Interactive Exploration of Image Collections on Mobile Devices

Gerald Schaefer[1], Matthew Tallyn[1], Daniel Felton[1], David Edmundson[1], and William Plant[2]

[1] Department of Computer Science, Loughborough University, Loughborough, U.K.
gerald.schaefer@ieee.org
[2] School of Engineering and Applied Science, Aston University, Birmingham, U.K.

Abstract. Image collections are ever growing and hence visual information is becoming more and more important. Moreover, the classical paradigm of taking pictures has changed, first with the spread of digital cameras and, more recently, with mobile devices equipped with integrated cameras. Clearly, these image repositories need to be managed, and tools for effectively and efficiently searching image databases are highly sought after, especially on mobile devices where more and more images are being stored. In this paper, we present an image browsing system for interactive exploration of image collections on mobile devices. Images are arranged so that visually similar images are grouped together while large image repositories become accessible through a hierarchical, browsable tree structure, arranged on a hexagonal lattice. The developed system provides an intuitive and fast interface for navigating through image databases using a variety of touch gestures.

1 Introduction

Due to advances in camera technology and the associated drop of equipment costs, many people nowadays own personal image collections of hundreds to thousands of images. Moreover, a significant proportion of these images is taken with mobile devices rather that bespoke cameras. Obviously these collections need to be managed so that the user is able to retrieve certain images at ease. However, as users typically refrain from annotating their image collections with e.g. keywords or descriptions [1], to automate this is a non-trivial task.

Common tools display images in a one-dimensional linear format where only a limited number of thumbnail images are visible on screen at any one time, thus requiring the user to search back and forth through thumbnail pages to view all images. Obviously, this constitutes a time consuming, impractical and exhaustive way of searching images, especially in larger collections. Furthermore, the order in which the pictures are displayed is based on attributes like file names that often do not reflect the actual image contents and hence cannot be used to speed up the search.

R. Huang et al. (Eds.): AMT 2012, LNCS 7669, pp. 288–296, 2012.

Recently, several approaches have been introduced which provide a more intuitive interface to browsing and navigating through image collections [2,3]. In this paper, we introduce such an image browsing system that allows for interactive exploration of image repositories on mobile devices such as smartphones. Since nowadays a large percentage of images are captured using and stored on such devices, the presented approach fills a highly needed gap of managing these collections. Our approach presents an image database in such a way that visually similar images (images of similar colour) are grouped together while large image collections are managed on a browsable hierarchical tree structure, arranged on a hexagonal lattice. User interaction is performed through various touch gestures leading to an intuitive way of exploring image datasets. An initial user study confirms the usefulness of the developed approach.

2 Image Browsing

Image database navigation systems have been shown to provide an interesting and useful alternative to image retrieval systems [2,3]. The idea here is to provide a visualisation of a complete image collection together with browsing tools for an interactive exploration of the database.

Visualisation methods for image repositories can be grouped into three main categories: mapping-based, clustering-based, and graph-based approaches [3,4]. Mapping-based techniques employ dimensionality reduction techniques to map high-dimensional image feature vectors to a low-dimensional space for visualisation. Typical examples examples use principal component analysis (PCA) [5,6], multi-dimensional scaling (MDS) [7], or non-linear embedding techniques [8] to define a visualisation space onto which to place images. Clustering-based visualisations group visually similar images together, often in a hierarchical manner. Among the systems that employ such an approach are [9,10,11,12,13,14]. In graph-based navigation systems, images are the nodes of a graph structure, while the edges of the graph show relationships between images (e.g., joint keywords or visual similarity). Graph-based systems include [15,16,17,18].

Once a database has been visualised, it should then be possible to browse through the collection in an interactive, intuitive and efficient way [3,19]. We can distinguish between horizontal browsing which works on images of the same visualisation plane, and includes operations such as panning, zooming, magnification and scaling, and vertical browsing which allows navigation to a different level of a hierarchically organised (often clustering-based) visualisation.

3 Mobile Image Browsing

In this paper, we present an effective and efficient tool for exploring and querying large image collections on mobile devices. Our system provides users with a visualisation of images generated solely from the image pixel data, overcoming the reliance on metadata. Image thumbnails are displayed on a hexagonal lattice, based on mutual colour similarity of the images. Users can effortlessly navigate through the visualisation using a variety of browsing operations to effectively explore the image collection. Large image collections are visualised in a hierarchical manner so that images are accessed in different layers, while usage of the visualisation space is maximised through the application of image spreading algorithms.

Arranging visually similar images close together within a visualisation space has been shown to support more efficient image retrieval when compared to retrieval from a random arrangement of images [20]. Similarly, it has been shown that image overlap, often occurring in mapping-based visualisations, leads to user confusion and that a regular arrangement of images supports faster navigation [20].

We follow these principles and arrange images according to their mutual colour content similarity. More precisely, we utilise the median colour in HSV space (which better correlates with human perception than RGB) of an image as an image descriptor, from which a position in the visualisation space is derived using the value V (describing brightness) and hue H descriptors. Hue is the perceptual attribute usually associated with 'colour' and is defined in HSV as [21]

$$H = \cos^{-1} \frac{0.5[(R-G)+(R-B)]}{\sqrt{(R-G)(R-G)+(R-B)(G-B)}} \tag{1}$$

where R, G, and B are red, green and blue pixel values. Hue constitutes an angular attribute, going from red to yellow to green to blue back to red. Each image is thus characterised by its H and V values, which in turn directly define the co-ordinates of the image within the (2-dimensional) visualisation space by mapping H to the x- and V to the y-axis. While the features employed might at first seem simple, average colour descriptors have been shown to work at least as well as higher-dimensional (and computationally more complex) image descriptors, such as colour distributions (histograms), for image database browsing [20].

Following [20], we also organise images on a regular lattice with no image overlap; however in contrast to there we do not employ a grid structure but rather follow [22] and utilise a hexagonal lattice onto which the images are placed. This has the advantage that, when the images are organised in a space-filling arrangement, each row and column of images is visually displaced from its neighbouring rows/columns. This would not be possible using a regular square grid structure where larger visual gaps are needed to delineate images clearly. The space saved as a result of using a hexagonal lattice enables larger or more images to be displayed within the visualisation. In addition, on a hexagonal lattice, the six neighbours of a hexagon are equidistant from the middle cell while on a square architecture, the neighbours at the diagonal are further away than the horizontal and vertical neighbouring cells [23]. Clearly, images are not of hexagonal shape and hence need to be cropped for display. This is done by inscribing a hexagon of maximal size at the centre of the image and cropping the part outside this area.

Only a limited number of images from the database can be displayed simultaneously in any visualisation approach. However, even personal image collections on mobile devices may contain hundreds or thousands of pictures. Akin to the approach in [14], we address this problem by employing a hierarchical approach to image database visualisation and browsing. Construction of the hierarchical data structure is performed in a recursive manner. At the root layer (which in our default implementation consists of 17×14 hexagons) we determine for each image into which of the cells it falls. If a cell contains only a single image, this is the image that is displayed in the hexagon. If more than one image is mapped to a cell, the image whose hue/value co-ordinates are closest to the cell centre is displayed as a representative image for that image cluster. Then at the next level of the hierarchy, this cluster is expanded by subdividing the hexagon into a set of smaller hexagons and performing the above procedure (that is, mapping each image in the cluster to a sub-cell, and - should multiple images still fall in the same cell - descend into the next level of the hierarchy) again. The complete image data structure can thus be regarded as a hierarchical tree where intermediate nodes correspond to image clusters and terminal nodes represent individual images. Equivalently, one can envisage the image database as being visualised as layers of different resolutions, with the user being shown one of these layers and having the ability to move up or down to other layers.

While this hierarchical approach allows effective and efficient access even to large image databases, it is inevitable that only a certain amount of cells will actually be filled with images. Similarly, if two images have very similar image descriptors (i.e. have similar median hue and value co-ordinates), such as two photographs of the same scene, these images are likely to be mapped to the same cell even on lower layers of the data structure. Therefore, we employ two image spreading algorithms in order to maximise the available visualisation space and to minimise the number of generated browsing layers.

To reduce the number of empty cells and thus distribute images within the visualisation more evenly, each empty cell in a layer is located. If a given empty cell has 4 or more neighbouring hexagons that contain images, a relative percentage of these images is moved to the empty cell. The images moved from each of the occupied cells will be those with hue and value co-ordinates closest to the hue and value of the centroid of the empty cell (i.e. those images closest to the borders between the two cells) as illustrated in Figure 1. The overall effect of this approach is that more cells are filled, hence making better use of the visualisation space. We repeat this process three times to arrive at a compromise between filling empty cells and maintaining the original positions of the images within the browsing environment.

In order to prevent the creation of too many layers, we employ a second spreading strategy which is designed to distribute images that fall within the same (parent) cell. In particular, we adopt the 'place', 'bump' and 'double-bump' strategies of [20] illustrated in Figure 2 if, for a particular parent node, fewer than 25% of the cells are filled. For cells that contain multiple images, the idea is to keep the representative image in the cell while moving the other images to a nearby cell with little change to the overall configuration. To do so, we initiate a spiral scan around the cell, checking whether any of its 6 neighbouring cells is

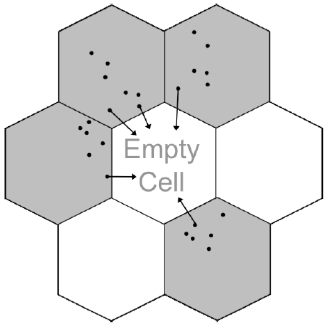

Fig. 1. Five images (represented as black dots) from the 4 neighbouring hexagons (which contain 25 images in total) will be moved to the vacant cell, so that each of the cells contains five images

empty where we can 'place' an image. Should all neighbouring cells be occupied, we move to the next ring (which contains 12 cells) looking for an empty hexagon. If one is found, the image of the closest cell in the first ring is placed there and an image from the original cluster is placed in the first ring cell ('bump'). If necessary, the same principle can be extended to the third ring (which contains 18 hexagons) by moving images from the second to the third and from the first to the second ring before placing an image ('double bump').

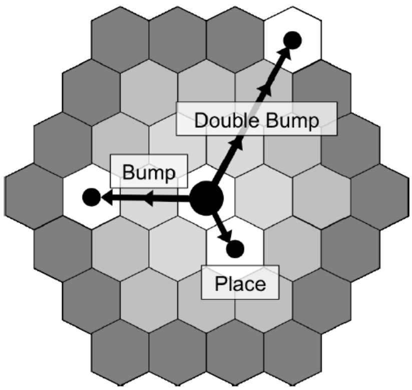

Fig. 2. A graphical representation of our 'place', 'bump' and 'double-bump' spreading strategies

While an effective visualisation paradigm is important for intuitive image database navigation, it is only through interaction with such a system that its true potential and usefulness becomes apparent. To support intuitive and efficient interaction, our system provides a variety of browsing tools and operations, all based on touch gestures of the mobile device. The user may effortlessly pan the current display through a dragging operation similar to those provided e.g. in document editing/viewing applications. Using such a pan operation, the user is able to focus on a different area in the visualisation space. Panning along the horizontal main axis will bring up images of different hues while panning along the vertical axis allows to explore brighter or darker images. As hue is a circular quantity, panning continuously along the main axis will eventually lead back to the starting point. Panning is supported not only on the root layer but on all layers of the visualisation hierarchy. In lower layers, it is possible to move from images belonging to one parent cell to those corresponding to neighbouring parent cells, thus enabling panning along the entire layer.

A continuous zoom operation is also implemented activated by a two-finger pinch gesture. This allows to zoom in on the current display, hence giving the user the possibility to inspect the images contained more closely, or to zoom out to a view where an overview of all images in the layer is possible.

While the above operations allow interaction within one particular layer of the visualisation (i.e. so-called horizontal browsing [19]), it is vertical navigation (that is, navigation from one layer to the next) that allows for intuitive management of large image collections. As detailed earlier, an image cluster is visualised through the use of a representative image. By double-tapping an image, the respective image cluster is expanded allowing the user to delve further into the browsing structure, and to bring up images that were not shown before.

4 Experimental Results

While the system as described above can of course be implemented on literally any mobile device with a touchscreen, our actual implementation is based on Apple's iOS environment[1] and our browser hence runs on Apple's iPhone and iPod Touch devices.

Figures 3 and 4 show two sample "screenshots" of our application in action. Figure 3 shows the root layer of the system running on a database of about 1400 images [24], while Figure 4 displays the application after a vertical browsing operation, i.e. after the user has selected one of the images on the previous layer to be expanded.

As has been shown [25], an objective evaluation of image browsing systems is difficult if not impossible to perform. We therefore performed an initial, user-based study to gather some feedback. For this, we asked a group of eight computer science undergraduate students to participate and installed the application on their smartphones. The users were hence able to browse through their own private image collections using our browsing systems. The initial feedback received during these sessions was very promising since all users commented that they found the developed application very useful.

In order to provide a more quantitative analysis, and on a larger database, we used the MPEG-7 Common Colour Dataset [26] which contains close to 5500

[1] `http://developer.apple.com/devcenter/ios`

Fig. 3. UCID image database [24] visualised in the image browser on an iPod Touch

images and 10 of its predefined ground truth queries, and count the number of user interactions that are needed to reach, starting from a query image, all its corresponding model images. We compare this to the equivalent number of user interactions that is required to do the same on a common file browsers as well as using the ImageSorter browser [27] which is also based on idea of arranging images by derived content features but on a single visualisation plane.

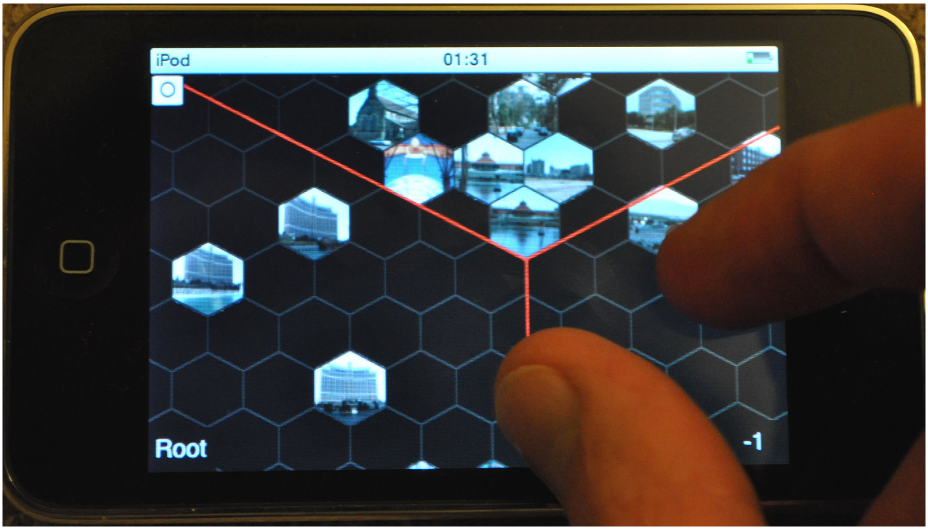

Fig. 4. Image browser after navigating into the next layer of the browsing structure

We found that for our image browser the average number of user interactions was 4.9 compared to 9.4 using ImageSorter and 36.5 using a file browser [22]. This clearly demonstrates the usefulness of our image browsing approach and also shows that employing a hierarchical approach to managing image collections leads to a more effective browsing interface.

5 Conclusions

In this paper, we have presented a content-based image browsing system for image collections stored on mobile devices such as smartphones. Images are described by colour features and arranged so that images with similar features are grouped together in the browsing display. Arrangement is performed on a regular hexagonal lattice which can be browsed through panning and zooming operations using intuitive touch gestures. Larger image collections are accessed through a hierarchical browsing structure akin to clustering where an image cluster can be opened thus revealing similar images not displayed before. An initial user study confirms that our approach provides a useful and intuitive system for managing personal image collections on mobile devices. Clearly, while this is certainly encouraging, a more formal study is necessary and is planned for future research.

References

1. Rodden, K., Wood, K.: How do people manage their digital photographs? In: SIGCHI Conference on Human Factors in Computing Systems, pp. 409–416 (2003)
2. Heesch, D.: A survey of browsing models for content based image retrieval. Multimedia Tools and Applications 40, 261–284 (2008)
3. Plant, W., Schaefer, G.: Visualisation and Browsing of Image Databases. In: Lin, W., Tao, D., Kacprzyk, J., Li, Z., Izquierdo, E., Wang, H. (eds.) Multimedia Analysis, Processing and Communications. SCI, vol. 346, pp. 3–57. Springer, Heidelberg (2011)
4. Plant, W., Schaefer, G.: Visualising image databases. In: IEEE Int. Workshop on Multimedia Signal Processing, pp. 1–6 (2009)
5. Moghaddam, B., Tian, Q., Lesh, N., Shen, C., Huang, T.: Visualization and user-modeling for browsing personal photo libraries. Int. Journal of Computer Vision 56, 109–130 (2004)
6. Keller, I., Meiers, T., Ellerbrock, T., Sikora, T.: Image browsing with PCA-assisted user-interaction. In: IEEE Workshop on Content-Based Access of Image and Video Libraries, pp. 102–108 (2001)
7. Rubner, Y., Guibas, L., Tomasi, C.: The earth mover's distance, multi-dimensional scaling, and color-based image retrieval. In: Image Understanding Workshop, pp. 661–668 (1997)
8. Nguyen, G., Worring, M.: Interactive access to large image collections using similarity-based visualization. Journal of Visual Languages and Computing 19, 203–224 (2008)
9. Krischnamachari, S., Abdel-Mottaleb, M.: Image browsing using hierarchical clustering. In: IEEE Symposium on Computers and Communications, pp. 301–307 (1999)

10. Chen, J., Bouman, C., Dalton, J.: Hierarchical browsing and search of large image databases. IEEE Trans. Image Processing 9, 442–455 (2000)
11. Bartolini, I., Ciaccia, P., Patella, M.: Adaptively browsing image databases with PIBE. Multimedia Tools and Applications 31, 269–286 (2006)
12. Gomi, A., Miyazaki, R., Itoh, T., Li, J.: CAT: A hierarchical image browser using a rectangle packing technique. In: 12th Int. Conference on Information Visualization, pp. 82–87 (2008)
13. Schaefer, G., Ruszala, S.: Image Database Navigation on a Hierarchical MDS Grid. In: Franke, K., Müller, K.-R., Nickolay, B., Schäfer, R. (eds.) DAGM 2006. LNCS, vol. 4174, pp. 304–313. Springer, Heidelberg (2006)
14. Schaefer, G.: A next generation browsing environment for large image repositories. Multimedia Tools and Applications 47, 105–120 (2010)
15. Chen, C., Gagaudakis, G., Rosin, P.: Similarity-based image browsing. In: Int. Conference on Intelligent Information Processing, pp. 206–213 (2000)
16. Heesch, D., Rüger, S.: NNk networks for content-based image retrieval. In: European Conference on Information Retrieval, pp. 253–266 (2004)
17. Dontcheva, M., Agrawala, M., Cohen, M.: Metadata visualization for image browsing. In: 18th Annual ACM Symposium on User Interface Software and Technology (2005)
18. Worring, M., de Rooij, O., van Rijn, T.: Browsing visual collections using graphs. In: Int. Workshop on Multimedia Information Retrieval, pp. 307–312 (2007)
19. Plant, W., Schaefer, G.: Navigation and browsing of image databases. In: Int. Conference on Soft Computing and Pattern Recognition, pp. 750–755 (2009)
20. Rodden, K.: Evaluating Similarity-Based Visualisations as Interfaces for Image Browsing. PhD thesis, University of Cambridge Computer Laboratory (2001)
21. Sangwine, J., Horne, R.: The Colour Image Processing Handbook. Chapman & Hall (1998)
22. Plant, W., Schaefer, G.: Image retrieval on the honeycomb image browser. In: 17th IEEE Int. Conference on Image Processing, pp. 3161–3164 (2010)
23. Sheridan, P., Hintz, T., Alexander, D.: Pseudo-invariant image transformations on a hexagonal lattice. Image and Vision Computing 18, 907–917 (2000)
24. Schaefer, G., Stich, M.: UCID - An Uncompressed Colour Image Database. In: Storage and Retrieval Methods and Applications for Multimedia 2004. Proceedings of SPIE, vol. 5307, pp. 472–480 (2004)
25. Plant, W., Schaefer, G.: Evaluation and benchmarking of image database navigation tools. In: Int. Conference on Image Processing, Computer Vision, and Pattern Recognition, vol. 1, pp. 248–254 (2009)
26. Moving Picture Experts Group: Description of core experiments for MPEG-7 color/texture descriptors. Technical Report ISO/IEC JTC1/SC29/WG11/N2929 (1999)
27. Barthel, K.: Automatic image sorting using MPEG-7 descriptors. In: Workshop on Immersive Communication and Broadcast Systems (2005)