens to search and list views?* Grid layouts are not the only choice for users – they can switch to other views when appropriate (e.g., in a file browser, a list view allows users to sort by date), and can also find items through a search interface. These alternate presentations, however, do not imply that a spatially consistent view is not practical – it would be simple to include a spatially consistent view as one of several presentations. In addition, it is also possible to use visualization techniques to add the functionality of these other views to a spatially consistent presentation. For example, highlighting could be used to show search results (as seen in the Mac OS X control panel), or to show recently-used files. Augmented views such as these could provide the benefits of spatial consistency and still give users the power of specific retrieval tools.

## CONCLUSION

Spatial consistency is a powerful organizing principle for interfaces, but everyday use involves many forms of view transformation. We conducted a study to improve understanding of how performance with spatially consistent views is influenced by different forms and magnitudes of display transformation. Results showed that users can quickly reorient their spatial understanding to all of the tested transformations, but that adaptation to rotation is much slower than the others. We then tested these findings in a real-world usage scenario, hypothesizing that performance with a file browser could be improved by replacing the reflow layout approach with a layout that scaled the view. Results showed substantially improved performance. The primary design implication of this work is that spatial consistency should be used as a fundamental design principle for interfaces and information displays.

## REFERENCES

1. Andrade, J. and Meudell, P. Short report: Is spatial information encoded automatically? *Quarterly J of Exp. Psych.* 46A (1993), 365-375.
2. Baddeley, A.D. *Human Memory*. Lawrence Erlbaum Associates, Hove, 1990.
3. Bryant, D.J. and Tversky, B. Mental representations of perspective and spatial relations from diagrams and models. *J. Exp. Psych.: Learning, Memory, Cognition* 25, 1 (1999), 137.
4. Bühlhoff, I. and Newell, F.N. The role of familiarity in the recognition of static and dynamic objects. *Progress in Brain Research*, Elsevier, 2006, 315-325.
5. Bundesen, C., Larsen, A. Visual transformation of size. *J. Exp. Psych.: Perception and Performance* 1, 3 (1975), 214-220.
6. Cockburn, A., Kristensson, P., Alexander, J. and Zhai, S. Hard Lessons: Effort-Inducing Interfaces Benefit Spatial Learning. in *Proc. CHI'07*, (2007), 1571-1580.
7. Cohen, J. Eta-squared and partial eta-squared in communication science. *Hum. Comm. Research* 28, 473-490 (1973), 56.
8. Cohen, J. *Statistical power analysis for the behavioral sciences*. Lawrence Erlbaum, 1988.
9. Cooper, L.A. Mental rotation of random two-dimensional shapes. *Cognitive psychology* 7, 1 (1975), 20-43.
10. Czerwinski, M., van Dantzich, M., Robertson, G. and Hoffman, H. The Contribution of Thumbnail Image, Mouse-Over Text and Spatial Location Memory to Web Page Retrieval in 3D. in *Proc. INTERACT'99*, (1999), 163-170.
11. Darken, R. and Sibert, J. Wayfinding strategies and behaviors in large virtual worlds *Proc. CHI'96*, 1996, 142-149.
12. Dix, A., Finlay, J., Abowd, G. and Beale, R. *Human-Computer Interaction* (1993).
13. Ehret, B. Learning Where to Look: Location Learning in Graphical User Interfaces. in *Proc. CHI'02*, (2002), 211-218.
14. Gutwin, C. and Cockburn, A. Improving List Revistation with ListMaps. in *Proc. AVI'06*, ACM, (2006), 396-403.
15. Hansen, W.J. User engineering principles for interactive systems *Proc. AFIPS'71 (Fall)*, 1971, 523-532.
16. Hart, S. and Staveland, L. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. in *Human Mental Workload*, 1988, 139-183.
17. Hick, W.E. On the rate of gain of information. *Quarterly J. of Exp. Psych.* 4 (1952), 11-26.
18. Hyman, R. Stimulus information as a determinant of reaction time. *J. Exp. Psych.* 45 (1953), 188-196.
19. Jiang, Y., Song, J.H. and Rigas, A. High-capacity spatial contextual memory. *Psychonomic Bulletin & Review* 12, 3 (2005), 524-529.
20. Milner, P.M. A model for visual shape recognition. *Psychological Review* 81, 6 (1974), 521-535.
21. Newell, A. and Rosenbloom, P.S. Mechanisms of Skill Acquisition and the Law of Practice. in *Cognitive Skills and their Acquisition*, Erlbaum, 1981, 1-55.
22. Nielsen, J. *Usability Engineering*. Morgan Kaufmann, San Francisco, 1993.
23. Postma, A. and De Haan, E. What Was Where? Memory for Object Locations. *Q. J. Exp. Psych.* 49A, 1 (1996), 178-199.
24. Robertson, G., Czerwinski, M., Larson, K., Robbins, D., Thiel, D. van Dantzich, M. Data Mountain: Using Spatial Memory for Document Management. *Proc. UIST'98*, (1998), 153-162.
25. Scarr, J., Cockburn, A., Gutwin, C. and Bunt, A. Improving command selection with CommandMaps. *Proc. CHI'12*, ACM, Austin, Texas, 2012, 257-266.
26. Shneiderman, B. *Designing the User Interface*, 1992.
27. Steven, G.V. and Allan, R. Mental rotations, a group test of three-dimensional spatial visualization. *Perceptual and motor skills* 47, 2 (1978), 599-604.
28. Tak, S., Cockburn, A., Humm, K., Ahlstrom, D., Gutwin, C. and Scarr, J. Improving Window Switching Interfaces. *Proc. INTERACT'09*, (2009), 187-200.
29. Tak, S., Scarr, J., Gutwin, C. and Cockburn, A. Supporting window switching with spatially consistent thumbnail zones: design and evaluation. *Proc. INTERACT'11*, (2011), 331-347.
30. Wertheimer, M. Laws of organization in perceptual forms. *A source book of Gestalt psychology* (1999), 71-88.