A Configurable Photo Browser Framework for Large Image Collections

Frode Eika Sandnes

Faculty of Engineering, Oslo University College, N-0130 Oslo, Norway
frodes@hio.no

Abstract. Image collections are growing at an exponential rate due to the wide availability of inexpensive digital cameras and storage. Current browsers organize photos mostly chronologically, or according to manual tags. For very large collections acquired over several years it can be difficult to locate a particular set of images – even for the owner. Although our visual memory is powerful, it is not always easy to recall all of one's images. Moreover, it can be very time consuming to find particular images in other peoples image collections. This paper presents a prototype image browser and a plug-in pattern that allows classifiers to be implemented and easily integrated with the image browser such that the user can control the characteristics of the images that are browsed and irrelevant photos are filtered out. The filters can both be content based and based on meta-information. The current version is only employs meta-information which means that large image collections can be indexed efficiently.

1. Introduction

High quality digital cameras are inexpensive and within the reach of most people. Storage capacity has grown exponentially and prices have dropped. Moreover, most mobile devices are equipped with powerful cameras. People therefore take more photographs than ever before, and image collections are therefore growing at an exponential rate.

Most camera owners have a have a gut feeling about what content they have in their personal collection and approximately where they can find particular images. However, for shared image repositories where multiple people contribute images and use the photographs it can be a huge challenge to find exactly the images one is looking for. The literature focuses on three approaches to handling large image collections, namely manual image tagging [1, 2], content based image retrieval [3-5] and image browsing [6-8]. Manual image tagging is the strategy whereupon an image or a collection is manually tagged with keywords or text. Tagged images can be retrieved using text-based queries. Many camera owners also organize their amateur images into folders where each folder is given a descriptive name. Although crude, the folder names can be useful when searching for images. Unfortunately, manual image tagging is very time consuming especially with large collections. Labeling of folders is more manageable than tagging individual photos. Another problem with manual tagging is that it is a subjective process dependent on how the observer perceives the image content, and two different people may tag images differently. In practice, however, people do not voluntary tag their images and several researchers have experimented with games to encourage the manual tagging of images [1, 2]. Several automatic tagging strategies have also been proposed such as the strategy used by the well known service Google image search, whereupon images on web pages are associated with the text on the pages in which they appear. Several strategies are also based on the automatic analysis of image contents.

Content based image retrieval is an active research topic where various approaches for automatically identifying the contents of images have been proposed [3, 9]. Content based image retrieval is a very complex process and the successful examples have been limited to special domains such as medical [5], landmark recognition [9] and general shapes [4] to mention at few.

Image browsing is yet another area that has received some attention, especially in the field of human computer interaction [10-12]. A browser allows the user access to all the images. As the human visual system is good at scanning, browsing appears to be a preferred method of accessing images as the user is in control. However, with large collections even browsing can be a challenge. Most commercial browsers are very simple in their approach, as they organize images chronologically and typically exploit folder names. Browsing applications such as Picasa has also employed techniques from image processing, in particular face recognition, to assist in the image browsing process. The merging of ideas from browsing and image processing is an emerging trend [6-8] and this is also the basis for the approach taken in this paper.

2. Ideation

The background for this work was the author's previous research on techniques for extracting geographical information from image collections [13-16] where the geographical information is extracted from the image meta data stored in the image EXIF headers [10, 17, 18]. This has for instance been applied to automatic tagging of images and providing summaries of collections in the form of image browsers for the blind [19].

The initial work started with a video sketch of a touch based image browsing application for children shown in Fig. 1. In this browser the child will construct a scene or scenario representing the images the child wants to see. The child is represented by a character that can be placed indoors or outdoors, representing indoor images or outdoor images, what time of day, that is, morning, noon, afternoon, evening and night, when the scene is inverted to represent the darkness of night. The weather conditions are indicated by flicking the sun where the user cycles through clear skies, partially cloudy days and cloudy days. Moreover, the season can be indicated by stretching or shrinking the sun. If the sun is shrunk it becomes winter and the house is covered in snow. Also the geographical location of the images is represented by the characters clothing, here represented by a Viking from northern Europe, a cowboy from America, a Chinese outfit, etc. The clothing is changed by flicking the character. The thumbnail images display would automatically change when scenario is changed. This interface is not dependent on text or technical characteristics of images, hence, making the interface suitable for children. The video sketch was created by first making the animated cartoon illustrating the various screens, then played back on an ordinary computer with the hand gestures enacted on top and the resulting scene filmed with a video camera.

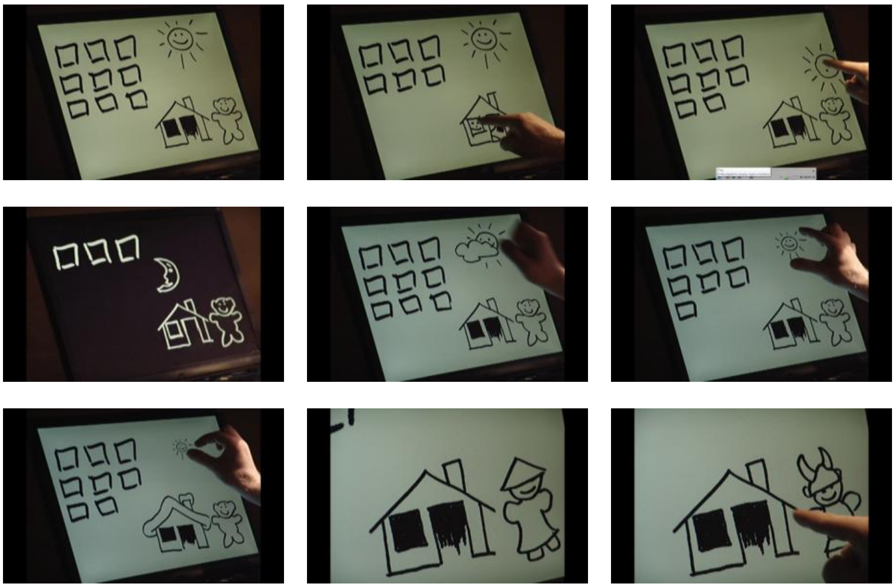

Fig. 1. A video sketch of a photo browser for children. a) the start screen, b) move the character indoors, c) changing the time of day, d) at night, e) chaning the weather conditions, f) shrinkning the sun, e) shrinkning the sun into winter, f) chaning the geographical location into South East Asia, g) chaning the geographical location into Northern Europe.

3. Image browsing framework

Fig. 2 shows an overview of the image browsing framework. The user is exposed to the graphical user interface that can be tailor made for a particular domain such as the children's image browser shown in Fig. 1. The GUI is built on top of the image browsing framework which maintains a database of all the images in a given collection. The image browsing database helps speed up browsing such that all the attributes of interest connected to a given image is easily retrievable and the interface can respond to user queries in real-time. Therefore, each time an image is browsed the system does not need to analyze the image. Images are only analyzed once when loaded into the system the first time.

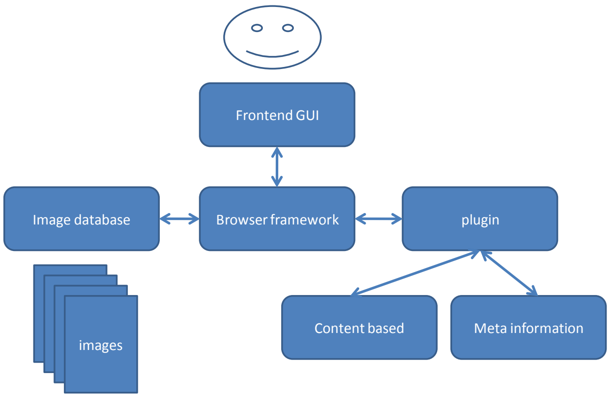

Fig. 2. Framework architecture

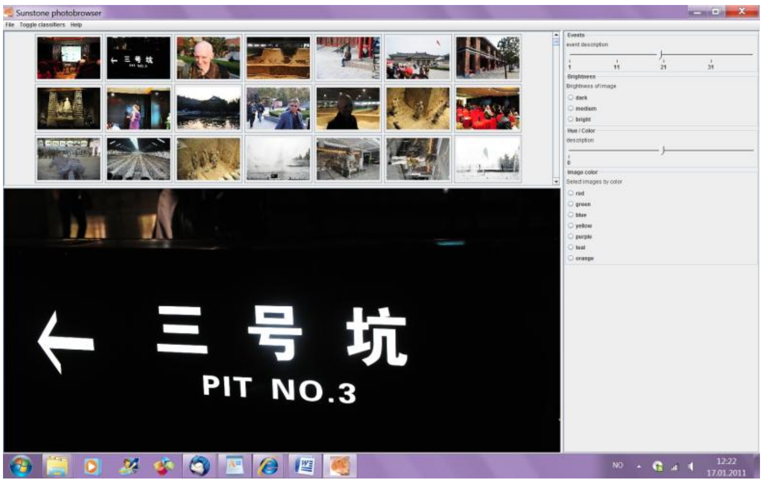

Fig. 3. The photo browser without any criteria selected.

3.1 Content-based versus meta-information based

The plug-ins are written for the framework, and a plug-in can be content based or based on meta-information. Image content based plug-ins typically analyses the image contents as a basis for providing a classification. Content based analysis can be everything from simple analysis of average pixel intensity and mean pixel hue to more complex image analysis such as object recognition. Content analysis is time-consuming, especially with high resolution images as images as analyzed at pixel level. The complexity of the image analysis algorithm will significantly affect the processing time, and the more sophisticated object recognition algorithm for instance, can have a high time-complexity. Moreover, the processing time is proportional to the size of the collections and the processing of a very large collection can therefore be computationally demanding.

Image meta-information can provide a vast range of valuable information that can be analyzed. Modern digital cameras store the state of the camera in so called EXIF headers [10, 17, 18], for example the shutter speed, aperture, film speed (ISO), the time and date the image was taken, lens focal length, etc. Some of these settings such as shutter speed, aperture and film speed can be used to compute more abstract quantities such as the exposure value [20]. The exposure value gives an absolute measurement of the light intensity at the scene and such information cannot be extracted fully from the content of images as the image measurements are relative. Moreover, even more abstract information such as geographical location [16] and descriptive information relating to local time zones [19] can be extracted based on these quantities. One strength of meta-information is that it can be exploited without having to inspect the image contents and is thus not related to the image size. Meta-information analysis is typically much less computationally intensive than content based analysis.

The framework supports a mixture of content and meta-information based image classifiers.

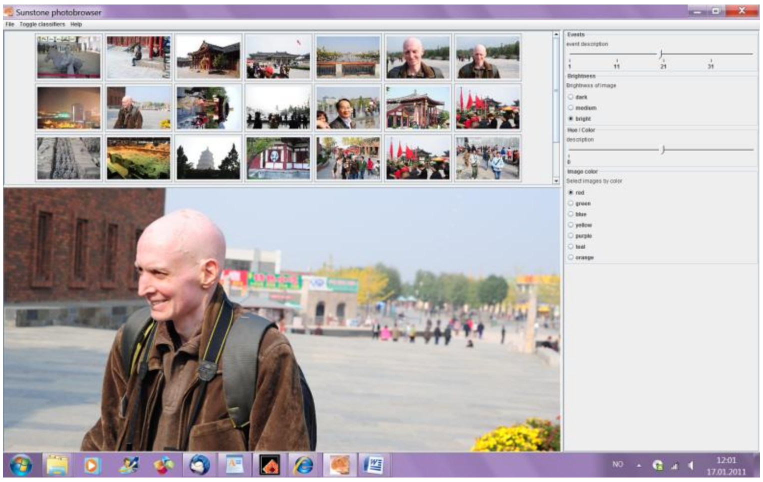

Fig. 4. The photo browser with the settings “bright” and “red”.

3.2 Image versus collections

The framework supports two further types of plug-ins namely individual image classifiers and collection classifiers. An individual image classifier is responsible for analyzing a single image and extracting a particular feature. For example, this could be the overall image intensity, mean hue, the month the image was taken, etc. However, features such as brightness level are characteristics that are only meaningful to connect to single images and not necessarily a collection of images. On the other hand image collection classifiers finds characteristics common to a set of images. For example, images can be grouped according to their temporal spread, that is, images taken within the same few days could be part of the same holiday, while images separated by more than a few days could belong to totally unrelated events. Moreover, images taken within one related event are typically taken in a limited geographical region, time-zone, etc. A collection classifier will thus typically classify groups of images.

3.3 Extendibility

The plug-ins are designed to take an image or image collection as input, that is, they have access to all the image characteristics, contents, and meta-information, as well as the output of other plug-ins, and then produce a single feature as output. A separate plug-in is thus needed for each feature that should be represented in the interface. Therefore, there is a precedence relationship between the plug-ins dictating their dependencies.

Fig. 5. The photo browser with settings “red” and “medium”.

The output of each plug-in is represented as a visual control in the user interface. For the default interface, implemented in the current framework this is either a category represented as a pull down menu, choices represented as radio buttons, or a slider representing a quantity. It is also possible to represent support plug-ins whose values are not displayed in the interface, but provided as input to other classifiers.

Each plug-in is run on each image once to expedite short response times. Once a plug-in has categorized an image the corresponding value is stored in the database. Plug-ins and images can be added dynamically, and once a new image is added, all the plug-ins are applied to that image. Likewise, if a new plug-in is added, the plug-in is applied to all the images.

The framework is illustrated in Figs. 3-5, 7 and 8. The criteria for the images are selected in the rightmost pane, the corresponding selection of images are shown in the top left pane and the selected image at any given time is shown in the bottom left pane.

The framework is written in Java. Plug-ins are also written in Java and the constructors are used to signal whether it is a collection classifier or single image classifier as well as its appearance in the GUI, with explanatory text. Extracts from an example classifier plug-in is shown in Fig. 6.

```
public ClassifierDayNight()
        {
        super( false,
                new Component("Day or night",
                                "Show photos of day or night",
                                new String[]{"Day",
                                                "Night" }. true ));
        }
   Public Object processElement(HashMap<String, Object> input)
        {
        double EV = 0;
                {
                Object tmp = input.get(exposureValueClassifier);
                if(tmp instanceof Double)
                        {
                        EV = (Double) tmp;
                        }
                else
                        {
                        throw new NullPointerException(
                                                "Need Exp. Value");
                        }
                }
        return (EV >= 11);
        }
```

Fig. 6. Extract of a classification filter implementation.

The various plug-ins can be added dynamically by adding their respective class-files to the application. However, they require a restart as the database is checked against the available images and plug-ins only during startup. During run-time the user can activate or de-activate various plug-ins via the user interface. This allows the user to display only the settings that are relevant in a given application and unnecessary clutter is avoided.

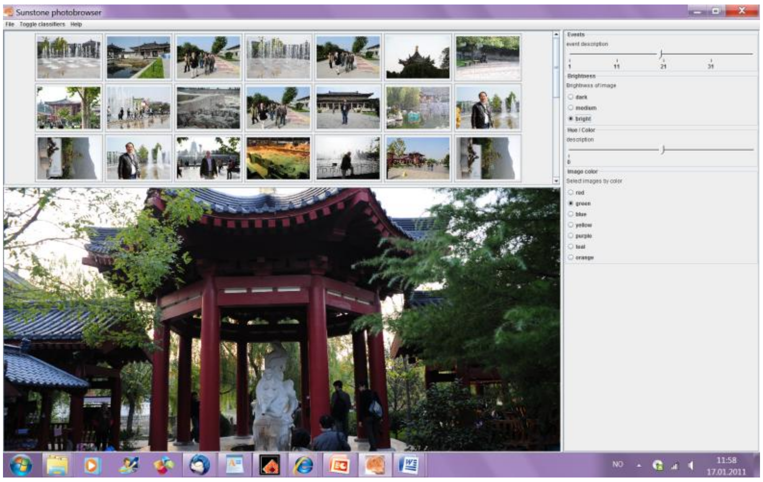

Fig. 7. The photo browser with settings “green” and “bright”.

4. Demonstration

The early prototype has not yet been evaluated on a panel of users, as the set of implemented classifiers are limited and the front end is yet crude. Figs. 3-5, 7 and 8 illustrate the prototype photo browser on a collection of 480 images taken at Xi’an China during the UIC/ATC’2010 conference in October 2010. Fig. 3 shows the browser when started. Initially all the images are displayed as no criteria are selected. In Fig. 4 the browser is set to show “red” images that are “bright”. Red images are images with some amount of red in them. Clearly, there is only a fraction of the images that satisfies these criteria which can be seen from the scrollbar. Next, in Fig. 5 the setting “red” with “medium” intensity is selected. Again, the set of images is much smaller and many of the images are indoors. Fig. 7 and 8 shows the images that are displayed when selecting “green” and “blue” images, respectively – both in the category “bright”.

5. Conclusions

This paper has presented a customizable image browser with an architecture that allows plug-ins to be designed for specific purposes based on image contents or image meta-information, or both. This strategy holds potential for browsing large image collections and especially image collections where the viewer is not the photographer. The strategy bridges techniques and the benefits from traditional image browsing with the recent advances in image processing and analysis. Future work involves testing the framework with novel image sorting categories on target groups such as children.

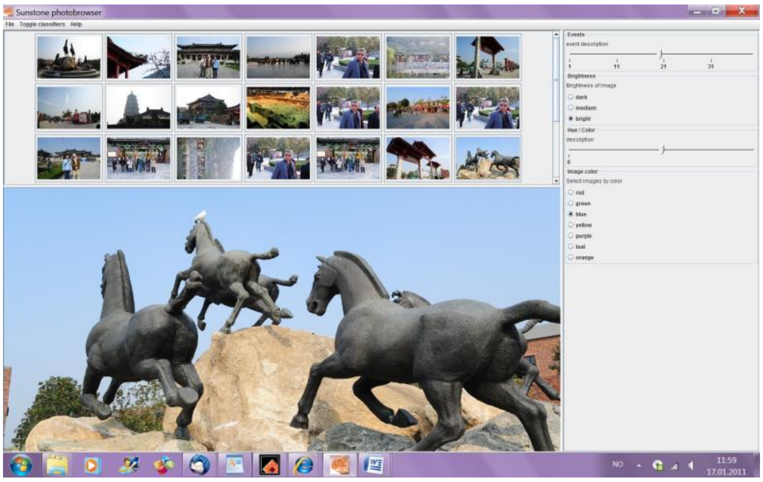

Fig. 8. Image browser with selections “bright”, “blue”.

Acknowledgements

The author is grateful for the hard work of Martin Haukeli, Jostein Haukeli and Marit Olsen who implemented the Sunstone browser and contributed to the design of the plug-in pattern.

References

1. N. Diakopoulos and P. Chiu, Photoplay: A collocated collaborative photo tagging game on a horizontal display., presented at UIST '07, 2007.
2. S. Golder and B. A. Huberman, The structure of collaborative tagging systems, Journal of Information Sciences 32(2), 198-208 (2006).
3. D. Carboni, S. Sanna, and P. Zanarini, GeoPix: image retrieval on the geo web, from camera click to mouse click, presented at Proceedings of the 8th conference on Human-computer interaction with mobile devices and services, 2006.
4. Y.-P. Huang, T.-W. Chang, and F. E. Sandnes, Efficient Shape-Based Image Retrieval Based on Grey Relational Analysis and Association Rules, International Journal of Pattern Recognition and Artificial Intelligence 22(4), 711-732 (2008).
5. G. Quellec, M. Lamard, G. Cazuguel, B. Cochener, and C. Roux, Wavelet optimization for content-based image retrieval in medical databases, Medical Image Analysis 14(2), 227-241 (2010).
6. P. Bruneau, F. Picarougne, and M. Gelgon, Interactive unsupervised classification and visualization for browsing an image collection, Pattern Recognition Letters 43(2), 485-493 (2010).

7. W.-T. Chu and C.-H. Lin, Consumer photo management and browsing facilitated by near-duplicate detection with feature filtering, Journal of Visual Communication and Image Representation 21(3), 256-268 (2010).
8. J. S. Jin, R. Kurniawati, G. Xu, and X. Bai, Using Browsing to Improve Content-Based Image Retrieval, Journal of Visual Communication and Image Representation 12(2), 123-135 (2001).
9. Y.-T. Zheng, Z. Ming, S. Yang, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T.-S. Chua, and H. Neven, Tour the world: Building a web-scale landmark recognition engine, presented at IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), 2009.
10. C.-J. Jang, J.-Y. Lee, J.-W. Lee, and H.-G. Cho, Smart Management System for Digital Photographs using Temporal and Spatial Features with EXIF metadata, presented at 2nd International Conference on Digital Information Management, 2007.
11. G. Schaefer, A next generation browsing environment for large image repositories, Multimedia Tools and Applications 47(1), 105-120 (2010).
12. M. Porta, Browsing large collections of images through unconventional visualization techniques, presented at AVI 2006, 2006.
13. F. E. Sandnes, Sorting holiday photos without a GPS: What can we expect from contents-based geo-spatial image tagging?, Lecture Notes on Computer Science 5879, 256-267 (2009).
14. F. E. Sandnes, Determining the Geographical Location of Image Scenes based on Object Shadow Lengths, Journal of Signal Processing Systems for Signal, Image, and Video Technology (2010).
15. F. E. Sandnes, Unsupervised and Fast Continent Classification of Digital Image Collections using Time, presented at ICSSE 2010, 2010.
16. F. E. Sandnes, Where was that photo taken? Deriving geographical information from image collections based on temporal exposure attributes, Multimedia Systems 16(4-5), 309-318 (2010).
17. P. Alvarez, Using Extended File Information (EXIF) File headers in Digital Evidence Analysis, International Journal of Digital Evidence 2(3) (2004).
18. N. L. Romero, V. V. G. C. G. Chornet, J. S. Cobos, A. A. S. C. Carot, F. C. Centellas, and M. C. Mendez, Recovery of descriptive information in images from digital libraries by means of EXIF metadata, Library Hi Tech 26(2), 302-315 (2008).
19. F. E. Sandnes, A Non-Visual Photo Collection Browser based on Automatically Generated Text Descriptions, presented at The Third International Conference on Software Development for Enhancing Accessibility and Fighting Info-exclusion (DSAI 2010), 2010.
20. S. F. Ray, Camera Exposure Determination, in The Manual of Photography: Photographic and Digital Imaging, R. E. Jacobson, S. F. Ray, G. G. Atteridge, and N. R. Axford, Eds.: Focal Press, 2000.