Learning User Interest for Image Browsing on Small-form-factor Devices

Xing Xie[1], Hao Liu[2], Simon Goumaz[1], Wei-Ying Ma[1]

[1]Microsoft Research Asia
5F, Sigma Center, No. 49, Zhichun Road
Beijing, 100080, P.R.China
{xingx, wyma}@microsoft.com, simongoumaz@hotmail.com

[2]Information Engineering
The Chinese University of Hong Kong
Shatin. N.T., Hong Kong
hliu@cuhk.edu.hk

ABSTRACT

Mobile devices which can capture and view pictures are becoming increasingly common in our life. The limitation of these small-form-factor devices makes the user experience of image browsing quite different from that on desktop PCs. In this paper, we first present a user study on how users interact with a mobile image browser with basic functions. We found that on small displays, users tend to use more zooming and scrolling actions in order to view interesting regions in detail. From this fact, we designed a new method to detect user interest maps and extract user attention objects from the image browsing log. This approach is more efficient than image-analysis based methods and can better represent users' actual interest. A smart image viewer was then developed based on user interest analysis. A second experiment was carried out to study how users behave with such a viewer. Experimental results demonstrate that the new smart features can improve the browsing efficiency and are a good compliment to traditional image browsers.

Categories & Subject Descriptors: H.5.2 [**Information Interfaces and Presentation**]: User Interfaces; H.1.2 [**Models and Principles**]: User/Machine Systems – human factors

General Terms: Algorithms, Human Factors

Keywords: Mobile image browsing, attention model, small display

INTRODUCTION

In recent years, mobile phones or PDAs with embedded digital cameras have undergone considerable progress. These days, a typical camera-equipped phone usually includes a 0.3M pixel digital camera, capable of taking still pictures with a 640x480 pixel resolution. It is expected that phones with 5M pixel cameras will become popular within several years, and that they will ultimately replace low-range and mid-range digital cameras.

Image browsing has become a "must-have" feature for this type of device because people want to check the pictures immediately after capture. In addition, with the steady increase of storage size on mobile devices, keeping the images directly on the device instead of saving them elsewhere is becoming more common.

Currently, most commercial image browsers on mobile devices only offer a simplified set of features directly ported from the desktop applications. Few browsers take the characteristics of small-form-factor devices into consideration, despite the fact that the user experience on mobile devices is quite different from that of desktop PCs. The differences include input capabilities, processing power and screen characteristics. Usually it is quite inconvenient to perform complex tasks on small devices because of the restricted input capabilities, such as selecting an image folder, or scrolling or zooming a large image. Currently, due to the limited processing power, it is very time-consuming or even impossible to open high-resolution images, browse large numbers of images at the same time, or edit large images. But the biggest difference from our point of view is the screen characteristics.

A brief overview of the current display capabilities of various mobile devices has been given in [8], but the authors focused more on improving color dithering and palettization in order to achieve better rendering results using reduced color bit-depths. In [3][7], the authors agreed with us that the small display problem is most critical. They proposed an attention model based image adaptation approach to address this. Based on an automatically extracted image attention model, they designed a number of novel features to aid or automate common image browsing tasks such as viewing thumbnails, zooming and scrolling.

We expect that processing power and display quality will cease to be a problem in the near future. But the input capabilities and display size will continue to be constraints

due to the portability requirements of mobile devices. It is therefore useful to know how users interact with an image viewer given these constraints, and how we can then improve the user experience by adding smart features.

The contributions of this paper include:

- We propose a new method for extracting user attention objects by analyzing how the user zooms and pans over an image.
- Based on this image attention object detection method, we develop a smart image viewer which incorporates new automatic browsing functions to facilitate browsing tasks on mobile devices.
- A user study is carried out to investigate how users behave with the new features.

The rest of the paper is organized in four parts. We first describe our user study of users' image browsing behavior with a basic mobile image viewer, and several conclusions were generated based on experimental observations. Second, we present a novel approach for analyzing the browsing log to generate an image attention model with a set of user attention objects. Then, based on this model, we design and evaluate a smart image viewer for mobile devices. Finally, we discuss two potential extensions to our work and conclude the paper with some remarks.

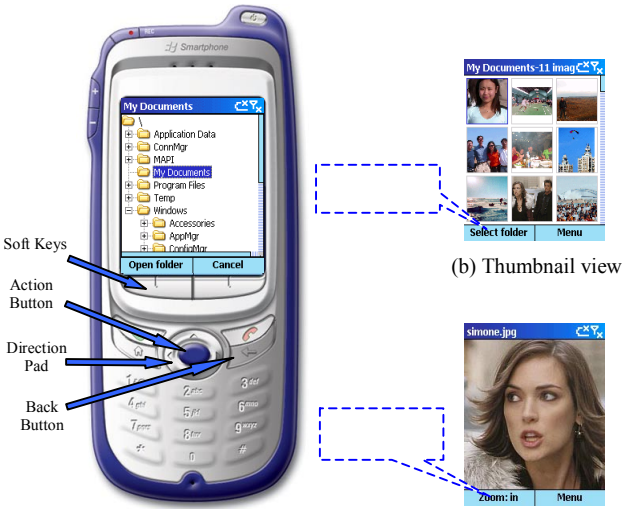
IMAGE BROWSING BEHAVIOR ON A BASIC IMAGE VIEWER

In this section, we relate the user study that was carried out to determine how users view images using basic image browsing functions, like zooming and scrolling. The user study results are analyzed and taken into consideration for designing a set of smart browsing functions later.

A Basic Image Viewer

There are already quite a few software packages available for image browsing and management on Windows CE or Palm OS-based handheld devices, such as ACDSee Mobile [1] and Resco Picture Viewer [11]. The basic features of these image browsers include a folder tree view to choose the folder containing images, a thumbnail view to quickly browse through a collection of images, and a detail view of the current image. On desktop applications, these views can be displayed on the screen simultaneously for more convenience [10], however, on most mobile image browsers, they are located in different windows because of the small size available. Some browsers also provide slide show, annotation (either sound or text) and communication features, but there is almost no effort to address the small display size problem. For now, it looks like developers simply tried to port desktop applications to the mobile devices, removing overly complex features and adding some functions to take advantage of the built-in capabilities of the devices like sound recording and handwriting recognition.

In order to study user browsing behavior, we implemented a basic image viewer on a Dopod 515 camera phone. The phone has a 133MHz Ti OMap processor, 64MB of RAM, a 320K pixel digital camera, and a 176x220/64K reflective TFT color display. The operating system is Microsoft Smartphone 2003. The viewer was implemented using Embedded Visual C++ 4.0 with the Smartphone SDK 2003. The program is optimized so that it can render image and interact with users' inputs smoothly.



(a) Folder tree view (b) Thumbnail view (c) Detail view

Figure 1. A screen capture of the basic image viewer.

Smartphone devices neither have a touch screen nor the screen size of Pocket PCs. The UI of most Smartphone applications relies on a combination of two soft keys (one of which usually triggers a menu), a direction pad, an action button and a 'back' button. As shown in Figure 1, our browser has most basic image browsing functions, including a folder tree view, a thumbnail view, and a detail view supporting scrolling and zooming. In Figure 1(c), a combination of left soft key, direction pad and action button is used for the detail view. The left soft key rotates between two different action modes: 'zoom: in' and 'zoom: out'. The action button activates the function associated to the left soft key. The direction pad scrolls the image in the given direction. The back button fits the image to the window, or returns to the thumbnail view if the fit-to-window view is the current view.

We record every action users perform in the image viewer in a browsing log, the structure of which is shown in Figure 2. Most of our analysis later is based on this log.

Each log file corresponds to all actions performed in browsing a single image, while each log item stands for one action and is attached to an action type, a current focus point, a current display ratio and a time stamp. The action type can be zoom in, zoom out and scroll (in four directions).

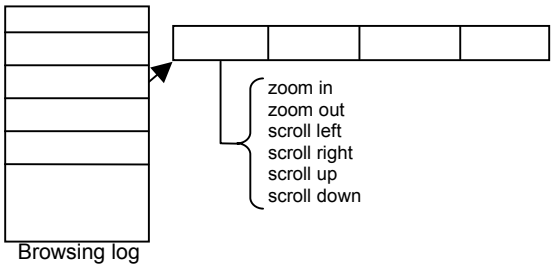


Figure 2. The structure of the browsing log.

User Study Settings

Ten subjects were selected to take part in our study, including seven males and three females. They were recruited from nearby universities. The criterion for selecting the subjects was that they should be very familiar with the use of computers and cell phones before the study.

We chose 26 images from various sources. Many of them are obtained from personal albums and popular Web sites. One of the images was provided to allow subjects to practice and get familiar with the user interface. The subjects were asked to explore the images in detail by employing manually the zooming and scrolling functions. The choice of the images aimed to include a variety of image types close to the one found in personal digital collections, which is likely to be the most common case of image viewing on mobile devices. There was thus a mix of outdoor and indoor images, with and without people, etc. We tried to pick out interesting images to encourage users, especially as the image content was not related to them as it is the case in personal collections. The image sizes ranged between 800x600 and 1600x1200. Images of sizes over 1600x1200 were not included to avoid overloading the device's limited memory and processing power.

Number of Actions

Based on our observation, on this basic image viewer, the subjects sometimes got "lost" when zooming and scrolling around to get content of the image. This is due to the fact that since only a portion of the image is visible and the image is relatively unfamiliar. Subjects usually scrolled in some direction only to discover that there is nothing interesting there, then backed off. It was found that on average, each subject used 46.6 zooming or scrolling actions per image, although it was quite inconvenient to perform such operations using the phone's buttons. On desktop PCs, zooming or scrolling is considerably less used because the screen size can display full image with sufficient detail. This leads to our first observation:

Observation 1: Due to the small display size, mobile users need to use more zooming and scrolling actions to catch the content of images than those of desktop users.

Table 1 lists the average number of actions per image. Scrolling was used the most, mainly due to the fact that the direction pad must be pressed repeatedly to reach a destination, i.e., the scrolling is discrete. A second reason is that the subjects seemed reluctant to modify the zoom ratio often. If the ratio was not optimal when viewing some region, additional scrolling was used. Some devices may provide continuous scrolling controls, but they can also be looked as a set of discrete actions, therefore, will not affect our following discussions.

Table 1. Average number of actions per image (basic viewer).

Action	Total	Scroll	Zoom in	Zoom out
Number	46.6	40.6	4.6	1.4
Percentage	100%	87.3%	9.9%	2.8%

Interaction Speed

Another observation of the study is about user interaction speed, i.e., the delays between browsing actions.

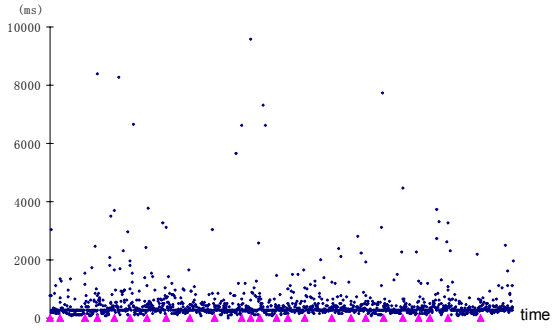


Figure 3. A sample sequence of actions and their durations.

Figure 3 is a plot representation of the action durations for one of the users' browsing on all 25 test images. They are in the original order of user interactions. The small triangles on the bottom represent the changing of images. There are 1460 actions in total for this user and an average of 58.4 actions for each image. The majorities of values are small and distributed in a relatively short range. Other values are bigger and exhibit a large variance. For this particular user, half of the action durations are below 270ms, while 5% of the durations are larger than 1200ms. The duration distributions for other users are very similar to Figure 3. This leads to our second observation:

Observation 2: The action durations in mobile image browsing exhibit a very unbalanced distribution. In other words, a small fraction of the values are much bigger than the others.

A phenomenon observed was that the physical arrangement of buttons on the Smartphone device influenced the time intervals between actions, for example:

- The intervals are longer when the user switches between actions, for example, between scroll and zoom, and when the zooming mode is changed.

- The last interval (at the end of the viewing) is significantly longer. Possible causes include the time for the user making his decision to stop the viewing and the need to use the device's back button, whose location is less convenient than that of the other buttons involved.

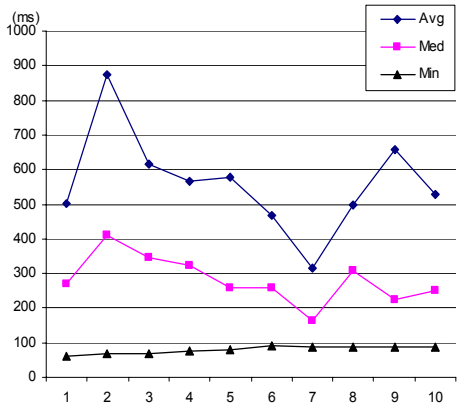


Figure 4. Interaction speed of different users.

Figure 4 shows the average action duration, median action duration and minimal action duration for all ten users. Although the action duration distribution is similar among users, different users interact at different speeds. A one way analysis of variance (ANOVA) is conducted on a log transform of the action duration data. It is found that there is a strong difference for the mean of action durations of different subjects, $F(9,11848)=18.07$, $p<0.001$. The minimal values are almost the same for different users, mainly due the limited system responding speed. In consequence, our third observation is:

Observation 3: Users interact with a mobile image viewer at different speeds, but the minimal action durations are similar.

Transition Action and Interesting Action

Observation 2 can be explained by introducing a distinction between "transition" actions and "interesting" actions. Most actions are transition actions, whose goal is to move the viewing focus to the next interesting region of the image. Interesting actions occur when the user reaches some interesting region and wants therefore to spend more time checking it.

Two main criteria were employed to identify interesting actions: display ratio and action duration. The first one is obvious, since the display ratio clearly states whether the user is looking at the detail or not. For the second one, we propose a heuristic algorithm to calculate the minimum duration $T_{interest}$ for each action to be considered as interesting. It is based on an initial hypothesis on the shape of the action duration distribution: we expect most actions (transition actions) to have a short, relatively similar duration, while rarer interesting actions should have a larger and more various durations. Because of Observation 3, we estimate $T_{interest}$ by looking at the median value T_{med} and the minimal value T_{min} of the original distribution.

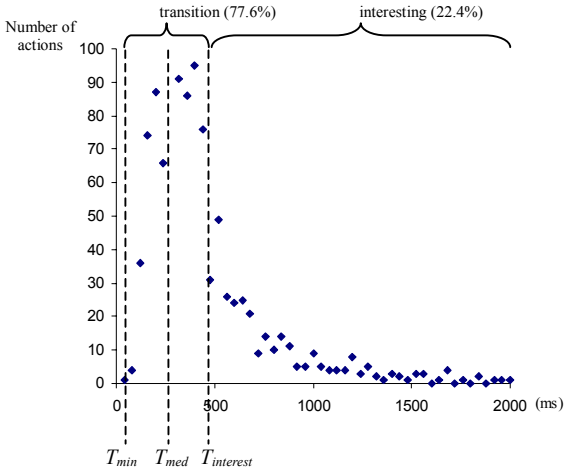


Figure 5. The duration distribution of transition and interesting actions for the example in Figure 3.

We firstly check whether the number of actions in the log is large enough, and the image resolution is sufficient to make the estimation significant. If the image size is too small, the use of zooming and scrolling will not be as necessary, so it is likely that the difference between the number of transition actions and that of interesting actions will decrease, biasing our estimation. Then, we use the median action duration as an indication of the average speed of transition actions, and assume that the distribution of transition action durations is symmetrical. Therefore, we can approximate the maximal action duration of transition actions, i.e., the threshold $T_{interest}$ as:

$$T_{interest} = T_{med} + T_{med} - T_{min} \quad (1)$$

We illustrate the above algorithm using Figure 5, which displays the action duration distribution for the example in Figure 3. Each data point in the figure stands for the number of actions with a 40ms interval, and we have cut the data for durations larger than 2000ms. For this example, $T_{interest}$ is 478ms and 22.4% of the actions are interesting actions, i.e., 327 interesting actions for all 25 images.

User Interest Map

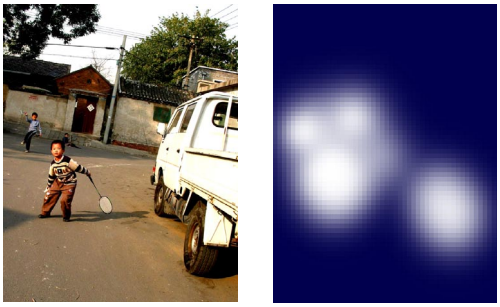
After transition action and interesting action detection, we already know the focus of interest of an image and display ratio associated with each interesting action, a user interest map can be computed to represent the interest of the user for the different image regions. The generating of user interest map is somewhat similar to the analysis of eye-movement traces which has been studied in another context [15][5]. Since the actions here have semantics and are easy to be recorded on mobile devices than eye-movements, it is regarded as a direct indication for users' attention.

For each image zone, of size 20x20 pixel, the following four parameters are calculated:

T_{view}, the total weighted viewing time of this zone
R_{view}, the best display ratio of this zone
T_{first}, the time the zone was first viewed
N_{view}, the number of different viewings of this zone

The detailed steps are, for each interesting action in the browsing log:

1. Check if the action duration goes over a fixed maximum value. It is possible that, sometimes, the action duration might have no relation to user interest for diverse reasons. For instance, the user has been interrupted by some events. As we have no way to be aware of it, we protect the decision procedure by limiting the action duration.
2. For each image zone in the user interest map corresponding to the action focus:
   a. If it's the first time the zone comes into view, update its T_{first} value.
   b. If the zone was not into view for the previous action, increase its N_{view}.
   c. If the action's associated display ratio is bigger than the zone's current R_{view}, update the latter. The maximum allowed ratio is of course 1:1.
   d. Update the viewing time T_{view} of the zone according to its distance to the focus point of the action. This is explained in more detail below.



(a) Original image (b) User interest map

Figure 6. A sample user interest map.

As the user will generally try to center the focus on the object he's interested in, the interest cannot be assumed equal for every zone in the viewing window. We favor the image zones close to the focus point to reflect this. T_{view} is updated as following:

$$T_{view} = T_{view} + T_{ActionDuration} \frac{1 + cos(d\pi / d_{max})}{2} \qquad (2)$$

where d is the distance from the zone to the focus point and d_{max} is the distance from the edge of the current viewing window to the focus point; $T_{ActionDuration}$ is the time user spend on current view window. Functions other than *cos*, such as *Gaussian*, might also work here. We plan to study the performance difference by using different distance functions in our future work. A sample user interest map is shown in Figure 6, where the intensity represents the value of T_{view}.

Shared Interest among Different Users

After examining the user interest maps by different users, we come to following observation:

Observation 4: Most important regions of image seem to be agreed upon among the different subjects, even though the exact attended points and zoom ratios often slightly differ. Secondary points of interest usually vary from subject to subject.

(a) Original image (b) Regions with maximal T_{view}

Figure 7. Regions with maximal T_{view} by different users.

Figure 7 shows an example and the regions with maximal T_{view} for different subjects, where red spots indicate attended points and spot size gives an indication of the total viewing time. The typical browsing sequence was quite similar across the subjects: zoom in to some main interest point, and then scroll around to see other points. The zoom ratio was seldom modified after the first zooming in, but could be in some cases.

BUILDING ATTENTION MODEL FROM BROWSING LOG

Although the user interest map is a very intuitive representation of users' interest, it is not efficient to store the whole map with the original image, whereas the computation of other smart features will be quite costly if directly based on the map. To solve this problem, we adopt an object-based image attention model similar to [3]. The image attention model is defined as a set of attention objects, which stand as an abstraction for the user's interest.

Definition 1: The attention model for an image is defined as a set of attention objects:

$$\{AO_i\} = \{(ROI_i, AV_i, MPS_i, MPT_i)\}, \quad 1 \le i \le N \qquad (3)$$

where AO_i, the *ith* attention object within the image
ROI_i, Region-Of-Interest of AO_i
AV_i, attention value of AO_i
MPS_i, minimal perceptible size of AO_i
MPT_i, minimal perceptible time of AO_i

Four attributes are assigned to each attention object (*AO*). They are Region-Of-Interest (*ROI*), attention value (*AV*), minimal perceptible size (*MPS*), and minimal perceptible time (*MPT*). The notion of 'Region-Of-Interest (ROI)' is borrowed from JPEG 2000 [4]. It is referred as a spatial region within an image that corresponds to an attention object. Attention value (*AV*) is a quantified value indicates the weight of each attention object in contribution to the information contained in the image. Minimal perceptible size (*MPS*) represents the minimal allowable spatial area of an attention object. It is meant as a threshold to avoid excessive sub-sampling when reducing display size. Minimal perceptible time (*MPT*) is a threshold for the fixation duration when browsing an attention object. If an attention object does not stay on the screen longer than *MPT*, it may not be perceptible enough to let users catch the information.

The original image attention model descried in [3] includes three types of attention objects, i.e., saliency, face, and text objects. Many image browsing tasks can be treated as manipulating attention objects [3][7] to provide as much information as possible under resource constraints, The image attention objects can be generated automatically by detection algorithm. However its performance depends heavily on detection algorithm. As soon as faces or clear and distinct saliency features are not available, the detection accuracy is greatly reduced due to the limited capabilities of the available algorithms. A good complementary approach is to learn from the user browsing history, thus leveraging the user himself to detect the attention regions. In the next sub-section, we introduce our approach to generate user attention objects from the user interest map.

User Attention Objects

A new type of objects, named user attention objects, is added to represent user-defined objects. The extraction of user attention objects takes the total viewing time T_{view} as the main factor, as it is clearly the value the most directly related to user interest. Though this problem looks quite similar to the attention area extraction introduced in [9], there are a number of differences. For example, we have much more information in the map, like R_{view} and N_{view}. Particularly, R_{view} can be used to determine the size of an object. Therefore, we adopt a more straightforward extraction method here.

The maxima in the interest map naturally give us a first set of *AO*s, which we call "primary" *AO*s. These correspond to the main interest points the user has attended during his viewing. However, they might not cover enough of the interest, mainly because of two common behaviors displayed by users:

- Users often spend some time looking at some region of the image at a certain zoom ratio, then zoom in further to check the main interest point closer. As the primary *AO* rectangle depends on the highest zoom ratio used, we will only consider the second focus.
- Before, in the middle of or after viewing some main interest point, users eventually scroll around, sometimes back and forth, to check nearby interesting zones. While these movements are significant to interest, they seldom produce maxima on our interest map and are therefore ignored by primary *AO*s.

Thus, when extracting each primary *AO*, we also try to check for nearby interesting zones it does not cover, by growing or shifting from the original *AO* rectangle (growing and shifting respectively address each of the two cases). This step produces another set of *AO*s, "secondary" *AOs*. Having prepared the two sets of *AO*s, we perform a correction step to adjust primary *AO*s' attention values, favoring the *AO*s that have been viewed early in the browsing and those that have been viewed many times. This reflects the fact that users generally focus first on the most interesting objects, and often come back to them after the initial viewing. Finally, secondary *AO*s are checked against primary ones and added if relevant.

Here is the step-by-step description of the algorithm:

1. A primary *AO* is created for each maximum T_{view} in the user interest map, if the maximum point is not already contained in an existing *AO* (Figure 8(a) and (d)). The size of this *AO* is decided by the point's R_{view} and its initial *AV* is set to the point's T_{view}.
2. A secondary *AO* is created as following, starting from the primary *AO*'s rectangle.
   a. We first try to grow around the initial rectangle (corresponding to the first case aforementioned). The growing is performed in four directions. If there is no sufficient T_{view} in one direction, the growing will be stopped in that direction while continued in other directions (Figure 8(b) and (c)).
   b. If growing results in a "bad" aspect ratio for the derived rectangle (too narrow or too flat, which means we are in the second case aforementioned), we shift the initial rectangle in the appropriate direction instead (Figure 8(e) and (f)).
3. Adjust the primary *AO*s' attention value if necessary. We add a fixed value to the attention value of the first viewed *AO* (i.e., with the minimal T_{first}), if it is not already the most important. We do a similar operation on the *AO*s that have been viewed more than once, increasing their attention values by a value proportional to N_{view}.
4. Check each secondary *AO* against the primary ones, adding it to the final set if relevant. A secondary *AO* is considered relevant if the center of its rectangle is not lie in primary *AO*s and 50% portion of its rectangle is not covered by primary *AO*s. In the particular case a secondary *AO* encloses a primary one, and their size difference is relatively small, the primary *AO*'s rectangle is replaced with the secondary.

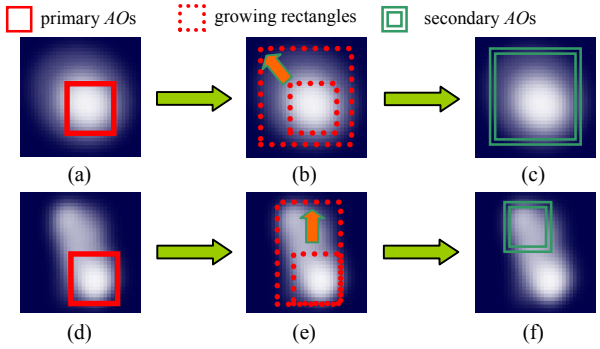


Figure 8. Primary and secondary user attention objects.

The other attributes of user attention objects are generated as following:

- The *AV* is adjusted to be as important as face objects by setting a relative high model weight for user attention objects. We consider the importance of user attention objects to be much more certain than that of saliency or text objects.
- The *MPS* is set to the minimal R_{view} in the *AO* rectangle. We want the user *AO*s to be displayed at a zoom ratio close to the one the users actually used when browsing manually, as we have no clue about the *AO*s' content.
- The *MPT* of a given user *AO* is set to be proportional to its *AV*. This is reasonable, as T_{view} is the determining factor when calculating *AV*.

Update of the Image Attention Model

Initially, the attention model of each image is detected and saved as metadata in the image as described in [3]. In the later browsing process, each time the user zooms in, scrolls, zooms out, etc. the browsing log is updated with the corresponding information. When the user is done viewing the image, the image browser performs the necessary computation to extract a set of user attention objects and updates the attention model and saves the changes.

A complete work flow is shown in Figure 9. We first check the user interaction speed, and then look at the list of interesting actions to generate the user interest map, from which the user *AO*s will be extracted.

The browser updates the image's original set of *AO*s when a new set is produced. Two issues need to be dealt with: we must be able to tell if new *AO*s are actually the same as existing ones, and we must eliminate those *AO*s that are equivalent to each other. Parasite *AO*s can be mistakenly produced for various reasons, for example, the user's browsing is interrupted in the middle of panning between two interesting regions, and then resumed. One feasible way to detect them is to check the sets of *AO*s produced by other viewings of the image. If the *AO* is never confirmed, that is, similar *AO*s never show up again, it is likely that it is a parasite *AO*.

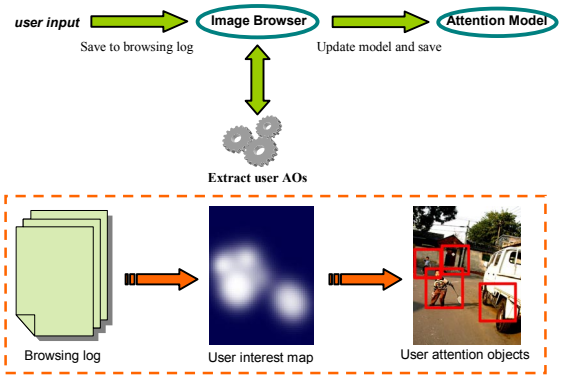


Figure 9. The work flow of updating image attention model.

Collaborative Update

According to Observation 4, different users tend to agree upon the most important regions of interest in an image. Therefore, for those shared images, it could be possible for users to take advantage of each other's browsing, and to share different user browsing logs to get a more complete set of user attention objects. We propose an approach to extend the previous scheme for image sharing over the Internet. Figure 10 illustrates the proposed scheme.

The images are first uploaded by the original user to the server, which proceeds to detect their face, text and saliency features and store the images with the resulting attention model in the collection (step 1 on figure). When mobile users access the server to view the collection, the attention-model-marked images are downloaded over to their device (step 2). The browsing of each image is done using our client, and when it finishes, the corresponding browsing log is sent back to the server (step 3). The server finally processes the feedback data to update the image's attention model adequately (step 4).

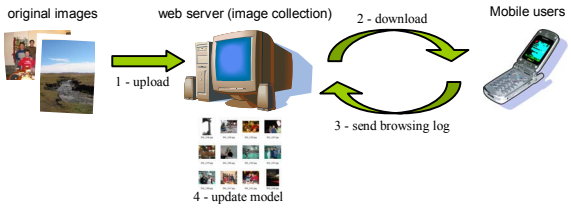


Figure 10. Collaborative update of the image attention model.

IMAGE BROWSING BEHAVIOR ON A SMART IMAGE VIEWER

Based on the image attention model described in the previous section, we implementation a new image viewer including a number of smart features to help users browse large images on the small screen more conveniently. We will observe through a user study how users browse having the full functionality of a smart image viewer at their disposal.

A Smart Image Viewer

The smart image viewer is built on top of the basic image viewer. Three smart features have been made available in addition to regular zoom and scroll. They are smart zoom-in, animation, and smart navigation:

- **Smart zooming-in (Zoom: Auto)**: it allows the user to go directly to the most informative part of an image with an appropriate display ratio. It is implemented as searching the optimal region which contains as many perceptible attention objects as possible under the display constraint. Here perceptible stands for that the display area of an attention object is larger than its *MPS*.
- **Animation**: an optimal browsing path similar to that in [7] is generated and then the window is automatically steered to browse through the different parts of an image. The image browsing path is defined as an iteration of fixation status (i.e. exploiting an interesting region) and shifting status (i.e. scrolling to the next focus). The problem can be transformed to a Traveling Salesman Problem: given a finite number of attention objects along with the distance moving between each pair of them, find the shortest way covering all the attention objects. Here we use a backtracking algorithm to enumerate all the possible paths and then find the best one among them. Some approximation algorithms can be applied to get a fast but sub-optimal solution when the number of attention objects is large.
- **Smart Navigation**: it is designed to speed up and ease the browsing while still giving the user some control. The attention objects are grouped into four regions, i.e. up, down, left, and right, by 45 degree diagonal and 135 degree diagonal lines starting from the current focus of the image. In smart navigation mode, pressing the direction pad will result in one of four possible directions, i.e., up, down, left, right, and trigger a smooth scrolling to the closest *AO* in the corresponding direction. If the next closest *AO* in the given direction is too similar to the current one, it is ignored.

The key board functions of the smart viewer do not differ much with the basic viewer. As shown in Figure 11, in the detail view, the left soft key rotates among four different modes: 'zoom: in', 'zoom: out', 'zoom: auto', and 'animation' if an animation path exists for the image. The action button activates the function associated with the left soft key. The right soft key triggers a menu which includes a switch of the 'smart navigation' mode and other system functions such as 'attention detection' and 'set to wall paper'.

During animation, the left soft key allows users to pause/resume the animation, and the 'back' button stops the animation. The right soft key and the direction pad are disabled during the animation.

Figure 11. Detail view of the smart image viewer.

User Study Settings

The same ten subjects that took part in previous user study were asked to browse through a collection of 11 images, which were different from those in the former study, while came from the same sources. The first image was provided to allow subjects to practice and get familiar with the new features. Each image already contained a set of *AO*s generated from previous browsing, so that most interesting parts of the image could be viewed using the automated functions. Although defining a satisfying set of *AO*s is subjective, as was indicated in Observation 4, users generally agree on the choice of the most important ones. The image selection was done using similar criteria as for the first study.

The subjects were instructed on the use of the different functions, and then asked to view the images using whatever methods they thought the most convenient and effective. The browsing log recorded all their actions, including the using of smart functions. After having viewed all images, they answered the following four subjective questions:

1. *How about your feelings on the browsing functions you just tried?*
2. *With the new browsing feature, if you still used the regular zoom and scroll, why?*
3. *Which of the browsing methods did you prefer most?*
4. *When will you use mobile devices to browse pictures?*

Finally, we also encouraged the subjects to give more comments about the browser and its features.

Number of Actions

Whereas subjects used an average of 46.6 actions per image using only manual zoom and scroll, 31.3 actions were necessary when using the full range of functions. Especially, the number of "non-smart" actions (scrolling, zooming in and zooming out) was reduced to less than 20, as shown in Table 2. When the results were analyzed by one way ANOVA with action type conditions, there were significant reductions on scrolling actions ($F(1,18)=36.6$, $p<0.001$) and zoom-in actions ($F(1,18)=28.2$, $p<0.001$), but no big difference for zoom-out actions ($F(1,18)=2.4$, $p=0.14$). Most of the smart actions were smart navigation while the other two functions were usually used once for each image.

Observation 5: The number of user actions is smaller on a smart image viewer. This indicates that the smart features can make the browsing more efficient.

Table 2. Average number of actions per image (smart viewer).

Action	Total	Scroll	Zoom in	Zoom out	Smart functions
Number	31.3	15.1	2.6	1.5	12.1
Percentage	100%	48.2%	8.3%	4.8%	38.7%

It was found that the subjects still used the manual zooming and scrolling for a good part of the images. This was expected, as it seems impossible to produce a set of *AO*s that is perfect for every user.

We do not use browsing time as a criterion because we believe that unlike that of searching, the efficiency of browsing cannot be simply evaluated by the time cost. For example, if the content is interesting, users generally like to spend more time checking it.

Complementarity

We collect the comments from the subjects after experiments. For the first question we asked the subjects, 60% of them thinking the new features are a big improvement over regular zooming and scrolling methods. While remaining 40% subjects think it nice to have such function although they are not necessary to use them under all circumstances.

To the second question, the general answer was that they felt more "in control" when using the manual (zoom and scroll) method. They were able to decide the time they focused on a point, and could browse more content this way. One subject said that if the image was familiar, he would prefer employing the smart functions, but if it was new, he liked to check it in detail manually.

For the third question, five subjects said they liked smart navigation the most, three preferred animation, two preferred manual zoom and scroll, and none preferred smart zooming-in. One subject said he found the number of foci in the animation to be usually insufficient, and that the fixation time was too short. Another would like to be able to set the animation speed, and to have some kind of "reverse" button when animating, so that he could come back to interesting points the animation just passed.

When we probe the people for the last question, the majority of users think they will use mobile devices to browse images while on the move, when time and dedication are not optimal (for instance, because of environmental factors), they will more likely take a "sneak peek" at the images than browse them in full detail. This should be regarded as a positive point for smart functions.

In summary, the following observation can be made:

Observation 6: Different users appreciate to have different methods at their disposal, without exclusively using one or the other. Although the smart methods will not replace manual browsing, the latter must be seen as a necessary complement to them.

DISCUSSIONS

In this paper, we only focused on studying the browsing behavior for a single image on mobile devices. We use this section to discuss two potential extensions of the work here, i.e. dealing with new images and image collections.

Browsing New Images: Generalized or Personalized?

Until now, we have been learning to extract attention objects for existing images. If we want to make the image attention model evolve with the user browsing and be able to detect attention objects in new images, visual features should be added and combined with the model. In this way, a mapping from user interest to low level image features can be established and therefore, facilitate the detection of interesting parts in new images. This is related to many CBIR problems [16].

If we look to the problem from a "general" perspective that is, building an evolving attention model through users' interaction that could be applied and benefit to all users, then it looks like a very difficult problem: as we have been able to observe in the user studies, sometimes even a human mind cannot predict what elements of a given image a random user will be interested in, even if the most obvious elements can usually be guessed. It would be very hard for this evolving model to exceed the performance that a saliency map [9] has already achieved.

But the previous observations lead to considering the problem from a "personalized" perspective. If we associate the attention detection process with a single user's interaction, with a single background, a single set of interests, etc., then it looks more relevant: if we get to learn, for example, that the user is usually more interested in a person with certain characteristics in images, maybe because they look like people he knows, then this might actually be useable to infer what he'll be interested in when browsing new images.

Browsing Image Collections: What's New?

There have already been many efforts addressing the problem of organizing, searching and visualization of image collections on desktop PCs [2][12]. Recently, authors of [6] has ported their timeline based image browser to a handheld device and studied its performance. It was reported that their browser is as effective for searching and browsing tasks as a traditional browser that requires users to manually organize their photos. In [14], smart thumbnails are proposed to improve image searching efficiency. A rectangle containing all the attention objects is cropped from the original image and shown as the new thumbnail. Therefore, a much bigger spatial area is used to present the information inside an image. A similar technique has also been applied to improve desktop thumbnails in [13].

Based on our observations, the main difference between image collections on a mobile device and on a PC may lie in the type of images that would exist on mobile devices. Initial market study shows that people tend to use camera phone to take pictures of other people (either familiar or unfamiliar) and interesting scenes. That would be a good hint to improve the image browsing experience on mobile devices. For example, images may be better organized by people and events than by time.

CONCLUSIONS

Designing image viewers for mobile devices is a lively topic today. In our experiments, we found that mobile users tend to use more zooming and scrolling actions while browsing to view interesting regions, mainly because of the limited display size. Leveraging users' browsing interactions, we proposed an adaptive technique to detect user attention objects of an image, and in turn, used them to design a set of smart features to help users' browsing process. User study results showed that the smart features can improve the browsing efficiency and are a good compliment to traditional browsing functions.

We plan to extend the image attention model to be personalized and represent the relationship between user interest and certain image features. A better organization of photos on camera-equipped phones is another interesting problem to explore. Since people usually use phones for personal communications, it is essential that the data can be organized, searched and visualized by people or events. We will continue to investigate these directions in the future.

ACKNOWLEDGEMENTS

We would like to express our special appreciation to Dave Vronay, Patrick Baudisch, Xin Fan, Zhigang Hua, and all anonymous reviewers for their insightful comments. We also thank all the participants in our user study experiments.

REFERENCES

1. ACD Systems. http://www.acdsystems.com
2. Bederson B.B.. PhotoMesa: a zoomable image browser using quantum treemaps and bubblemaps.ACM UIST 2001, Orlando, FL, USA, Nov. 2001.
3. Chen, L.Q., Xie, X., Fan, X., Ma, W.Y., Zhang, H.J., and Zhou, H.Q. A visual attention model for adapting images on small displays. ACM Multimedia Systems Journal, Vol. 9, No. 4, Oct. 2003.
4. Christopoulos, C., Skodras, A., and Ebrahimi, T. The JPEG2000 still image coding system: an overview. IEEE Trans. on Consumer Electronics, Vol. 46, No. 4, pp1103-1127, Nov. 2000.
5. DeCarlo, D., and Santella, A. Stylization and Abstraction of Photographs, SIGGRAPH 2002, San Antonio, Texas, USA, Jul. 2002
6. Harada, S., Naaman, M., Song, Y.J., Wang, Q.Y., and Paepcke, A. Lost in memories: interacting with large photo collections on PDAs. Technical report, Stanford University, Oct. 2003. http://dbpubs.stanford.edu/pub/2003-30
7. Liu, H., Xie, X., Ma, W.Y., and Zhang, H.J. Automatic browsing of large pictures on mobile devices. ACM Multimedia 2003, Berkeley, CA, USA, Nov. 2003.
8. Luo, J., Singhal, A., Braun, G., Gray, R.T., Seignol, O., and Touchard, N. Displaying images on mobile devices: capabilities, issues, and solutions. ICIP 2002, Rochester, NY, Sep. 2002.
9. Ma, Y.F., and Zhang, H.J. Contrast-based image attention analysis by using fuzzy growing. ACM Multimedia 2003, Berkeley, CA, USA, Nov. 2003.
10. Plaisant, C., Carr, D., and Shneiderman, B. Image browsers: taxonomy, guidelines, and informal specifications. IEEE Software, Vol. 11, No. 1, pp33-52, Mar., 1995.
11. Resco. http://www.resco-net.com
12. Rodden, K., and Wood, K. How do people manage their digital photographs?. ACM CHI 2003, Fort Lauderdale, FL, USA, Apr. 2003.
13. Suh, B., Ling, H., Bederson, B.B. and Jacobs, D.W. Automatic thumbnail cropping and its effectiveness. ACM UIST 2003, Vancouver, Canada, Nov. 2003.
14. Wang, M.Y., Xie, X., Ma, W.Y., and Zhang, H.J. MobiPicture - browsing pictures on mobile devices. ACM Multimedia 2003 demo, Berkeley, CA, USA, Nov. 2003.
15. Wooding, D.S. Fixation maps: quantifying eye-movement traces. Eye Tracking Research and Applications Symposium (ETRA 2002), New Orleans, LA, USA, Mar. 2002.
16. Yu, K., Ma, W.Y., Tresp, V., Xu, Z., He, X., Zhang, H.J., and Kriegel, H.P. Knowing a tree from the forest: art image retrieval using a society of profiles. ACM Multimedia 2003, Berkeley, CA, USA, Nov. 2003.