Creating Better Thumbnails
CS 638 Project Paper

Chris Waclawik
cwaclawik@wisc.edu

May 13, 2009

Abstract

When a user wants to find a particular image in a set, they will often scan a table of thumbnails instead of flipping through the full-sized images. The smaller the thumbnails, the more images can be displayed on one screen, and the shorter it (theoretically) takes to find a particular image. Once a thumbnail is small enough, however, the loss of detail can make it difficult to recognize the the original image, lessening the thumbnails effectiveness. A smarter way would be to first select the the most recognizable, or salient, part of the image, and then shrink it. This project implements a thumbnail creator that creates a saliency map for a given image, and crops/scales it down to a specified size.

Introduction and Motivation

Computer screens have a limit on the number of full-resolution images they can display simultaneously. If we consider, for example, a 4" x 6" printed photograph, the average laptop screen could display three vertically oriented photos. Given a large set of images, it is impractical to look through each of them one by one to find a specific image. It is sometimes possible to use image metadata to accelerate this search, such as file names, the date the photo was taken, or user-supplied image tags. In most cases, however, this metadata has to be supplied by the user (barring continued advances in computer vision, specifically object detection and recognition). Furthermore, supplying this metadata requires the user to go through the entire image set one by one, which becomes impractical once the set grows large enough.

Thumbnailing is a common technique used to assist the navigation of large image set. A traditional thumbnail is a scaled, smaller copy of an existing image. Their reduced size allows more thumbnails to fit concurrently on the screen than full-sized images. They are found in nearly any application that handles images such as iPhoto, Adobe Photoshop Album, and Picasa. Web applications like Flickr, Google Image Search, and Facebook, use thumbnails to save bandwidth:

a full-sized image will be downloaded from the server only if it's the specific image a user requested [3]. Most operating systems now include thumbnails by default when viewing a folder containing images.

Traditional thumbnails are very computationally cheap to compute: the image needs to be scaled and (optionally) interpolated. This technique, however, exhibits shortcomings when one tries to fit too many thumbnails on the screen: the smaller an image, the more difficult it is to recognize details within it. Once a thumbnail is small enough that the original image can't be quickly recognized from it, the thumbnail fails to serve its purpose. Is it possible, however, to create thumbnails in another fashion, so that the original image can be recognized (even as thumbnail size diminishes)?

The key to answering this question is realizing that not every part of a picture is required to identify it, and the some parts will be more recognizable, or "salient," than others. On the other hand, this is the same thing that makes the problem very difficult: just how do we determine what portions of an image are the most salient? And furthermore, how do we preserve the most salient features when creating a thumbnail?

Related Work

Various methods have been devised to compute image saliency. A good overview of visual attention (and modelling it computationally) can be found in [5]. Most methods can be characterized as "top-down" or "bottom-up." The bottom-up approach relies on low-level, image-dependent cues. These methods, including [6], [9] and [10], rely on contrast to compute saliency. The top-down approach requires information about high-level features within the image, and, optionally, about the task that is being accomplished (e.g., recognizing faces). These approaches require complex object detection and are currently not as general-purpose as the bottom-up method.

Once the salient portions of an image have been determined, there are several proposed methods for preserving these portions. One method uses fish-eye view warping to enlarge the most salient part of an image while preserving the rest of it (albeit distored) [8]. Other methods "push" the salient regions closer to one another [11], [1]. The simplest method, however, is to use the saliency map to find the best crop for an image before creating a thumbnail [12]. This implementation is based on this method.

Method

The thumbnail creation process consists of two parts:

1. Creating the image saliency map.
2. Cropping the image.

Figure 1: The original image.

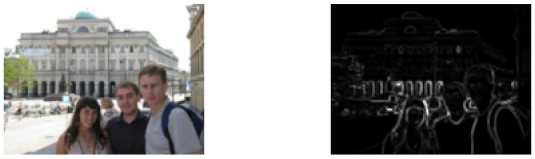
Creating the image saliency map

The method for creating an image saliency map is based on work by Liu et al. [9]. The method first creates a scale-invariant saliency map, then enhances this map using region data for the image. The scale-invariant map requires creating a contrast map at varying image scales; the contrast map at each scale will highlight the most salient features matching that scale. The individual steps, illustrated in figure 2, are as follows:

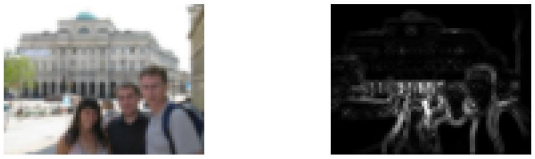
- Convert the image to LUV colorspace. LUV is perceptually uniform, so the distance between two points in the colorspace corresponds to the difference perceived with human vision.
- Construct a Gaussian pyramid for the image [2].
- Create the contrast map for each level of the pyramid as described by Ma et al. [10]. The map is calculated on a pixel-by-pixel basis with the following equation:

$$
\begin{aligned}
C_{i,j} &= \sum_{q \in \Theta} w_{i,j} d(p_{i,j}, p_q) \\
w_{i,j} &= 1 - \frac{r_{i,j}}{r_{max}}
\end{aligned}
$$

(a) Level 1 (b)



(c) Level 2 (d)



(e) Level 3 (f)

(g) Level 4 (h)

Figure 2: Creating the region-enhanced scale-invariant saliency map. Shown is each level of the Gaussian pyramid and its corresponding contrast map.

Figure 3: The scale-invariant saliency map

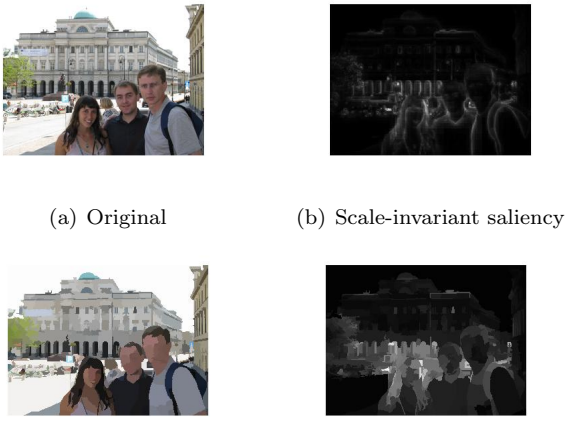
Where C is the contrast of a pixel (i, j), Θ is the neighborhood of the pixel, w is the saliency weight, and d is the L^2 norm of two pixels in LUV colorspace. The saliency weight w depends on r, the distance of a pixel from the center of the image, and r_{max}, the largest possible such distance. The weighting accounts for the fact that the salient portions of an image are often close to the center.

- Rescale all the contrast maps back to their original size using nearest-neighbor.
- Add all the contrast maps together to create the scale-invariant saliency map.

The saliency map is then enhanced using region information. This information can be calculated using whichever preferred image segmentation method; this paper uses mean shift [4] to compute image regions, as is described in the original paper [9]. Existing MATLAB code was found online and used with permission [7]. The saliency of a region is defined as the average scale-invariant saliency over the region.

The region-enhanced salinecy can be calculated with the following equation:

$$S_{i,j} = \frac{1}{\|q \in R_{i,j}\|} \sum_{q \in R_{i,j}} T_q$$



(a) Original (b) Scale-invariant saliency

(c) Mean shift (d) Region-enhanced saliency

Figure 4: Enhancing the saliency map with region data.

Where S is the region-enhanced saliency of a pixel (i, j), $R_{i,j}$ is the region the pixel belongs to, and T is the scale-invariant saliency of a pixel.

An example of the region-enhancing process is shown in figure 4.

Cropping the image

Once the saliency map for an image is computed, the problem remains of selecting the optimal crop. We define the saliency threshold of a thumbnail as the sum of the saliency in the thumbnail divided by the total saliency of the image. Given a saliency treshold λ and thumbnail output dimensions (w, h) we want to find the smallest possible crop (proportionate to w and h) that has a saliency threshold greater than λ.

Finding this optimal solution is computationally expensive, but an approximate solution can be found using a simple, greedy heuristic. The regions are sorted by saliency, and each region is added to the crop one by one until the threshold saliency is reached. In more detail, while the cropped image's saliency threshold is less than λ:

- $P \leftarrow$ most salient unused region
- $R' \leftarrow$ smallest box containg P
- $R_C \leftarrow R_C \cup R'$
- Grow R_C as necessary to have the correct proportions
- Calculate the saliency treshold

Experimental Results

Figure 2 shows an example of the calculations for scale-invariant saliency, and figure 4 shows how this is combined with region data to create the region-enhanced saliency map. The mean shift algorithm has a number of parameters that can be tweaked by the user, most notably "minimum region area," the smallest number of pixels possible in a region. For the most part, however, the default values produced satisfactory results and were left unaltered.

Figure 5 shows the a subset various stages of the cropping process for three images. During each iteration, the red box represents the minimum bounding box for the saliency regions included so far, and the green is that box expanded to meet the target aspect ratio. For the first image (a-f), if we let $\lambda \approx .5$, we get a good crop highlighting the faces in the photo; expanding the box to meet our target aspect ratio prevents parts of the faces from getting chopped off.

The second image (g-l) shows the process for a different input image; the target aspect ratio has also been changed to square. Once again, we seem to get optimal results when $\lambda \approx .5$

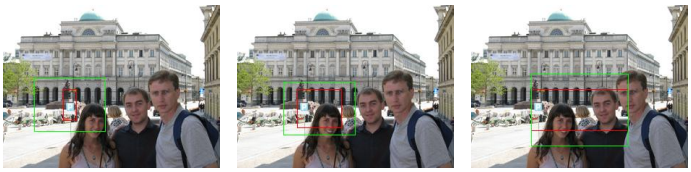
The third image (m-r), however, shows that we cannot chose the same λ for every thumbnail. If we let $\lambda \approx .5$, we'll end up with a crop that does not include the entire boat. A better result is obtained when we let $\lambda \approx .8$. It appears that if the saliency is extremely concentrated in one portion of the image, we must increase λ to ensure that all salient portions will appear in the thumbnail.

Concluding Remarks

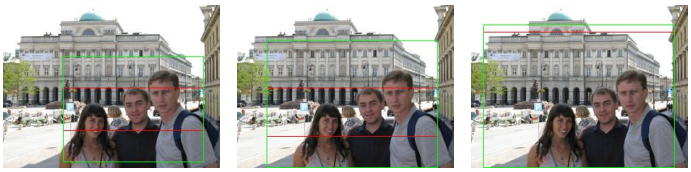
There are a number of modifications that could enhance the performance of this method. One would be to experiment with different algorithms for determining saliency. Our algorithm consists of two replaceable components: a different saliency algorithm could be used to highlight low-level features. One could also use a different image segmentation algorithm when computing thesaliency map. It is possible that there is a method better than mean shift at preserving "regions of interest." Furthermore, the mean shift method doesn't account for any high-level features: a more intelligent method would recognize salient objects (e.g., faces) and give them extra weight.

A better method would find the optimal λ automatically instead of requiring the user to supply it. One way to do this would be a gradient search over the threshold/area plot [12]. Figure 6 shows two such plots for the picture of the bridge. If we consider the plot on the left first, the cropping process moves from left to right. A steep slope means the algorithm significantly increased the area of the crop without increasing the saliency by much. One heuristic would be to select the crop immediately preceeding the steepest slope. A similar idea can be applied to the second plot (although this one shows the minimum saliency included in the crop). The difference is that the algorithm preceeds from high minimum saliencies to lower ones.

Most importantly, the effectiveness of this method needs to be tested with



(a) λ=.248 area=.109 (b) λ=.298 area=.110 (c) λ=.440 area=.200

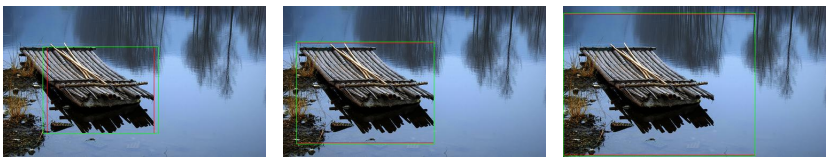


(d) λ=.670 area=.422 (e) λ=.823 area=.618 (f) λ=.926 area=.777

(g) λ=.188 area=.037 (h) λ=.291 area=.068 (i) λ=.392 area=.111

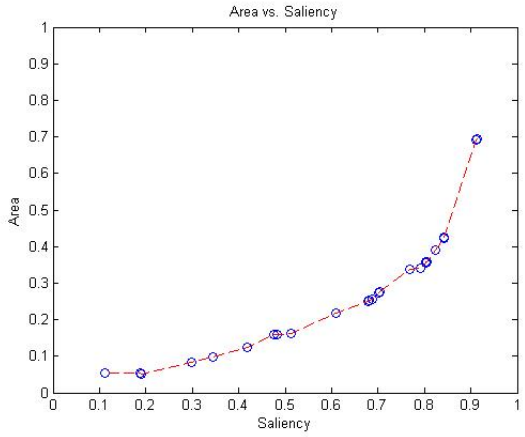
(j) λ=.543 area=.198 (k) λ=.659 area=.303 (l) λ=.733 area=.381

(m) λ=.201 area=.053 (n) λ=.417 area=.124 (o) λ=.609 area=.217

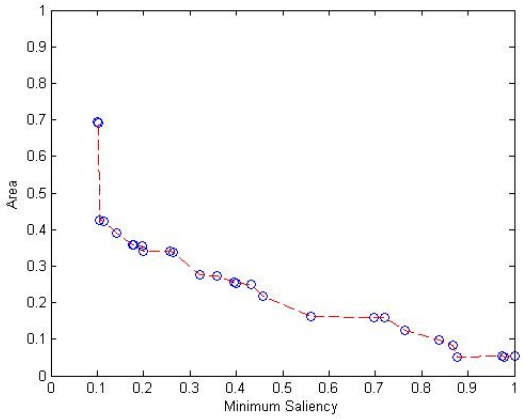


(p) λ=.681 area=.254 (q) λ=.802 area=.356 (r) λ=.914 area=.694

Figure 5: Various stages of the cropping process. The area has been normalized (so that the area of the orginial image is 1. The minimal bounding box is in red, while the box in green has been expanded to match the aspect ratio of the target output.



(a) Area vs. λ



(b) Area vs. Minimum Saliency

Figure 6: Determining the optimal crop

actual users. The ultimate goal of thumbnails is to aid in image navigation and selection; no matter how good the preliminary results, the algorithm has little value if it does not aid users in performing these tasks more efficiently. One such study has already been performed [12], and showed a significant reduction in browsing time (when using their own similar thumbnail cropping method).

References

[1] Avidan, S., and Shamir, A. Seam carving for content-aware image resizing. *ACM Trans. Graph. 26*, 3 (2007), 10.

[2] Burt, P. J., and Adelson, E. H. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications COM-31,4* (1983), 532–540.

[3] Burton, C., Johnston, L., and Sonenberg, E. Case study: an empirical investigation of thumbnail image recognition. *Information Visualization, IEEE Symposium on 0* (1995), 115.

[4] Comaniciu, D., and Meer, P. Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 24*, 5 (2002), 603–619.

[5] Itti, L., and Koch, C. Computational modelling of visual attention. *Nature Review Neuroscience 2*, 3 (March 2001), 194–203.

[6] Itti, L., Koch, C., and Niebur, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 11 (1998), 1254–1259.

[7] Lankton, S. http://www.shawnlankton.com/2007/11/mean-shift-segmentation-in-matlab/, 2009.

[8] Liu, F., and Gleicher, M. Automatic image retargeting with fisheye-view warping. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology* (New York, NY, USA, 2005), ACM, pp. 153–162.

[9] Liu, F., and Gleicher, M. Region enhanced scale-invariant saliency detection. *Multimedia and Expo, IEEE International Conference on 0* (2006), 1477–1480.

[10] Ma, Y.-F., and Zhang, H.-J. Contrast-based image attention analysis by using fuzzy growing. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia* (New York, NY, USA, 2003), ACM, pp. 374–381.

[11] Setlur, V., Takagi, S., Raskar, R., Gleicher, M., and Gooch, B. Automatic image retargeting. In *MUM '05: Proceedings of the 4th international conference on Mobile and ubiquitous multimedia* (New York, NY, USA, 2005), ACM, pp. 59–68.

[12] Suh, B., Ling, H., Bederson, B. B., and Jacobs, D. W. Automatic thumbnail cropping and its effectiveness. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology* (New York, NY, USA, 2003), ACM, pp. 95–104.