

Evaluating the PileBar technique

Paolo Brivio^{*}, Marco Tarini^{*}, Federico Ponchio[•],
Paolo Cignoni[•], Roberto Scopigno[•]

* <name.surname@uninsubria.it>

• <name.surname@isti.cnr.it>

November 7, 2011

Abstract

Image browsing is an ubiquitous task, which becomes an issue as the number of images raises over a few tens. Amongst the wide spectrum of innovative image arrangements proposed in literature, conventional browsers still adopt regular layouts, which are more simple to interact with, but less suited for hundreds or thousands of images. We compare this kind of arrangement with that of PileBars, which is designed to be as intuitive, but more scalable. In our experiments, we evaluate if the Pile-Bar image arrangements into ordered dynamic clusters effectively improve user navigation.

1 Introduction

Browsing image datasets composed of thousands of images is a well known problem. This issue is exacerbated in those applications in which the vast majority of the screen is devoted to represent something else, like a high resolution version of the currently selected image in the browser, or a 3D model of the depicted scene.

In this report, we present some *preliminary* quantitative and qualitative comparison among the PileBar [1] interface and the ones of a couple of conventional image-browsers. In our tests, we aim to measure how fast users navigate thousands of images with image-browsers covering only a small part of the screen. In particular, the PileBar browser is compared with a horizontal thumbnail-bar and a browser with a grid layout.

1.1 Experiments setup

In choosing the most appropriate PileBar contenders, we considered a number of alternatives, selected among several free tools available online. As the browsing interface of most of those browsers was out of our control and as we could not resize them properly, we finally resorted on the FastStone [2] thumbnail-bar, and on the Microsoft Windows Explorer grid layout. In particular, we adapted the PileBar, the Explorer, and the FastStone windows to the same 1600x235 pixels working area, thus covering

but a fraction of the whole 1600×900 screen, which elsewhere was filled with a blank background. All experiments were performed on a laptop with a *17inch* screen, a *2.5GHz* dual processor, and *4GB* of RAM.

All interfaces support browsing through the mouse-wheel. However, the PileBar is the only one implementing a dynamic layout, enabling thumbnail selection, dragging, and preview, while the other browsers interactions are based on scrollbars.

A set of 16 students and young researchers volunteered to participate in two tests. However, due to the long time required to complete the first one, only 10 of them accepted to participate also in it. All had a normal or corrected to normal vision with no color blindness, and experiments took place under the same lighting conditions in a silent room.

2 Browsers evaluation

As all participants were new both to the PileBar and to the FastStone interface, each one was allowed a preliminary 5 minutes test-run on each browser, using a different training dataset. During these test-runs, users were carefully instructed using one sheet with illustrated instructions about the complete set of functionalities of either tool.

Then, we asked each user to perform the experiments, each constituted of a sequence of tasks on a specific dataset. At the begin of every experiment, the Explorer and FastStone browsers figured the thumbnail of the first image of the dataset, while the PileBar browser focused at the middle of it. Tasks were described in written assignments, and users were totally unassisted while performing them. Timings were taken after each task was read and understood and until they pointed at the target image. After each task was completed, the browser was reset to its initial configuration.

2.1 Experiment one

First, we evaluated how the considered browsers are efficacious in browsing an image dataset in which an explicit total ordering is naturally imposed (i.e. an arbitrary ordering would make the browsing cumbersome).

Dataset the dataset has been generated procedurally: given the set of integers from 1 to 28,000, 20 disjoint subsets of random size (253 on average) have been randomly extracted. Then, every subset number has been associated an image featuring that number. Finally, the test dataset has been filled with all of these 5,061 images. Numbers were written in white color on a dark background with the same font and size. Inside the dataset they were ordered from the smallest to the largest and this ordering was respected in each browser.

Task users were asked to locate, in a random order, each of the first 28 multiples of 1,000, or, if not present, its two nearest numbers. To further minimize the influence of user knowledge of the dataset on his/her performance, half of the tasks was performed with PileBar first, whereas the other half with Explorer first. In either case, all tasks were performed with FastStone last.

Results and discussion compared results are shown in Tab. 1. We used the R system to compute statistics on the timing data. For each independent variable (i.e. the browser adopted during the experiment), we considered two dependent variables: the time to complete each task, and the order of browser utilization (the latter does not apply for FastStone, of course). All dependent variables resulted normally distributed with respect to each independent variable with Shapiro-Wilk normality test.

A t-Student two-tailed paired test shows that the measured differences of the user performances between PileBar and Explorer and between PileBar and FastStone is significant at least at level $p < 0.01$ (p-value = 0.0036 and p-value = $9.66e - 08$, respectively). On average, while using PileBar, participants took 8.5 seconds (s.d. 5.8) to complete, while 10.3 seconds (s.d. 3.2) with Explorer, and 13.1 seconds (s.d. 5.0) with FastStone.

Considering user performances when using Explorer, we notice that there is no significant difference when using Explorer first or second for $p < 0.05$ (p-value = 0.8817). On the other hand, the difference between using PileBar first and PileBar second is significant even for $p < 0.001$ (p-value = 0.0008). Participants were slower when using PileBar first, taking a mean time of 9.78 seconds (s.d. 1.78), compared to 7.19 seconds (s.d. 0.98).

Table 1: Experimenting with an explicitly ordered dataset. For each of the 10 participants, timings were recorded to complete each of the 28 tasks, performed with PileBar, Explorer, and FastStone image browsers. Here, each cell contains the value of the ratio between the time to perform tasks with PileBar over Explorer (first column), and with PileBar over FastStone (second column).

User	PileBars/Explorer	PileBars/FastStone
User 1	66.9%	60.7%
User 2	69.0%	62.6%
User 3	73.8%	73.8%
User 4	93.8%	74.9%
User 5	98.8%	67.9%
User 6	83.4%	62.8%
User 7	94.3%	70.6%
User 8	85.2%	56.1%
User 9	77.5%	56.5%
User 10	94.6%	65.0%
Average	83.7%	65.1%

When all tasks were completed, each participant was asked to score the three tools from 1 (minimum) to 10 (maximum), answering the following questions:

Q1. how much did you feel comfortable with each tool?

Q2. how much did you think each tool was helpful for these tasks?

The average scores are shown in Tab. 2. Considering Q1, with a t-Student test no significant difference can be observed between PileBar

and Explorer for $p < 0.05$ (p-value = 0.7670). Their average scores are similar, too. However, the differences between PileBar and FastStone are very significant (p-value = 0.0001). In case of Q2, the measured differences between PileBar and Explorer, and between PileBar and FastStone are significant for $p < 0.001$ (p-value is $9.0e - 4$, and $4.29e - 9$, respectively).

Table 2: User scores. The table reports the users average scores for the browsers used for the first test (1 is minimum, 10 is maximum).

Question	PileBar	Explorer	FastStone
Q1	7.1	6.9	2.9
Q2	8.3	5.7	2.4

Finally, participants were asked for qualitative additional comments on the tools. Three of them reported that they found confusing the PileBar arbitrary image ordering on the vertical direction, while all agreed that the thumbnail-bar of the FastStone interface was the less effective tool for browsing large image datasets. In addition, most participants argued that, with more training with the PileBar interface, they would have probably performed better with it. This is also partially confirmed by the above questionnaire scores, as users stated that they felt equally comfortable with the PileBar and the Explorer interfaces, but they thought that the former has a higher potential for helping people in locating images.

2.2 Experiment two

After the first experiment, we investigated users performances on a dataset with an explicit image-clustering function defined, but without a semantically significant order. Given the results of the former experiment, we chose to compare only PileBar and Explorer image browsers.

Dataset images have been retrieved with Google by searching for 22 species of different domestic animals and their various races, discarding outliers. The resulting images were joint in a sequence of 3,784 animals grouped by species and race. In other words, Rottweilers were separated from Chihuahuas, but no cat could appear among dogs. Note that tags have no semantic order (i.e. there is no cue to predict if cats come before or after dogs). Image tagging was used by PileBar to cluster images, while the other two browsers showed the plain sequence of animals. Furthermore, for each cluster of images in the PileBar browser it was computed an image ordering based on image color distribution and color spatial layout.

Task we differentiated between two types of task: *to locate a species of animal*, and *to locate a specific animal of a species*. The tasks were:

- T1. to locate a turtle (11 images in total);
- T2. to locate a Dalmatian (11 images in total);
- T3. to locate a red Canary pictured on a uniform white background (4 items are present);
- T4. to locate a cat pictured on a red background (4 items are present);

To prevent the task execution ordering to significantly influence the results, half of the participants performed T1 and T3 with PileBar first, and T2 and T4 with Explorer first. The other half, instead, did the opposite.

Results and discussion The timings to complete each task are summarized in Tab. 3. Notice that in this case data distributions are not normal. Thus, we analyze them with a paired Wilcoxon signed rank test.

Considering each browser separately, there is no significant difference – for $p < 0.05$ – in performing the first two tasks ($p\text{-value}_{PB} = 0.4164$, $p\text{-value}_{EX} = 0.06084$), nor in performing the last two ($p\text{-value}_{PB} = 0.1533$, $p\text{-value}_{EX} = 0.5134$). This confirms that the choice of T1 and T2 as exemplars of the first type of task, and of T3 and T4 as exemplars of the second type of task did not influence the results.

Comparing the timings measured with the two browsers, it results that the difference between PileBar and Explorer is significant even for $p < 0.001$ both for the first two tasks ($p\text{-value} = 8.315e - 7$) and for the last two ($p\text{-value} = 5.207e - 6$). Overall, the direction of the difference is always clear, as with PileBar the recorded timings are from 2 to 20 times faster than with Explorer.

Table 3: Results of experiment two. This table reports the average timings (in seconds) to complete each of the four tasks listed above, using PileBar and Explorer image browsers. Significance levels are computed with a Wilcoxon test.

Task	PileBar		Explorer		p-value
	Time (secs)	s.d.	Time (secs)	s.d.	
T1	2.4	0.8	49.1	34.8	4.814e-4
T2	2.8	1.3	25.6	14.2	4.803e-4
T3	8.0	5.6	19.1	12.5	2.913e-3
T4	4.8	3.1	15.3	7.6	4.782e-4

3 Conclusion

Browsing thousands of images with conventional image browsers has proven to be time consuming. In our preliminary experiments, the PileBar interface has been generally appreciated by the users, as it allowed them to locate arbitrary images faster and without linearly scanning the whole dataset. Our collected data confirms the improvements that the PileBar novel image arrangements were meant to bring out.

However, the differences in the results of the two experiments reveal that further experiments need to be conducted before an overall conclusion on PileBar effectiveness could be carried out. Some questions remain unanswered: in which measure does the specific thumbnail arrangement (i.e. number of images per pile, distance among piles, number of piles per column, etc.) affect the bar effectiveness? If users were let change image ordering and clustering, how much would their performances vary? It would be sensible to tailor new experiments on these aspects, which indeed constitute a key aspect for the design of image-browsers in practical cases.

References

- [1] P. Brivio, M. Tarini, F. Ponchio, and P. Cignoni. Pilebars: Scalable dynamic thumbnail bars. Technical Report 2011-TR-006, ISTI-CNR, Pisa, Italy, 2011.
- [2] FastStone-Soft. Faststone. <http://www.faststone.org/>, 2004.