An Intelligent User Interface for Browsing and Search MPEG-7 Images using Concept Lattices

Peter Eklund, Jon Ducrou, and Tim Wilson

School of Information Technology and Computer Science
University of Wollongong
NorthFields Avenue, New South Wales, Australia
email: {jond,peklund}@uow.edu.au, timwilson1@optusnet.com.au

Abstract. This paper presents the evaluation of a design and architecture for browsing and searching MPEG-7 images. Our approach is novel in that it exploits concept lattices for the representation and navigation of image content. Several concept lattices provide the foundation for the system (called IMAGE-SLEUTH) each representing a different search context, one for image shape, another for colour and luminance, and a third for semantic content, namely image browsing based on a metadata ontology. The test collection used for our study is a sub-set of MPEG-7 images created from the popular *The Sims 2*^{TM} game. The evaluation of the IMAGE-SLEUTH program is based on usability testing among 29 subjects. The results of the study are used to build an improved second generation program –

isii – but in themselves indicate that image navigation via a concept lattice is a highly successful interface paradigm. Our results provide general insights for interface design using concept lattices that will be of interest to any applied research and development using concept lattices.

1 Introduction

The application objective of this research is to offer a different way of searching and navigating digital images – images annotated with MPEG-7 multimedia descriptors – and presenting them in a way that is easily understood by humans. The interface we develop presents a new way to graphically represent relationships between images so that navigation across a collection of images occurs in a non-linear or serendipitous way. A suitable method for achieving this is through the application of formal concept analysis. The images (as objects) are organised as a conceptual hierarchy via the formal concept analysis of their image and metadata attributes. The concept lattice that results provides the information space over which the interface navigates. The paper tests the success of this idea through software evaluation in a usability trial.

The images used to test the approach are derived from the popular computer game *The Sims 2*^{TM}. This collection is made interesting by taking into account the properties given to these objects in *The Sims 2*^{TM} game play. The game playing properties are elements such as: how an object addresses *The Sims 2*^{TM} character's needs — like hunger, bladder comfort, tiredness, hygiene, etc; how the object develops *The Sims 2*^{TM} character's skills in different skill areas – e.g. logic, cooking, mechanical skills and creativity.

In addition to this metadata associated with the game play, MPEG-7 feature descriptors are also used so that the images can be navigated according to their colour and shape.

In related work [1] we address the issue of the design theory underlying a Web-based FCA system for browsing and searching MPEG-7 images called IMAGE-SLEUTH. This paper, which emphasises the usability of IMAGE-SLEUTH, is structured as follows. In order to be self-contained, an introduction to formal concept analysis is provided in Section 1. Because the image format we use contains semantic attributes as well as an image signature we give the reader a brief introduction to MPEG-7 in Section 2. We illustrate the idea of image browsing using FCA with a collection of images from the *The Sims 2TM* and in Section 3 we present both a synopsis of the *The Sims 2TM* game play and details of the MPEG-7 image content. In Section 4 we present our approach to image navigation based on the design of edge traversal in the concept lattice. Our main results are presented in Section 5 where we present the usability test script, the survey instrument and the results of our evaluation for the image browsing software. An important purpose to our usability study was to learn how to improve the performance and design of our software. In Section 6 we show how our findings conditioned the development of IMAGE-SLEUTH2, in particular the way that conceptual scaling is handled and the introduction to IMAGE-SLEUTH2 of distance and similarity metrics for approximate matching. In Section 7 we discuss work in progress on extending our ideas to searching and browsing a dynamic data collection: namely the Amazon catalog.

Formal Concept Analysis Background

Formal Concept Analysis [2] has a long history as a technique of data analysis ([3], [4]) conforming to the idea of Conceptual Knowledge Processing. Data is organized as a table and is modeled mathematically as a many-valued context, (G, M, W, I_w) where G is a set of objects, M is a set of attributes, W is a set of attribute values and I_w is a relation between G, M, and W such that if $(g, m, w_1) \in I_w$ and $(g, m, w_2) \in I_w$ then $w_1 = w_2$. In the table there is one row for each object, one column for each attribute, and each cell is either empty or asserts an attribute value.

A refined organization over the data is achieved via conceptual scales. A conceptual scale maps attribute values to new attributes and is represented by a mathematical entity called a formal context. A formal context is a triple (G, M, I) where G is a set of objects, M is a set of attributes, and I is a binary relation between the objects and the attributes, i.e. $I \subseteq G \times M$. A conceptual scale is defined for a particular attribute of the many-valued context: if $\mathbb{S}_m = (G_m, M_m, I_m)$ is a conceptual scale of $m \in M$ then we define $W_m = \{w \in W | \exists(g, m, w) \in I_w\}$ and require that $W_m \subseteq G_m$. The conceptual scale can be used to produce a summary of data in the many-valued context as a derived context. The context derived by $\mathbb{S}_m = (G_m, M_m, I_m)$ w.r.t plain scaling from data stored in the many-valued context (G, M, W, I_w) is the context (G, M_m, J_m) where for $g \in G$ and $n \in M_m$

$$\begin{aligned} gJ_m n \ \Leftrightarrow: \exists w \in W : \ (g, m, w) \in I_w \\ \text{and } (w, n) \in I_m \end{aligned}$$

Scales for two or more attributes can be combined in a derived context. Consider a set of scales, S_m, where each $m \in M$ gives rise to a different scale. The new attributes supplied by each scale can be combined:

$$N := \bigcup_{m \in M} M_m \times \{m\}$$

Then the formal context derived from combining these scales is:

$$gJ(m,n) \;\Leftrightarrow: \exists w \in W : \; (g,m,w) \in I_w \text{ and } (w,n) \in I_m$$

Several general purpose scales exist such as ordinal and nominal scales. A nominal scale defines one formal attribute for each value that a many valued attribute can take. An ordinal scale can be used to interpret an attribute whose values admit a natural ordering, for example the $\leq$ ordering over numbers.

A concept of a formal context (G, M, I) is a pair (A, B) where $A \subseteq G$, $B \subseteq M$, $A = \{g \in G \mid \forall m \in B : (g,m) \in I\}$ and $B = \{m \in M \mid \forall g \in A : (g,m) \in I\}$. For a concept (A, B), A is called the extent and is the set of all objects that have all of the attributes in B, similarly, B is called the intent and is the set of all attributes possessed in common by all the objects in A. As the number of attributes in B increases, the concept becomes more specific, i.e. a specialization ordering is defined over the concepts of a formal context by:

$$(A_1, B_1) \leq (A_2, B_2) :\Leftrightarrow B_2 \subseteq B_1$$

In this representation more specific concepts have larger intents and are considered "less than" ($<$) concepts with smaller intents. The analog is achieved by considering extents, in which case, more specific concepts have smaller extents. The partial ordering over concepts is always a complete lattice [2].

For a given concept $C = (A, B)$ and its set of lower covers $(A_1, B_1)...(A_n, B_n)$ with respect to the above $<$ ordering the object contingent of C is defined as $A - \bigcup_{i=1}^{n} A_i$.

2 MPEG-7 Images

Accepted as an ISO standard in 2001, MPEG-7[1] allows the storage of physical and semantic descriptors for use in content management, organization, navigation, and automated processing of images [5]. MPEG-7 is extensible, being based on XML, and can therefore support a broad range of applications.

MPEG-7 comprises Description Tools made up of the metadata elements, along with their structure and relationships, which are used to form Descriptors and Description Schemas. *Descriptions* can then be used by applications for effective and efficient access to multimedia content. These descriptions accommodate a range of abstraction levels, from low-level signal characteristics to high-level semantic information. In this paper, we are interested in both low-level image descriptors, more specifically color descriptors and shape descriptors, as well as high-level semantic metadata by extending the MPEG-7 format to store customised details for each object.

[1] *http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm*

Colour descriptors in MPEG-7 consist of seven more specific descriptors: Colour space, Colour Quantization, Dominant Colours, Scalable Colour, Colour Layout, Colour-Structure, and GoF/GoP Colour. Three of the MPEG-7 visual descriptors are used in this research, these are Colour Layout, Scalable Colour and Edge Histogram and these are extracted from the image collection as described below.

3 Feature Extraction on the Example Collection

The example collection for our Web-based image browser is based on items from the popular computer game *The Sims 2TM*. Created by Electronic Arts, *The Sims 2TM* is "the sequel to the best-selling PC game of all time"[2]. In short, the game is a real-life simulation; the player is given control over a suburban neighbourhood and the people in it, shaping their careers, friendships, houses, children, and controlling mundane tasks: such as directing them to cook meals, have showers and go to bed. These simulations of the people who populate the neighbourhood — the characters of the game — are referred to as *Sims*.

Each Sim has 8 *needs* that affect their well-being. These are hunger, comfort, hygiene, bladder, energy, fun, environment and social. Sims also have 7 types of skills which they can practice and refine. These skills are cleaning, charisma, creativity, body, logic, mechanical and cooking. As well as looking after their needs and lives, a player can build a house for their Sim and purchase different household items to furnish it. These items include furniture, plumbing, appliances, decorations, electronics, plants, lighting and much more. Household items can directly affect a Sim's needs and skills when in use[3]. Hunger, for example, will be alleviated to a lesser degree when cooking with a cheap microwave than when using an expensive oven. Some of the items can also have a negative impact on a Sim. For example, a coffee machine increases energy, but will also decrease bladder comfort. A bookcase will allow the Sim to study and increase cooking skills, while an artist's easel will allow the Sim to produce artworks and increase their creativity.

3.1 *The Sims 2TM* Image Collection

Our collection is based on virtual household items that can be bought and sold in the *The Sims 2TM*. The basis for this choice is the dual nature of the items. A household item must aid in the successful playing of the game – as well as have aesthetic appeal – perhaps matching other furnishings already in place. Therefore, a household item has physical properties such as colour and shape, as well as properties that effect a Sim's life and well-being .

The colour layout descriptor in MPEG-7 breaks the image into an 8×8 grid and represents each grid square by the dominant color in YCbCr format[4]. The scalable colour descriptor gives a measure of colour distribution over the entire image. The edge

[2] *http://thesims2.ea.com/*

[3] The exception here is the "social" dimension which is only affected by social interactions.

[4] YCbCr is a family of colour spaces used in video systems and similar to that used in colour television.

```
<VisualDescriptor xsi:type="ColorLayoutType">
   <YDCCoeff>5</YDCCoeff>
   <CbDCCoeff>30</CbDCCoeff>
   <CrDCCoeff>31</CrDCCoeff>
   <YACCoeff63>
      13 23 15 12  5 20  9 14 19 17 16 17 21 18 15 17 18 12 16 11 13
      16 14 15 15 15 17 13 16 15 17 14 20 15 17 16 18 15 16 15 15 12
      14 15 16 15 16 14 16 15 16 16 17 16 15 15 14 15 15 15 16 17 16
   </YACCoeff63>
   <CbACCoeff63>
      16 15 16 16 17 15 16 16 15 15 16 15 15 15 16 15 15 16 15 16 16
      16 16 16 16 16 15 16 16 16 15 16 15 16 15 16 15 16 16 16 16 16
      16 16 15 16 16 16 16 16 15 16 15 15 16 16 16 16 16 16 16 15 15
   </CbACCoeff63>
   <CrACCoeff63>
      16 16 16 16 16 16 15 15 16 16 15 16 16 16 15 15 16 16 16 15 15
      16 15 16 15 15 15 16 16 16 16 16 16 15 16 16 16 15 16 16 16 16
      15 15 16 16 16 15 16 15 16 16 16 16 15 15 16 16 16 16 16 16 16
   </CrACCoeff63>
</VisualDescriptor>
```

Fig. 1. An example of the Colour Layout extracted values for an image.

histogram defines a 4×4 grid and gives the strength of the non-homogeneous texture for each grid square in 4 directions and an overall strength.

```
<VisualDescriptor
   xsi:type="ScalableColorType"
   numOfBitplanesDiscarded="0"
   numOfCoeff="64" >
   <Coeff>
      -202  58  40  41  -7  12  20  14   6  13  11  22   1  16  21   9
         0   1   0   2  -1   5   0   0  -9  -2  -2   9 -15   3  -1 -19
         0   0   0   1   0   0   1   2   1   1   1   3   1   2   4   5
         1  -3   2  -2   2  -1  -8  -2   0 -15   0  -4   1  -2  -3 -15
   </Coeff>
</VisualDescriptor>
```

Fig. 2. Scalable Colour Type extracted values.

To extract shape and colour information for a household item, a feature extraction tool, Caliph [7], is used to generate colour layout [8] and edge descriptors [9]. These are then stored in MPEG-7 using the appropriate tags as shown in Figs. 1 and 2. A secondary feature extraction process, which analyses the resulting descriptors, is then run to produce more user-friendly colour descriptors. The secondary colour descriptors use a reduced form of the standard HTML colour set to assign a meaningful colour property. The set of colour names have a hierarchy in which parent colours are more general (e.g. Red → Dark Red → Maroon). These secondary descriptors are added into the MPEG-7 datastore using a custom mark-up.

The edge histogram descriptor is a measure of the edge distribution within an image [10]. In a similar method to that used by the colour layout descriptor, an image is

Fig. 3. The 5 edge types used in the edge histogram descriptor.

	Price	Needs::Hunger	Needs::Comfort	Needs::Hygiene	Needs::Bladder	Needs::Energy	Needs::Fun	Needs::Environment	Skills	Function	Room Type
4 by 4 Designer Chandelier	§120	0	0	0	0	0	0	1		Lighting	Dining, Living, Bathroom, Bedroom
Absolutely Nothing Special	§850	0	0	0	0	0	0	1		Lighting	Kids, Study, Dining, Living, Bedroom
Ad-a-Quaint Barstool	§285	0	3	0	0	0	0	1		Comfort	Living, Kitchen
Ad-a-Quaint Coffee Table	§140	0	0	0	0	0	0	1		Surfaces	Study, Living
Astrowonder Telescope	§550	0	0	0	0	0	4	0	Logic	Hobbies	Outside
Zenu Meditation Sleeper	§950	0	4	0	0	4	0	2		Comfort	Bedroom

Table 1. A fragment from the 'Item Properties' sub-context.

broken down into a series of non-overlapping square blocks. An edge histogram is then generated on each of these blocks. The descriptor defines 5 values to represent the edge histogram for each block. These 5 values describe the vertical, horizontal, 45 degree and 135 degree edges as well as a non-directional edge. A nondirectional edge is one that has no apparent direction (e.g. a curve).

The colour and shape descriptors are then complemented with the various semantic metadata derived from the *The Sims 2^{TM}* game information. Needs and skills are an attribute hierarchy where more specific attributes in the hierarchy imply more general. For example, Needs is implied by Fun, Fun is implied by Fun:1 to 5. This gives some level of encapsulation to the attributes because in order to have Fun appear as an attribute, Needs must have also been included. Other item properties such as object price and function and suitability to a given room type are also included. On function, objects are grouped by the game into one of 11 functional categories: electronics, lighting, miscellaneous, comfort, aspiration rewards, career rewards, decorative, plumbing, hobbies, appliances and surfaces. Items are also given a room property based on which room the item would most likely be placed. An item may have one or more values for the room property, meaning that it is suitable in several different rooms, or it may have no room value at all, meaning that it can be put anywhere. The room types are: kids, study, dining room, outside, living room, bathroom, bedroom and kitchen. An fragment of the underlying formal context based on *The Sims 2^{TM}* household objects is shown in Table 1.

3.2 Basis for Selection of *The Sims 2^{TM}* dataset

In games such as *The Sims 2^{TM}*, where collectible items affect gameplay, much effort is put into the game's design in terms of the balance and distribution of items with respect to item properties. This design principle provides an excellent base for testing ideas that use query-by-example, as for most items there is only a single exact item, and varying cluster types associated with it. For example, there may be only 1 curved, 3-seater, blue couch with Comfort:8, but there is a collection of other blue couches with different comfort levels, a collection of different coloured and shaped couches with Comfort:8, and a matching blue coffee table that associates with the couch. These are all acceptable responses to query-by-example for the 3-seater, blue couch with Comfort:8 because in some way they are all household objects of the same grouping — but collected based on different facets of the data.

4 Conceptual Design of the Image Browser

4.1 Problem Decomposition

Our approach decomposes the overall lattice – generated from the formal context fragment, a fragment of which is shown in Table 1 – into smaller sub-lattices. These sub-lattices are created by combining attributes compatable meaning. In the case of *The Sims 2^{TM}* data, three sub-lattices are evident; colour properties (including all attributes regarding the colours used in the MPEG-7 images), Item properties (including all the game play properties) and edge properties (including a human readable form of the MPEG-7's generated EdgeHistogramType classifier). Decomposing the lattice into sub-lattices in this way allows for more overall generality per concept for each concept of each sub-lattice. This is necessary given the unique nature of computer game items used in our image collection, but also allows for query-by-example type searching.

4.2 Interface Design

At any one time the user will be placed at either a single formal concept of a sub-lattice or at a single object (an image). The formal concept is displayed as a neighbourhood showing the current extent as thumbnailed images (in arbitrary order), and the attributes which allow movement to other formal concepts in the neighbourhood. Movement from the current formal concept or image object can be via either specialisation or generalisation. Specialisation is achieved by adding attributes and moving down in the lattice structure (via an interface control called *include*). Alternatively, generalization is achieved by removing attributes via an upward movement between formal concepts (via an interface control called called *remove*). For *include*, the attributes that can be added are displayed, and for *remove*, the attributes that can be removed are displayed (see Fig. 4 (left)).

Our design philosophy is that the presentation of attributes belonging to the upper and lower neighbour formal concepts allows the current state of the interface to move across the concept lattice in an intuitive way. Further, that this approach to navigation is often preferred to showing a complete list of possible attributes to add, or all attributes

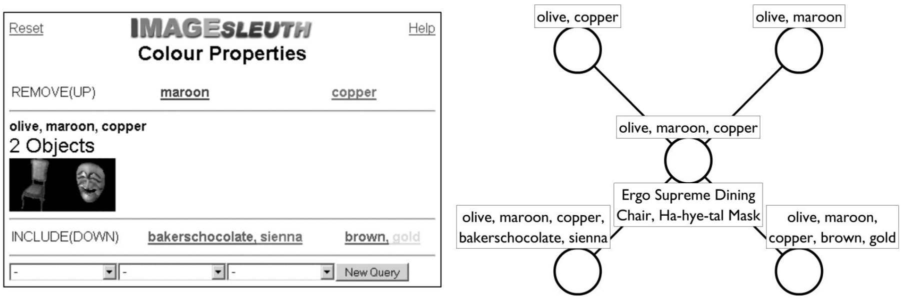


Fig. 4. An example screenshot of *Image Sleuth* and the lattice representation of the corresponding neighbourhood

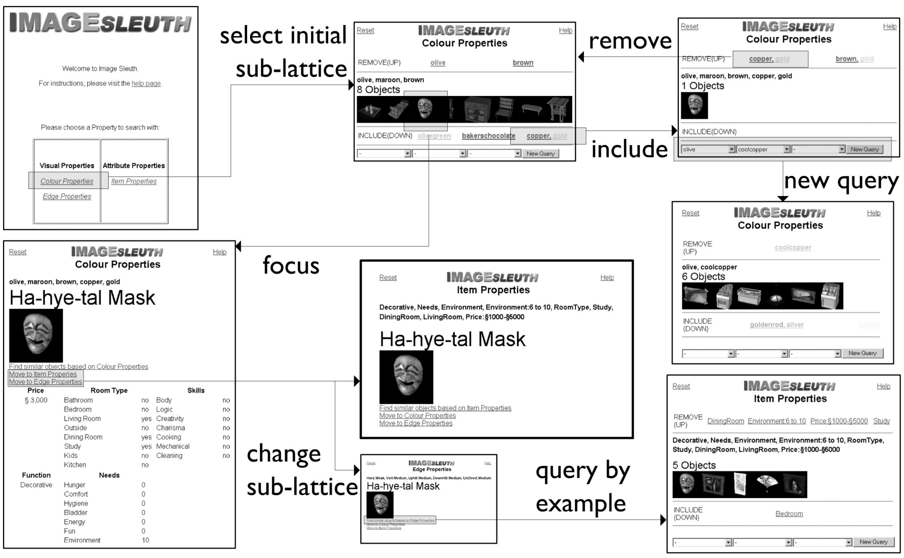


Fig. 5. A navigation overview of the system. (Note: Some screenshots have been resized and are slightly out of scale.)

that can be removed from the current view. One of the consequences of this design is that it is impossible to navigate to a concept with an empty extent (i.e. no images), because two mutually exclusive attributes can never be selected during navigation via upper and lower concepts in the way described[5]. By using the concept lattice structure as the focus for navigation in the interface, the users' perspective is concentrated on conceptual changes that are minimal and incremental. Furthermore, navigation actions in the interface conform directly to the definition of edge traversal in a concept lattice

[5] Also, the bottom-most concept is considered inaccessible and the user cannot navigate to it unless it has an extent size greater than zero.

and movment through the information space is therefore directly expressed in the theory of Formal Concept Analysis.

This form of navigation helps to reduced the complexity per ‘decision’, as attributes will often be hidden by others because of implications or attribute hierarchies, whether data-emergent or artificial. For example, given the data has an attribute hierarchy over colours, with more specific colour descriptions being children of more general terms, the attributes ‘Dark Blue’ and ‘Light Blue’ will not be visible as *include* attributes until the user has included the ‘Blue’ attribute. Conversely, if the ‘Dark Blue’ attribute is visible as a *remove* attribute, ‘Blue’ will be hidden. Complexity is also reduced by attribute equivalence. For example, if ‘Dark Green’ is the only type of green in the data it will appear in the *include* attributes as a combined pair of attributes ‘Green, Dark Green’ as there is equivalence between them, (i.e. images with ‘Dark Green’ will also be ‘Green’ as well).

As well as navigation via traversal of the concept lattice the IMAGE-SLEUTH interface also provides a traditional query interface that allows direct positioning into the concept lattice. The query interface restricts the user to terms that are attributes of the current sub-lattice. The query interface takes the submitted attributes and finds the most specific concept that has all query terms, namely the query interface performs the equivalent of the double prime operation. This method ensures that at all times the user is positioned at a formal concept. In the event that the user selects attributes for which no formal concept exists, no images are returned.

IMAGE-SLEUTH also supports the direct selection of an image of interest by clicking on it. In this case the user is presented with the exact set of attributes for the given image and the option of changing sub-lattice or querying-by-example. Any single object can act as a connection point between contexts, and by changing contexts the user is presented with the attributes this object has within the new sub-lattice. Query-by-example uses the current images’ attributes to relocate to the most specific concept associated with the image. This will then show all other objects with the same attributes. This means that an object can be found using one sub-lattice, then be used in another sub-lattice to find similar objects, but within a different area of interest. For example, using *The Sims 2* collection, a user may find a bed that suits their in-game requirements, then swap sub-lattices to find matching furniture for that bed using colour or shape information. The architecture for the IMAGE-SLEUTH system in shown in Fig. 6.

5 Usability

Rozanaski and Haake define several attributes of usability with regards to a user interface [11]: (i) learnability; (ii) efficiency; (iii) memorability; (iv) amount of error; (v) satisfaction. This usability evaluation attempts to measure the success of IMAGE-SLEUTH in meeting these attributes. Usability will be measured through an empirical study that consists of two parts, a test script and a participant survey. The results are also compared with interaction logs which independendly validate that the participant achieved the correct state.

To perform the study, postgraduate and honours students were recruited to participate in the usability tests. Each of the 29 testers were based in one of 6 Faculties at

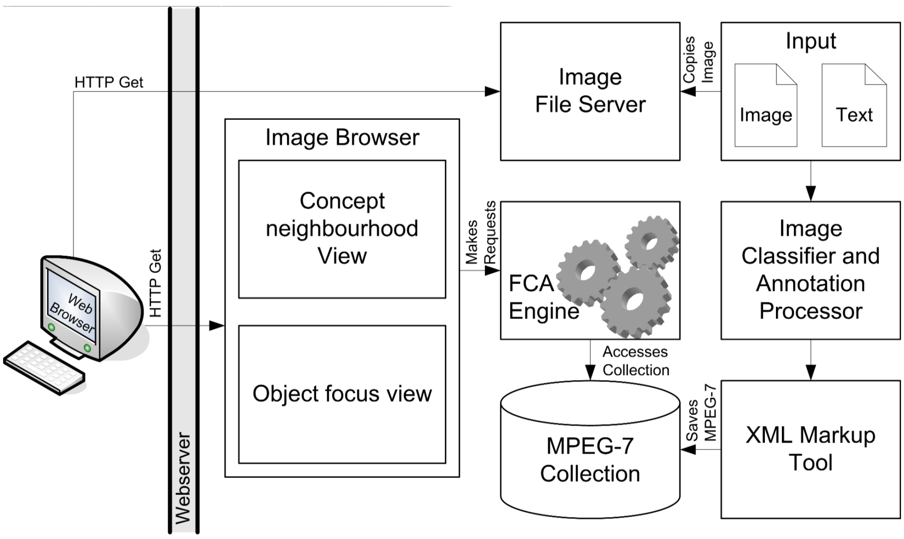


Fig. 6. A technical overview of the system.

the University of Wollongong in Australia. The students were primarily drawn from Informatics (49%) and Commerce (32%) Faculties, with others from Law, Education, Engineering and Arts. All could be said to exhibit a high degree of computer literacy.

5.1 Test Script

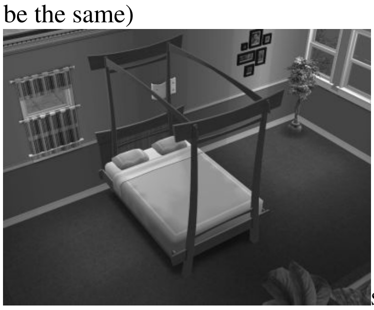
The test script consists of three sections the first two of which returned quantitative results and the third qualitative results. The first section included directions to be followed using Windows Explorer to browse the image collection. Images can be viewed as thumbnails within a folder, and sorted by various criteria such as file name and creation date. Tasks included finding particular items, finding items that matched certain criteria, observations of item groups with certain features and so on. Some tasks in this section – while possible to complete – may have proven time-consuming or difficult to accomplish using Windows Explorer. Windows Explorer is not designed as an image browser, but it can be used to browse images and is used as the base level functionality because all participants are familiar with it. If IMAGE-SLEUTH proved no better than Windows Explorer then this would be a powerful argument to abandon the design.

The second section of the test started with identical tasks to those in the first but now completed using IMAGE-SLEUTH. As the study supervisors were not permitted to assist participants, the steps to be followed using IMAGE-SLEUTH were ordered in such a way as to expose participants to the various functionality of IMAGE-SLEUTH gradually. Designing the test script in this way assists the participant in learning the new navigation style without needing special training. Tasks in the latter half of the second section of the test script were designed to make participants perform more complex interactions in order to solve problems. For example, showing a black and white image of an object in a setting at a different orientation and asking participants to identify its colour by finding the corresponding image in IMAGE-SLEUTH. These tasks extended the test script and had no comparable task in the first section of the script. A complete list of the second sections tasks are shown below.

The first and second sections were issued in reverse order to half of the participants. This way, familiarity with the object set did not give an unfair advantage to IMAGE-SLEUTH.

1. Identify how many images have the environment attribute.
2. Identify how many images have a price higher than 5000.
3. Search for all images that are Decorative. Then include the outside room type. How many images are there?
4. Identify how many images can be used in both the Dining Room and Kitchen.
5. Identify how many of the plumbing images also have an environment attribute. What types of objects are these?
6. How many images do not have a price at all?
7. Identify how many images are green using the colour properties.
8. There are 2 beds with a high level of horizontal edges. Which are they?
9. Are there more red, navy or green images?
10. Find the name of the chair in the following image:

    
    Shown in colour.
11. Find the name of the bed in the following image: (Note - The bedspread need not be the same)

    
    Shown in colour.

12. Find the name of the bath/shower in the following image. What colour is the curtain around it?
Shown in black and white.
13. This object is not in *The Sims 2^{TM}* collection.
Shown in colour.
It has attributes of: *Comfort* = 5, *Energy* = 4, *Environment* = 3.
Find one image that is similar in design and one that has similar attributes while not being expensive.
14. Find the object that can be used outside, builds logic and has a price between 1000 to 5000. How many images have similar colours?
15. Aspiration rewards and career rewards are special objects in *The Sims 2^{TM}*. Using IMAGE-SLEUTH to browse the objects, what can you say about them, in 50 words or less?

The final section of the test script consisted of a "free exploration" of IMAGE-SLEUTH: encouraging participants to discover features without any particular goal in mind. This allowed participants to gain an understanding of the features without explicit direction. Participants were subsequently asked to provide their positive and negative thoughts regarding the features of the program.

5.2 Survey

The survey asked participants questions on their personal background and experience with IMAGE-SLEUTH. Background information included the faculty of study, experience with other image browsers/viewers and the methods of organisation used for personal image collections. Likert scales were used collect details of their experience with IMAGE-SLEUTH and Windows Explorer. This was followed by a series of questions to assess the participants understanding of IMAGE-SLEUTH and how it worked. Fig. 7 shows a complete list of questions asked in the survey.

Participant Survey

- Short Response
  - Which faculty does your university degree belong?
  - Are you colour blind? If so, did you experience difficulty in completing the test script?
  - Have you used image management applications before? If so, which?
  - Do you already sort your images based on specific criteria (e.g. date, location, etc.)? If so, what?
- Likert Scale Statements (0 to 10, disagree to agree)
  - I am familiar with the PC Game *The Sims 2^{TM}*.
  - I found it easy to complete tasks with:
    - Windows Explorer
    - IMAGE-SLEUTH
  - I feel that IMAGE-SLEUTH has a strong advantage over Windows Explorer.
  - I feel that IMAGE-SLEUTH has a strong advantage over other image browsers.
  - IMAGE-SLEUTH allows me to recognise relationships between images that I may not have noticed previously.
  - IMAGE-SLEUTH is a tool that gives more power over searching and browsing catalogs of images.
  - I found that the ______ properties were accurate.
    - Colour
    - Edge
  - My overall experience with IMAGE-SLEUTH was a positive one.
- Multiple Choice
  - What features of IMAGE-SLEUTH did you find assisted most when completing the tasks in the test script? (circle all that apply)
  - What features of IMAGE-SLEUTH made it difficult to complete the tasks in the test script? (circle all that apply)
- Long Response
  - Which features of IMAGE-SLEUTH could be used to improve the image browsing experience in the future, and why?
  - In your own words, describe the 4 main components of the IMAGE-SLEUTH interface and what they do. Name the 3 different types of searches and what they do?
  - Do you understand what the Remove(up) and Include(Down) sections mean in IMAGE-SLEUTH?
  - In your own words, please describe what the Include and Remove sections allowed you to do, and comment on whether or not this tool helped you to complete the allocated tasks in the test script.
  - Could you see this application being used in the real world? If so, where?
  - Do you have any other comments?

Fig. 7. Complete list of questions on the survey.

5.3 Interpreting the Usability Results

The second section of the test script (where users are first exposed to IMAGE-SLEUTH) proved difficult for participants, but after completion of first few questions, most subjects became acquainted with the user interface and its functionality. On average, once participants had attempted 6 – 7 tasks the number of correct responses increased considerably, even though the tasks became progessively more difficult.

Average correct completion for the Explorer tasks was 70.5%, and 74.5% for the equivalent IMAGE-SLEUTH tasks. The correct completion rate for the entire IMAGE-SLEUTH test script was 73.4%.

Of the respondents, 24 had previously used image management programs. Of these, 17 stated that they sorted their images based on specific criteria and over half the testers had some familiarity with *The Sims 2*^{TM}. Not surprisingly, all but 3 testers found that they could complete the test script easier with IMAGE-SLEUTH than with Windows Explorer and 23 testers stated that they felt more comfortable using IMAGE-SLEUTH. All testers believe that IMAGE-SLEUTH was better than Windows Explorer for browsing images and all but 2 testers thought that IMAGE-SLEUTH had advantages over photo browsing applications they had encountered.

Question 6 asked participants to rank the ease of task completion for both Windows Explorer and IMAGE-SLEUTH. It can be seen in Fig. 8 that most subjects found IMAGE-SLEUTH easier to use. On a scale to 10, the average for IMAGE-SLEUTH is 7.3, while Windows Explorer's average is 3.7; almost half that of IMAGE-SLEUTH.

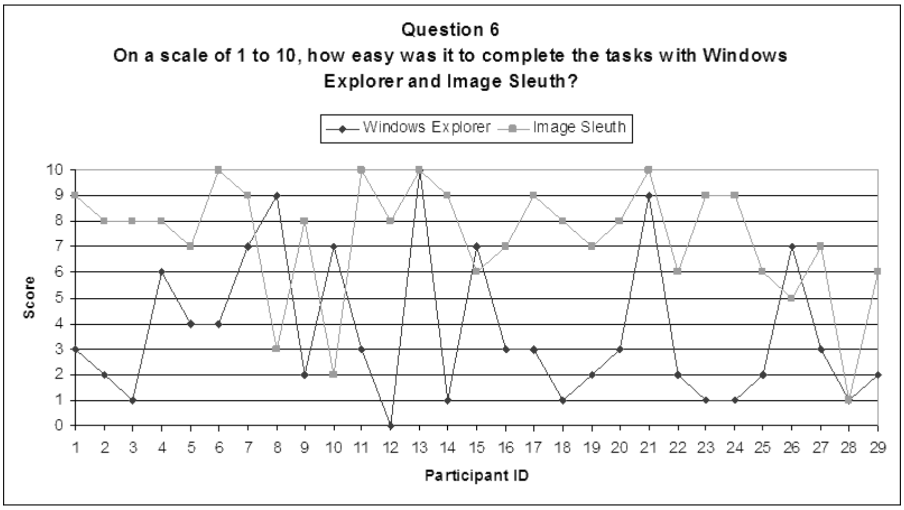


Fig. 8. Comparison of the ease of task completion between Windows Explorer and IMAGE-SLEUTH

Question 12 asked "*What features of* IMAGE-SLEUTH *did you find assisted most when completing the tasks in the test script? (circle all that apply)*". Results (shown in

Fig. 9) indicated that the ‘include/remove’ and attribute search controls were found to be most useful.

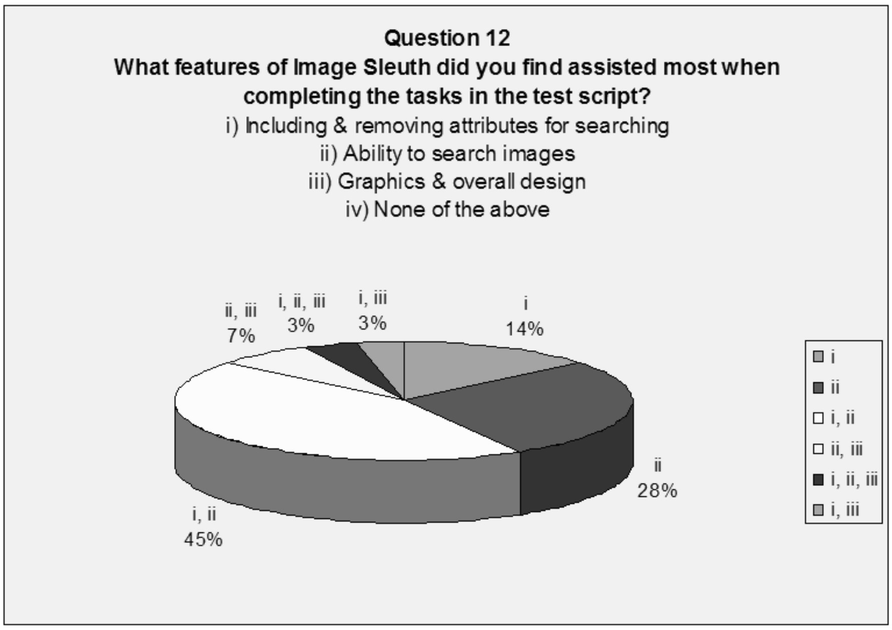


Fig. 9. The most useful features of IMAGE-SLEUTH

Many participants were not happy with the IMAGE-SLEUTH interface, claiming it appeared primitive and difficult to navigate with the search functions at the bottom of the page (See Fig.10). Another frequently mentioned negative was the accuracy of the colour property. Participants did not seem to agree with some of the colours that were returned by IMAGE-SLEUTH for some images and suggested the inclusion of a colour palette (or legend) so that testers could identify by label the colour they were searching for.

Of the positive comments, the most common involved the include and remove controls, the ability to find specific images quickly and the consistency in design. Testers found it extremely useful to be able to remove single attributes rather than having to perform new searches from scratch.

Participants for the most part understood the navigation paradigm very well, some even using the terms ‘*narrow*” and “*move down*” to describe the include control and “*broaden*” and “*move up a level*” to describe the remove control. Other testers felt that the remove was more like the ‘Back’ button in a Web browser that allowed you to navigate back in multiple ways, for instance, “*...allowed me to reverse a search term without having to go back and redo the search again*” and “*... similar to the function of*

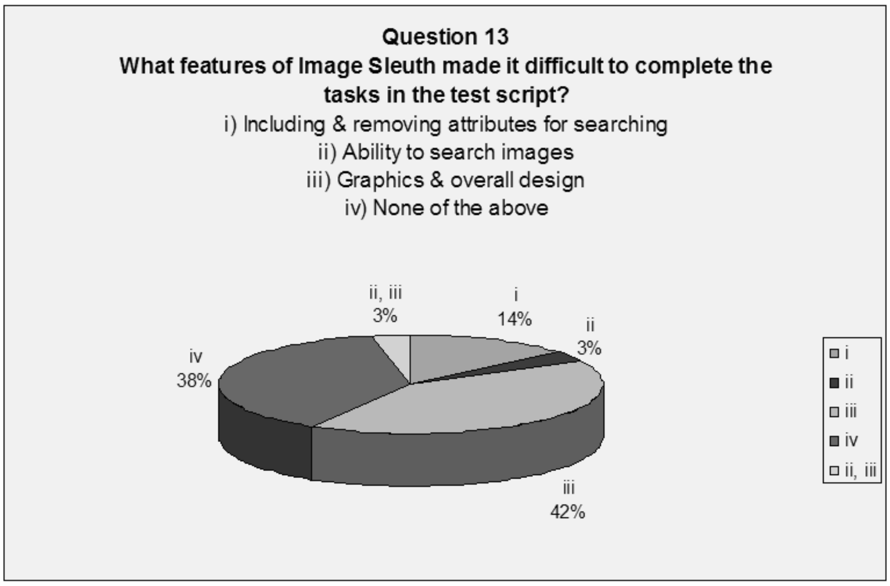


Fig. 10. Features difficult to use in IMAGE-SLEUTH

back in Internet Explorer. The difference is you can choose which step you have before (sic) easily".

Review of interaction logs found that the most common pattern for participants was to start each task with a term search in the appropriate sub-lattice (colour, shape or game play) and then navigate from this concept using the include/remove controls. This method was appropriate but showed an inherent flaw in the term search approach. When attributes are used in a term search and there is no object with all the attributes IMAGE-SLEUTH returns the empty-extent concept. This appeared to leave participants confused, and many reported that the task was unsolvable and moved on to the next task. It is an observation that argues for some form of approximate matching when the result set is empty (discussed below in Section 6).

5.4 Usability Conclusions

The 'include/remove' controls for navigation was very successful, further 'term searching' was liked by participants but sometimes caused empty results which led to confusion. Testers responded well to the idea that images were 'grouped' at all times, the groupings being concept extents. Many participants commented on the ease-of-use of IMAGE-SLEUTH, or the fact they had understand the functionality of the software quickly. Most understood or had an idea about how each interaction affected the state of their navigation. This indicates that the navigation paradigm is intuitive.

6 Applying Usability Results

After the analysis of the usability results work on a second version of IMAGE-SLEUTH began. The aims of the second version were to address the problems and issues revealed in the usability testing.

A significant change is to allow overlapping sub-contexts, to combine attributes from colour, shape and game play in a more fluid way, so that a more dynamic notion of the sub-lattices that could be created and navigated emerges. In the first version, there were three exclusive contexts concentrating on different facets of the information space. This was changed to one context, with a set of 'perspectives' (conceptual scales) over the formal context. These perspectives can then be used singularly or in combination, and added/removed as necessary during use of the system. This reduces the restrictive nature that separate sub-contexts caused and allows users to see all attributes pertinent to their navigation needs. The original contexts were “Item Properties”, “Color Properties” and “Edge Properties”, but IMAGE-SLEUTH2 [6] has 10 perspectives;

- SimpleColours (16 colour set.)
- AdvancedColours (216 colour set + SimpleColours. Equivalent to “Colour Properties”)
- Needs
- Skills
- Price
- Function
- RoomType
- NeedsAndSkills (combination of Needs and Skills.)
- Gameplay (all game play related attributes. Equivalent to “Item Properties”.)
- SimpleGameplay (same as GamePlay, excluding the lowest level of detail.)

This use of a library of concept scales (as perspectives or view over the image collection) means sub-contexts, and concept lattices derived from them, can be drawn from any combination of the scales (e.g. AdvancedColours and SimpleGameplay). A screenshot showing IMAGE-SLEUTH2 is shown in Fig. 11.

The IMAGE-SLEUTH interface received much criticism from participants, primarily focusing on poor organisation. To address this IMAGE-SLEUTH2 has fixed positions for each interface component.

One method for dealing with the return of empty-extents from term search search is to provide users with a list of the terms entered so that they can be incrementally removed terms to unconstrain the search. Another method is to apply a vector space model of MPEG-7 images [12] and then apply similarity measures for multi-dimensional feature spaces. IMAGE-SLEUTH2 explores a different approach by using variations on defined distance [13] and similarity [14] metrics in the FCA literature in order to find relevant concepts.

The similarity metric we applied uses the size of the common objects and attributes of the concepts. For two concepts (A, B) and (C, D):

[6] IMAGE-SLEUTH2 can be trialed by visiting *http://130.130.112.18/jon/test/framebuilder.exe*

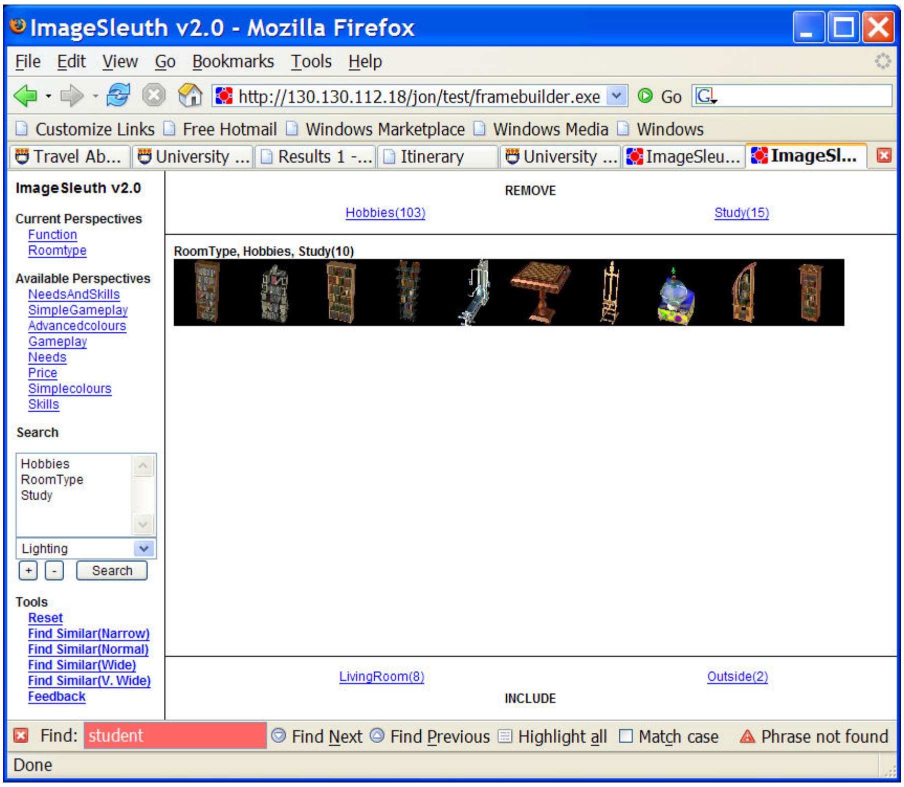


Fig. 11. An example screenshot of IMAGE-SLEUTH2 showing the neighbourhood of the formal concept represented by the images with exactly the attributes Hobbies and Roomtype=study. The scales (or perspectives) that condition for lattice are Function and Roomtype.

$$similarity((A,B),(C,D)) := \frac{1}{2}\left(\frac{|A \cap C|}{|A \cup C|} + \frac{|B \cap D|}{|B \cup D|}\right).$$

The distance metric uses the size of the total overlap of the intent and extent normalised against the total size of the context. For two concepts (A, B) and (C, D):

$$distance((A,B),(C,D)) := \frac{1}{2}\left(\frac{|A \setminus C| + |C \setminus A|}{|G|} + \frac{|B \setminus D| + |D \setminus B|}{|M|}\right).$$

When a search contains attributes which are not manifest in a single object, IMAGE-SLEUTH2 creates a semi-concept with the searched terms as the intent. This semi-concept is used to prime a traversal of the lattice structure – the traversal applies the distance and similarity metrics to calculate a relevance score. The traversal is bounded by a maximum distance (see Fig. 12). The user is shown the most relevant concepts (with objects as thumbnails) allowing users to decide the concept that best matches their search. This relevance ranking this traversal method is accessible from any concept to find closely matching concepts. A screenshot of results is shown in Fig. 13. This provides a powerful tool for finding similar concepts and objects from a given starting concept.

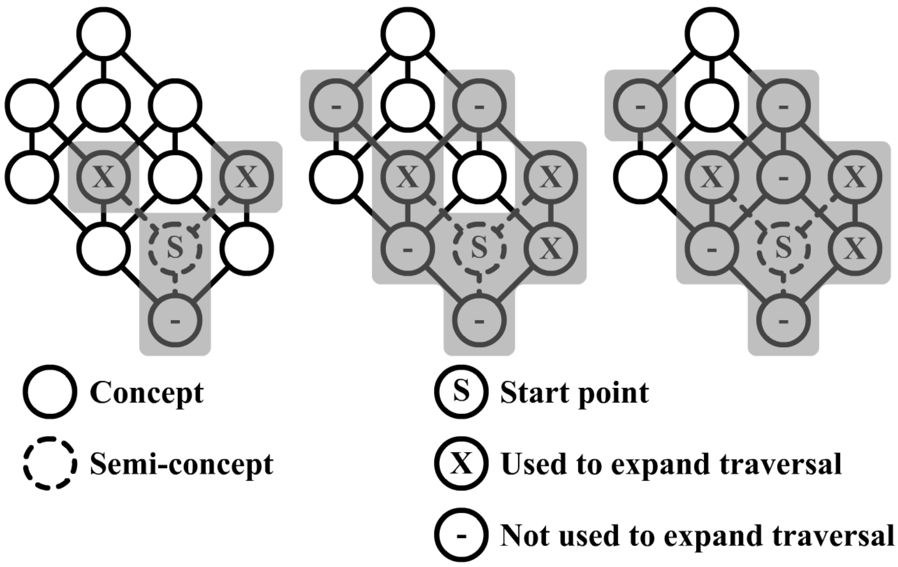


Fig. 12. An example of lattice traversal starting from a semi-concept. The traversal in this example is complete in 3 steps. The shaded area shows the computed concepts at each step.

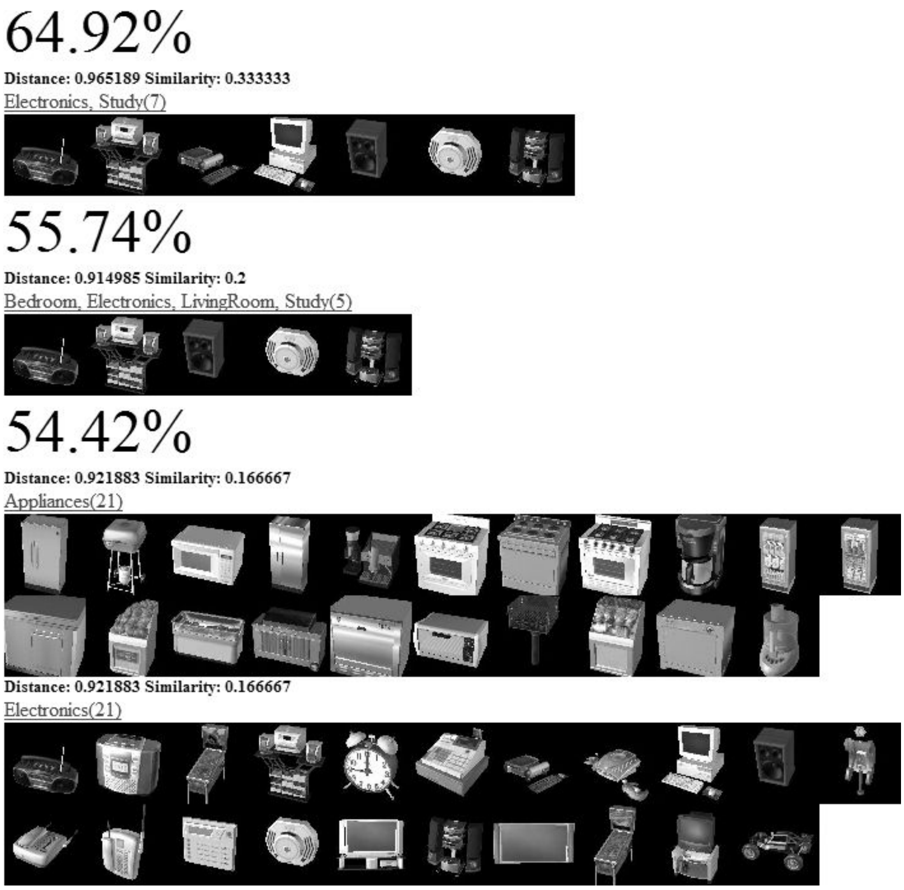


Fig. 13. Results of a concept traversal from the query "*Appliances, Electronics, Study*" using the perspectives "*Function, RoomType*". *Appliances* and *Electronics* are mutually exclusive.

7 Future Directions

IMAGE-SLEUTH2 is currently being extended to utilise DVD information collected from Amazon's Web store. This allows a DVD to be represented by the front cover of its case, and attributes to be created from the accompanying details (e.g. genre, actor, director, etc). Performing a relevance queries on a DVD allows users to find closely related DVD's based on whichever facets of the data considered important. The architecture of that tool, called DVDSLEUTH, is quite different from IMAGE-SLEUTH2 because the context is dynamic and grows in various ways depending on the directions taken in the navigation in the Amazon catalog.

8 Conclusion

The design theory underlying a Web-based FCA system for browsing and searching MPEG-7 images was introduced in Ducrou et al. [1]. This paper has presented the evaluation of an architecture and implementation of a browsing and search interface for MPEG-7 images that exploits concept lattices for the representation and navigation of image collections. Sub-contexts provide the foundation for the IMAGE-SLEUTH system, each representing a different search view: one for image shape, another for colour and luminance, and a third for semantic content. In this way the initial IMAGE-SLEUTH would navigate over three concept lattices. In the subsequent versions of IMAGE-SLEUTH, a library of conceptual scales (called perpectives) are introduced to allow the more fluid creation of different concept lattices for navigation. The main results of the usability study are the confirmation of the suitability of the concept lattice as a navigation paradigm for image browsing. We also demonstrate how distance and similarity measures within the concept lattice can be used for approximate matching when search terms do not result in a precise match to a formal concept. The experience with the iterative development of IMAGE-SLEUTH has lead to new insights in search using concept lattices which are being realised for the creation of dynamic contexts and the navigation of Web content.

References

1. Ducrou, J., Vormbrock, B., Eklund, P.: Fca-based browsing and searching of a collection of images. In: Proceedings of 14th International Conference on Conceptual Structures. LNAI4068, Springer Verlag (2006) 203–214
2. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin (1999)
3. Vogt, F., Wille, R.: TOSCANA - a graphical tool for analyzing and exploring data. In Tamassia, R., Tollis, I., eds.: Proceedings of the DIMACS International Workshop on Graph Drawing (GD'94). Lecture Notes in Computer Science 894, Berlin-Heidelberg, Springer-Verlag (1995) 226–233
4. Kollewe, W., Skorsky, M., Vogt, F., Wille, R.: TOSCANA - ein werkzeug zur begrifflichen analyse und erkundung von daten. Begriffliche Wissensverarbeitung: Grundfragen und Aufgaben (1994) 267–288

5. Ohm, J.R.: The MPEG-7 visual description framework - concepts, accuracy, and applications. In: Lecture Notes in Computer Science. Volume 2124. (2001) 2
6. Sutanto, D., Leung, C.: Automatic index expansion for concept-based image query. In: Lecture Notes in Computer Science. Volume 1614. (1999) 399
7. Lux, M.: Caliph & emir : Mpeg-7 photo annotation & retrieval. `http://caliph-emir.sourceforge.net` Accessed 30/08/05 (2005)
8. Cieplinski, L.: MPEG-7 color descriptors and their applications. In: Lecture Notes in Computer Science. Volume 2124. (2001) 11
9. Won, C.S.: Feature Extraction and Evaluation Using Edge Histogram Descriptor in MPEG-7. In: Lecture Notes in Computer Science. Volume 3333. (2004) 583 – 590
10. Park, D.K., Jeon, Y.S., Won, C.S.: Efficient use of local edge histogram descriptor. In: MULTIMEDIA ’00: Proceedings of the 2000 ACM workshops on Multimedia, New York, NY, USA, ACM Press (2000) 51–54
11. Rozanski, E.P., Haake, A.R.: The many facets of hci. In: CITC4 ’03: Proceedings of the 4th conference on Information technology curriculum, New York, NY, USA, ACM Press (2003) 180–185
12. Park, R.W., Lee, D.H.: Full-automatic high-level concept extraction from images using ontologies and semantic inference rules. In: The Semantic Web - ASWC 2006. LNAI4185, Springer (2006) 307–321
13. Lengnink, K.: Ähnlichkeit als Distanz in Begriffsverbänden. In Stumme, G., Wille, R., eds.: Begriffliche Wissensverarbeitung: Methoden und Anwendungen, Springer (2000) 57–71
14. Saquer, J., Deogun, J.: Concept approximations based on rough sets and similarity measures. In: Int. J. Appl. Math. Comput. Sci. Volume 11. (2001) 655–674