Automatic Thumbnail Cropping and its Effectiveness

Bongwon Suh*, Haibin Ling, Benjamin B. Bederson*, David W. Jacobs
Department of Computer Science
*Human-Computer Interaction Laboratory
University of Maryland
College Park, MD 20742 USA
+1 301-405-2764
{sbw, hbling, bederson, djacobs}@cs.umd.edu

ABSTRACT
Thumbnail images provide users of image retrieval and browsing systems with a method for quickly scanning large numbers of images. Recognizing the objects in an image is important in many retrieval tasks, but thumbnails generated by shrinking the original image often render objects illegible. We study the ability of computer vision systems to detect key components of images so that automated cropping, prior to shrinking, can render objects more recognizable. We evaluate automatic cropping techniques 1) based on a general method that detects salient portions of images, and 2) based on automatic face detection. Our user study shows that these methods result in small thumbnails that are substantially more recognizable and easier to find in the context of visual search.

Keywords
Saliency map, thumbnail, image cropping, face detection, usability study, visual search, zoomable user interfaces

INTRODUCTION
Thumbnail images are now widely used for visualizing large numbers of images given limited screen real estate. The QBIC system developed by Flickner *et al.* [10] is a notable image database example. A zoomable image browser, PhotoMesa [3], lays out thumbnails in a zoomable space and lets users move through the space of images with a simple set of navigation functions. PhotoFinder applied thumbnails as a visualization method for personal photo collections [14]. Popular commercial products such as Adobe Photoshop Album [2] and ACDSee [1] also use thumbnails to represent image files in their interfaces.

Current systems generate thumbnails by shrinking the original image. This method is simple. However, thumbnails generated this way can be difficult to recognize, especially when the thumbnails are very small. This phenomenon is not unexpected, since shrinking an image causes detailed information to be lost. An intuitive solution is to keep the more informative part of the image and cut less informative regions before shrinking. Some commercial products allow users to manually crop and shrink images [20]. Burton *et al.* [4] proposed and compared several image simplification methods to enhance the full-size images before subsampling. They chose edge-detecting smoothing, lossy image compression, and self-organizing feature map as three different techniques in their work.

In quite a different context, DeCarlo and Santella [8] tracked a user's eye movements to determine interesting portions of images, and generated non-photorealistic, painterly images that enhanced the most salient parts of the image. Chen *et al*. [5] use a visual attention model as a cue to conduct image adaptation for small displays.

In this paper, we study the effectiveness of saliency based cropping methods for preserving the recognizability of important objects in thumbnails. Our first method is a general cropping method based on the saliency map of Itti and Koch that models human visual attention [12][13]. A saliency map of a given image describes the importance of each position in the image. In our method, we use the saliency map directly as an indication of how much information each position in images contains. The merit of this method is that the saliency map is built up from low-level features only, so it can be applied to general images. We then select the most informative portion of the image.

Although this saliency based method is useful, it does not consider semantic information in images. We show that semantic information can be used to further improve thumbnail cropping, using automatic face detection. We choose this domain because a great many pictures of interest show human faces, and also because face detection methods have begun to achieve high accuracy and efficiency [22].

In this paper we describe saliency based cropping and face detection based cropping after first discussing related work from the field of visual attention. We then explain the

UIST '03, Vancouver, BC, Canada

design of a user study that evaluates the thumbnail methods. This paper concludes with a discussion of our findings and future work.

RELATED WORK

Visual attention is the ability of biological visual systems to detect interesting parts of the visual input [12][13][16][17] [21]. The saliency map of an image describes the degree of saliency of each position in the image. The saliency map is a matrix corresponding to the input image that describes the degree of saliency of each position in the input image.

Itti and Koch [12][13] provided an approach to compute a saliency map for images. Their method first uses pyramid technology to compute three feature maps for three low level features: color, intensity, and orientation. For each feature, saliency is detected when a portion of an image differs in that feature from neighboring regions. Then these feature maps are combined together to form a single saliency map. After this, in a series of iterations, salient pixels suppress the saliency of their neighbors, to concentrate saliency in a few key points.

Chen *et al*. [5] proposed using semantic models together with the saliency model of Itti and Koch to identify important portions of an image, prior to cropping. Their method is based on an attention model that uses attention objects as the basic elements. The overall attention value of each attention object is calculated by combining attention values from different models. For semantic attention models they use a face detection technique [15] and a text detection technique [6] to compute two different attention values. The method provides a way to combine semantic information with low-level features. However, when combining the different values, their method uses heuristic weights that are different for five different predefined image types. Images need to be manually categorized into these five categories prior to applying their method. Furthermore, it heavily relies on semantic extraction techniques. When the corresponding semantic technique is not available or when the technique fails to provide a good result (e.g. no face found in the image), it is hard to expect a good result from the method. On the other hand, our algorithm is totally automatic and works well without manual intervention or any assumptions about the image types.

THUMBNAIL CROPPING

Problem Definition

We define the thumbnail cropping problem as follows: Given an image I, the goal of thumbnail cropping is to find a rectangle R_C, containing a subset of the image I_C so that the main objects in the image are visible in the subimage. We then shrink I_C to a thumbnail. In the rest of this paper, we use the word "cropping" to indicate thumbnail cropping.

In the next subsection, we propose a general cropping method, which is based on the saliency map and can be applied to general images. Next, a face detection based cropping method is introduced for images with faces.

A General Cropping Method Based on the Saliency Map

In this method, we use the saliency value to evaluate the degree of informativeness of different positions in the image I. The cropping rectangle R_C should satisfy two conditions: having a small size and containing most of the salient parts of the image. These two conditions generally conflict with each other. Our goal is to find the optimal rectangle to balance these two conditions.

An example saliency map is given in Figure 1:

Figure 1: left: original image, right: saliency map of the image shown left

Find Cropping Rectangle with Fixed Threshold using Brute Force Algorithm

We use Itti and Koch's saliency algorithm because their method is based on low-level features and hence independent of semantic information in images. We choose Itti and Koch's model also because it is one of the most practical algorithms on real images.

Once the saliency map S_I is ready, our goal is to find the crop rectangle R_C that is expected to contain the most informative part of the image. Since the saliency map is used as the criteria of importance, the sum of saliency within R_C should contain most of the saliency value in S_I. Based on this idea, we can find R_C as the smallest rectangle containing a fixed fraction of saliency. To illustrate this formally, we define candidates set $\Re(\lambda)$ for R_C and the fraction threshold λ as

$$\Re(\lambda) = \left\{ r : \frac{\sum_{(x,y)\in r} S_I(x,y)}{\sum_{(x,y)} S_I(x,y)} > \lambda \right\}$$

Then R_C is given by

$$R_C = \arg\min_{r\in\Re(\lambda)}(area(r))$$

R_C denotes the minimum rectangle that satisfies the threshold defined above. A brute force algorithm was developed to compute R_C.

Find Cropping Rectangle with Fixed Threshold using Greedy Algorithm

The brute force method works, however, it is not time efficient. Two main factors slow down the computation. First, the algorithm to compute the saliency map involves several series of iterations. Some of the iterations involve convolutions using very large filter templates (on the order of the size of the saliency map). These convolutions make the computation very time consuming.

Second, the brute force algorithm basically searches all sub-rectangles exhaustively. While techniques exist to speed up this exhaustive search, it still takes a lot of time.

We found that we can achieve basically the same results much more efficiently by: 1) using fewer iterations and smaller filter templates during the saliency map calculation; 2) squaring the saliency to enhance it; 3) using a greedy search instead of brute force method by only considering rectangles that include the peaks of the saliency.

```
Rectangle GREEDY_CROPPING (S, λ)
thresholdSum ← λ * Total saliency value in S
R_c ← the center of S
currentSaliencySum ← saliency value of R_c
WHILE currentSaliencySum < thresholdSum DO
  P ← Maximum saliency point outside R_c
  R' ← Small rectangle centered at P
  R_c ← UNION(R_c, R')
  UPDATE currentSaliencySum with new region R_c
ENDWHILE
RETURN R_c
```

Figure 2: Algorithm to find cropping rectangle with fixed saliency threshold. S is the input saliency map and λ is the threshold.

Figure 2 shows the algorithm GREEDY_CROPPING to find the cropping rectangle with fixed saliency threshold λ . The greedy algorithm calculates R_C by incrementally including the next most salient peak point P. Also when including a salient point P in R_C, we union R_C with a small rectangle centered at P. This is because if P is within the foreground object, it is expected that a small region surrounding P would also contain the object.

This algorithm can be modified to satisfy further requirements. For example, the UNION function in Figure 2 can be altered when the cropped rectangle should have the same aspect ratio as the original image. Rather than just merging two rectangles, UNION needs to calculate the minimum surrounding bounds that have the same aspect ratio as the original image. As another example, the initial value of R_C can be set to either the center of image, S, or the most salient point or any other point. Since the initial point always falls in the result thumbnail, it can be regarded as a point with extremely large saliency. When the most salient point is selected as an initial point, the result can be optimized to have the minimum size. But, we found that to begin the algorithm with the center of images gives more robust and faster results even though it might increase the size of the result thumbnail especially when all salient points are skewed to one side of an image.

Find Cropping Rectangle with Dynamic Threshold

Experience shows that the most effective threshold varies from image to image. We therefore have developed a method for adaptively determining the threshold λ .

Intuitively, we want to choose a threshold at a point of diminishing returns, where adding small amounts of additional saliency requires a large increase in the rectangle. We use an area-threshold graph to visualize this. The X axis indicates the threshold (fraction of saliency) while the Y axis shows the normalized area of the cropping rectangle as the result of the greedy algorithm mentioned above. Here the normalized area has a value between 0 and 1. The solid curve in Figure 3 gives an example of an area-threshold graph.

A natural solution is to use the threshold with maximum gradient in the area-threshold graph. We approximate this using a binary search method to find the threshold in three steps: First, we calculate the area-threshold graph for the given image. Second, we use a binary search method to find the threshold where the graph goes up quickly. Third, the threshold is tuned back to the position where a local maximum gradient exists. The dotted lines in Figure 3 demonstrate the process of finding the threshold for the image given in Figure 1.

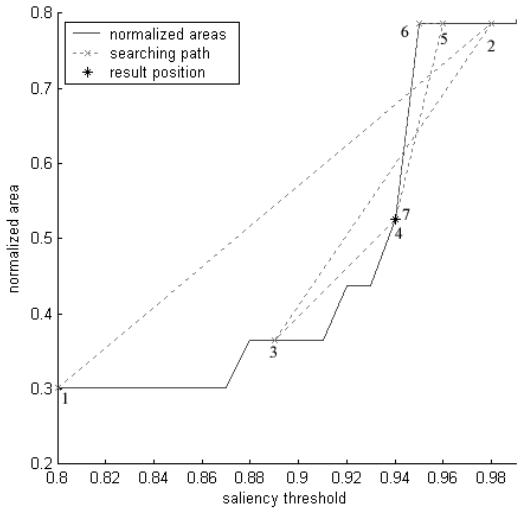

Figure 3: The solid line represents the area-threshold graph. The dotted lines show the process of searching for the best threshold. The numbers indicate the sequence of searching

Examples of Saliency Map Based Cropping

After getting R_C, we can directly crop the input image I. Thumbnails of the image given in Figure 1 are shown in Figure 4. It is clear from Figure 4 that the cropped thumbnail can be more easily recognized than the thumbnail without cropping.

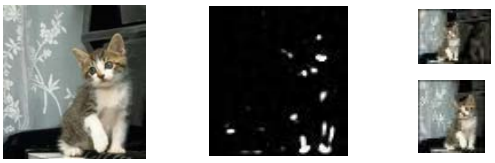

Figure 4 (left): the image cropped based on the saliency map; (middle): the cropping rectangle which contains most of the saliency parts; (right top): a thumbnail subsampled from the original image; (right bottom): a thumbnail subsampled from the cropped image (left part of this figure).

Figure 5 shows the result of an image whose salient parts are more scattered. Photos focusing primarily on the subject and without much background information often have this property. A merit of our algorithm is that it is not sensitive to this.

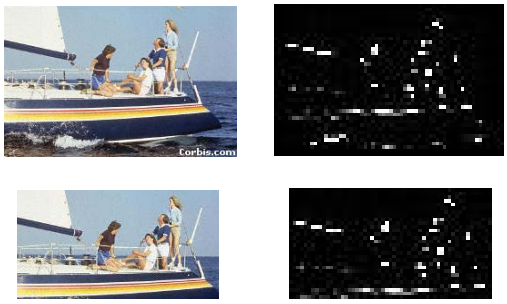

Figure 5 (left top): the original image (courtesy of Corbis [7]); (right top): the saliency map; (left bottom): the cropped image; (right bottom): the cropped saliency map which contains most of the salienct parts.

Face Detection Based Cropping

In the above section, we proposed a general method for thumbnail cropping. The method relies only on low-level features. However, if our goal is to make the objects of interest in an image more recognizable, we can clearly do this more effectively when we are able to automatically detect the position of these objects.

Images of people are essential in a lot of research and application areas. At the same time, face processing is a rapidly expanding area and has attracted a lot of research effort in recent years. Face detection is one of the most important problems in the area. [22] surveys the numerous methods proposed for face detection.

For human image thumbnails, we claim that recognizability will increase if we crop the image to contain only the face region. Based on this claim, we designed a thumbnail cropping approach based on face detection. First, we identify faces by applying CMU's on-line face detection [9][19] to the given images. Then, the cropping rectangle R_C is computed as containing all the detected faces. After that, the thumbnail is generated from the image cropped from the original image by R_C.

Figure 6 (left): the original image; (middle): the face detection result from CMU's online face detection [9]; (right): the cropped image based on the face detection result.

Figure 6 shows an example image, its face detection result and the cropped image. Figure 7 shows the three thumbnails generated via three different methods. In this example, we can see that face detection based cropping method is a very effective way to create thumbnails, while saliency based cropping produces little improvement because the original image has few non-salient regions to cut.

Figure 7: Thumbnails generated by the three different methods. (left): without cropping; (middle): saliency based cropping; (right): face detection based cropping.

USER STUDY

We ran a controlled empirical study to examine the effect of different thumbnail generation methods on the ability of users to recognize objects in images. The experiment is divided into two parts. First, we measured how recognition rates change depending on thumbnail size and thumbnail generation techniques. Participants were asked to recognize objects in small thumbnails (Recognition Task). Second, we measured how the thumbnail generation technique affects search performance (Visual Search Task). Participants were asked to find images that match given descriptions.

Design of Study

The recognition tasks were designed to measure the successful recognition rate of thumbnail images as three conditions varied: image set, thumbnail technique, and thumbnail size. We measured the correctness as a dependent variable.

The visual search task conditions were designed to measure the effectiveness of image search with thumbnails generated with different techniques. The experiment employed a 3x3 within-subjects factorial design, with image set and thumbnail technique as independent variables. We measured search time as a dependant variable. But, since the face-detection clipping is not applicable to the Animal Set and the Corbis Set, we omitted the visual search tasks with those conditions as in Figure 8. The total duration of the experiment for each participant was about 45 minutes.

Thumbnail Technique	Animal Set	Corbis Set	Face Set
Plain shrunken thumbnail	√	√	√
Saliency based cropping	√	√	√
Face detection based cropping	X	X	√

Figure 8: Visual search task design. Checkmarks (√) show which image sets were tested with which image cropping techniques.

Participants

There were 20 participants in this study. Participants were college or graduate students at the University of Maryland at College Park recruited on the campus. All participants were familiar with computers. Before the tasks began, all participants were asked to pick ten familiar persons out of fifteen candidates. Two participants had difficulty with choosing them. Since the participants must recognize the people whose images are used for identification, the results from those two participants were excluded from the analysis.

Image Sets

We used three image sets for the experiment. We also used filler images as distracters to minimize the duplicate exposure of images in the visual search tasks. There were 500 filler images and images were randomly chosen from this set as needed. These images were carefully chosen so that none of them were similar to images in the three test image sets.

Animal Set (AS)

The "Animal Set" includes images of ten different animals and there are five images per animal. All images were gathered from various sources on the Web. The reason we chose animals as target images was to test recognition and visual search performance with familiar objects. The basic criteria of choosing animals were 1) that the animals should be very familiar so that participants can recognize them without prior learning; and 2) they should be easily distinguishable from each other. As an example, donkeys and horses are too similar to each other. To prevent confusion, we only used horses.

Corbis Set (CS)

Corbis is a well known source for digital images and provides various types of tailored digital photos [7]. Its images are professionally taken and manually cropped. The goal of this set is to represent images already in the best possible shape. We randomly selected 100 images out of 10,000 images. We used only 10 images as search targets for visual search tasks to reduce the experimental errors. But during the experiment, we found that one task was problematic because there were very similar images in the fillers and sometimes participants picked unintended images as an answer. Therefore we discarded the result from the task. A total of five observations were discarded due to this condition.

Face Set (FS)

This set includes images of fifteen well known people who are either politicians or entertainers. Five images per person were used for this experiment. All images were gathered from the Web. We used this set to test the effectiveness of face detection based cropping technique and to see how the participants' recognition rate varies with different types of images.

Some images in this set contained more than one face. In this case, we cropped the image so that the resulting image contains all the faces in the original image. Out of 75 images, multiple faces were detected in 25 images. We found that 13 of them contained erratic detections. All erroneously detected faces were included in the cropped thumbnail sets since we intended to test our cropping method with available face detection techniques, which are not perfect.

Thumbnail Techniques

Plain shrinking without cropping

The images were scaled down to smaller dimensions. We prepared ten levels of thumbnails from 32 to 68 pixels in the larger dimension. The thumbnail size was increased by four pixels per level. But, for the Face Set images, we increased the number of levels to twelve because we found that some faces are not identifiable even in a 68 pixel thumbnail.

Cropping Technique and Image Set		Ratio	Variance
Saliency based cropping	Corbis Set	61.3%	0.110
	Animal Set	53.9%	0.127
	Face Set	54.3%	0.128
	All	57.6%	0.124
Face detection based cropping (Face Set)		16.1%	0.120

Figure 9: Ratio of cropped to original image size.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.