

Computer Science Technical Report

iMap - A Stable Layout for Navigating Large Image Collections with Embedded Search

Chaoli Wang, John P. Reese, Huan Zhang, Jun Tao, Robert J.
Nemiroff

Michigan Technological University
Computer Science Technical Report
CS-TR-12-01
January 11, 2012

MichiganTech

Department of Computer Science
Houghton, MI 49931-1295
www.cs.mtu.edu

iMap - A Stable Layout for Navigating Large Image Collections with Embedded Search

Chaoli Wang, John P. Reese, Huan Zhang, Jun Tao, Robert J. Nemirow

January 11, 2012

Abstract

Effective techniques for organizing and visualizing large image collections are in growing demand as visual search gets increasingly popular. We present our solution for image search and clustering based on the evaluation image similarity using both visual and textual information. To lay out images, we introduce iMap, a treemap representation for visualizing and navigating image search and clustering results. iMap not only makes effective use of available display area to arrange images but also maintains stable update when images are inserted or removed during the query. We also develop an embedded visualization that integrates image tags for in-place search refinement. Using an online astronomy archive with thousands of images, we show the effectiveness of our approach by demonstrating experimental results and conducting a comparative user study.

iMap - A Stable Layout for Navigating Large Image Collections with Embedded Search

Chaoli Wang, John P. Reese, Huan Zhang, Jun Tao, Robert J. Nemiroff

Abstract—Effective techniques for organizing and visualizing large image collections are in growing demand as visual search gets increasingly popular. We present our solution for image search and clustering based on the evaluation image similarity using both visual and textual information. To lay out images, we introduce iMap, a treemap representation for visualizing and navigating image search and clustering results. iMap not only makes effective use of available display area to arrange images but also maintains stable update when images are inserted or removed during the query. We also develop an embedded visualization that integrates image tags for in-place search refinement. Using an online astronomy archive with thousands of images, we show the effectiveness of our approach by demonstrating experimental results and conducting a comparative user study.

Index Terms—Image layout, large image collection.



1 INTRODUCTION

With the booming of digital cameras and image archiving and photo sharing websites, browsing and searching through large online image collections is becoming increasingly popular. An emerging trend is that images are now often tagged with names, keywords, hyperlinks and so on to improve the search and understanding. In this paper, we strive for innovation on the organization and interaction aspects of image search rather than the search algorithm itself. Specifically, we explore how to arrange images in a layout for better viewing and how to leverage the connection between image and text for better interaction.

Many existing applications provide overviews of image collections by presenting a set of thumbnail images arranged in a spreadsheet-like interface. We advocate a more attractive way for image browsing to enable effective sense-making of large image collections through image ranking and clustering, and intuitive presentation and interaction. We analyze image content and design measures to evaluate their similarity using both visual and textual information. We arrange similar images close to each other and leverage a treemap representation to visualize image search and clustering results to facilitate the understanding. This visual means brings several benefits such as effective screen space utilization, occlusion minimization, and stable update. In addition, we develop an embedded visualization that integrates image tags for in-place search refinement.

We experiment with our approach using the Astronomy Picture of the Day (APOD) [5], a popular online astronomy archive. Everyday APOD features a picture of our universe, along with a brief explanation written by a professional astronomer. Since its debut in June 1995, APOD has archived thousands of handpicked pictures, which makes it the largest collection of *annotated* astronomical images on the Internet. This makes it perfect for us to include textual information into similarity analysis and interaction design. Our work complements the state-of-the-art image search and exploration techniques with new interface and interaction that enable effective sense-making of a large image collection and guide users to sift through the collection and identify images of interest.

2 IMAGE DISTANCE MEASURE

Measuring the similarity or distance between two images is central to any image searching or clustering tasks. Images themselves provide direct cues to visual comparison. Textual information associated with images, whenever available, gives additional hints for us to evaluate their similarity or difference. We therefore compare images using their visual and textual information from multiple perspectives and define five partial distances (\mathbb{D}_G , \mathbb{D}_F , \mathbb{D}_H , \mathbb{D}_K , and \mathbb{D}_L). The overall distance measure is a weighted summation of all the partial distances.

2.1 Visual Distance

Different images in an image collection come with different dimensions, types, and formats. For simplicity, we convert all resulting images to the same type and format, and scale them down to a fixed resolution (256×256). We consider three aspects of images, namely, grayscale image distance, spectrum image distance, and color histogram distance, for calculating their visual distance. Note that our solution to visual similarity analysis is by

-
- Chaoli Wang, John P. Reese, Huan Zhang, and Jun Tao are with the Department of Computer Science, Michigan Technological University, Houghton, MI 49931. Email: {chaoliw, jpreese, huanz, junt}@mtu.edu.
 - Robert J. Nemiroff is with the Department of Physics, Michigan Technological University, Houghton, MI 49931. Email: nemiroff@mtu.edu.

no means ideal. Rather, we seek a cost-effective solution to serve the basic visual search need.

Grayscale Image Distance. Intuitively, the similarity between two images can be evaluated by identifying their structural similarity. The structural similarity index proposed by Wang et al. [9] considers *luminance*, *contrast*, and *structure* information of the two images. We use the grayscale version of images for this evaluation and will consider color information separately when computing the color histogram distance. Given two grayscale images, we take a local 8×8 window, which moves pixel-by-pixel over the entire image, to evaluate their similarity. For two corresponding image blocks B_a and B_b , we compute their similarity as

$$\mathbb{S}_B(B_a, B_b) = \frac{(2\mu_a\mu_b + c_1)(2\sigma_{ab} + c_2)}{(\mu_a^2 + \mu_b^2 + c_1)(\sigma_a^2 + \sigma_b^2 + c_2)}, \quad (1)$$

where μ_a and μ_b are the means of B_a and B_b respectively, σ_a and σ_b are the standard deviations of B_a and B_b respectively, and σ_{ab} is the covariance of B_a and B_b . Small constants c_1 and c_2 are included to avoid instability when μ_a , μ_b , σ_a , and σ_b are very close to zero. We set $c_1 = (0.01 \times L)^2$ and $c_2 = (0.03 \times L)^2$ where L is the number of levels in the grayscale images. We define the distance between two grayscale images G_a and G_b as

$$\mathbb{D}_G(G_a, G_b) = 1.0 - \frac{1}{m} \sum_{i=1}^m \bar{\mathbb{S}}_B(B_{ai}, B_{bi}). \quad (2)$$

where m is the total number of image blocks considered and $\bar{\mathbb{S}}$ denotes the normalized similarity value.

Spectrum Image Distance. The power spectrum of an image is a representation of the magnitude of its various frequency components that has been transformed using the Fourier transform. The power at each location in the spectrum indicates the frequency and orientation of a particular feature in the image. We use the grayscale version of the spectrum image after the log transform. Given two grayscale frequency spectrum images F_a and F_b , we compute their similarity through evaluating their block-wise Pearson linear correlation. Again, we take a local 8×8 window, which moves pixel-by-pixel over the entire image. For two corresponding image blocks B_a and B_b , we compute their correlation as

$$\mathbb{P}_B(B_a, B_b) = \frac{1}{n} \sum_{j=1}^n \left(\frac{p_{aj} - \mu_a}{\sigma_a} \right) \left(\frac{p_{bj} - \mu_b}{\sigma_b} \right), \quad (3)$$

where p_{aj} and p_{bj} are the j -th pixel values of B_a and B_b respectively, μ_a and μ_b are the means of B_a and B_b respectively, σ_a and σ_b are the standard deviations of B_a and B_b respectively, and n is the number of pixels in the block. With Equation 3, we compute the distance $\mathbb{D}_F(F_a, F_b)$ similar to Equation 2 where the *absolute* correlation values $|\mathbb{P}_B(B_{ai}, B_{bi})|$ are used instead.

Color Histogram Distance. Image colors provide additional information for similarity comparison. Given an image, we compute its color histogram by sampling each of the R, G, B channels into eight levels, which leads to a color histogram of 512 entries. Given two normalized

color histograms H_a and H_b , we use the Kullback-Leibler divergence (KLD) to evaluate their difference

$$\mathbb{K}_H(H_a || H_b) = \sum_{k=1}^b h_a(k) \log \frac{h_a(k)}{h_b(k)}, \quad (4)$$

where $h_a(k)$ and $h_b(k)$ are the heights of the k th bin in H_a and H_b respectively, and b is the number of bins in the color histogram.

Notice that the KLD is not a true metric, i.e., $\mathbb{K}_H(H_a || H_b) \neq \mathbb{K}_H(H_b || H_a)$. We thus use the symmetric Jensen-Shannon divergence (JSD) measure instead

$$\mathbb{D}_H(H_a, H_b) = \frac{\mathbb{K}_H(H_a || H_m) + \mathbb{K}_H(H_b || H_m)}{2}, \quad (5)$$

where $H_m = (H_a + H_b)/2$.

2.2 Textual Distance

To obtain textual features, we extract meta-tagged keywords in the HTML header. We also extract hyperlinks in the explanation. These hyperlinks refer to URLs and similar hyperlinks indicate that their corresponding images may also share similarity. Then, we convert all uppercase letters to lowercase ones for extracted keywords and hyperlinks and apply the bag-of-words model [4] for textual similarity measurement. We point out that semantic-based methods for detecting text similarity [3] can classify text based on the same semantic focus such as an object or action. Nevertheless, the simple keyword extraction and similarity evaluation technique we propose performs well as all images in our case share the same theme of astronomy.

Keyword List Distance. We treat two given keywords k_a and k_b as strings and calculate their edit distance by computing the minimum number of inserts, deletes, and substitutions required to transform one string into the other. The distance between k_a and k_b is calculated as

$$\mathbb{D}_k(k_a, k_b) = \frac{c_i n_i + c_d n_d + c_s n_s}{|k_a| + |k_b|}, \quad (6)$$

where c_i , c_d , and c_s are the costs of insertion, deletion, and substitution operations respectively, n_i , n_d , n_s are the numbers of times that these three operations occur respectively, and $|k_a|$ and $|k_b|$ are the lengths of keywords k_a and k_b respectively. We set $c_i = 2.0$, $c_d = 2.0$, and $c_s = 3.0$ since the cost of deletion or insertion operation must be greater than half of the cost of substitution operation. Otherwise, two strings k_a and k_b will be matched using the deletion and insertion operation only.

To calculate the distance between two lists of keywords, we take into account all keyword pairs from the two lists. Our solution is to calculate the average of the *minimum* distance values of keyword pairs, i.e.,

$$\mathbb{D}_K(K_a, K_b) = \frac{\mathbb{A}_K(K_a, K_b) + \mathbb{A}_K(K_b, K_a)}{2}. \quad (7)$$

where $\mathbb{A}_K(K_a, K_b)$ is the average of the minimum distances (Equation 6) for each keyword in list K_a to any keyword in list K_b . All \mathbb{D}_K are normalized for use.

Hyperlink List Distance. Given two hyperlinks, we compute their similarity by first determining if we are comparing an *internal* link (i.e., a URL in the same website) to an *external* link (i.e., a URL in a different website). For simplicity, we only check these links not the actual content these links refer to. If the two hyperlinks are internal and external links, we define their similarity as 0. If both links are internal, the similarity is 1 if they match exactly; otherwise, the similarity is 0. If both links are external, we ignore any directories and only take into account each hyperlink’s subdomains. Each hyperlink is first split into a list, where each element in the list is a subdomain (for example, `www.nasa.gov` is split into the list `www`, `nasa`, and `gov`). We then compare each subdomain in the first hyperlink to each subdomain in the second hyperlink starting from the end of each list and moving backwards. At any step, if two subdomains being compared do not match exactly, then we do not proceed further. Let l_a be a hyperlink with $|l_a|$ subdomains, l_b be a hyperlink with $|l_b|$ subdomains, and n be the number of subdomains matched. We define the similarity between two hyperlinks l_a and l_b as

$$\mathbb{S}_l(l_a, l_b) = \frac{n}{\max(|l_a|, |l_b|)}. \quad (8)$$

To find the similarity between two lists of hyperlinks, we use the average of the *maximum* similarity values, i.e.,

$$\mathbb{S}_L(L_a, L_b) = \frac{\mathbb{A}_L(L_a, L_b) + \mathbb{A}_L(L_b, L_a)}{2}. \quad (9)$$

where $\mathbb{A}_L(L_a, L_b)$ is the average of the maximum similarity values (Equation 8) for each hyperlink in list L_a to any hyperlink in list L_b . We normalize all similarity values to $[0, 1]$ and define the distance between L_a and L_b as $\mathbb{D}_L(L_a, L_b) = 1.0 - \mathbb{S}_L(L_a, L_b)$.

3 IMAGE RANKING AND CLUSTERING

With the overall image distance defined, we build a symmetric distance matrix recording the distance between any two images in the collection. During image search, the user selects a query image and all other images in the collection are ranked accordingly. The user can change the weights for partial distances to update the distance matrix and search results.

For image clustering, we apply the hierarchical quality threshold algorithm due to its simplicity and efficiency. The algorithm uses a list of distance thresholds with increasing values $\{\delta_0, \delta_1, \delta_2, \dots, \delta_l\}$ to create a hierarchy up to $l + 1$ levels ($\delta_0 = 0$, $\delta_l = 1$). Initially, each image in the collection is in its own cluster. At the first iteration, we build a candidate cluster for each image I by including all images that have their distance to I smaller than δ_1 . We save the cluster with the largest number of images as the first true cluster and remove all images in this cluster from further consideration. In the true cluster, image I is treated as its representative. We repeat with the reduced set of images until all images are classified. In the following iterations, the input is

all representative images gathered from the previous iteration. We continue this process for the following iterations until we finish the l th iteration or until we only have one cluster left in the current iteration.

4 IMAGE LAYOUT AND INTERACTION

Once image are ranked, image layout is important as it determines how images should be arranged for viewing. For a large image collection, it is desirable to maintain good visual overview while allowing flexible exploration and detailed examination. An appropriate image layout should fulfill the following criteria:

- *stable layout*: the layout should accommodate image ordering and maintain stable update when images are inserted or removed during the query;
- *screen utilization*: for efficiency, the layout should display as many images as users can comfortably view them;
- *occlusion minimization*: for effectiveness, images displayed should not occlude each other in the layout or their overlap should be minimized;
- *in-place interaction*: the layout itself should also serve as an interface for in-place interaction to attract user attention and facilitate image identification.

4.1 iMap

Since most displays and images are in the form of rectangle, we opt to use the rectangular shape for image layout so that the available display area can be best utilized. We propose to use the treemap [6] to visualize a large image collection due to its simplicity and effectiveness. We refer to the treemap representation of image collections as the *iMap* (“i” stands for image). Each node in iMap corresponds to a rectangle that displays an image thumbnail. The sizes of these rectangles can be determined by the importance of their images, such as search rank or hit count.

Layout Design. The original “slice-and-dice” treemap layout generates rectangles of arbitrary aspect ratios. Squarified treemaps [2] create rectangles with smaller aspect ratios but give up on node ordering. Ordered treemaps [7] offer a good tradeoff among stable updates, order preserving, and low aspect ratios. Quantum treemaps developed for PhotoMesa [1] guarantee that the regions showing groups of photos have dimensions that are integer multiples of the dimensions of the photos (i.e., they must be sized to contain quantum, or indivisible contents). Spiral treemaps [8] place nodes along the spiral pattern which guarantees that neither the overall pattern nor the specific node ordering will change drastically as the data evolve over time.

For iMap, we propose a hybrid layout that combines the advantages of both quantum and spiral treemaps: the use of quantum with a fixed aspect ratio simplifies the layout for images of different aspect ratios; and the adoption of spiral pattern maintains stable update when

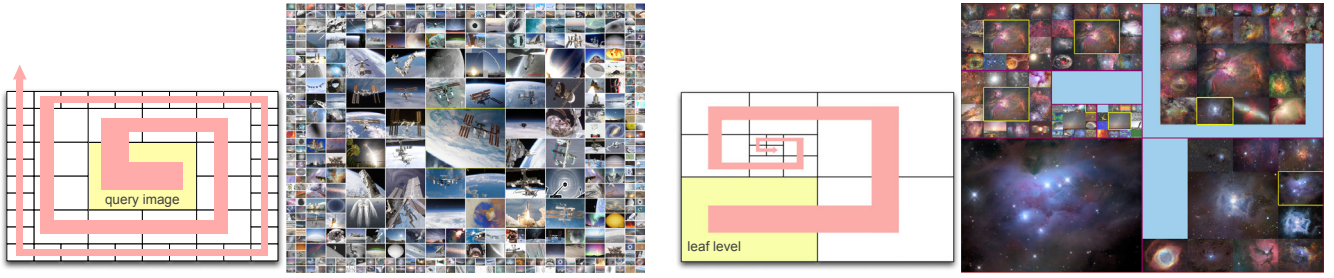


Fig. 1. iMap with the spiral layout. Left: image search results. Right: hierarchical clustering results.

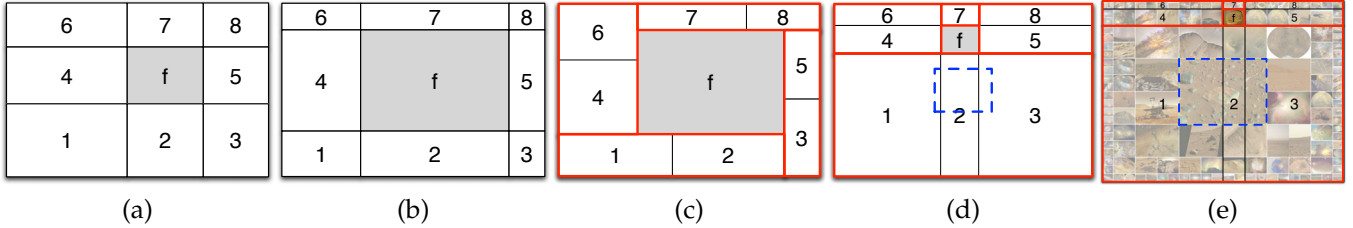


Fig. 2. F+C Visualization. (a) edge expansions of the focus divide the entire space into eight areas. (b) adjust each area independently. (c) adjust areas by group. In (d) and (e), one image, shown in the dashed rectangle, crosses three areas.

we insert or remove images. The 1D spiral also accommodates image ordering, such as the chronological order or rank order. Our layout results with image search and clustering are given in Figure 1. To organize clustering results, we apply a two-level layout: the quad-tree layout for different levels of hierarchy and the spiral layout for images within each level. Much as in human vision, we display the focus image at the center of display area (focal point) and arrange less important images in its surrounding (periphery). By default, we display the focus image in normal size and reduce the width and height by half for the successive layers. The user can adjust the number of repetition layers r needed to keep the current width and height. In Figure 1, we set $r = 2$ for image search and $r = 1$ for image clustering.

iMap Interaction. The interaction with image search function includes the following: left mouse click for showing the overlay full-resolution image in its original aspect ratio; middle mouse click for showing metadata information; and right mouse click for updating search with respect to the image selected. Each image comes with an ID and we also store the number of pixels similar to a list of given representative colors. This allows us to enable “search by ID” (selecting an image ID via slider) and “search by color” (picking a representative color via radio button) functions for updating the query image besides “search by image” (directly clicking on an image displayed in iMap). The interaction with image clustering function is the same except that right mouse click on an image is for displaying its next level of images in the hierarchy.

We also implement a focus+context (F+C) function to further improve the readability of iMap as the user mouses over the images. Our F+C approach aims to apply a simple strategy to achieve an acceptable result for a single focus. As shown in Figure 2, in a general case,

the expansions of the four edges of the focused image divide the entire space into eight areas. When the focus is enlarged, its edges move accordingly, so do the edges of the eight areas. However, if we simply scale each area to fit into the new layout, the four areas adjacent to the focus, i.e., Areas 2, 4, 7, and 5, will suffer from more serious distortion than other areas (Figure 2 (b)). Taking Areas 1 and 2 for example, when the focus grows, the horizontal edges of Area 2 will expand while its vertical edges shrink; but for Area 1, both its horizontal and vertical edges will shrink. Thus, the aspect ratio of Area 2 changes more significantly than that of Area 1. To balance these changes, we group each area adjacent to the focus to the one at the corner, i.e., Areas 1 and 2, 3 and 5, 4 and 6, 7 and 8 will be grouped together, respectively. Then we apply the transformation to each group, so that the overall distortion can be reduced (Figure 2 (c)). However, for the spiral layout, an image that is larger than the focus might cross three areas, as shown in Figure 2 (d). Figure 2 (e) shows an example where the current query image at the center crosses Areas 1, 2, and 3. In this case, Areas 1, 2, and 3 must be in one group to ensure that every image is still a rectangle. For other areas, we will group Areas 4 and 6, 5 and 8, respectively, and leave Area 7 ungrouped (since it is the smallest one).

In our implementation, the focused image will be scaled up to the same size as the centered query image before the deformation. To better preserve the original aspect ratio for the areas out of focus, the center of the focused image might move. Take the horizontal direction for example, the left and right boundaries of the focused image after deformation are decided in a way such that the left and right remaining areas are squeezed proportionally to their origin widths. Once the boundaries of the focused image are determined,

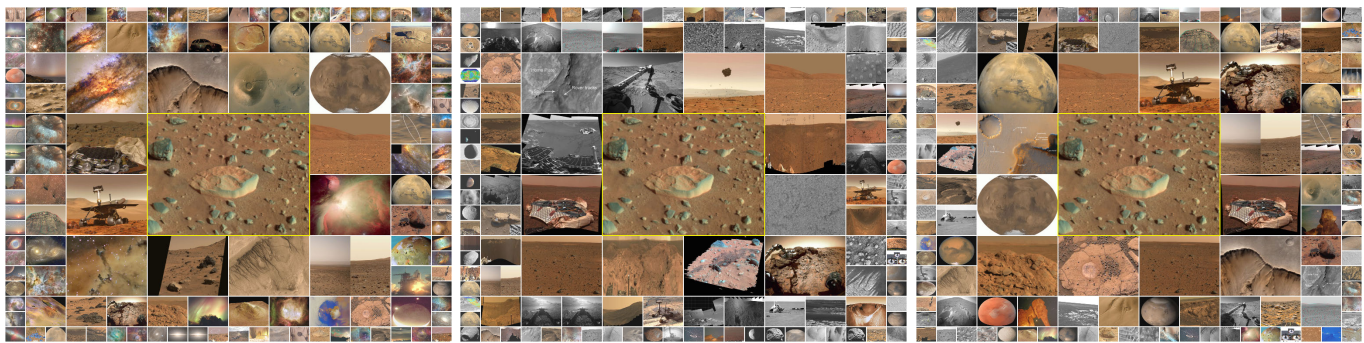
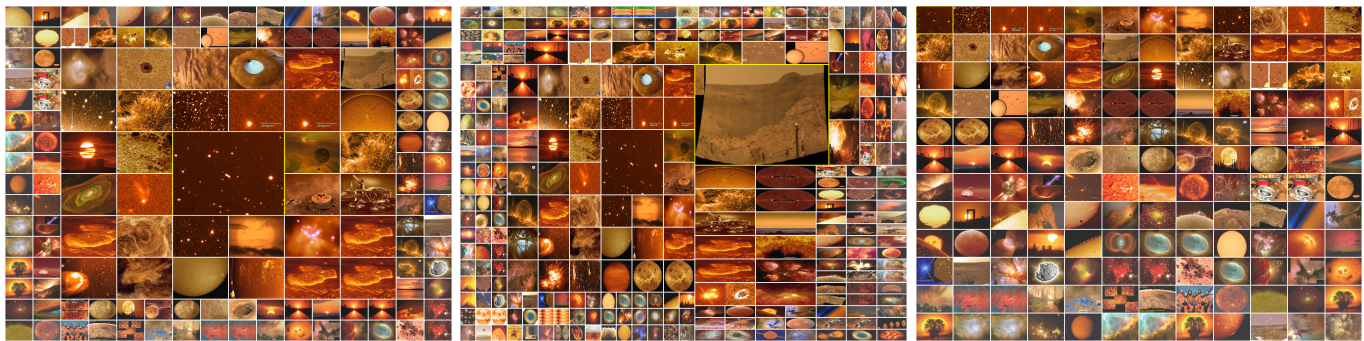


Fig. 3. Left to right: iMap showing search results with the use of visual distance only, textual distance only, and a combination of visual and textual distances, respectively.



(a) spiral, 5 layers, 145 images

(b) spiral, 7 layers, 289 images

(c) row-and-column, 144 images

Fig. 4. Query by color: images are ranked according to their percentages of brown pixels. F+C visualization is shown in (b) where the focused image is highlighted in the yellow boundary.

we compute the boundaries of each area and uniformly deform the context images within each area.

4.2 Integrating Text into Image Search

Building the overall distance matrix that includes the keyword and hyperlink distances for image search *implicitly* utilizes the textual information. In this case, all keywords or all hyperlinks associated with images must be taken into account. Nevertheless, these keywords or hyperlinks can also be *explicitly* utilized to customize or refine the search. We present three different ways to explicitly integrate text into image search.

Keyword Input. In this mode, the user inputs a keyword into a text widget and we search and display images that contain such a keyword. The keyword input could be partially or exactly matched with image keywords and/or text explanations in the HTML files. Multiple keywords are allowed with logical operators (OR, AND). Images founded can be ordered by their IDs or other attributes such as hit count. They can also be arranged according to their rank order from the previous image search results.

Tag Cloud. Unlike keyword input, tag cloud displays pages of keyword from which the user clicks on a keyword of interest to find related images. The order of keywords in the tag cloud can be determined by their alphabetical or frequency order. Their visual attributes such as size can be determined by their frequency.

Embedded Text. Both keyword input and tag cloud display text in another widget or window separate from iMap. Another different design is to embed image tags for *in-place* selection and search, which obviates the need to use a list of words separately. This concept is similar to PhotoMesa [1], a zoomable image browser (in terms of in-place interaction) and scented widgets [10] (in terms of improving navigation cues). Zoomable interfaces make navigation straightforward and avoid getting lost. Improving navigation cues lowers the cost structure of seeking and accessing information. We therefore advocate a solution that integrates tag information into iMap for further interaction and embedded search. For the APOD collection, tags are only associated with the image, not specific objects. We thus overlay a layer of tags for user selection and search refinement. Much as in tag clouds, the size, color, or order of tags for an image can be adjusted to provide additional hints such as how many images will match selected tags. The user can select multiple tags to add into the current search.

5 RESULTS

We collected APOD webpages that contain meta-tagged keywords (since Sep. 26, 1997) till a cut-off date (Apr. 3, 2010). Occasionally, APOD runs videos instead of images. In this experiment, we did not consider videos and therefore excluded those webpages from our collection. For images in the GIF format though, we extract



Fig. 5. Image search via keyword input. Left to right: the search results corresponding to keyword(s) “spiral”, “galaxy”, “spiral” OR “galaxy”, and “spiral” AND “galaxy”, respectively. There are 222, 461, 474, and 208 images matched from left to right, respectively. The first 145 images, ordered by their dates, are shown in each search.

mars moon saturn sun
 star formation spiral galaxy earth
 jupiter dust galaxy planetary nebula
 infrared x-ray emission nebula comet
 venus milky way spacecraft supernova
 reflection nebula aurora
 supernova remnant orion stars black hole
 nebula galaxies space shuttle cassini iss
 asteroid star cluster open cluster mercury
 globular cluster meteor lmc rings
 meteor shower galactic center star
 solar eclipse clouds cosmology telescope
 crater solar system pleiades water titan
 volcano sunset universe galaxy cluster

starscape starspot starspots statue of liberty statuses steins
 stellar collisions stellar encounter stellar evolution stellar flare
 stellar life cycle stellar nursery stellar wind stellar winds
 stephan's quintet stereo stereo image stereo mission
 stereo satellite stereogram stickney crater still life strom circles
 stonehenge stonehouse storm storm alley storms stormy story
 strange streak streak string theory structure sts-100 sts-101 sts-100
 sts-105 sts-108 sts-109 sts-112 sts-114 sts-115 sts-116 sts-117
 sts-119 sts-121 sts-122 sts-124 sts-125 sts-128 sts-130 sts-41-b sts-64
 sts-107 sts-109 sts-123 sts-126 sts-128 sts-98 subaru sublimation
 suborbital subury suitsat sulfur sulfuric acid summer
 summer solstice summer triangle sun sundog sundogs
 sun halo sun pillar sun-like star sundial sundial sundog
 sungrazer sunrise sunset sunspot sunspots
 super earth super-earths superbubble supergalactic wind supergiant
 supermassive black hole supernova supernova 1987a
 supernova 1994d supernova remnant
 supernova remnants supernovae superovas supersymmetry



Fig. 6. iMap and tag cloud. Left: the page of the most frequent keywords. Middle: a page of keywords in the alphabetical order. Right: all images that contain the keyword “sun” are ordered by their IDs and displayed.

the first frame as the representative of the entire image sequence. The resulting data set consists of 4560 images with text information including keywords, hyperlinks, and explanations extracted from the HTML files. At preprocessing, we computed five distance matrices (\mathbb{D}_G , \mathbb{D}_F , \mathbb{D}_H , \mathbb{D}_K , and \mathbb{D}_L) to record the distance for all pairs of images. These distance matrices are used at runtime to update the final overall distance.

Image Layout. Figure 1 shows iMap layout. For image search, the query image of the International Space Station is displayed at the center of iMap as the focus. Results based on both visual and textual distances are ranked and arranged along the spiral circling out. The effectiveness of image search can be verified by the similar images retrieved and displayed. For image clustering, the user explores the cluster hierarchy by clicking the image of interest which will be highlighted with a yellow bounding box, and its next level of detail is displayed. By displaying all levels of hierarchy currently explored, we give the user the freedom to jump between non-consecutive hierarchical levels during the exploration. In Figure 3, we can see that using only visual distance picks up images of similar brownish-yellow colors while using only textual distance pulls up the ranks for grayscale images related to the Mars. A combination of both visual and textual distances finds a good balance in between.

Figure 4 shows an example of “search by color” where images are ranked according to how much percentages of brown pixels they contain. Compared to the row-and-column layout, the spiral layout effectively high-

lights top-ranked images while maintaining stable update when the number of image layers changes. We also show the results with different numbers of layers displayed and the effect with F+C visualization. Note that our F+C strategy might result in larger distortion when the focused image is close to the corner. In this case, some divided areas could be small, whose size will change dramatically during the deformation. A more sophisticated F+C visualization for this kind of application remains an open problem. To keep the shape of each image as a rectangle, it will be challenging to minimize the distortion while preserving their relative positions without creating voids. The variation of image size in the spiral layout makes the problem more complicated, since those large images will greatly limit the possible moves we can take. Producing a smooth animated transition of the deformation could be even more difficult.

Image and Text. Figure 5 shows an example of interactive image filtering via keyword input. The user can choose either partial or exact keyword match in the search. Figure 6 shows the use of tag cloud in iMap. Tag clouds organize all keywords in a certain order and the user can go over pages to identify the keyword of interest. As an option, the frequency count for each keyword may also be displayed. When the user clicks on the keyword, iMap updates the search result and displays all images containing the selected keyword. Figure 7 shows an example where the user first searches images based on the use of visual and textual distances as usual. Upon identifying the images



Fig. 7. iMap with embedded search. Left: image search results using visual and textual distances. Middle/right: the user clicks on the keyword “space shuttle”/“columbia” from the embedded list for result filtering.

of interest, the user can then turn on the embedded keyword list associated with the query image to refine the search by clicking on a certain keyword. All images that do not contain the selected keyword will be filtered out from the query result while the rank order from the previous search is utilized to maintain the relative stable update. The embedded search allows users to perform in-place interaction to refine their search results without shifting their focus among different windows, which makes it easy for users to follow and take actions.

Performance. For the APOD data set, the one-time computation of the five distance matrices took about a few days to complete on a single Intel Xeon 2.0GHz CPU. The dominant timing was spent on calculating \mathbb{D}_G and \mathbb{D}_F where we computed \mathbb{S}_B (Equation 1) and \mathbb{P}_B (Equation 3) on a local 8×8 window which moves pixel-by-pixel over the entire image. This timing can be significantly improved by taking an approximate solution: increasing the window size and/or setting a step size larger than one pixel for moving the local window. Furthermore, we can leverage the GPU to perform distance computation in parallel for multiple images simultaneously. At run time, only the clustering step takes a few minutes to complete. All other tasks and interactions are interactive.

6 USER STUDY

We performed a user study to evaluate the effectiveness of iMap by comparing it with the existing image search functions (archive, index, and text search) provided by online APOD. We used a hybrid study design of 2 conditions (iMap vs. online APOD) \times 3 tasks (text, image, and image + text). To avoid any learning effect, we did not allow one image search task to be conducted by the same user for both conditions. We assigned a target image for each of the six combinations except for Task 1, where users were asked to identify two images with very different numbers of image retrieved. So, a total of eight images were selected. These images cover different topics: astronaut, aurora, black hole, Earth, Jupiter, Mars, Moon, and Sun.

Hypotheses. We postulated four hypotheses for the study. Since the users’ respond time varies for each

task, hypotheses about response time of search for both conditions will be considered based on different tasks. Furthermore, we only consider the overall accuracy due to the high probability of finding the exact image.

- *Hypothesis 1.* Given the keywords and description only (Task 1), iMap is faster to search than online APOD.
- *Hypothesis 2.* Given the image only (Task 2), iMap is faster to search than online APOD.
- *Hypothesis 3.* Given the image and keywords (Task 3), iMap is faster to search than online APOD.
- *Hypothesis 4.* Overall, for image search, using iMap makes fewer errors than online APOD.

Interactions. Detailed interactions with iMap are described in Section 4. For online APOD, three search modes are provided: archive, index, and text search. The archive mode provides the dates and titles of all images and arranges them in the reverse chronological order. The index mode offers various keywords organized by category. Clicking on any keyword shows keyword-related thumbnail images with their dates, titles, and short explanations. The text search mode provides OR or AND search for multiple keywords. Since the way to search online APOD is similar to other websites, we assume that all users are familiar with it.

Tasks. Three tasks were implemented to compare the performance of iMap and online APOD. In each task, users were asked to identify the three most related or similar images. Ideally, the exact image should be found and if not, they were asked to find up to three most related ones.

- *Task 1.* Text search: given a short text description and several recommended keywords, users were asked to find the three most related images. Two images will be tested for each condition, one search generated a large number of retrieved images while the other generated a small number.
- *Task 2.* Image search: given target images without any keyword or description, users were asked to find the three most similar images. We assumed that the users could figure out the content of the images with their very basic astronomy knowledge.
- *Task 3.* Image + text search: given target images with

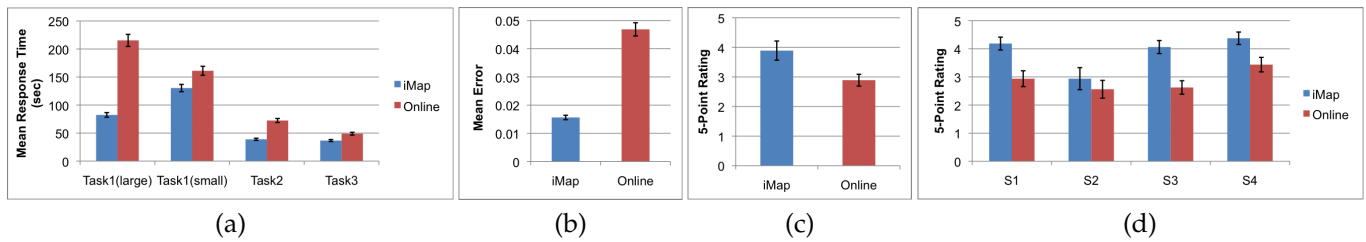


Fig. 8. Comparing iMap to online APOD. (a) mean response time by task. (b) mean error rate. (c) mean overall rating. (d) mean rating by statement. The four statements are: (S1) It was enjoyable to use; (S2) It was easy to search images with only keywords given; (S3) It was easy to search images with only images given; and (S4) It was easy to search images with keywords and images given.

several keywords, users were asked to find the three most similar images.

Procedure. For iMap, there was a 15-minute training session and a further 5-minute free exploration time (practice search) preceding the actual tasks. This is the same for online APOD, except that there was no training. A post-test survey for user preference and comments was conducted immediately after a user finished all the tests. A total of 16 users (eight graduate and eight undergraduate students) participated in this study and each user used both iMap and online APOD. Users were required to finish tasks in the order given. Users record the date of images they select, and we help them record the starting and ending time for completing each task. Each user was asked to perform 8 trials, and therefore, we had a total of 128 (16×8) trials. Each experiment was conducted individually and took approximately 40 minutes, including the training, practice task, experimental tasks for both conditions, and questionnaire.

Results and Discussion. We present results from the study in three aspects: completion time, accuracy, and subjective preferences. A paired t -test with a standard significance level $\alpha = 0.05$ was performed to determine statistical significance between the two conditions.

Completion Time. Figure 8 (a) shows the mean response time results when comparing iMap to online APOD for each task.

- Task 1: iMap was significantly faster than online APOD with only keywords given and a large number of images retrieved (iMap: 82.39s, online APOD: 215.36s; $p = 0.007$). This is because iMap provides a good overview of the image collection to facilitate the visual search. However, no significant difference was found with only keywords given and a small number of images retrieved. Possible reasons include: the text descriptions were confusing to some users; users were more proficient for online search while it took time for them to adjust to using iMap; and users needed to switch among multiple views or tabs in iMap while there was only one view in online APOD.
- Task 2: iMap was significantly faster than online APOD with only images given (iMap: 38.86s, online APOD: 72.29s; $p = 0.0127$). Two reasons explain this. First, using online APOD, users were able to

scroll up and down the result page to view about 10 images at a time, while they could view hundreds of image simultaneously using iMap. Second, neighboring images in iMap are similar while neighboring images have no connection at online APOD.

- Task 3: No significant difference in response time was found with keywords and images given. Since the images we selected for this task got a very small number of images retrieved (10 to 20 images), it was easy for users to search for both conditions.

Therefore, Hypothesis 1 was fully supported when the number of images retrieved is large. Hypothesis 2 was also fully supported while Hypotheses 3 was not supported.

Accuracy. Since almost all users were able to find the exact image correctly, we compute the error as the number of failed trials (i.e., image misidentified) over the number of total trials. With online APOD, 4.69% (6 out of 128) tests failed and with iMap, 1.56% (2 out of 128) tests failed. Figure 8 (b) shows the average error results when comparing iMap and online APOD. In terms of accuracy, iMap was not distinguishable from online APOD, which contradicts Hypothesis 4. We found that almost every user could find the exact image in both conditions. Incorrect answers mainly came from three users. One of them made four errors because he only tried to find similar images rather than the exact image. We identified this reason immediately and for the following users, we stressed that they should try their best to find the exact image. The error rate dropped dramatically in the later experiments.

Subjective Preferences. The 16 users completed a survey after their experiment. They were asked which condition they preferred overall and which interface they perceived to be more useful for each of the three tasks. Each statement was answered with a 5-point scale (1 = strongly disagree, 5 = strongly agree). We ran a paired t -test with a standard significance level $\alpha = 0.05$ and found a significant effect for iMap.

Figure 8 (c) shows the mean overall rating when comparing iMap to online APOD. The rating for iMap is significantly higher than online APOD (iMap: 3.89, Online APOD: 2.89; $p = 0.039$), which indicates that the users prefer using iMap over online APOD.

Figure 8 (d) shows the mean rating when comparing iMap to online APOD for the four statements:

- For Statement 1, iMap was judged to be more enjoyable to use (iMap: 4.19, online APOD: 2.94; $p = 0.0017$).
- For Statement 2, iMap was not significantly easier to search images with only keywords given (iMap: 2.94, online APOD: 2.56; $p = 0.5544$).
- For Statement 3, iMap was significantly easier to search images with only images given (iMap: 4.06, online APOD: 2.634; $p = 0.0001$).
- For Statement 4, iMap was significantly easier to search images with keywords and images given (iMap: 4.38, online APOD: 3.44; $p = 0.0098$).

- [10] W. Willett, J. Heer, and M. Agrawala. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1129–1136, 2007.

7 CONCLUSIONS AND FUTURE WORK

We have presented iMap, an analysis and visualization framework that supports effective searching, browsing, and understanding of a large image collection. iMap strikes a good balance among simplicity, intuitiveness, and effectiveness by addressing issues such as stable layout, screen utilization, and in-place interaction. Our user study confirms that iMap provides a more effective solution for image search, ranking, and identification compared with traditional archive and keyword search methods. We are currently deploying a web version of iMap for APOD so that any users can easily access our system online. In the future, we will improve our image distance measure by considering image aesthetics measures, high-level image features, and relationships between words. We will also incorporate animated transitions to help users better perceive the changes as a new query image is chosen.

REFERENCES

- [1] B. B. Bederson. Photomesa: A zoomable image browser using quantum treemaps and bubblemaps. In *Proceedings of ACM Symposium on User Interface and Software Technology*, pages 71–80, 2001.
- [2] M. Bruls, K. Huizing, and J. J. van Wijk. Squarified treemaps. In *Eurographics/IEEE TCVG Symposium on Visualization*, pages 33–42, 1999.
- [3] V. Hatzivassiloglou, J. L. Klavans, and E. Eskin. Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning. In *Proceedings of Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 203–212, 1999.
- [4] D. Lewis. Naïve (Bayes) at forty: The independence assumption in information retrieval. In *Proceedings of European Conference on Machine Learning*, pages 4–15, 1998.
- [5] R. J. Nemiroff and J. T. Bonnell. Astronomy picture of the day: <http://antwrp.gsfc.nasa.gov/apod/astropix.html>. In *Bulletin of the American Astronomical Society*, page 1291, 1995.
- [6] B. Shneiderman. Tree visualization with tree-maps: A 2D space-filling approach. *ACM Transactions on Graphics*, 11(1):92–99, 1992.
- [7] B. Shneiderman and M. Wattenberg. Ordered treemap layouts. In *IEEE Symposium on Information Visualization*, pages 73–78, 2001.
- [8] Y. Tu and H.-W. Shen. Visualizing changes of hierarchical data using treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1286–1293, 2007.
- [9] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

IMAGE SIMILARITY ANALYSIS

Content-based image analysis is at the heart of modern image searching and retrieval. Its primary goal is to organize digital image archives by their visual content. Image features capture visual properties of an image, either globally or locally. To extract image features that help perform meaningful classification and retrieval, researchers have utilized key visual contributions such as color [7], [5], [6], texture [3], [5], [6], shape [1], and salient points [4]. Advances have been made in both deriving new features and constructing signatures based on these features [2]. We leverage the color and spectrum information together with the grayscale version of images for similarity analysis.

REFERENCES

- [1] A. D. Bimbo and P. Pala. Visual image retrieval by elastic matching of user sketches. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):121–132, 1997.
- [2] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2), 2008.
- [3] B. S. Manjunath and W.-Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, 1996.
- [4] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.
- [5] K. Rodden, W. Basalaj, D. Sinclair, and K. Wood. Evaluating a visualisation of image similarity as a tool for image browsing. In *Proceedings of IEEE Symposium on Information Visualization*, pages 36–43, 1999.
- [6] K. Rodden, W. Basalaj, D. Sinclair, and K. Wood. Does organisation by similarity assist image browsing? In *Proceedings of ACM SIGCHI Conference*, pages 190–197, 2001.
- [7] M. Swain and B. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

IMAGE COLLECTION ORGANIZATION

The most common way to organize a large collection of images is based on a two-dimensional grid of thumbnails, but it enforces a uniform thumbnail aspect ratio. Furthermore, only parts of the dataset can be seen within line of sight when the image collection is excessively large. Over the years, different new solutions have been proposed to improve the organization of image collections. Chen et. al. [2] leveraged the Pathfinder network scaling technique, originally developed for the analysis of proximity data in psychology, to organize a collection of images based on their color labels, texture, shape orientation etc. Torres et al. [4] introduced a focus+context approach based on spiral and concentric rings for exploring query results in an image database. Yang et al. [5] developed a scalable semantic image browser (SIB) based on the semantic content of images. The multidimensional scaling layout based on semantic similarities was used for image overview and the value and relation layout was used for content overview. Gomi et al. [3] presented clustered album thumbnails (CAT) for hierarchical browsing large image collections which shows representative images when zooming out and individual images when zooming in. Brivio et al. [1] proposed a dynamic image browsing mechanism in which the arrangement of the thumbnails is based on weighted anisotropic Voronoi diagrams and Lloyd relaxation.

REFERENCES

- [1] P. Brivio, M. Tarini, and P. Cignoni. Browsing large image datasets through Voronoi diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1261–1270, 2010.
- [2] C. Chen, G. Gagaudakis, and P. Rosin. Content-based image visualization. In *Proceedings of International Conference on Information Visualisation*, pages 13–18, 2000.
- [3] A. Gomi, R. Miyazaki, T. Itoh, and J. Li. CAT: A hierarchical image browser using a rectangle packing technique. In *Proceedings of International Conference on Information Visualisation*, pages 82–87, 2008.
- [4] R. Torres, C. Silva, C. Medeiros, and H. Rocha. Visual structures for image browsing. In *Proceedings of International Conference on Information and Knowledge Management*, pages 49–55, 2003.
- [5] J. Yang, J. Fan, D. Hubball, Y. Gao, H. Luo, W. Ribarsky, and M. Ward. Semantic image browser: Bridging information visualization with automated intelligent image analysis. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, pages 191–198, 2006.

VISUALIZATION AND PRESENTATION MODES

Common visualization schemes for image collections include *relevance-ordered* (e.g., Google Images), *time-ordered* (e.g., the time quilt [2]), *clustered* (e.g., the design gallery layout using multidimensional scaling [3]), *hierarchical* (e.g., Google Image Swirl), and *composite* (the mix of two or more of the preceding forms). In terms of user presentation, there are three modes: *static* (i.e., no motion is involved whatsoever), *moving* (i.e., constant motion even without interaction), and *interactive* (i.e., motion triggered only under user interaction). A recent study has shown that static presentation is better than moving presentation in terms of recognition success and user preference [1]. We design a layout for organizing a large image collection using the composite visualization scheme and interactive presentation mode.

REFERENCES

- [1] K. Cooper, O. de Bruijn, R. Spence, and M. Witkowski. A comparison of static and moving presentation modes for image collections. In *Proceedings of International Working Conference on Advanced Visual Interfaces*, pages 381–388, 2006.
- [2] D. F. Huynh, S. M. Drucker, P. Baudisch, and C. Wong. Time quilt: Scaling up zoomable photo browsers for large, unstructured photo collections. In *Proceedings of ACM CHI Extended Abstracts*, pages 1937–1940, 2005.
- [3] J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. Design galleries: A general approach to setting parameters for computer graphics and animation. In *Proceedings of ACM SIGGRAPH Conference*, pages 389–400, 1997.