DynamicMaps: Similarity-based Browsing through a Massive Set of Images

Yanir Kleiman[1], **Joel Lanir**[2], **Dov Danon**[1], **Yasmin Felberbaum**[2], **Daniel Cohen-Or**[1]
[1]Tel-Aviv University, Haim Levanon 55, Tel-Aviv Yafo, Israel
[2]University of Haifa, Mt. Carmel, Haifa, 31905, Israel
yanirk@gmail.com, ylanir@haifa.ac.il, dov84d@gmail.com, jas86f@gmail.com, dcor@tau.ac.il

ABSTRACT

We present a novel system for browsing through a very large set of images according to similarity. The images are dynamically placed on a 2D canvas next to their nearest neighbors in a high-dimensional feature space. The layout and choice of images is generated on-the-fly during user interaction, reflecting the user's navigation tendencies and interests. This intuitive solution for image browsing provides a continuous experience of navigating through an infinite 2D grid arranged by similarity. In contrast to common multidimensional embedding methods, our solution does not entail an upfront creation of a full global map. Image map generation is dynamic, fast and scalable, independent of the number of images in the dataset, and seamlessly supports online updates to the dataset. Thus, the technique is a viable solution for massive and constantly varying datasets consisting of millions of images. Evaluation of our approach shows that when using DynamicMaps, users viewed many more images per minute compared to a standard relevance feedback interface, suggesting that it supports more fluid and natural interaction that enables easier and faster movement in the image space. Most users preferred DynamicMaps, indicating it is more exploratory, better supports serendipitous browsing and more fun to use.

Author Keywords

Image browsing; Image search; similarity browsing; relevance feedback;

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous. H.3.3. Information search and retrieval.

INTRODUCTION

As huge image collections become common in the Web and various digital libraries, it is increasingly important to allow users to easily search and browse these collections in fast and intuitive ways. Many commercial Web search engines have developed technologies to allow users to search for images, mostly focusing on keyword search with images presented in a grid ordered by some sort of relevance measure. While text-based directed search can be effective for many image search tasks, studies have shown that image search is often more exploratory in nature than Web search, and that browsing is an essential strategy when looking for images [1, 6, 21]. Still most commercial systems lack support for exploratory search and do not provide means for serendipity in the search process [13, 21].

To address this gap, various research systems have looked into browsing as a complementary tool to text-based search methods [8, 23]. One useful way of browsing through images is by using similarity. Users often look for images that are similar to a given image, and browsing according to similarity between images has been shown to be useful [20, 25]. A prominent method for browsing images according to similarity uses relevance feedback techniques, which refine search results according to a selection of preferred images made by the user [37]. At each relevance feedback step, the user is presented with a new set of images based upon past selections. However, the navigation experience with this approach is not continuous and it requires the user to go over a large collection of images and select the relevant or irrelevant ones at each step.

A possibly more intuitive approach is to lay out the images on a two-dimensional grid allowing users to navigate over them in a continuous manner. We present DynamicMaps, a novel system to intuitively navigate through a massive, dynamic set of images. Navigation is done with a metaphor of an infinite two-dimensional canvas, where the images are presented and browsed through according to similarity. Figure 1 illustrates the browsing process. The user views a local subset of images, ordered such that similar images are next to each other (A). The user decides to pan towards images on the bottom right corner (B), and new images similar to the ones on the boundaries of the images in (B) appear to reveal another patch of the map (C). The currently displayed map can be figuratively viewed as a window that shows a local patch of an infinite canvas.

CHI 2015, April 18 - 23 2015, Seoul, Republic of Korea
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3145-6/15/04□$15.00
http://dx.doi.org/10.1145/2702123.2702224

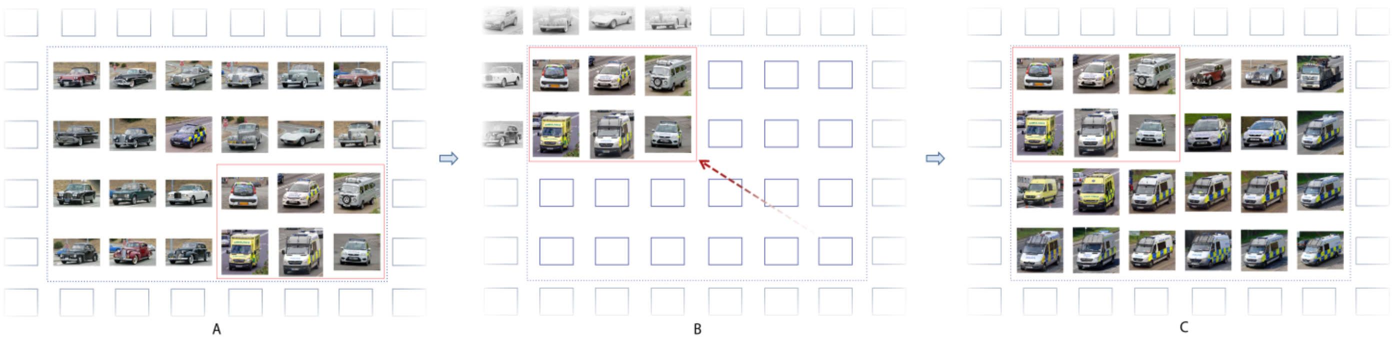

Figure 1. Browsing images using DynamicMap. A region of photos is displayed ordered by similarity (A). Dragging the map to the upper left corner (B) reveals new images which are similar to images in the dragging direction (C).

In order to support millions of images and more, given the high dimensionality of the image space (i.e., images have many different features that can be used: color, spatial structure, composition, texture, etc.), the challenge is in creating a cohesive grid that preserves the relations among all images. Conventional dimensionality reduction methods such as SOM, that can organize images on a 2D plane according to similarity, use global computations and are thus not scalable for a very large set of images [18]. We claim that for image-based navigation, the global requirements can be relaxed. In our solution, the local view is generated during navigation in response to user interactions, such that the relative positions of images respect only local high-dimensional relations.

We build on our previous work of browsing 3D shapes [19], extending and implementing it to support the browsing of images. The specific contributions of the current paper are as follows. First, images suggest a very different feature space than shapes. Dealing with images requires very different semantics and the variation between images is much larger. Second, we tested a significantly larger and richer collection of objects (1 million vs. 4500) enabling us to examine claims of supporting a massive set, as well as the behavior of the system when dense data is available. Third, image search is a much more ubiquitous domain than shape search. Forth and for most, we focus here on the UI considerations rather than the technical solution, reporting on a controlled empirical user study that evaluates the presented method and discuss its advantages and limitations, specifically in context of image search.

Evaluation of DynamicMaps, comparing it to a standard relevance feedback technique, showed that such similarity-based navigation enables an open-loop exploration, where users can quickly and seamlessly direct the search towards relevant images of their choice without the need to sequentially go over the images and select the relevant ones. Using DynamicMaps, users viewed many more images per minute compared to a relevance feedback interface, affording a quicker and more natural method of interaction. Our results indicate that most users preferred DynamicMaps thinking it is more exploratory, better supports serendipitous browsing in which the user explores unknown regions of a large dataset, and more fun.

RELATED WORK

The majority of work done on image retrieval focuses on the back end of the search in image indexing and content-based image retrieval (CBIR) (see survey in [9]). However, as our major contribution is with the design and evaluation of an image browsing user interface, we focus on the user experience side and first review systems that support image browsing and specifically similarity-based image browsing. We then briefly review related work on relevance feedback and dimensionality reduction methods.

Image browsing

Images have several characteristics that makes image search different than text-based search. Unlike text documents, the content of an image can be grasped at a glance, and a large number of images can be presented to a user at once. In image search, often the user does not have an exact target in mind [7]. Furthermore, images often lack textual cues and might have many different meanings embedded in a single image [30], making them difficult to support with only keyword-based search. For example, if the user is looking for a scenery image to add to a presentation, the user would not necessarily know how to phrase the search terms or even exactly which image he or she is looking for. Moreover, images presented in the first page of a text-based search result are not necessarily better than those presented in the following pages. Consequently, users have to sequentially scan these results spending considerable effort finding relevant images. Still, most current systems focus on providing text-based image querying rather than navigational support even though studies have shown that image browsing can improve in achieving user's search needs [8, 20, 23].

To address these needs, some research systems focus on supporting various browsing capabilities to enable navigating through images. For example, browsing specific clusters of images [23], browsing hierarchies that are automatically built according to visual and semantic similarities [17], or browsing along conceptual dimensions according to hierarchical faceted metadata [36]. Similarly, some commercial systems added interactive visual content-based search methods that allow browsing by similar shape and/or color. The “similar images” feature, allows users to search for images similar to a certain image, utilizing relevance feedback methods.

Laying out images on a large canvas allows users to browse the images according to some organization of their structure using pan and zoom interactions. In [8], the results of an initial query can be browsed on a zoomable user interface (ZUI). In [23], images were clustered into conceptual regions. The user can continuously pan across this plane and zoom in or out of any particular region. In JustClick [12] a topic network is first generated and browsed through. Representative images of a topic are then organized on a 2D hyperbolic plane according to similarity.

In the works above, the images are laid out according to some measure of distance (in similarity) between them. However, when browsing images, there is no need for an accurate representation of the original distances between images. In fact, an even spread of images over the canvas can be more beneficial than an accurate representation of the original geometry [25], especially in cases where the original data includes very distinctive clusters which may appear too far apart for easy navigation. Indeed, the most common way to lay out a set of images is on a two-dimensional grid. Studies have found that arranging a set of thumbnail images on a single-page grid according to their similarity can be useful for users in an image browsing task [20]. Strong and Gong [32, 33] employed this idea and organized a collection of images based on similarity using an SOM-based algorithm. Users could browse the image collection using pan and zoom interactions. According to the authors their system could support browsing with up to 10,000 images. Similarly, in PhotoMesa [3], images are laid on a large 2D grid. Users can browse through a large collection of images, panning to browse horizontally or vertically through the image collection. Here, zooming out enabled seeing the photos semantically grouped into pre-organized categories.

The systems mentioned above work with a limited number of images and are not scalable beyond several thousands of images. Thus, they are not suited for large repositories that exist in the Web today. Our work builds upon the idea of browsing images on a large 2D canvas, and the works in [20, 24, 32] that present similar images together on a grid. However, we apply it to a dataset of virtually unlimited size, finding solutions for interacting in such a large image space.

Relevance feedback

Many recent search and retrieval systems, including image retrieval, utilize relevance feedback [27], a method to refine search results using selection of preferred elements. Suditu and Fleuret [34] presented an image retrieval system that features iterative relevance feedback for a very large set of images. At each step, the user is presented with a set of images, and selects a single image that is the closest match to the desired query. Then a new set of images is displayed and the process is repeated.

While this process may be effective at filtering relevant images, the use of relevance feedback in commercial search interfaces is still relatively rare [28]. One possible explanation is that it requires users to make relevance judgments on each item, which is an effortful user task [28]. Relevance feedback tends to work best when the user selects multiple objects as relevant as well as some objects as irrelevant. However, selecting multiple objects is cumbersome for most users. This is amplified in image search where extractable low-level features (e.g., color, texture, shape) may not necessarily match high-level perception-based human interpretation [37].

Dimensionality reduction

Dimensionality reduction is a wide area that has been extensively researched over the years. Common techniques such as multidimensional scaling (MDS) [4] or locally linear embedding (LLE) [26] create a global map that aims to preserve the distances among the high dimensional data points. In image search, a similarity measure between images is first computed, after which a visual map of the image collection is constructed according to the projection of the features to a 2D space. A number of papers use these techniques to map and then browse an images space according to the global relations among images [5, 23]. In order to better organize images, layout methods have been applied to MDS results to put them on a 2D grid [24]. Self-organizing maps (SOM) [18] is a dimensionality reduction method that produces a grid which preserves similarity between elements without preserving the distance. Works such as [29, 32, 33] utilize SOM to visualize a relatively small set of images in a global cohesive map.

Dimensionality reduction methods such as MDS or SOM work very well for small datasets. However they do not scale well and are too computationally intensive to be effective for massive datasets. In [16], an SOM was used to organize millions of documents. Due to the large volume of the dataset, special tools and methodologies had to be developed, yet still, several weeks of computation time were required. In addition, in order to add or remove images to the dataset, the computation process needs to be redone. Since our method relies only on local relationships, our technique is computationally inexpensive and thus highly scalable, as well as dynamic, allowing the addition and removal of data during execution.

DYNAMICMAPS

We first describe the map generation process (for a detailed description see [19]). We then describe the way we computed the image nearest neighbors that is used in the map generation process. Next, we describe the zoom levels mechanism that was built to support browsing through different levels of similarity. Finally, we explain how users can focus on a single image.

Map generation

The grid-like map is instantly generated during user interaction to keep a sense of continuity. The generation of local neighborhoods in the map is based on the assumption that for high dimensional data such as images, short distances are more accurately measured than long distances. Even for a human observer, the task of deciding which images are more similar to each other is easier for a set of similar images than for a set of very different images. This carries over to automatically computed distance measures as well. We thus use only the shortest distances between images in our dataset. Only the distances to k nearest neighbors (with k being a small positive integer) of each image in the dataset are considered. Images farther apart relate to each other by a sequence of nearest neighbors that connect them, utilizing short distances in the whole set. The optimization problem becomes one of maximizing the similarity of nearby images, such that each image is surrounded by similar images. The result is a continuous map in which images show a gradual change over local neighborhoods. A dense set is expected to have shorter distances than a sparse set, hence our method is especially suitable for massive datasets.

The input to the map generation process is a precomputed list of nearest neighbors and their similarity score (see next section) for every image in the dataset. The map can be then seeded around a specific image or constrained by any number of images. As the user is navigating by panning the map, the map is extended locally to the region of interest, using previously placed images as constraints. The map is generated by iteratively filling in empty cells in the grid. Each cell is assigned with the most compatible image by calculating the best match of the closest neighbors of images that are already assigned to adjacent cells. Images that already appear on the map are excluded.

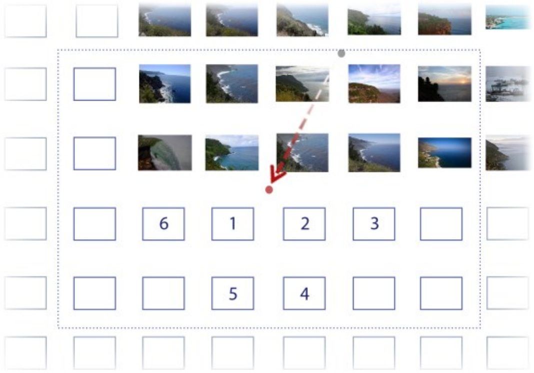

Figure 2: The order in which cells in the map are filled is relative to the direction of browsing. In this example, the user dragged the map two images up and one image to the right. The gray dot and red dot, respectively, mark the previous and new center of the viewport. The numbers state the order in which the first six images on the map are filled.

The order in which empty cells are selected has a great effect on the mapping. We select the vacant cells in accordance with the user's actions; in general, we give precedence to cells that have as many reference images as possible. However, since the map is a regular grid, often there will be ties and many cells will have the same number of reference images. We break ties by selecting the cell which is closest to the direction the user panned to. This causes the grid to start growing from the user's focus area on and outwards into the rest of the map. Figure 2 illustrates the order in which empty cells in the grid are filled. The user drags the map two images up and one image to the right. The center of the user's viewport thus moves on the map in the opposite direction; two cells down and one cell to the left. The cells marked with numbers 1 to 6 are closest to the direction of movement and therefore will be filled in their respective order, followed by the rest of the cells on the grid. Existing images which are closer to the panning direction effectively have more weight in the map generation, since their neighbors are selected first.

To maintain stability of view such that images that were already displayed are not repeated during browsing, we keep images that were placed on the map in their relative global position and prevent them from reappearing on the map again. That is, we save the 2D structure of the view of the created map, so if the user returns back to a previously visited location (e.g., pans right and then pans left), the map is not recreated but rather the previously shown map is presented. The 2D structure is kept until the map is rearranged around a new image. The stability of view supports the important guideline of reversal of actions [31], since users can easily return back on their steps.

Image nearest neighbors

The map generation is decoupled from the nearest neighbors computation. In our implementation, we find the k nearest neighbors of every image using three image metrics, or image *descriptors*. Each of the following three descriptors is computed for each image in the dataset. The distance between two images in each descriptor space is the Euclidean distance between the image descriptors. Average color and color histogram are popular descriptors used in image retrieval [10]. We used them as described below combined with a third descriptor. In other implementations, it is possible to choose different descriptors or different weights for each descriptor.

Average Color. The image is divided into 16 segments, a four by four grid, and the average color in each segment is computed. Similar images in this metric tend to have a similar composition. Of course, the image partitioning does not necessarily need to be four by four, but we find this partitioning appealing in the sense that it seems fine enough to distinguish between images with significantly different compositions, yet sufficiently coarse to ignore small changes in composition of similar images.

Color Histogram. A joint color histogram for RGB values is computed. Each color channel is divided into four bins, to create a total of 64 bins for every color combination. The number of pixels that fall in each bin is counted and divided by the total number of pixels in the image. Similar images in this metric have similar color distributions, which suggest similar atmosphere or surrounding. This descriptor is less sensitive to translation, rotation or reflection of the images compared with the average color descriptor.

Spatial Envelope. The spatial envelope was described in [22] and named gist descriptor since it captures the gist or context of a scene. The gist descriptor describes the spatial structure of a scene using a set of spectral signatures which are specifically tailored for the task of scene recognition. It was shown that in the gist descriptor space, scenes that belong to the same context are projected close to each other. We use the code provided by the authors to compute the gist descriptor of every image in the dataset.

The three descriptors are calculated for each image, and k nearest neighbors are found for each descriptor space separately. The distance from the image to each nearest neighbor in each descriptor space is kept as well. The three lists are then merged to a single list of k nearest neighbors by computing a normalized score for each candidate that appears in one or more lists.

Zoom levels

Our system supports semantic zooming by allowing zooming out to see a wider variety of images as well as enhance navigation capabilities, and zooming in to see more similar images. Zooming out brings the user to a higher level where images are less similar to each other, and thus it is possible to browse further away in the similarity matrix, and zoom back in whenever reaching an area of interest. This enables more diversity within the results, which was shown to be important in image search [14]. When the user zooms out, he or she browses through the higher level which is much smaller in size and thus much more diverse than the original level.

Zoom levels are implemented by automatically preselecting high-level delegates for every image in the dataset from its nearest neighbors list. To create a smaller, higher level delegate map, for each image in the dataset, we check whether one of its nearest neighbors is already a high-level delegate. If none of the nearest neighbors of the image is a delegate, the image itself becomes a delegate for all of its neighbors. The same process can be done when adding a new image to an existing dataset. We then create a higher level map using the chosen delegates. This process is repeated recursively to create multiple zoom levels. For example, if starting at a 1,000,000 images, and using k=20 nearest neighbors, since every delegate represents around 20 images, the second level will include 50,000 delegates, the third level 2500 and the fourth and final zoom level will include 125 images.

Zooming is always done according to a reference image. The user hovers over an image of his or her choice and uses the mouse scroll to change the zoom level. We regenerate the map around that image in a lower (or higher) zoom level, and vice versa for zooming out. To keep the user oriented, when zooming, the reference image does not get replaced by its delegate. Figure 3 illustrates the zooming out process. In addition, there is also a zoom widget in the control panel which, if chosen, zooms in or out according to a click with the center image as the reference image.

Focusing on an image

We provide the user with the option to focus on a single image by double clicking on it. This regenerates the map around the clicked image in the lowest zoom level (maximum similarity). The images in the rebuilt map are

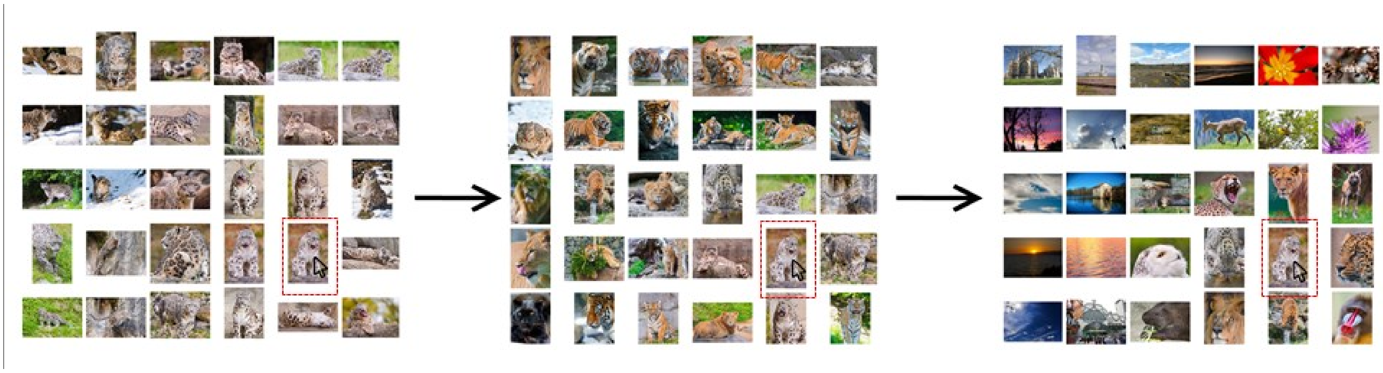

Figure 3. Zooming out. The user hovers over an image and uses the mouse scroll to zoom out. The image stays as a reference point and the images around it are retrieved from a higher zoom level (red box is only for illustration and is not part of the interface).

then more likely to be similar to the image in focus rather than the images around it. This option also provides the user with another way to quickly zoom in from higher levels.

EVALUATION

In order to evaluate DynamicMaps, we compared it to a standard relevance feedback method. Relevance feedback (RF) was chosen as the most prominent method for similarity-based browsing, and the only one we are aware of, that can support a corpus of millions of images. Another possibility was to compare DynamicMaps to another global approach (such as SOM). However, this would have limited the evaluation to the level of thousands of images (other global approaches do not scale more), not fully evaluating the utility of our system. For simplicity, we decided to implement a standard RF method rather than a more complex one (i.e., that might include negative feedback). We employed a within-subject design to compare performance and attitudes of participants. The main variable *interface* describes the search interface used: DynamicMaps (DM), or relevance feedback (RF).

Methodology

Participants. A total of 24 participants took part in the study, 11 were male and 13 were female with an average age of 27.1 (*SD* = 5.1). Participants were mostly students of a large university from a wide range of departments and faculties. All had normal or corrected-to-normal eye vision. 15 participants reported searching on the Web for images every week, while 5 participants reported searching every two weeks or so and 4 reported a lower rate. Image search task reported including finding images for presentations, looking for images for study purposes, looking for products and more. Most participants indicated using Google images as their main image search tool.

Interfaces. Both interfaces show a grid of 6x5 images at any given time. For the DM interface, we used the system as described above, initialized with the starting image at the center, around which the algorithm builds the initial screen grid of 30 images. For the RF interface, the system initializes showing the starting image on the upper left corner followed by the 29 closest neighbors on a grid. The user can then select up to 3 images and click a button (labeled "more images") to fetch the next set of images closest in similarity to the chosen images (ordered by similarity). At any time, the user can press the back button and return to the previous screen. Both interfaces included a "restart" button that returned the view to the initial screen formed by the starting image. As a starting point, users could enter an image number in a provided textbox around which the system initializes as mentioned above.

Tasks. Tasks were designed to be open-ended and reflect real-world search needs (similar to [24, 36]). Two general tasks were defined for the within-subject design. In each task participants were asked to find images that would best fit text slides of a given presentation. For example, the first presentation was on a non-profit organization titled "the society of preservation of nature". The initial slide was a title slide, the second slide talked about the organization's mission, the third slide talked about the history of the organization and the final slide talked about the major active projects the organization employs today. All slides included only text with no color or graphic design. Participants were asked to find up to three images that would best fit each slide. The second presentation was similar in nature and had to do with architecture. For each task, participants were given four starting points in the interface. This emulated four possible keyword search queries. The starting points were chosen as single images relevant to the task (for example, images of animals or nature for the previously mentioned task).

Dataset. We downloaded one million images from the free-to-use (creative commons) Flickr image hosting service. The image collection spans photos with an upload date within a range of 400 days, where for each day in the range a few thousands of random images were selected. This has resulted in a diverse dataset which contains images of many different types, such as landscapes, urban areas, people, wildlife, birds, vehicles and more. Computing the *k* nearest neighbors for each image was done as a preprocess using Matlab with *k*=20 and took a few hours for the entire dataset.

Procedure. Participants were seated in front of a 22" screen with 1440x900 screen resolution. Participants were then presented with one of the two interfaces. The user interface features were first explained to them, after which participants performed one practice task on which they were instructed to use the interface until they felt comfortable with it. Participants were then given one of the two tasks and were asked to perform the task as best as possible. No time limit was given for the task. After they completed the first task using the first interface participants were asked to fill in a questionnaire asking their subjective opinion of the interface they just used. Participants were then given the second interface on which they completed the same procedure using the second task. All interactions with the interfaces were logged and later analyzed. At the end of the experiment a comparative questionnaire was given and participants were asked to comment on each interface. The order of interfaces (which interface was first used), as well as which task set was used with which interface was fully counterbalanced, creating 4 different configurations (six participants in each configuration).

Results

Order effects

To rule out order effect (whether participants started with the DM or the RF interface), we performed a between-subject ANOVA with interface order as the independent variable on both task completion time and on number of unique images seen. No effect was found for both variables. Next, to ensure there were no differences between tasks we performed a within-subject ANOVA with task as an independent variable. Again, no effect was found for both task completion time and number of images seen.

Completion time

On average, it took participants 805.8 seconds (~13.5 minutes) to complete the task in the DM interface (*SD* = 334) and 761.5 seconds (~12.7 minutes) in the RF interface (*SD* = 348). A one-way repeated-measures ANOVA on task completion time did not find these differences to be significant, F(1,23) = .55, *p* = .47.

Amount of Interaction

We compared the amount of user interaction with each of the interfaces. In the DM interface, an interaction is performed either by dragging the mouse to pan the view in order to bring up more images (number of pans), by zooming in or out (number of zooms), or by double-clicking on a single image to bring it to the center. In the RF interface, an interaction translates into a “more images” or “back” press which brings up the next or previous set of images (number of presses). Thus, we compared the number of pans + zooms + double clicks in the DM interface with the number of combined “more” and “back” presses in the RF interface. Results indicate that there were many more interactions per task in the DM interface (*M* = 158.4, *SD* = 93.3) than in the RF interface (*M* = 35.9, *SD* = 31.4). A one-way within subject ANOVA on number of interactions showed these differences were significant, F(1,23) = 90.9, *p* < 0.001.

Amount of images seen

Analyzing the log files, we summed up the amount of images seen in each interface. We examined both the total amount of images seen in a specific task, and the total amount of unique images seen, since some images may appear several times during the same task. With the RF interface, the total amount of images seen is equal to the number of interactions (as listed above) plus 1 (for the initial screen) times 30 (each screen showed a grid of 6×5 images). In the DM interface, each pan adds a different amount of images to the screen depending on the pan position. We counted the 30 initial images, and then added the newly filled images in each pan. A zoom, restart, or doubleClick event brought 30 more images. For the unique images seen, in both interfaces, we counted the unique images presented from the beginning till the end of the task. Because there were large individual differences in task completion time, we normalized these results over time and measured the total number of unique and non-unique images seen per minute. Results, presented in Table 1, indicate a large, significant difference in both total and unique number of images seen per minute. Users using the DM interface have seen significantly more total images per minute than when using the RF interface, F(1,31) = 98.2, *p*<0.001. Users using the DM interface have also seen more unique images per minute than users using the RF interface, F(1, 23) = 107.6, *p* < 0.001.

	DM	RF	F	p
Task completion time (seconds)	805.8 (334.4)	761.5 (348.9)	0.55	0.47
Average number of images seen per minute	230.0 (51.9)	104.5 (51.0)	98.2	<0.001
Average number of unique images seen per minute	98.4 (21.1)	49.3 (18.6)	107.6	<0.001

Table 1. Task completion time and average number of unique and non-unique images seen per-minute (and standard deviation) per interface

Zooming

All participants used the zooming feature often, with an average of 39.6 times per session (or 2.94 zoom events per minute). To better understand the usage of DynamicMaps, we analyzed the use of the zooming levels. Figure 4 shows the number of pans made and number of (non-unique) images seen in each zoom level. As can be expected, most interaction was done in the first zoom level, with interaction dropping heavily after the first level.

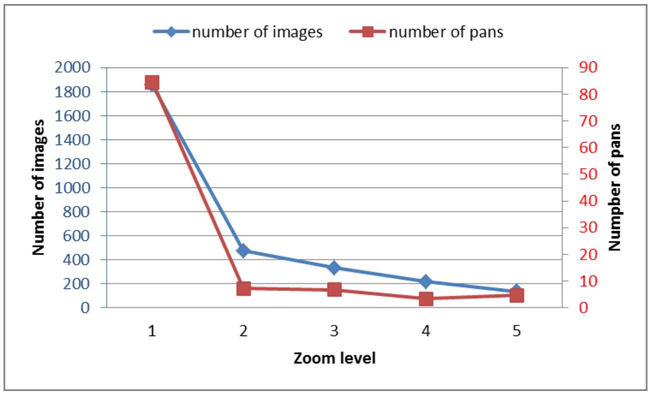

Figure 4. Average number of pans made and number of images seen in each of the five zoom levels (level 1 being images that are most similar to each other).

Subjective opinions

After using each interface, participants were presented with a set of statements and were asked how much they agreed with each one on a 7-point Likert scale ranging from strongly disagree (1) to strongly agree (7). Table 2 presents these statements and the visitors’ responses with both interfaces. A Wilcoxon Signed Ranked non-parametric test did not find significant differences in ranking of any of the statements between the two interfaces.

At the end of the experiment, we presented participants with a final questionnaire asking them for their preference of interface on a list of criteria (Table 3). Results indicate a

Statement	DM	RF
The system was efficient for the search tasks	4.20 (1.14)	4.15 (1.22)
The system limited my options	4.75 (1.69)	4.91 (1.28)
The search was fun	4.63 (1.24)	4.33 (1.16)
I quickly understood how to use the interface	5.87 1.19)	5.87 (1.11)
During the search I felt frustrated	3.25 (1.64)	3.54 (1.84)
I am satisfied with the images I picked for the presentation	5.10 (1.25)	4.83 (1.00)

Table 2. Participants average ratings (and standard deviation) per interface on a 7-point Likert scale (N=24).

general preference toward the DM interface, although preference was not absolute. Most participants thought the DM interface was more efficient and preferred it overall. It is interesting to note that most participants thought the DM interface is better when there is a vague idea of the search target and for seeing a wide variety of images, while there was a general preference for RF when the target is clear.

DISCUSSION

Our results indicate that DynamicMaps provide a more interactive experience for the users and allows them to view a wider variety of images than previous methods. Participants viewed many more images (both unique and non-unique) per time with the DM interface compared to the RF interface. While the way of interacting in the two conditions is quite different, still, many more interaction events were measured in the DM compared to the RF interface. Thus, it seems that participants viewed more images by actively interacting more with the interface. It should be noted that we cannot be sure that participants actually saw all the images that were displayed on screen. RF actually forces the user to more closely examine each image, while DM better supports scanning through images. This may help to explain the large difference in the amount of presented images.

Statement	DM	RF	No pref
Which system was more efficient?	12	8	4
When you have a vague idea of the search target, which system is better?	15	7	2
When the search target is clear, which system is better?	7	15	2
Which system is best to see a wide variety of images?	16	5	3
Which system is easier to learn?	6	7	11
Which system do you prefer overall?	15	8	1

Table 3. Number of participants preferring each interface on a list of criteria (N=24).

DynamicMaps provides immediate and continuous interactive feedback that does not require the user to make conscientious sequential selections, but rather asks the user to visually choose a direction to follow based on general perceptive cues. Thus, it affords easier and faster movement in the image space, with less sense of commitment, enabling the user to see a wider variety of images (a fact also realized by participants in the subjective preference questionnaires). This can also be look at from a cognitive load perspective. Cognitive load in the information retrieval context can be seen as a measure of information processing effort a user expends to comprehend the visual stimuli and interact with the system [15]. Using the RF interface, the user needs to go over every image and explicitly provide a relevance judgment on the image, a process that requires a high state of cognitive load [2]. DynamicMaps is less cognitively demanding since the user does not need to make a decision regarding each and every image, but can rather follow general visual cues. As one participant wrote "*I prefer DM. Less mouse clicking. Dragging is easier then thinking of which images will bring me closer. In DM you can see a larger range of images at once without the need to choose and click over and over*".

Having easier interaction capabilities and viewing more images per time unit is more useful when the search is vague and it might be difficult to select specific images that lead directly to the target. It is then easier to experiment, and follow one or more visual search directions than to select specific images. Another advantage of faster and more interactive browsing is that it can better support serendipity in the search process, since users interact more and may stumble upon different areas. It is easy for users to explore regions they may not have envisioned. This was reflected in a statement of a participant: "*It is possible to reach different directions, thoughts and ideas that I have initially not thought about*".

Zooming was often used and was referred to by participants as being very useful. The Zooming option enabled the users to step back and get a wider view of the current corpus. It also supports getting a more diverse view, with the diversity level controlled by the user. Furthermore, using zoom out and pan, the user can view the different topics and content available in the current corpus using simple interactions. This can be useful to get an overview of the image corpus.

No overall significant difference between the interfaces was found for task completion time. Completion time is often looked at as a measure of efficiency. However, in the current study, the task was open-ended and participants were asked to take as much time as needed to find the best possible images. Thus, we do not think that in this case completion time is an indicator of efficiency or quality. On the contrary, it might be that more time spent on the task indicates that the interface was more engaging and caused users to search more thoroughly. Similarly, other studies have found no correlation between task completion time and quality of results or user satisfaction [24].

Finally, we note that many participants mentioned that DynamicMaps was enjoyable and the interaction with it was much more smooth and fun to use than the RF interface (e.g., "*The [DM] system is enjoyable, it is easy to operate and it naturally flows*"). We believe that this will be highlighted even more when using the system with touch-based interfaces. With its pan-based interaction, DynamicMaps should be ideal for searching images on a Tablet computer, for which the playfulness of DynamicMaps would be even more prominent.

Limitations and merits of the local solution

One of the most prominent features of our approach is locality. The k-nearest neighbors technique and the greedy map generation process are critical for accomplishing scalability, and as a result we are able to seamlessly work with massive data sets. However, there are certain attributes, such as distance, which are not preserved, and altogether from an algorithmic point of view, our solution technique does not seek a global solution of a well-defined optimization problem. Rather, we navigate through low dimensional representations of the image space, and rely on user input to scan through the images. A potential limitation of this approach is that it is difficult to assess the quality of the outcome, and we cannot rigorously prove optimality or near optimality. Work in the future will include further investigation of the effects of locality.

On the other hand, the local solution brings with it several strong merits: it is computationally inexpensive, highly scalable, flexible and dynamic. The local nature of the algorithm yields an efficient computational procedure. Computing a global solution may be computationally prohibitive and may present an over-constrained problem that leads to many conflicts, resulting in a solution that is not necessarily better than the one obtained by a local search. When considering massive sized image repositories, a computationally inexpensive approach which nonetheless produces high quality results is critical. DynamicMaps is also dynamic, and can easily handle frequent changes in the dataset. The local nature of the algorithm allows for a seamless addition of images, and other on-the-fly changes. This cannot easily be accomplished by other global feature-preserving techniques.

Another limitation of our system is that it may be difficult to find non-dominant concepts or particular images. If a concept rarely appears, the user will be unlikely to find it as it will be hidden within another area. This can be partly addressed by relaxing the stability of view allowing an image to appear multiple times in different contexts. However, a complete solution for this issue would have to involve combining our system with direct or faceted search.

Combining browsing with direct search

For browsing to be effective it needs to be complementary to direct keyword search, since using keywords is still the preferred method in image retrieval systems [11]. Furthermore, in order to browse, the user needs a way to approach the image space, which can be best done with direct search. One way to combine DynamicMaps with keyword search is by starting with a regular grid of images triggered by keyword search (such as provided by Google images), and allowing the user to choose an image around which the Map will be created to start browsing from. This is similar to how relevance feedback ("similar images") is combined with keyword search today. Thus, keyword search can be used to first reach an area of interest, and then the search can be refined using browsing according to similarity (similar to the orienteering strategy [35]). Another way, which we started experimenting with, directly embeds keyword search within the current DynamicMaps interface. The user types in the search query, and the system presents the results (that match the query) already ordered according to similarity. The challenge lies in seamlessly extending browsing to images that are not directly related to the search query.

ACKNOWLEDGEMENTS

We would like to thank Ofer Arazy for his insightful comments during the writing of the paper.

CONCLUSIONS

We have presented DynamicMaps, a system for browsing a very large set of images on a 2D grid according to similarity using a metaphor of an infinite canvas. DynamicMaps enables a smooth, fast and more interactive experience that is best suitable for exploratory search, when the search target is vague. It is also useful for serendipitous browsing in exploring image regions not envisioned by the user and for getting a wider view of the image corpus. Further work would explore using semantic information in the similarity measures as well as combine DynamicMaps with keyword search.

REFERENCES

1. André, P., Cutrell, E., Tan, D. S., & Smith, G. Designing novel image search interfaces by understanding unique characteristics and usage. *INTERACT* '09 (2009), 340-353.
2. Back, J., & Oppenheim, C. A model of cognitive load for IR: implications for user relevance feedback interaction. *Information Research 6* (2) (2001).
3. Bederson, B. B. PhotoMesa: a zoomable image browser using quantum treemaps and bubblemaps. *In Proc. UIST 2001*, ACM Press (2001), 71-80.
4. Borg, I. and Groenen, P. *Modern multidimensional scaling: Theory and applications*. Springer, 2005.

5. Chen, C., Gagaudakis, G., & Rosin, P. Similarity-based image browsing. *Int. conference on intelligent information processing* (2000), 206-213.
6. Chew, B., Rode, J. A., & Sellen, A. Understanding the everyday use of images on the web. *In Proc. NordiCHI'10.* ACM Press (2010), 102-111.
7. Chung, E., & Yoon, J. Image needs in the context of image use: An exploratory study. *Journal of Information Science* (2011).
8. Combs, T. T., & Bederson, B. B. Does zooming improve image browsing? *In Proceedings of Digital libraries '99*. ACM Press. (1999), 130-137.
9. Datta, R., Li, J., & Wang, J. Z. Content-based image retrieval: approaches and trends of the new age. *In Proc. of SIGMM*. ACM Press (2005), 253-262.
10. Deselaers, Thomas, Daniel Keysers, and Hermann Ney. Features for image retrieval: an experimental comparison. *Information Retrieval 11*, 2 (2008), 77-107.
11. Eakins, J. P., Briggs, P., & Burford, B. Image retrieval interfaces: A user perspective*. In Image and Video Retrieval,* Springer Berlin Heidelberg (2004). 628-637.
12. Fan, J., Keim, D. A., Gao, Y., Luo, H., Li, Z. JustClick: Personalized image recommendation via exploratory search from large-scale Flickr images. *IEEE Transactions on Circuits and Systems for Video Technology*, *19*, 2 (2009), 273-288.
13. Hearst, M. *Search user interfaces*. Cambridge University Press. 2009.
14. Hoque, E., Hoeber, O., & Gong, M. Balancing the trade-offs between diversity and precision for web image search using concept-based query expansion. *Journal of Emerging Technologies in Web Intelligence*, *4*, 1 (2012) 26-34.
15. Hu, P. J. H., Ma, P. C., & Chau, P. Y. Evaluation of user interface designs for information retrieval systems: a computer-based experiment. *Decision Support Systems, 27*, 1 (1999) 125-143.
16. Igarashi T, Hinckley, K., Speed-dependent automatic zooming for browsing large documents, *Proceedings of UIST 2000.* ACM Press (2000), 139-148.
17. Jing, Y., Rowley, H., Wang, J., Tsai, D., Rosenberg, C., & Covell, M. Google image swirl: a large-scale content-based image visualization system*. Proceedings of WWW.* (2012), 539-540.
18. Kohonen, T. *Self-organizing maps* (Vol. 30). Springer. 2001.
19. Kleiman, Y., Fish, N., Lanir, J., & Cohen-Or, D. Dynamic maps for exploring and browsing shapes. *Computer Graphics Forum 32*, 5 (2012), 187-196.
20. Liu, H., Xie, X., Tang, X., Li, Z. W., & Ma, W. Y. Effective browsing of web image search results. *In Proc of SIGMM*. ACM Press (2004), 84-90.
21. Markkula, M., & Sormunen, E. End-user searching challenges indexing practices in the digital newspaper photo archive. *Information retrieval*, *1*, 4, (2000) 259-285
22. Oliva, A., and Torralba, A. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision 42*, 3 (2001) 145–175.
23. Pečenovic, Z., Do, M. N., Vetterli, M., & Pu, P. Integrated browsing and searching of large image collections. *In Advances in Visual Information Systems.* Springer Berlin Heidelberg. (2000). 279-289.
24. Rodden, K., Basalaj, W., Sinclair, D., & Wood, K. Does organization by similarity assist image browsing? *In Proc CHI.* ACM Press (2001). 190-197
25. Rodden K, Basalaj D W Sinclair, Wood K. Evaluating a visualization of image similarity as a tool for image browsing. *IEEE symposium on information visualization* (1999) 36–43
26. Roweis, S., and Saul, L. Nonlinear dimensionality reduction by locally linear embedding. *Science 290*, 5500, (2000), 2323–2326.
27. Rui, Y., Huang, T. S., Ortega, M., & Mehrotra, S. Relevance feedback: a power tool for interactive content-based image retrieval. *Circuits and Systems for Video Technology, IEEE Transactions on*, *8,* 5 (1998), 644-655.
28. Ruthven, I., & Lalmas, M. A survey on the use of relevance feedback for information access systems. *The Knowledge Engineering Review*, *18*, 2 (2003) 95-145
29. Sakamoto, Y., Kuriyama S., and Kaneko, T. Motion map: image-based retrieval and segmentation of motion data. *In Proc SIGGRAPH/ Eurographics symposium on Computer animation*, (2004).
30. Shatford-Layne, S. Some issues in the indexing of images. *Journal of the American society of Information Science*. *45*, 8 (1994) 583-588
31. Shneiderman B. *Designing The User Interface: Strategies for Effective Human-Computer Interaction*, 4/e. Pearson Education India, 2003.
32. Strong, G., Gong, M.. Browsing a large collection of community photos based on similarity on GPU. *Advances in Visual Computing*. (2008), 390–399.
33. Strong, G., Hoeber, O., & Gong, M. Visual image browsing and exploration (Vibe): User evaluations of image search tasks. *Active Media Technology.* Springer Berlin Heidelberg (2010), 424-435.
34. Suditu, N., & Fleuret, F. HEAT: Iterative relevance feedback with one million images. *In ICCV 2011* IEEE (2011), 2118-2125.
35. Teevan, J., Alvarado, C., Ackerman, M. S., & Karger, D. R. The perfect search engine is not enough: a study of orienteering behavior in directed search. *In Proc CHI 2004*. ACM Press (2004), 415-422
36. Yee, K. P., Swearingen, K., Li, K., & Hearst, M. Faceted metadata for image search and browsing. *In Proc CHI 2003*. ACM Press (2003), 401-408.
37. Zhou, X. S., & Huang, T. S. Relevance feedback in image retrieval: A comprehensive review. *Multimedia systems*, *8*, 6 (2003), 536-544.