PhotoCube: Towards Multi-Dimensional Image Browsing

Hlynur Sigurþórsson, Grímur Tómasson,
Björn Þór Jónsson, Laurent Amsaleg

School of Computer Science
Reykjavík University
January 2012

TECHNICAL REPORT / RUTR-CS12001

ISSN 1670-5777

PhotoCube: Towards Multi-Dimensional Image Browsing

Hlynur Sigurþórsson*, Grímur Tómasson*,
Björn Þór Jónsson*, Laurent Amsaleg†

Technical Report RUTR-CS12001, January 2012

Abstract: As the scale of personal photo collections keeps growing, the simple methods commonly used today for photo browsing quickly become inadequate. We describe PhotoCube, a novel 3D photo browser based on the recently proposed ObjectCube data model. The data model allows users to seamlessly involve a large number of meta-data dimensions in each browsing session and captures a rich set of browsing actions. The architecture of the PhotoCube browser is highly extensible, as different browsing modes can be implemented and the user can easily swithch between browsing modes. In this paper, we describe the architecture of the prototype in detail and present the user experience. We then present results from a preliminary user study, which indicate that while experienced photo browser users find the prototype interface to have a steep learning curve, they are excited about the potential of the underlying data model.

Keywords: Photo Browsing; Multi-Dimensional Analysis; User Interface; ObjectCube.

This work is partially supported by Icelandic Research Fund Grant 070005023.

* Reykjavik University, Menntavegi 1, IS-101 Reykjavík, Iceland. bjorn@ru.is
† IRISA–CNRS, Campus de Beaulieu, 35042 Rennes, France. laurent.amsaleg@irisa.fr

PhotoCube: Að margvíðri myndaskoðun

Hlynur Sigurþórsson, Grímur Tómasson,
Björn Þór Jónsson, Laurent Amsaleg

Tækniskýrsla RUTR-CS12001, Janúar 2012

Útdráttur: Þar sem stærð einkamyndasafna fer sífellt vaxandi, verða þær einföldu aðferðir sem almennt eru notaðar í dag við myndaskoðun fljótlega ófullnægjandi. Við lýsum PhotoCube, nýjum þrívíðum myndaskoðara, sem byggður er á margvíða gagnalíkaninu ObjectCube. Gagnalíkanið leyfir notendum að nýta sér margar víddir lýsigagna við skoðun mynda á samfelldan hátt og býður upp á margar skoðunaraðgerðir. Högun PhotoCube myndaskoðarans er mjög sveigjanleg, þar sem hægt er að útfæra marga skoðunarhætti og notandinn getur auðveldlega skipt milli skoðunarhátta. Í þessari tækniskýrslu lýsum við ítarlega högun frumgerðarinnar og notendaviðmótinu. Þá lýsum við niðurstöðum fyrstu notendaprófana, sem benda til þess að þótt vönum notendum þyki viðmót frumgerðarinnar fremur óþjált, þá séu þeir spenntir yfir möguleikum gagnalíkansins.

Lykilorð: Myndaskoðun; Margvíð greining; Notendaviðmót; ObjectCube.

Contents

1 Introduction

Today, photos and images are stored as binary files on computers and are viewed using image browsing applications. The digital switchover has enabled us to move our images from the cumbersome paper albums and shoeboxes onto computers. In theory, this should have enabled us to exploit the computer's power and flexibility to organize and browse our image collections in a more efficient manner. Indeed, myriad image browsers have been developed to help people with the task of browsing their photo collection. Finding images in a large collection, however, can still be as difficult and frustrating a task as before.

1.1 Browsing Digital Images

Consider, for example, the following two plausible browsing scenarios for a relatively large image collection.

Scenario 1 *We want to browse all images that were a) taken somewhere in Europe, b) include our friends and c) have a brightness value in a certain range. This scenario might be interesting if the user had traveled through Europe with friends, and only wanted to view images that were neither over- nor underlit.*

Scenario 2 *We want to browse all images of our family members containing at least one parent and one child, regardless of the time period, grouped by parents, children, and the location where the photos were taken. This scenario might be interesting if the user wanted to study co-occurences of family members in the images.*

The more advanced image browsers on the market support several strategies for organizing and browsing images. Many of them allow the application of tags and other meta-data to images. Some also offer construction of virtual folder hierarchies within the browsers without modifying the underlying folder structure of the images. Furthermore, some more advanced browsers offer automatic analysis of the content, such as face recognition and geographical tagging. Most of these browsers allow users to search for images using user-generated tags and meta-data information, as well as using automatically generated tags. Images are most often presented using a two dimensional grid, but many offer time line browsing and slide shows generation.

Despite their advanced features, none of these browsers adequately supports the two browsing scenarios above, since a) only one categorization (e.g., folders or time-line) can be used at a time, b) relationships between tags are not supported, neither grouping of tags into concepts nor organization of tags into hierarchies, and c) grouping of images based on their contents is not adequately supported. We can therefore draw the conclusion that the digitization of photography has largely

resulted in our computers becoming digital shoeboxes. Today's image browsers lack the flexibility and expressiveness required for browsing large photo collections, even for the simple browsing scenarios above.

1.2 Contributions

In this paper we present the PhotoCube prototype, a novel 3D image browser which is based on the recently proposed ObjectCube multi-dimensional data model. Unlike other image browsers, PhotoCube is flexible and expressive enough to handle the browsing scenarios above, as well as many other similar scenarios. It has the following key properties:

- The graphical user interface supports three browsing dimensions, some or all of which can be used simultaneously. The underlying data model also supports pivoting dimensions on and off the screen, resulting in practical multi-dimensional image browsing.
- The multi-dimensional data model captures a rich set of browsing actions, such as drilling down through tag hierarchies and applying various filters to dimensions to focus on particular sets of images.
- The architecture of PhotoCube is highly extensible, as different plug-ins are used to define the potential actions to take in many circumstances, e.g., when examining a set of images in detail.

In the remainder of this paper, we first consider the state of the art in photo browsing in more detail (Section 2). We then review the ObjectCube data model and its browsing operations (Section 3), before describing the architecture of the PhotoCube prototype (Section 4). We then describe how PhotoCube supports the browsing scenarios above (Section 5), before presenting briefly results from a preliminary user study with experienced photo browser users, which indicate that they find the data model of the browser both useful and engaging (Section 6). Finally, we conclude with a discussion of multiple avenues for future work (Section 7).

2 Background

In this section, we consider the state-of-the-art in image browsing tools. We start by describing the various meta-data types that can be associated with images, and hence potentially used in browsing scenarios. Then we consider both commercial/freeware browsers and research prototypes, before briefly introducing multi-dimensional image browsing.

2.1 Image Meta-Data

Images are typically described using various meta-data attributes, which are commonly categorized by their origin:

Photo Header Attributes: Camera settings and scene information are often recorded by cameras and written into the image files. Examples of such information are shutter-speed, date and time, focal length, exposure compensation, metering pattern, and flash usage.

Calculated Attributes: Various attributes can be calculated directly from the images. These include extraction of elementary characteristics in images such as shapes, colors, and texture, as well as more advanced analysis methods such as face recognition.

User-Generated Attributes These are attributes that a user relates to the image content. The typical form of such attributes is a text tag, that may be linked to the image or a part of the image. This type of attribute, along with some of the photo header attributes, is most commonly used in today's image browsers.

Turning back to the browsing scenarios in the introduction, we observe that the first two conditions of Scenario 1 involve attributes that may either be user-generated or calculated (location could be based on GPS coordinates supplied in the photo header). Neither condition, however, refers directly to particular tags, but rather to a higher-level concept (Europe or friends). The last condition of Scenario 1, however, refers directly to a range of values for a photo header attribute. Scenario 2 refers to similar attributes, but focuses on the grouping of images based on those attributes.

2.2 Current Image Browsers

As discussed above, myriad commercial/freeware image browsers have been developed to help people with browsing scenarios such as those presented in the introduction. In this discussion, we divide these browsers roughly into simple browsers and advanced browsers.

We term simple image browsers those that merely offer browsing of file system folders and viewing of images within them. In these browsers, images can only be categorized by a single criterion; this categorization is performed outside the browser by storing images in folders. Image collections are most often browsed using the folder tree and images presented on a two dimensional grid. Examples include early versions of several photo-specific browsers, as well as traditional file-system browsers. Clearly neither browsing scenario above can be served by such simple browsers.

We term advanced image browsers any commercial or freeware browsers that offer more effective strategies for organizing and browsing images. Many allow application of tags and other meta-data to images. Some also offer construction of virtual folder hierarchies within the browsers without modifying the underlaying structure of the images. Furthermore, some more advanced browsers offer automatic analysis of the content, such as face recognition and geographical tagging. Most of these browsers allow users to search for images using user-generated tags and meta-data information,

as well as using automatically generated tags. Images are most often presented in a similar fashion as in the simple browsers, namely using a two dimensional grid, but many also offer time line browsing and slide show generation.

Examples of advanced browsers are ACDsee, Adobe LightRoom, Apple iPhoto, and Google Picasa. Despite their advanced features, none of these browsers adequately supports the browsing scenarios above, for the following reasons.

First, although images of friends can be tagged with their names, there is no support for describing relationships between tags. The concept of a friend is thus not implicitly understood; what is needed is some means to organize tags into concept hierarchies. An industrious user can try to get around this limitation, for example by adding a specific tags to encode the hierarchy of friends and adding these to all images containing any friends, or by embedding a hierarchy within the textual tags. Such attempts, however, are labor-intensive and rarely sustainable.

Second, searching tags using a numerical range, as is done in Scenario 1, is not supported in these browsers as they only support searching for specific tags. Again, an industrious users may list all tags in the range to retrieve the correct set. This might be viable for small ranges, but it is impractical for large ranges and impossible for irrational values.

Finally, none of these browsers adequately support grouping of images based on contents. Some might support grouping based on a single concept, but Scenario 2 requires three-dimensional grouping. Furthermore, the scenario requires using different parts of the same attribute hierarchy along two different browsing dimensions.

2.3 Research prototypes

Many research projects have considered image browsing and have proposed many interesting methods for browsing. We now review the most relevant research prototypes.

PhotoMesa [Bed01] provides a zoomable interface to multiple directories of images at once, grouping the images from the folders into clusters for maximal use of the screen. With PhotoMesa, however, both browsing scenarios are impossible, as the browser operates solely on the folder structure and only allows filtering by people tags.

Camelis [Fer07] uses co-occurrences of tags in images to deduce relationships, and uses those relationships to facilitate browsing. Camelis may potentially be set up to return the required set of images, but constructing the query to do so would be very complex. Furthermore, images are presented linearly without any categorization.

Scenique [BC09] is conceptually the browser most similar to PhotoCube, as it allows browsing images by orthogonal dimensions (called facets) in 3D browsing rooms.[1] Scenique can in fact handle browsing Scenario 1 quite well, but browsing Scenario 2 is impossible. In Scenique each

[1] In addition to tag hierarchies, Scenique also offers hierarchies based in image similarity, which in turn is based on calculated image characteristics; we plan to add such support to PhotoCube in the future.

facet can only be viewed on a single axis of the three-dimensional space, while we need to view the same facet on more than one axis, and at a different location in the hierarchy.

Thus these research projects do not provide an acceptable solution to image browsing either.

2.4 Multi-Dimensional Image Browsing

Together, the image meta-data attributes can be considered to define a multi-dimensional image hyperspace. Selecting dimensions and placing them together for viewing images is called multi-dimensional browsing. The earliest mention of multi-dimensional photo browsing that we are aware of was by Hardarsson and Jónsson in 2007 [HJ07]. In that work the authors presented a very limited 3D image browser prototype based on the Partiview browser [SL04], originally developed to browse images of galaxies.

In the future work section of [HJ07], however, a wish list of browsing features was presented; most of these features correspond to operations used in on-line analytical processing (OLAP) applications, in the area of business intelligence. The ObjectCube data model, which is built on the traditional OLAP data model, was subsequently developed in [Tóm11]. In that work, the aim was to create an efficient multi-dimensional data model and a browsing engine that could be used as a back-end for multimedia browsers, such as a image- or file-browsers. Since the ObjectCube model is the underlying foundation of PhotoCube, we present the ObjectCube model in some depth in following section.

3 ObjectCube

ObjectCube is a generic multi-dimensional data model[2] and a browsing engine based on the concepts of the well known multi-dimensional analysis. In the ObjectCube data model, browsing operations are used to zoom into the media collection and construct multi-dimensional browsing sets, or cubes, similar to *OLAP* cubes but with objects instead of numerical facts. These cubes are then displayed using the PhotoCube interface.

The ObjectCube model is formally defined in [Tóm11], but in the following we review the model and the browsing engine architecture, focusing on the core aspects that are necessary for understanding the PhotoCube prototype.

[2] Although we focus on image browsing, ObjectCube is not bound to any specific media type.

3.1 The Data Model

The core concepts of *ObjectCube* are objects, tags, tag-sets, hierarchies, and filters. Together, these concepts are used to construct multi-dimensional browsing cubes presented to users. We now describe these concepts to give a better understanding of the ObjectCube model.

Objects: An object is any entity that one might be interested in storing information about, e.g., any file type that can be described by meta-data. The actual data is not stored as a part of the object, only a reference to it.

Tags: A Tag is any meta-data that can be associated with an object. For instance, we might have tags for individuals in a given photo or its brightness value. The ObjectCube prototype provides the four following tag types: Alphanumerical, Numerical, Time and Date.

Tag-sets: A tag-set groups together tags that are in some way cohesive; we can think of a tag-sets as a category for tags. In a tag-sets named "People," for instance, the tags can be the names of people.

Tag-sets give context to their tags and thereby reduce ambiguity greatly by allowing us to distinguish between synonymous tags in multiple tag-sets. For instance, the tag "Mouse" might occur in two separate tag-sets, "Computer equipment" and "Animals". Note that each tag-sets can only contain tags of one type.

Hierarchies: A hierarchy is a tree structure that adds organization to a (non-strict) subset of the tags in a given tag-sets. Each hierarchy belongs to one tag-sets and can only use tags from that tag-sets. One tag from the tag-sets is selected as the root node, whereas other tags from the same tag-sets can be used to extend the tree.

Filters: In ObjectCube, a filter is a very important concept. Filters are used to constraint the object set returned to the user. A given object passes through a filter if it has one or more tags associated with it that pass through the filter. A filter can be applied to any dimension, tag-sets or hierarchy, regardless of whether it is being shown or not. Traditional *OLAP* operations are, in fact, performed by adding or removing filters on the dimensions of an image collection. These filters restrict which images appear in the constructed cube. ObjectCube provides three types of filters, namely tag filters, range filters, and hierarchical filters.

Applying a tag filter constrains the set of objects retrieved to only those objects associated with that particular tag. Applying a tag filter to the tag "Jack" in the "People" tag-sets, e.g., would retrieve only images tagged with "Jack".

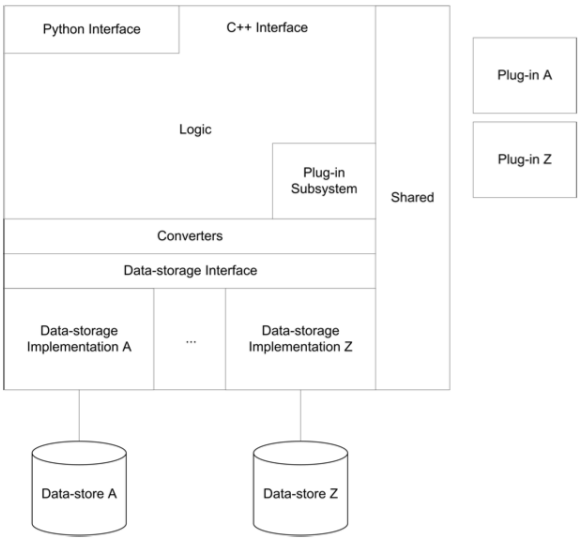

Figure 1: The *ObjectCube* prototype architecture [Tóm11].

A range filter constrains the set retrieved to only those objects with one or more tags with a value within the value range of the filter. For scenario 1, e.g., we must apply a suitable range filter to the "Brightness" tag-sets.

A hierarchical filter on a node in a hierarchy selects the entire subtree of the node, and constrains the set retrieved to only those objects associated with one or more tags in the subtree. Hierarchy traversal is implemented by using a stack of hierarchical filters of increasing depth. For Scenario 1, e.g, we must apply hierarchical filters to the "Location" hierarchy, eventually selecting the node "Europe".

3.2 ObjectCube Architecture

A prototype of the ObjectCube model has been implemented. Figure 1 depicts the overall architecture of the implementation. The architecture of the prototype is presented in detail in [Tóm11], but here we discuss the aspects that are most important for PhotoCube.

Python Interface: The Python interface is a thin wrapper on top of ObjectCube using Boost Python. This layer allow us to easily interface with ObjectCube in Python code. This is convenient since the PhotoCube image browser is written entirely in the Python programming language.

Data Stores: The prototype implementation enables developers to select their own data storage. By default, the prototype supports two data storage implementations, namely for SQLite and MonetDB. If developers wish to use another data storage, they must provide the implementation.

Plug-Ins: The prototype implementation provides a powerful plug-in architecture which allows developers to add automated analysis methods to ObjectCube. This operates as follows. When images are added to ObjectCube, they run through a pipeline of plug-ins. These plug-ins have access to the raw data of the image and can perform any analysis provided by the developer. ObjectCube annotates the image with all tags returned by plug-ins.

A prime example is the recently implemented plug-in for face recognition [Rún11]. When images pass through the plug-in, a face recognition algorithm is executed and known faces tagged with appropriate tags.

3.3 ObjectCube Performance

A performance evaluation of ObjectCube prototype has been performed. In this evaluation image retrieval was measured for the three available filter types, namely tag-filter, range-filter and hierarchical filter, to uncover any scalability or query complexity weaknesses of either the prototype or underlying data stores.

Three different data stores were considered; SQLite, MonetDB and one widely used commercial relational database. Since MonetDB performed the best of the data stores, we only discuss those results here. The threshold for acceptable performance was set at one second. Note that only the meta-data was retrieved, as the retrieval time for the images is independent of the method used for meta-data retrieval.

Even when using a very high selectivity of 10%, the prototype performed acceptably for 40-50,000 images. All filter types performed similarily and the overhead the prototype imposes on the underlying data store was measured to be 140 ms or less for 1,000 retrieved images.

The key result, however, was that the performance was more dependent on the number of objects retrieved than the size of the underlying data-set. Using a very reasonable selectivity of 1%, the prototype performed acceptably for data-sets of over 100,000 images. Note that even 1,000 images may be an unreasonably large number of images for a user to assimilate at once.

4 PhotoCube Architecture

In this chapter we present the architecture of PhotoCube. We first discuss the overall architecture of PhotoCube, and then focus on some of the most important aspects from the point of view of usability and performance.

4.1 Architecture

PhotoCube is written entirely in the Python programming language and uses the Panda3D framework for GUI controls and image presentation. It consists of a number of Python packages; Figure 2 depicts the package structure. Each package has its own purpose and provides a thin service layer for other packages within the application. We now briefly describe these packages in terms of their purpose.

Mode: This module handles the browsing modes available. When the browser is executed all available modes are added to a mode service which can be used to switch between modes during a browsing session.

ObjectCube: The ObjectCube module is a service layer on top of the ObjectCube Python layer. All communication with ObjectCube goes through this module.

Cube: This module handles cube construction and places images into the correct cells. For cells containing a large set of images, it also candidates for presentation in the cell; currently these candidates are selected randomly.

Dialogs: This module contains contains a number of pre-defined dialogs that other modules can utilize, as well as a generic interface for dialog management.

Common: The common modules contains code which is shared with all other modules. This modules contains, e.g., the browser settings and system- and action logging.

Devices: Handles peripheral device communications.

4.2 Key Aspects

In this section we discuss the key features in our architecture. The aim of these features is to make the browser efficient, flexible and extensible.

Browsing modes: A browsing mode is a method for visualizing a set (a cube) of images. The basic browsing mode is the 3D visualization of the cube; another potential browsing mode is a slide show. Within a browsing session, the PhotoCube architecture allows for easy transitions between browsing modes, where cells that are selected in one browsing mode can be taken into another browsing mode. Each browsing mode supports a well-defined API; new browsing modes can thus be added relatively easily, making the prototype very extensible.

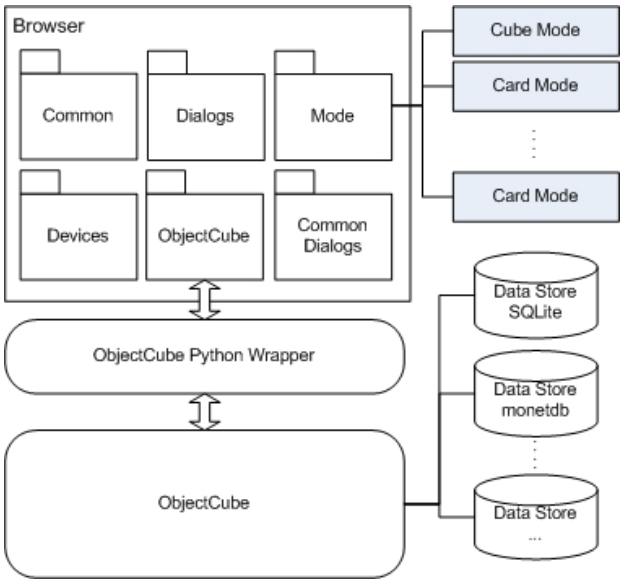

Figure 2: The architecture of PhotoCube.

Action Traces: Since browsing essentially boils down to adding and removing filters, as well as determining dimensions to visualize, it is easy to store action traces representing all the browsing actions in the session. These traces can subsequently be read from disk and replayed, similar to an animated slide-show.

Thumbnail creation: When images are added to PhotoCube, a thumbnail is created and stored on disk. When an image is requested, the thumbnail version is loaded first and stored in memory for later usage. When a cube is constructed, thumbnails are presented. If further inspection of images requires the loading of the full-sized image, the thumbnail is swapped out for the original image.

Image loading: One of the main criteria in our architectural design was that images should be presented as fast as possible to users. For these purposes, we implemented the thumbnail mechanism mentioned above, as well as image buffering and asynchronous image loading. All images are fetched through a singleton image manager. A buffer of a predefined size is initialized and the image manager keeps the most used images in memory, based on retrieval frequency. When images are fetched the image manager is notified and it starts loading the images asynchronously into memory. It takes into account images from a pervious browsing session that are in memory, thus avoiding reloads.

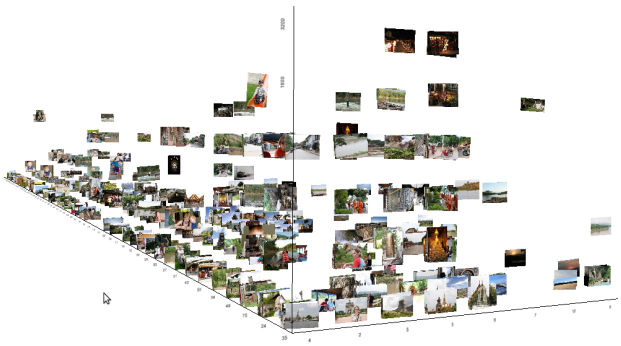

Figure 3: Screenshot of PhotoCube in cube mode.

5 User Interface and Browsing

We now discuss the browsing modes that are currently available, before showing how the two browsing scenarios from the introduction can be realized in PhotoCube.

5.1 Browsing Modes

Currently, there are three browsing modes available for PhotoCube. They are:

Cube Mode: This mode is a direct 3D representation of the ObjectCube data model, and is the main plug-in for the browser. In this mode, users can build an image cube by applying all the features supported by ObjectCube, such as filtering, pivoting, drill-down and roll-up. Figure 3 shows a screenshot of the cube mode. More details of the implementation of the cube mode, including the placement of image cells, may be found in [Sig11].

Card Mode: This mode is used to view images from selected cells in more details. The images are presented as a row of standing cards which can be browsed easily. Figure 4 shows a screenshot from PhotoCube in card mode.

Alternative Modes: As a proof of concept, we implemented a simple arcade-style game, where the selected images are used as shooting targets. Further modes could include slide-shows, web publishing and other such features.

Figure 4: Screenshot of PhotoCube in card mode.

Figure 5: Browsing Scenario 1: Friends hierarchy.

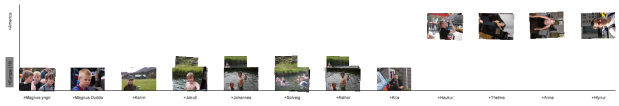

Figure 6: Browsing Scenario 1: Location added.

5.2 Browsing Scenario 1

In Scenario 1, we were interested in images of our friends, taken in Europe, that had a suitable brightness value range. This is done by selecting these three dimensions in cube mode, namely the people hierarchy and the location hierarchy, and drilling down to “Friends” and “Europe” (thus adding hierarchical filters). Finally, a range filter can be added on the exposure concept, resulting in the correct set of images. For experienced users, this is an easy task.

Figure 5 shows the result of the first step; we can see that images of our friends appear on the front axis. Figure 6 shows the result of the next step; there we have placed a location hierarchy on the up axis. We can see that our images have been taken in two continents, namely in Europe and in America. We can also see that our friends from America never appear on images from Europe.

Finally, Figure 7 shows the final step; there we have placed the Exposure concept on the in axis and added a range filter for the values of interest.

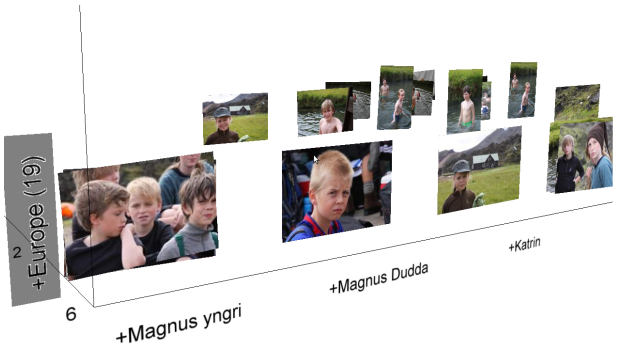

Figure 7: Browsing Scenario 1: Final result.

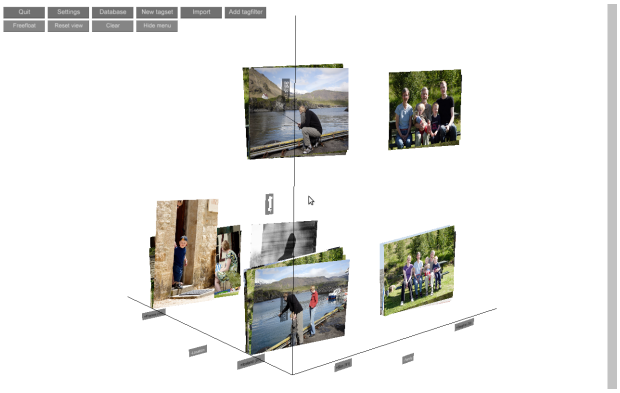

Figure 8: Browsing Scenario 2: Final result.

5.3 Browsing Scenario 2

In Scenario 2, we were interested in family photos, organized by parents, children, and location. In this case, we must add the "Location" hierarchy to one browsing axis and the "Family" hierarchy to the remaining two axes. Then, we must drill down into the Family hierarchy, in one case to the children, and in the other to the parents. The images will then automatically be grouped appropriately on screen. Figure 8 shows the final result in PhotoCube.

5.4 Discussion

The discussion in this section has demonstrated that using the multi-dimensional concepts of PhotoCube enables a rich set of browsing scenarios such as the ones described in the introduction. Since

it is a research prototype, however, many basic features, such as slide-shows, are still missing. Further avenues for future work are described in the conclusions.

6 User evaluation

Since PhotoCube offers a completely different browsing model which might not seem natural at first to many users, we decided to perform a user evaluation. We chose to focus the participants towards computer- and image-browsing-savvy users, to side-step the traditional effects of learning curve associated with research prototypes. The main research question of interest was *how these users like the multi-dimensional data model*, but we also gathered information about many other aspects, such as the interface experience. In this chapter we describe the user evaluation protocol and its outcome. We describe the experimental setup and experimental process, before discussing the results.

6.1 Experimental Setup

We considered a narrow set of users: experienced computer users with some knowledge of multi-dimensional analysis methods. The reason for using such a skewed sample was twofold. First, the focus of this experiment was on the model, rather than the user-interface, and more experienced computer users are more likely to understand the difference between those two concepts. Second, this sample of users is likely to have used other image browsing tools.

User sample: In this experiment we asked five advanced computer participants to participate in our research. Four participants were graduate students of computer science or software engineering, while one participant had more than a decade's experience of working industry. All participants are thus experienced computer users, but their experience of image browsers is varied. One participant did not possess any digital images and had never used a image browser, three participants owned between 500 and 10,000 images, while one participant is a professional photographer and estimated his collection at about 200,000 images. All participants were males and the average age was around 30 years.

Image Collection: The image collection used in the evaluation contained images from a five day trip along a well-known hiking trail. The hiking group consisted of 9 adults and 9 children from 5 different families. The collection consisted of $1,140$ images. Aside from 19 meta-data dimensions extracted from the photo headers, there were 126 tags in 7 tag-sets with a total of 8 hierarchies (one user-generated tag-set had no hierarchies, while one tag-set had two). The tag-sets were: "Events"; "Days"; "Locations"; "People"; "Objects"; "Animals"; and "Impression" (which contained the single tag "beautiful"). The reason for using this set, although it was not familiar to the participants, is that it was already well tagged.

Indicator	Meaning
A	Participant was able to finish the task with no problems.
B	Participant was able to finish the task but with multiple tries.
C	Participant was able to finish the task with a minor help from the instructor.
F	Participant was not able to finish the task.

Table 1: Task performance indicators.

Browsing Tasks: The participants were asked to perform the following tasks:

1. Show images of kids by location.
2. Show images that contain a sheep.
3. Show image that contain hiking shoes and have Aperture value on the range 4 - 5.
4. Show images of people in football.
5. Show images which contain a well-known person, grouped by F-number, ISO-Speed and location.
6. Show me some images which you think are cool.

The purpose of the final task was to allow the users to "play around" with the prototype on their own.

6.2 Experimental Protocol

The experiment was executed as follows. First, each participant was asked to fill in a consent form and a background questionnaire which asked, in addition to basic demographic questions, whether users owned a digital image collection. If so, they were asked whether they used any browsing applications to manage and browse this collection, and asked to state the pros and cons of these browsing applications.

Task Performance: Then the participant was given a short presentation of the ObjectCube data model and the PhotoCube prototype, and shown how to apply PhotoCube for browsing. He was then asked to solve the predefined tasks above using the prototype. The performance on the tasks was noted by the experimenter. Four different performance indicators were used for the performance, namely A, B, C and F; the meaning of these indicators is shown in Table 1.

	Tasks					
Participant	**1**	**2**	**3**	**4**	**5**	**6**
1	B	B	B	A	A	A
2	B	B	A	A	A	A
3	A	A	B	A	A	A
4	C	B	A	A	A	A
5	C	C	A	B	A	A

Table 2: Task performance results.

Usability Factors: Then the participant was asked to fill in a usability questionnaire, based on the AttrakDiff questionnaire (see http://www.attrakdiff.de/), which asked whether he found the prototype to be (on a scale of 1 to 7): Comfortable; Enjoyable; Easy to use; Complicated; Appealing; Pliable; Encouraging; Imaginative; and Useful.

Interview: Finally, the instructor asked a few open questions about the data model and interface.

6.3 Results

We now present the main results from the user evaluation, in the same order as the experimental protocol.

Task Performance: Table 2 shows the outcome of the task performance evaluation. Each row contains the task performance of one individual participant. The table shows that all participants, except for participant 3, experienced some difficulties with the first two tasks. This is most likely due to the learning curve of the model and the prototype. Two users experienced significant difficulties with the first tasks and had to ask for clarifications. The instructions they received, however, consisted solely of reminding users of functionalities of the model, available operations in the browsers and information about the dataset. No direct help instructions were given; yet we do not see any F labels.

In Table 2 we see an interesting pattern emerging. Overall, participants experienced the most difficulties with task 1, where only one user was able to finish this task without problems. In task 2 the performance was slightly better, but still only one user was able to finish the task without any problems. In task 3 the performance improved further, as two users were able to finish the task without problems and the others without any assistance. By task 5 all participants were able to complete the task without any problems, even though that is a relatively difficult task. These results indicate that there is a learning curve for the browser, but that a few browsing sessions may be sufficent to overcome it.

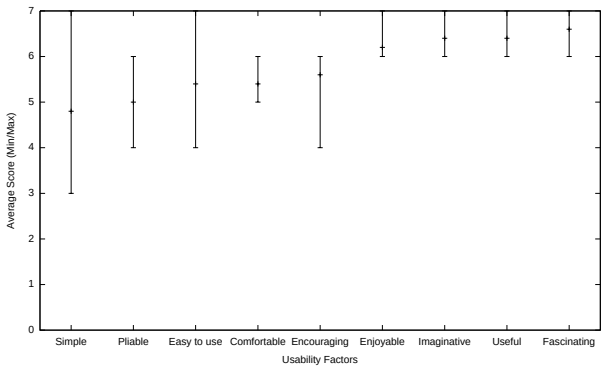

Figure 9: Results from the usability questionnaire

Usability Factors: Figure 9 shows the results of the usability questionnaire. The x-axis shows the measured usability factors, whereas the y-axis shows the average score across all participants. This figure also shows the range of the scores (the minimal and maximal scores). The usability factors are ordered from the lowest score to the highest score.

Overall, we observe that the scores are rather high across the board, ranging from 4.8 to 6.6. There are notable difference between the score for different factors, as the scores can essentially be divided into two parts. The factors on the left side, Simple, Pliable, Easy to use, Comfortable and Encouraging all have a comparably low average score, ranging from 4.8 to 5.6, with a wide range of scores. The widest range is for the Simple factor, where the average value is the lowest, and five and the scores range from 3 to 7. On the other hand, the factors on the right, Enjoyable, Imaginative, Useful and Fascinating, all have comparably high average scores of 6.2 through 6.6. Furthermore, the participants all seem to agree on these values, as the range of scores is quite narrow.

These results indicate that our participants find the prototype rather complicated and cumbersome to use, while finding it at the same time highly enjoyable, imaginative, and useful. The reason for the lower average values for the factors on the left side can possibly be explained by the status of the prototype. The focus was to get most of the features of the data model into the prototype at the cost of a poorer user interface. These features, on the other hand, might be the reason for such high values for the factors on the right side, as they can be considered exotic and innovative.

Interview: Many useful and interesting points were brought forward in the interviews. Overall, the participants made 19 different comments on the advantages and disadvantages of the prototype. Of these, 11 comments were positive and 8 comments negative, but they were worded constructively.

Many of the comments highlight the results of the task performance and usability evaluation. Some participants stated the there was a steep learning curve when solving the initial tasks. They also

said that they found it difficult to distinguish between tag-sets and hierarchies, but after a number of tries the difference became clearer to them. This mirrors the results of the task performance evaluation.

Also, participants commented on the user interface. These comments were mostly on user controls, location of components and lack of action feed-backs. This confirms the low value of the factors that are related to the user experience. Many of the comments from the participants were also on how much they liked the browsing model, the representation of the images and how well the browser showed the connection between images and their attributes. This explains the high average values of the usability factors that focus on enjoyment and usefulness. At the end of the interview participants were asked whether they would like to use this model, or a similar model, to browse their own image collection; they all answered positively.

6.4 Summary

Today we have a working prototype which contains a large number of the features that we were interested in implementing, but this user evaluation has shown that the interface requires significant work. The research question put forth, however, focused on the data model: how well users liked the model? The user sample is skewed towards computer-savvy participants, and hence the research questions cannot be answered for general users. The evaluation results indicate, however, that the model has significant potential to improve the state-of-the-art in image browsing, which is very encouraging for our future work on PhotoCube.

7 Conclusions

We have presented PhotoCube, a novel three-dimensional image browser based on the recently proposed ObjectCube data model. The data model allows users to seamlessly involve a large number of meta-data dimensions in each browsing session and captures a rich set of browsing actions. The architecture of the PhotoCube browser is highly extensible, as different browsing modes can be implemented and the user can easily swithch between browsing modes. In this paper, we described the architecture and user experience of the prototype in detail. We then presented results from a preliminary user study, which indicate that while experienced photo browser users find the prototype interface to have a somewhat steep learning curve, they are excited about the potential of the underlying data model.

There are very many potential avenues for future work. One line of work involves developing a smooth tagging process, which is imperative for the usefulness of the browser. This includes: further integrating state-of-the-art image analysis methods, such as face recognition; careful handling of image meta-data dimensions containing date and time information; and effective implementation of concept and hierarchy management. In order to evaluate an improved prototype, extensive user evaluations using regular users, and comparing to state-of-the-art browsers, are necessary.

Furthermore, careful attention must be paid to the efficiency of handling large image collections. While the ObjectCube model may, in many cases, return a very large set of images, a user can probably only handle a few hundred at a time, at most. Furthermore, a large concept can result in a very sparse cube. How to aggregate the information of large sets is a very interesting open question.

Finally, we plan to stream-line the installation process and make the browser accessible to the world at large.

References

[BC09] I. Bartolini and P. Ciaccia. Integrating semantic and visual facets for browsing digital photo collections. In *Proc. of SEBD*, Camogli, Italy, 2009.

[Bed01] B. B. Bederson. PhotoMesa: A zoomable image browser using quantum treemaps and bubblemaps. In *Proc. UIST*, Orlando, FL, USA, 2001.

[Fer07] S. Ferré. Camelis: Organizing and browsing a personal photo collection with a logical information system. In *Proc. CLA*, Montpellier, France, 2007.

[HJ07] K. Hardarsson and B. T. Jónsson. Breaking out of the shoebox: Towards having fun with digital images. In *Proc. CVDB*, Beijing, China, 2007.

[Rún11] K. Rúnarsson. A face recognition plug-in for the PhotoCube browser. Master's thesis, Reykjavik University, 2011.

[Sig11] H. Sigurþórsson. Photocube: A multi-dimensional image browser. Master's thesis, Reykjavik University, 2011.

[SL04] D. Surendran and S. Levy. Visualizing high dimensional datasets using Partiview. In *Proc. INFOVIS*, Austin, TX, USA, 2004.

[Tóm11] G. Tómasson. Objectcube: Multi-dimensional Model for Media Browsing. Master's thesis, Reykjavik University, 2011.

Technical Report RUTR-CS12001
ISSN 1670-5777

School of Computer Science
Reykjavík University
Menntavegi 1
101 Reykjavík, Iceland
Tel. +354 599 6200
Fax +354 599 6201
www.ru.is