A Method for Resizing Images by Content Perception

Anish Patankar, Joy Bose
Samsung R&D Institute, Bangalore 560037, India
{anish.p, joy.bose}@samsung.com

Abstract

With the popularity of mobile devices having smaller screens, it is important to focus on the clarity of the displayed content for the end users. Currently, there is no link between the layout or size of displayed image and the information content in those images. In this paper, we present a method for optimally arranging and displaying a group of images, where the relative size of the images is proportional to the number of objects in the images, or the number of faces. In our system, the images are resized such that the relative size of the images is proportional to the number of objects in the images, or the number of faces. This ensures that the user gets to view faces or objects in images equally well regardless of the original size of the images. The method can be used for a number of applications in a mobile device, such as the gallery, web browser and video applications. We present implementation details to resize and layout a group of images displayed on a web page in a web browser. We also present results of a user study to see if such an approach might be desirable.

Problem Statement

- When viewing images in an image application such as the Gallery app, the relative size of the images is not related to the information contained in those images.
- This leads to a situation where the user may not be able to see clearly those images where there is a lot of information e.g. more people in the image.
- This problem is more in devices where the display size is limited, for example on smaller mobile phones or wearable smartwatches.

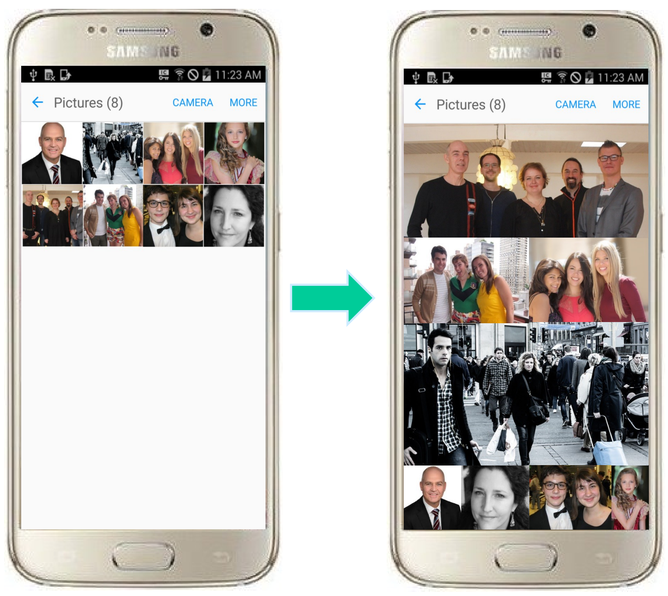


Photo gallery display: Resize as per the relative number of people, Better utilization of space and convey more information to user

Related Work

- Automatic photo organization frameworks (Abdel-Mottaleb, ICME 2004, Cooray, VIE 2006).
- Book by Shao et al (Multimedia Interaction and Intelligent User Interfaces, Springer, 2010) mentions dynamic layouts of photos.
- Wang (SPIE 2013) discuss various layouts in image search results based on clustering similar images.
- Luo et al (PICS 2003) mention selection of salient images in a group of images based on criterion such as size, sharpness, skin area etc.

Object detection and counting is, however, missing as a criterion in these works.

Algorithm

- Scale the images so that perceptible objects or faces occupy same amount of screen area

ALGORITHM 1: DETERMINATION OF IMAGE SIZES

1. Run face and object detection algorithm on each image in the given set of images and determine bounding rectangles of the faces and objects
2. After all images have been processed, determine mean size (*m*) of faces/objects in all the images of the set.
3. Compute scaling factor for i^{th} image as m /m_i, where m_i is mean size of faces in i^{th} image.
4. Scale each image in the set by the scaling factor determined in step 3.

Layout

- Arrange resized images in the same display area
- Essentially a knapsack problem
- Practically we go for bucket approach as described below.
- Computationally lightweight and gels well with modern designs giving good user experience

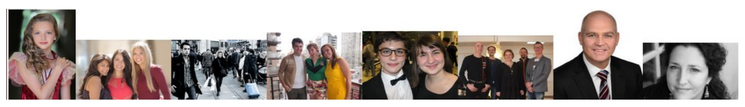
ALGORITHM 2: LAYOUTING IMAGES: BUCKET APPROACH

1. Determine number of buckets (*n*) and image bounding box sizes for each bucket (s_0, s_1, .. s_n)
2. For each image in the given set, determine which bucket it falls into, depending on the size of the image and which bucket's bounding box range it falls into
3. Resize the image to fit into the bounding box range for the selected bucket
4. Layout and display all the images of bucket *1*, then all images of bucket *2*, and so on till bucket *n*

Implementation details

We execute JavaScript to grab IMG elements from the webpage. We apply a simple rule based filter to grab only the thumbnails to exclude logos and ads and other image elements.

We execute a face and object detection algorithm on the images, which returns co-ordinates and sizes of the bounding boxes for detected objects and faces.



Images arranged horizontally at original aspect ratio and same scaling factor

Images arranged horizontally at original aspect ratio but adaptive scaling factor

The size estimator module computes the most suitable size for a given image based on detected objects within the image.

The layout engine puts the resized images into a layout such that overall dimensions of all images put together are maintained as the original.

A component called styler applies styles to specific elements i.e. actual images and their containers to achieve the overall desired effect using DOM APIs along with styles.

We have used a version of the YEF real time object detection algorithm as explained by Abramson et. al [3]. and implemented by LiuLiu [4] as a Javascript library, using canvas and DOM APIs.

User Study

We conducted a study on 330 people to ascertain their preference for the feature.

- Most people preferred the resizing of images
- Most of the users found the current image sizes to be too small for comfort, even though most people experienced little or no eye strain while viewing the images.

Use Cases

This method can be used to resize images in proportion to number of people, in:

- A video conference

- Image related SNS applications like Pinterest

- Image search results

Conclusion

In this paper we have presented the design of a system to resize images as per the number of objects or faces in each image. This system has been implemented for faces in a webpage using existing face detection plugins in a web browser.

Future Work

In future, we plan to extend the implementation of our system for objects other than faces and for other applications on mobile devices .

Selected References

1. Alexe, Bogdan, Thomas Deselaers, and Vittorio Ferrari. "What is an object?." Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010
2. Abdel-Mottaleb M, Chen L. Content-based photo album management using faces' arrangement. InMultimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on 2004 Jun 27 (Vol. 3, pp. 2071-2074). IEEE.
3. Abramson, Y., Steux, B. and Ghorayeb, H., 2007. Yet Even Faster (YEF) real-time object detection. International Journal of Intelligent Systems Technologies and Applications, 2(2-3), pp.102-112.
4. LiuLiu. JavaScript Face Detection Explained. Feb 19, 2012. Available: liuliu.me/eyes/javascript-face-detection-explained/
5. Anish Anil Patankar, Joy Bose, "Electronic device for displaying a plurality of images and method for processing an image", WIPO Patent WO 2016204449 A1, filed 9 June 2016.