Design and Evaluation of R-Zoom, a New Focus+Context Visualization Technique

J. Urquiza-Fuentes J.Á. Velázquez-Iturbide C.A. Lázaro-Carrascosa

ViDo Research Group, Rey Juan Carlos University. C/Tulipán s/n, 28933 Móstoles, Spain
{jaime.urquiza, angel.velazquez, carlos.lazaro}@urjc.es

Abstract. The paper presents the design and evaluation of R-Zoom, a focus+context information visualization technique. R-Zoom was designed to deal with large sets of items with specific structural properties (sequential order, heterogeneous sizes), and behavior requirements (flexible interaction, maximizing screen space, minimizing changes in screen). It was evaluated and compared to an overview+detail interface with 43 participants. Experienced users of R-Zoom completed tasks faster and with fewer errors than users of the overview+detail interface. Furthermore, a satisfaction questionnaire showed that users felt more comfortable with R-Zoom.

1 Introduction

We have developed a novel approach to the effortless generation and maintenance of program animations within an IDE [1]. A key feature of the approach is the availability of smaller versions of static visualizations of the execution steps, so that the user can select the most relevant ones. A critical issue here consists in preserving the comprehension of reduced visualizations. Typically, there are a large number of visualizations, so we need some technique to cope with them. After experimenting with different issues [2], we identified the requirements of the technique.

Our first approach was an overview+detail interface. However, in a general usability evaluation [3], we realized that users did not use this interface, even after watching the instructor using it. Consequently, we decided to use a focus+context interface. However, existing focus+context techniques do not satisfy such requirements, so we designed a novel technique called R-Zoom.

This paper describes the R-Zoom technique. The next section briefly states its requirements and behavior. The features and findings of an evaluation are described in sections 3 and 4, respectively. Finally, our conclusions are drawn in section 5.

2 The R-Zoom Technique

A description of R-Zoom is given in this section. A preliminary, comparative explanation of the technique can be found elsewhere [4].

2.1 Requirements

R-Zoom was designed to facilitate the tasks of visual search and selection in a sequence of reduced images. It provides a trade-off among space filling, context visibility and images comprehension. It also minimizes distraction of user's attention by trying to keep invariant global and relative locations of elements.

Being more specific, R-Zoom is intended for the following domain: (1) an ordered sequence of images reduced in size, (2) the number of images may be arbitrarily large, (3) images are heterogeneous in size and proportions, and (4) the relative proportions of reduced images must be preserved. Being an f+c technique, (5) any image can be brought into focus and displayed at a larger size.

From the point of view of user interaction, flexibility must be provided with respect to: (6) the reduction factor applied to images, and (7) the size of the window. Both facilities may lead to the situation where, for some reduction factors, all the miniatures cannot be visible at the same time. Here, (8) scrolling must be allowed.

Finally, the technique is intended to satisfy two properties: (9) screen space is minimized so that as many images as possible can be watched simultaneously, and (10) changes in the screen produced by a change of focus are minimized.

2.2 Related Work

An overview of related information visualization techniques that deal with large number of items is given in this subsection. Three families were taken into account: zooming+panning (z+p) techniques [5], overview+detail (o+d) techniques [6, chapter 4], and focus+context (f+c) techniques [6, section 3.3].

Z+p interfaces represent the simplest approach. The user interacts by zooming into items (loosing context) and panning to recover context. We discarded this approach because of the loss of context.

O+d interfaces use two separate windows, one to display context and the other for a detailed view of one item. As described above, users did not use this interface [3]. It has been argued [7] that, in some circumstances, o+d interfaces are slower than zooming ones, probably due to the mental effort required to switch between the detail and the overview windows.

F+c interfaces use one only window. When an element is focused to display it in detail, the remaining elements are redistributed within the window. Our first requirement states to deal with a sequence of images. Consequently, techniques which do not work with sequences were discarded, such as ConeTrees [8], Tree-maps [9], and Continuous Zoom [10]. We also discarded techniques which work with sequences of elements but unfocused ones are distorted in non-uniform ways, as Bifocal Lens [11], Perspective Wall [12] and Document Lens [13].

Other f+c techniques exhibit more similarities. Fisheye Views [14,15] and the Rubber Sheet [16] allows distorting the focused element without distorting the rest. The requirement of keeping as invariable as possible the location of all the elements is (almost) satisfied by Flip Zoom [17], which works with sequences of elements, where non focused elements are distorted uniformly way. However, its user interaction facilities are very poor.

A number of evaluations can be found in the literature, all of them giving better results for the f+c interface. Within the field of 3D environments, f+c has been compared to a conceptually z+p interface [18]. DateLens [19] is an f+c interface to use in a calendar application for PDAs, also compared to standard calendar software which implements the z+p approach. Finally, f+c screens have been compared with o+d and z+p interfaces [20].

2.3 Description of R-Zoom

Let us call miniature to a reduced image and focus to a selected image. Ideally, the focus is at its original size. At most there is one focus at a time, which is highlighted by putting it into a frame. This ensures that it can be distinguished, because it is typical for small images to have the same size both as a miniature and a focus.

Elements are always placed in left to right and top to bottom order. Each row contains as many miniatures as possible, thus maximizing space filling (see Fig. 1).

R-Zoom switches between two states: no-focus state and focus state. The initial state is no-focus, where all the elements are miniatures scaled with the default reduction factor (see Fig. 1a). Focus unselection restores reduced size and location of the focus, as well as the location of miniatures following the focus.

When a miniature is selected, R-Zoom switches to focus state. Then, the row of the focus is split into two rows, one with the miniatures previous to the focus and other with miniatures after it. (R-Zoom stands for Row-splitting Zoom.) If the focus fits the first row, it is located there (see Fig. 1b). If it only fits the second row, it is located there (see Fig. 1c). Otherwise, a scaled version of the focus is placed in the row with more free space available. If a second miniature is selected, the operation is equivalent to unselection of the previous focus followed by selection of the new one.

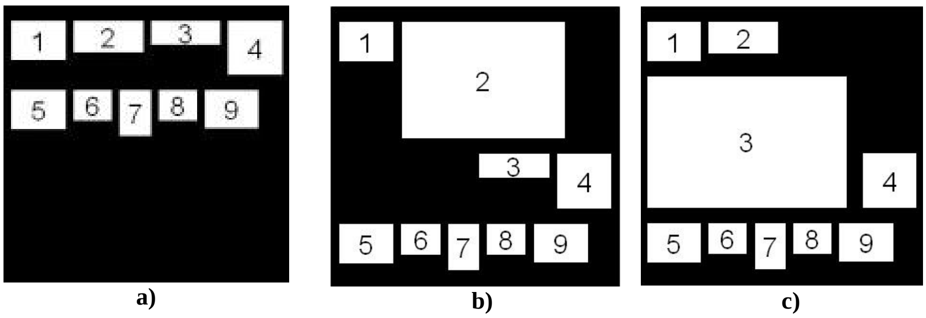

Fig. 1. Distribution of miniatures in: a) no-focus state, b) focus state with the focus located in the first row, ant c) focus state with the focus located in the second row.

Properties (9) and (10) are satisfied by the choices adopted for splitting and laying images out. In summary, these choices split a row into only two rows, use alignment to facilitate user reading, and place images so that changes in focus within a row restrict screen changes to that row.

Miniatures in the first row are aligned to the left and the top, and those in the second row are aligned to the top, maintaining their horizontal location. In addition, a focus placed in a first row is aligned to the top with the miniatures placed before, but a focus placed in a second row is aligned to the bottom with miniatures following it.

Screen changes due to focus selection in a row are restricted to the two resulting rows by placing the following row in a vertical location equal to the maximum of:

- The maximum enlarged size of the visualizations at the top row, plus the maximum size of the miniatures at the bottom row.
- The maximum size of the miniatures at the top row plus the maximum enlarged size of the visualizations at the bottom row.

Other possible interactions are: scrolling, change of scale factor, and resize of the window. These interactions affect state transition as summarized in Fig. 2. Scrolling behavior is provided only if all the miniatures do not fit the window. A change in the scale factor produces a new distribution of miniatures in rows, and updates the scrolling properties. A change in the width of the window produces redistribution, and a change of either width or height updates the scrolling properties.

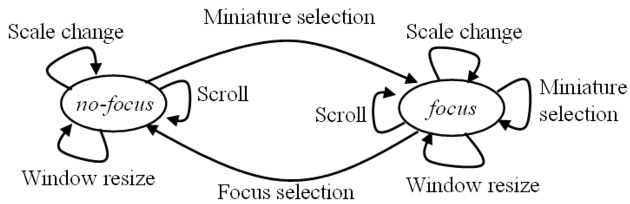

Fig. 2. State transitions and interactions of R-Zoom

3 Description of the Evaluation

We conducted a controlled evaluation where users had to complete a number of visual-search tasks. Visual-search performance of two interfaces was compared: R-Zoom and an o+d one. It attempted to find differences in effectiveness, efficiency and user's satisfaction between both interfaces. Although R-Zoom is a general-purpose visualization technique, notice that some questions are domain-dependent, namely, about animation construction. A set of demonstration videos about the evaluation is available via web: http://www.escet.urjc.es/~jurquiza/rzoom/demos.htm

3.1 Apparatus

Computers used in the evaluation were Pentium III 933 MHz processors with 256MB of RAM running Microsoft Windows XP Professional, 17" Hitachi CM620ET monitors at 1024x768 resolution, and Intel 82815 AGP 32MB video cards.

We used two applications for the experimental session: the monitor software that was used to time tasks and log users' errors, and the interfaces with which users worked (i.e. f+c and o+d). Both interfaces were integrated within the same IDE. Both the monitor software and the IDE were developed using Borland Delphi 5.

Both interfaces were maximized to use the whole screen space available. In addition, a reduction factor of 50% was applied to all the images used for the evaluation. This factor is a trade off between number of images simultaneously visible and their degree of comprehension. It has been proved [21] that reduction factors smaller than 50% has no effect in form recognition.

3.2 Participants

All the subjects that participated in the evaluation were students of a first year course of CS at Rey Juan Carlos University. There were three shifts of this course. Each shift was divided into two groups, experimental and control (EG and CG, respectively, for the rest of the paper). Students were randomly assigned to a group (either EG or CG). Participation in the evaluation was voluntary. A total number of 43 students completed the tasks: 17 in CGs and 26 in EGs. All of them had normal or corrected-to-normal vision.

3.3 Method, Procedure and Experimental Design

The participants' tasks involved image magnifying and zooming, and navigating through collections of reduced images seeking specific ones. Target images are cues to the tasks; as they represent execution stages of an algorithm, the user could guess their relative position in the collection. A task was complete when the correct index number of the target image was written in the text box of the monitor software and the user clicked on the "next" button. Then, a new task was presented.

The instructor explained each interface and then the participants trained. Training consisted of one drill-and-practice exercise cued in precisely the same way as the logged tasks. Collections used for both training and logged tasks were visualizations resulting from the execution of different algorithms.

Nine logged tasks were completed with each interface: three different tasks (which had targets at the initial, half and end position, respectively) with three different types of collections: short (one screen), medium (one screen and half), and large (three screens). Participants were not informed that targets were a discrete level of distance away, and the order of exposure was previously fixed for each task. Collection and target image associated to each task were the same independently from the group the user belonged to. To avoid users getting blocked in tasks, they were allowed to fail four times per task, after which the monitor software noticed the user the correct index number of the target image and continued with the next task.

Participants in the control group first completed nine tasks using the o+d interface. Simultaneously, the same tasks were completed by the experimental group using the R-Zoom interface. After that, each group changed the interface and completed nine additional tasks. For the second interface, different images but the same collections were used.

Dependent variables of the evaluation are: number of errors in each task and task completion time. The maximum number of errors allowed for each task is four; an error occurs when the user types an incorrect index number in the textbox and clicks on the “next” button. Task completion time is the time taken between the beginning of the task (last click on the “next” button of the previous task) and the end (click on the “next” button of the current task with the correct index number in the text box). We analyzed data using T-student (*pt*), Willcoxon Signed Ranks (*pw*) and Mann-Whitney (*pm*) tests, depending on the type of distribution of data.

After completing all the tasks, participants were asked to comment on the interfaces. They filled a questionnaire to rank the interfaces by several subjective measures: overall preference, usefulness, easiness of use, advantages, drawbacks, and the necessity of zooming interfaces in program animation construction.

4 Results

In this section, data are analyzed in two ways: group-interface-based and task-based. In the group-interface-based analysis (GIB, for the rest of the paper), data are referred by the group they belong to, C(control) or E(experimental), and the interface used, RZ (R-Zoom) or OD(o+d). Thus, four groups were formed: CRZ, COD, ERZ and EOD. In the task-based analysis (TB, for the rest of the paper), data are referred by the GIB reference plus a collection index (three collections) and an image index (three images per collection). Thus, task COD21 is the task where users from the control group, using the o+d interface, had to find the first image of the second collection.

In addition to using information on the group and interface used, experience was taken into account. Note that the pair group-interface denotes experience with tasks. The control group firstly used the o+d interface, so COD denotes no more experience than training (1 task); the same occurs for ERZ. EOD denotes more experience, because they had previously used the R-Zoom interface (1 training task + 9 R-Zoom tasks); the same occurs for CRZ.

Results analyze separately effectiveness and efficiency in target acquisition, and users’ satisfaction.

4.1 Verifying Quality of Data

Ideally, subjects in the same group but in different shifts completed the tasks under the same circumstances. Thus, time and error data can be processed independently from the shifts division of participants. Belonging to the same population has been tested for data obtained from participants in the same group but in different shifts. Error data analysis supported this hypothesis ($pm > 0.3$). But significant differences were found in time data for EOD11, COD22, CRZ11 and CRZ12.

Another anomaly was detected in EOD11 and CRZ11. Time spent working with the second interface over the first image of the first collection was considerably greater (over 10 times) than time for the rest of images of the same collection. Table 1 shows time in seconds spent by both groups working with the second interface on the first collection, divided by images found. We conclude that the difference was due to users having to learn a new interface. Another training task should be included after completing tasks with the first interface.

Table 1. Average task completion time for each group, in images of the first collection, in the second interface

Group/Image	1st	2nd	3rd
EG	482.61 s	34.11 s	42.88 s
CG	356.94 s	33.88 s	34.88 s

We have accounted both situations: significant differences among shifts of the same group, and first image of the first collection in the second interface. We have decided to ignore for the rest of the analysis, time results obtained in tasks involved: CRZ11, EOD11, COD22 and CRZ12.

4.2 Effectiveness in Target Acquisition

Of the 774 tasks (43 participants using 2 interfaces with 3 collections, and 3 images per collection), 59 failed: 55 one time and 4 two times.

The GIB-analysis found significant differences. Table 2 shows efficiency data and the significance difference analysis. For each pair of group-interface, efficiency is measured as: number of errors, error rate (per task) and error probability (computed as the division of the number of errors between the total possible errors, 4 per task). Although both interfaces are highly effective, experienced users with R-Zoom (CRZ) are those who obtained least error rate and error probability.

Table 2. GIB-analysis of significance differences for effectiveness, (italicized cells are pw, rest are pm; shadowed cells are no significant differences)

Group/Interface	# of errors	Errors / Task	Error prob.
COD	13	0.085	0.021
ERZ	24	0.102	0.025
CRZ	4	0.026	0.007
EOD	22	0.094	0.024

Significance differences			
	EOD	**CRZ**	**ERZ**
COD	0.251	*0.040*	0.168
ERZ	*0.770*	0.018	
CRZ	0.006		

Significant differences analysis in table 2 shows that adding experience to the use of R-Zoom (ERZ-CRZ) helps users in making 72.0% fewer errors. Note that this is not true for the o+d interface (COD-EOD). Experienced users also got better error rates and probability: an improvement of 70.8% if they use R-Zoom (CRZ-EOD). Finally, no differences were found for users with little experience (COD-ERZ).

Results of the TB-analysis (*pw* and *pm*) were greater than 0.1, so differences in effectiveness come from the whole group of tasks and not from individual ones.

4.3 Efficiency in Target Acquisition

Efficiency is measured as task completion time in seconds. The GIB-analysis found three significant differences of the six possible comparisons. Table 3 summarizes task completion time and significant differences analysis.

Table 3. GIB-analysis for time data and significance differences for efficiency, (italicized cells are pw, rest are pm; shadowed cells are no significant differences)

	Time average	Std. Deviation	Significance differences		
			EOD	**CRZ**	**ERZ**
COD	87.66 s	61.323	0.119	*<0.05*	0.200
ERZ	84.01 s	71.265	*0.295*	0.010	
CRZ	66.30 s	51.153	0.044		
EOD	78.98 s	59.609			

Experience has impact on efficiency results for R-Zoom (ERZ-CRZ), the improvement rate is 21.1%; but not for o+d (COD-EOD). Inexperienced users (COD-ERZ) got same results, independently of the interface they used. Experienced users (CRZ-EOD) that used R-Zoom got better results (16.06%). Rest of results (CRZ-COD and ERZ-EOD) supports previous ones.

TB-analysis has been applied to the results of the GIB-analysis, so only the pairs COD-CRZ, ERZ-CRZ and CRZ-EOD have been taken into account. Of the 20 possible comparisons (3 pairs with 9 tasks per pair, minus 7 ignored tasks), 11 gave significant differences, all of them in favor of the R-Zoom interface. Tables 4, 5 and 6 show tasks compared *pt* and the improvement rate.

In table 6, two values of *pt* (0.069 and 0.057) are above, but close to, the limit of 0.05. To clear up these doubts, an overlapping range analysis has been made. No overlapping was detected in both cases, therefore, significant differences exist.

This analysis evidences that the improvement rate, in average, is bigger in the second and third images (33.3% and 34.0%) than in the third image (15.0%). Also, half of the tasks with significant difference are related with the second image. The rationale of this result is the main feature of R-Zoom, it maintains as far as possible location of context elements. Thus, after locating the first image, the user have had the opportunity of viewing the surrounding context, where second and third images could be found, or at least, other images actuating as cues to locate the target ones.

Table 4. TB-analysis of effic. for COD-CRZ

Task	p_t	Impr.
21	0.007	14.7%
23	0.005	39.6%
31	0.045	21.8%
32	0.027	36.2%

Table 5. TB-analysis of effic. For ERZ-CRZ

Task	p_t	Impr.
21	0.013	8.5%
22	0.044	34.4%
23	0.032	30.5%
32	0.012	28.7%

Table 6. TB-analysis of effic. For CRZ-EOD

Task	p_t	Impr.
22	0.069	28.0%
23	0.020	31.9%
32	0.057	29,7%

4.4 User's Satisfaction

Data about users' satisfaction were collected with a questionnaire. First, we asked about the necessity of zooming interfaces to build algorithm animations with our IDE. 80% of users totally agreed with this need, 18% agreed. The second question was about users' preferences. 86% of them preferred R-Zoom, against the 12% who preferred the o+d interface. Users' opinion about ease of use was collected in the third question. 86% of users thought that R-Zoom is easier, or much easier, to use than o+d,

, while 6% thought that both interfaces were equally easy to use, and only 8% thought that o+d was easier or much easier to use. The opinion about usefulness was asked in question five. 42% of users answered that R-Zoom was much more useful than o+d, 22% answered that R-Zoom was more useful, 34% said that both were equally useful, and 2% said that o+d was much more useful.

Finally, users were asked to identify advantages and drawbacks in the interfaces they have used; here, the more frequent comments are reported. From 43 users, 38 of them wrote any comments. Two common comments are advantages of R-Zoom: 40% said that it offered a global vision of the collection and 24% said that they felt comfortable viewing the focused element and the context ones in the same window. Another comment is a drawback of o+d: 20% said that they did not feel comfortable viewing the focused element and the context ones in separate windows. Three more comments were advantages: 17.1% said that it is difficult to get lost with R-Zoom, 17,1% said that o+d offered a global vision of the collection and 14% said that they liked cursor navigation possibilities of R-Zoom. A last comment is a drawback: 11.4% said that is easy to get lost with the o+d interface.

5 Conclusions

The design and evaluation of R-Zoom, a focus+context information visualization technique, has been presented. We have described the requirements of the technique which try to provide the user with a trade-off among space filling, context visibility and images comprehension, and with minimizing the distraction of user's attention.

To evaluate R-Zoom, a controlled session has been conducted. In this evaluation, R-Zoom was compared to an overview+detail interface. Students used both the R-Zoom and the overview+detail interfaces. The evaluation revealed that experienced users using R-Zoom got better results than those using the overview+detail interface. Improvement rates for effectiveness and efficiency were 70.8% and 16.06%, respectively. We also found that experience did not have any impact on users of the overview+detail interface, while it caused improvements in users of R-Zoom. Effectiveness and efficiency improvement rates were 72.0% and 21.1% respectively.

These measurements were corroborated by users' satisfaction results. 86% of users preferred R-Zoom, 86% thought that R-Zoom was easier or much easier to use than overview+detail, and 64% thought that R-Zoom was more or much more useful. Finally, the two most frequent comments modestly support our decision of adopting a focus+context interface: they identified to be an advantage the one-window interface of focus+context versus the two-windows interface of overview+detail.

Acknowledgements

This work was supported by project TIN2004-07568 of the Spanish Ministry of Education and Science. Authors thank students who participated in the evaluation, and the reviewers of INTERACCIÓN 2006 by their valuable comments.

References

[1] J.Á. Velázquez-Iturbide, C. Pareja-Flores & J. Urquiza-Fuentes, An approach to effortless construction of program animations, Computers & Education, in press.

[2] F. Naharro-Berrocal, C. Pareja-Flores, J. Urquiza-Fuentes & J.Á. Velázquez-Iturbide, Approaches to comprehension-preserving graphical-reduction of program visualizations, Proc. 2002 Symp. Applied Computing, New York, NY, 2002, 771-777.

[3] M.A. Medina-Sánchez, et al., Empirical evaluation of usability of animations in a functional programming environment, Complutense University of Madrid, Tech. Rep. 141/04, 2004.

[4] J. Urquiza-Fuentes & J.Á. Velázquez-Iturbide, R-Zoom: A visualization technique for algorithm animation construction, IADIS Intl. Conf. Applied Computing, Lisbon, Portugal, 2005, 145-152.

[5] B.B. Bederson & J.D. Hollan, Pad++: a zooming graphical interface for exploring alternate interface physics, Proc. 7th Annual ACM Symp. UIST, New York, NY, 1994, 17-26.

[6] S.K. Card, et al., Information Visualization (San Francisco, CA: Morgan Kaufmann, 1999).

[7] K. Hornbæk, et al., Navigation patterns and usability of zoomable user interfaces with and without an overview, ACM Trans. Comput.-Hum. Interact., 9(4), 2002, 362-389.

[8] G.G. Robertson, et al., Cone Trees: Animated 3D visualizations of hierarchical information, Proc. of CHI'91, New York, NY, 1991, 189-194.

[9] B. Johnson & B. Shneiderman, Tree-maps: A space-filling approach to the visualization of hierarchical information structures, Proc. IEEE InfoVis, Los Alamitos, CA, 1991, 284-291.

[10] L. Bartram, et al., The continuous zoom: A constrained fisheye technique for viewing and navigating large information spaces, Proc. 8th Annual ACM Symp. UIST, New York, NY, 1995, 207-215.

[11] R. Spence, & M. Apperley, Data base navigation: an office environment for the professional, In S.K. Card, et al. (eds.), Information Visualization (San Francisco, CA: Morgan Kaufmann, 1999), 331-340.

[12] J.D. Mackinlay, et al., The Perspective Wall: Detail and context smoothly integrated, Proc. of CHI'91, New York, NY, 1991, 173-176.

[13] G.G. Robertson & J.D. Mackinlay, The Document Lens, In S.K. Card, et al. (eds.), Information Visualization (San Francisco, CA: Morgan Kaufmann, 1999), 526-569.

[14] G.W. Furnas, The FISHEYE view: A new look at structured files, In S.K. Card, et al. (eds.), Information Visualization (San Francisco, CA: Morgan Kaufmann, 1999), 311-330.

[15] M. Sarkar and M.H. Brown, Graphical fisheye views of graphs. Proc. SIGCHI Conference on Human Factors in Computing Systems, New York, NY, 1992, 83-91.

[16] M. Sarkar, et al., Stretching the Rubber Sheet: A metaphor for viewing large layouts on small screens, Proc. 6th Annual ACM Symp. UIST, New York, NY, 1993, 81-91.

[17] S. Björk, et al., A framework for focus+context visualizations, Proc. IEEE InfoVis, Los Alamitos, CA, 1999, 53-56.

[18] H. Shiaw, et al., The 3D vase museum: a new approach to context in a digital library, Proc. 4th ACM/IEEE-CS Joint Conference on Digital Libraries, New York, NY, 2004, 125-134.

[19] B.B. Bederson, et al., DateLens: A fisheye calendar interface for PDAs, ACM Trans. Comput.-Hum. Interact., 11(1), 2004, 90-119.

[20] P. Baudisch, et al., Keeping things in context: a comparative evaluation of focus plus context screens, overviews, and zooming, Proc. of CHI'02, New York, NY, 2002, 259-266.

[21] R.A. Rensink, The invariance of visual search to geometric transformation, Journal of Vision, 4(8), 2004, 178.