



SALERO

The System Design and Implementation of the Context-Based Search System

SALERO Deliverable D5.5.4



The System Design and Implementation of the Context Based Search System

SALERO identifier: SALERO-D5.5.4-UG-ContextBasedSearchSystem-
v05.doc

Deliverable number: D5.5.4

Author(s) and company: R. Villa, P. Swamy, V. Strathopoulos, Y. Feng,
H. Misra, J. Jose (UG) M. Tuomola, T. Korpilahti
J. Pesonen (TAIK), T. Bürger (LFUI), W. Weiss (JRS),
S. Miarnau (AM)

Work package / task: WP05

Document status: Final

Confidentiality: Public

DOCUMENT HISTORY

Version	Date	Reason of change
1	2009-04-30	document created
2	2009-05-08	Partners added to sections
3	2009-06-12	Draft, all content added, before version for internal review
4	2009-06-17	Version for internal review
5	2009-06-26	Final report for submission

The work presented in this document was partially supported by the European Community under the Information Society Technologies (IST) priority of the 6th framework programme for R&D.

This document does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of its content.

This document contains material, which is the copyright of certain SALERO consortium parties, and may not be reproduced or copied without permission. All SALERO consortium parties have agreed to full publication of this document. The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the SALERO consortium as a whole, nor a certain party of the SALERO consortium warrant that the information contained in this document is capable of use, nor that use of the information is free from risk, and does not accept any liability for loss or damage suffered by any person using this information.

Table of Contents

1	Executive Summary	1
2	Introduction	2
2.1	Purpose of this Document	2
2.2	Status of this Document	2
2.3	Related Documents	2
3	Architectural Overview	3
3.1	Introduction	3
3.2	Search and Indexing Services	4
3.2.1	Search Process	4
3.2.2	Indexing Process	4
3.3	SOAP Web Interface	5
3.4	External Web Searches	6
3.5	Conclusions	6
4	The Backend Retrieval System	7
4.1	System Requirement	7
4.2	Features Used in the Backend	8
4.2.1	Edge Histogram Descriptor	8
4.2.2	Contour Shape Descriptor	9
4.2.3	Block Ratio	10
4.2.4	Centroid Profile	10
4.2.5	Object Signature with reference to Axis of least inertia	10
4.2.6	Horizontal and Vertical projection profile	11
4.3	Fusion techniques	12
4.4	Evaluation of the Retrieval System with the TAIK Collection	14
4.5	Summary	17
5	The AspectBrowser Interface	19
5.1	The AspectBrowser Search interface	19
5.1.1	Executing a text query	21
5.1.2	Executing an image query	22
5.1.3	Using multiple aspects	23
5.2	The Indexing Interface	23
5.3	Conclusions	25
6	User Evaluation of the AspectBrowser	26
6.1	Research Questions	26
6.2	Complex Search Tasks	26
6.3	Experimental Design	27
6.4	Interfaces	27
6.4.1	The aspectual interface	27
6.4.2	The baseline interface	27
6.5	Tasks	28
6.6	Procedure	29
6.7	Results	29
6.7.1	Search performance	29
6.7.2	User perceptions of search task difficulty	32

6.7.3	<i>Interface features and user perceptions</i>	32
6.8	Discussion	33
6.9	Conclusions and Future Work.....	35
7	Integration with the Semantic Search and Annotation Systems	36
7.1	Interface Integration	36
7.2	Annotation and Search Application	37
7.3	Search System	38
7.3.1	<i>Semantic Search System</i>	38
7.3.2	<i>Content Based Search System</i>	38
7.3.3	<i>Retrieval Integration</i>	39
7.4	Experimental Results	40
7.5	Conclusion.....	42
8	Usage in Experimental Productions	43
8.1	TAIK: Alan01 and AlanOnline	43
8.1.1	<i>The Concept: Alan01 / AlanOnline</i>	43
8.1.2	<i>The Alan01 Installation</i>	44
8.1.3	<i>AlanOnline</i>	46
8.2	AM: Video Disk Jockey.....	47
9	Conclusions	50
10	References	51
11	Glossary	54

1 Executive Summary

The long term needs of a user while searching for some information are more likely to be satisfied by the context in which the search is performed. However, current systems typically do not make use of contextual information while presenting the results and “context-based search systems” are a promising alternative to bridge this gap.

The present deliverable describes the progress made in the development of a context-based search system in the last six months of the SALERO project. This report is divided into six main content chapters.

Chapter 3 gives an overview of the architecture of the proposed system developed for context-based search. In this chapter, the software modules provided in the system for search and indexing services are explained along with the SOAP Web interface which enables the system utilization via Web services (as explained in Chapter 8). Finally, the capabilities of the system while searching external resources such as Flickr and YouTube are also described.

In Chapter 4, several new low-level features which have been developed and incorporated into the system are described. A few of these features are motivated by the specific requirements of SALERO experimental productions (e.g. the TAIK collection used in Chapter 8). An evaluation of these new features and the investigation of fusion strategies was carried out on TAIK data, and an analysis is presented about the challenges imposed by TAIK collection while retrieving images.

In Chapters 5, the AspectBrowser Interface for search and indexing are explained; Chapter 6 covers the AspectBrowser Evaluation. In particular, while the section on the AspectBrowser Interface highlights the ways in which the browser can be used under different situations, the chapter on the AspectBrowser Evaluation deals with its use in some broad and complex tasks. Through user evaluations the effectiveness of the proposed context based search system is demonstrated.

Chapter 7 gives a description of our efforts in integrating semantic search with content based search. After providing motivations for integrating semantic and content based search methodologies, the design of our proposed system is presented. The evaluation of the system is done on the PDP data and the future research directions are identified.

Chapter 8, the final content chapter, is about SALERO Experimental Productions (EP). Two EPs, TAIK and Video Disk Jockey, are explained in this chapter, both of which use the search system in their operation, in different ways. In particular, Video Disk Jockey proposes the use of AspectBrowser system for searching videos similar to a given video query.

The conclusions of this comprehensive report are drawn in Chapter 9.

2 Introduction

2.1 Purpose of this Document

This document describes the design and implementation of the context based search system, including the search interface, the backend search system, and provides an introduction into the systems use in other SALERO experimental productions.

2.2 Status of this Document

This is the final version of Deliverable D5.5.4

2.3 Related Documents

Before reading this document it is recommended to be familiar with the following documents:

- D2.3.1 User Requirements Document
- D5.5.1 Context-based retrieval system and user interface
- D5.5.2 Retrieval algorithms based on contextual features

3 Architectural Overview

In this chapter, we give a high-level overview of the architecture of the context-based search system and provide the software details. In the following chapters of this report, the underlying techniques used by the different parts of the software are fleshed out further.

3.1 Introduction

The high-level architecture of the context-based search system is shown in Figure 3.1, below. The core of the system is the search and indexing services which run on a central server. This is composed of two services, one for searching, the other indexing, both of which expose a Java Remote Method Invocation (RMI) interface. These provide the content-based search and retrieval functions which are used by the rest of the system, and maintain the corresponding text and visual indexes. These services also access remote search databases such as yahoo, Flickr and YouTube, enabling the searching of these remote databases via the same java interface.

The Java RMI interfaces exposed by the search and indexes services are intended for use locally, within the same network. To allow remote systems to use the content-based search, such as the semantic search, AlanOnline and Video disk jockey, a SOAP interface is exposed via a Web server. This interface allows these other SALERO modules to use the full functionality of the content-based search and retrieval system. In addition, Web interfaces (the AspectBrowser and Indexing interface) are also exposed, for use by end users.

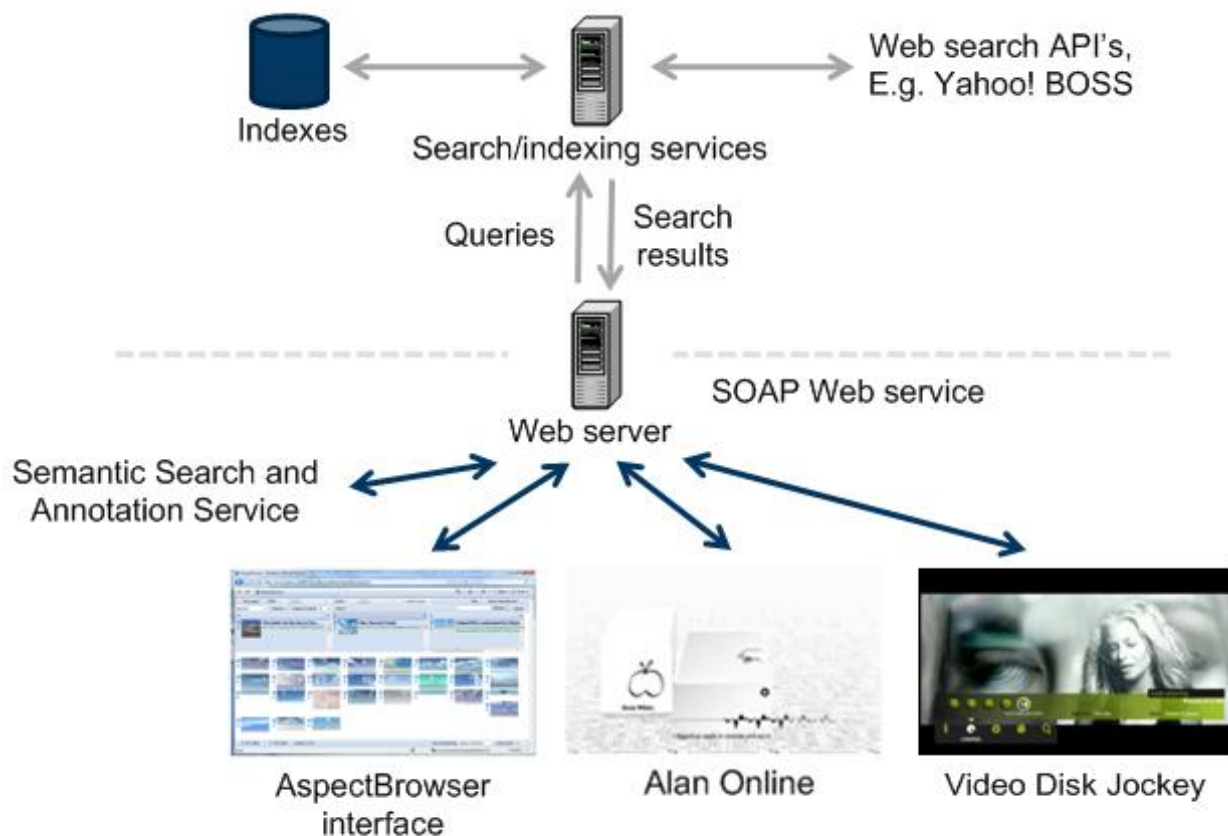


Figure 3.1: The high-level architecture of the context-based search system

In the following sections, we introduce each of these separate modules.

3.2 Search and Indexing Services

The core of the context-based search and retrieval system is the search and indexing services. These expose two Java Remote Method Invocation (RMI) interfaces which allow clients to index and search multimedia content. Both of these services run as separate processes for reliability: if the indexing process crashes, for example, the search process will continue to run.

3.2.1 Search Process

The search process exposes the following Application Programming Interface (API):

getAvailableCollections()

Returns a list of the collections which can be currently searched

getAvailableFeatures(String collectionID)

For the given collection, this method will return the list of features which can be used to search that collection. Each collection may be indexed by different features, either due to the preference of the indexer (some features are more appropriate for some types of collection), or due to the available data (e.g. image collections have no text, and therefore cannot be searched using the text feature)

loadCollection(String collectionID)

Makes the collection available for use, which will ensure the collection has been initialised and is ready to be searched.

searchLite

Will carry out a search, returning a list of results as document identifiers (which are simple strings). The parameters required are as follows:

- collectionID: The collection which will be searched
- textQuery: A text query which will be used if a text index is available
- features: A list of the features which should be used in the search
- examples: Document ID's of example images
- images: Base-10 encoded images which will be used in the search
- startFrom: The starting index of the first result
- numberOfResults: The number of results required

searchFull

This will carry out a search in the same way as searchLite, but will return objects of class Document, which contain extra meta-data such as URL, thumbnail image, title, etc.

searchFusion

Carries out a search as for searchFull, but enables the setting of either one or two extra "method" parameters, which allows the type of fusion to be set (either for fusing different examples, or different features from different examples)

getImageURL(String collectionID, String documentID)

Returns the image URL for a document in a specified collection.

getThumbnail(String collectionID, String documentID)

Returns the thumbnail URL for a document in a specified collection.

The most important methods in the above interface is the search methods (searchLite, searchFull and searchFusion), which can all be used to search a collection using both a text query and image examples. Other methods allow a client to find out the features available to search a particular collection, the list of collections which can be searched, etc.

3.2.2 Indexing Process

The indexing process exposes the following Application Programming Interface (API):

getAvailableCollections(String userID)

Returning a list of collections which have been created by a user.

createCollection(String userID, String collectionID)

Creates a new stub collection which is owned and accessible by the supplied user. Use setUsersPermissions to allow other users to search the new collection. This collection will be empty – data should be transferred into a specified directory by FTP before indexing is started

index(String userID, String collectionID)

Will start the indexing process for a collection.

indexingStatus(String userID, String collectionID)

Indexing may take some time, depending on the size of the collection indexed. This methods will return the percentage of the collection which has been indexed so far (which will be 100% when finished).

setUsersPermission(String userID, String[] users, String collectionID)

Sets the users which will be able to search a collection.

getAvailableFeatures()

Returns a list of the features that the indexing service can handle, i.e. the complete list of visual features which can be indexed and then subsequently searched.

getAvailableFeatures(String collectionID, String userID)

Returns the features which were used to index the specified collection. A collection may not be indexed with all available features, either due to the nature of the collection or by choice.

The above interface is designed to allow a client to create an empty collection, index it, and also get information about how far the indexing process has progressed. This latter information is important for implementing progress bars for end users, since indexing a large collection may take some time. One aspect which is *not* dealt with in the above interface is the transfer of the actual data (images, videos etc) to be indexed to the server. In testing it was found that transferring large quantities of data using such a programmatic interface was unreliable (particularly with the SOAP interface described in the next section). For this reason, a decision was made to use File Transfer Protocol (FTP) for uploading the data, which is more appropriate for moving around large quantities of data. The createCollection method creates an FTP directory on the server which can then be used by client programs or users to upload data to. Calling index will then automatically index all uploaded content.

3.3 SOAP Web Interface

In addition to the internal interfaces described in the previous two sections, a SOAP interface, which is exposed to the wider world via a Web service, was also defined, which contains the following methods:

search

Searches a specified collection (as searchLite from Section 3.2.1)

searchFuse

Searches a specified collection allowing the fusion techniques to be specified (as searchFuse from Section 3.2.1)

getAvailableCollections

Returns all the collections available for searching (as getAvailableCollections from Section 3.2.1)

getAvailableFeatures

Gets the features which can be used in a search for a collection (as getAvailableFeatures from Section 3.2.1)

getThumbnail

Get the thumbnail URL of a search result (as getThumbnail from Section 3.2.1)

For security reasons, no indexing functionality is included in this interface. Instead an indexing user interface is provided (Section 5.2), to enable users to create and index new collections. The above soap interface is used by the search system in the semantic search and annotation, the video disk jockey (Section 8.2) and also AlanOnline (Section 8.1).

3.4 External Web Searches

In addition to indexing and searching image and video content which is stored on the local server, the backend retrieval system is also able to carry out searches on external resources. Three are currently provided by the system:

- Flickr image search, as provided by the Flickr services (<http://www.flickr.com/services/>)
- Yahoo Web search, which searches textual Web documents. Provided by the yahoo BOSS API (<http://developer.yahoo.com/search/boss/>)
- YouTube video search, provided via the YouTube Data API (<http://code.google.com/apis/youtube/overview.html>)

All three of these services can be searched in the same way as other multimedia content. When searching these resources, the following restrictions apply:

- Only text queries can be executed due to the limitations of the respective APIs
- There may be maximum number of results for some of the services which stops the retrieval of very large lists of results

Despite this, the ability to search these external resources using the same simple interface which is used to search other multimedia content makes it very easy for client interfaces to enable users to search multiple collections. An example of this is the functionality provided by the AspectBrowser interface in Chapter 5.

3.5 Conclusions

In this Chapter we have provided a high-level overview of the context-based search and retrieval system. In the following two chapters we outline in more details the different parts of the system, starting with the techniques used in the backend retrieval system, with an emphasis on the developments made to better match the requirements of the SALERO experimental productions. This is followed in Chapter 5 with a description of the two user interfaces for search and Indexing.

4 The Backend Retrieval System

In this Chapter techniques used in the backend retrieval system will be described. This include some of the low level image features which can be used for search and retrieval, along with a variety of new techniques for fusing results from different features and examples. The work reported here bulids upon the work reported in the deliverable D5.5.3, and has been developed to better match the requirements of the SALERO experimental productions.

In D5.5.3 we presented a detailed description on the features we used for retrieval in our system such as colour histograms, edge histograms, homogenous texture, etc. and also some additional local visual features (e.g. SIFT), followed by a short exploratory section discussing 3D features.

Due to inclusion of new datasets from partners, specifically TAIK data, used in the experimental productions *Alan01* and *AlanOnline*, a number of image features were investigated for type of data used, each of which is described in the following sections.

4.1 System Requirement

The first point to be made is that the visual retrieval system is only required to generate a single result: a single result is used by *Alan01* as the initialising point for the sequence of other actions; in the *AlanOnline* production, though it is possible to display multiple results, to provide multiple responses to the input, this is not considered here. This requirement calls for high precision results, with the added requirement that the system must always generate a result – even if there is nothing in the collection which is similar to the input query, the *closest* should always, ideally, be selected.

Secondly, the matching required by the productions is largely visual in nature – each query input, whether generated by the touch screen in *Alan01* or the online drawing canvas of *AlanOnline*, is only required to be visually similar to the result image in some manner. But this visual similarity should be *understandable*, i.e. users should be able to intuitively see why a result was produced for a query, or be able to determine a property of the search result which matches the input query. If the system cannot return results which cannot be interpreted as similar, the risk is that those interacting with the productions will be less likely to engage with them.

The retrieval system is expected to retrieve the most semantically similar image from a small collection of 50 images, but for an arbitrary hand drawing generated by *Alan01/ AlanOnline* via their respective sketch interfaces.

Since the collections we are using in our case do not belong to a specific domain, it is difficult to predict if the boundary based or region based algorithms will perform better. Additionally, since we have a very small collection of shapes and the queries are expected to be generated by *Alan01* and *AlanOnline* through an interface where a user can draw any arbitrary image, the similarity retrieval of an object from the collection becomes a big challenge. For instance, the image in the database for the object snake shown in Figure 4.2(a) is very different from the user drawn snake shown in Figure 4.2(b) and yet the system should ideally retrieve the snake shown in Figure 4.2(a).

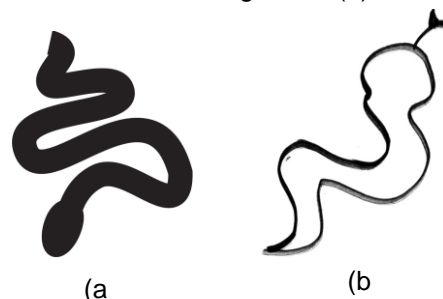


Figure 4.2: Examples of collection and query images

The problem of recognising arbitrary shaped images irrespective of various geometric transformations such as, orientation, scaling, translation and shearing effects has been tackled in the fields of Robotics, object recognition and computer vision, among others. Object recognition with shape features deals with finding a match between certain features obtained from the shape of a query submitted with that of the different instances of the model objects in the database.

A number of approaches have been proposed for 2D object recognition based on shape features. Depending on the feature extraction mechanism, the proposed methodologies can be categorised into two groups: boundary based methods or region based methods [Gonzalez, 2002].

The boundary based methods extract global/local features from the outer boundaries of an object. The simplest way of representing a shape is via a chain code representation, where the direction of the neighbour pixel to a reference pixel is recorded while a boundary is traversed in a clockwise or a counter clockwise direction. Shape boundaries, also known as contours, are represented, by many researchers, using statistical moment variants [Kim, 2000a; Kim, 2000b; Khotanzod, 1990].

Despite the fact that boundaries are very helpful to preserve the shape of an object, they are very sensitive to noise – the presence of additional pixels or the absence of a few pixels on the boundary completely varies the representative feature of the object. Most methods which work on boundaries are sensitive to starting points and also to the directions in which the boundary is traversed. They generally work in only limited setups. In addition to these disadvantages, there are many applications where the boundary features are not relevant and hence demand the use of region characteristics, i.e., the characteristic of an object within the boundary of the object.

Statistical moments being applicable to all areas in general, were also used to compute region features by a few researchers [The, 1986; Taubin, 1991]. Other, features such as those of the MPEG7 framework was used in [Prasad, 2004]. A few simpler region features can be found in [Gonzalez, 2002].

Almost all methods proposed so far for shape retrieval, have a fixed model dataset, containing images which may be occluded, distorted or otherwise transformed. The free style drawings in the collection open up a new research question in the field of shape representation and retrieval. Hence, we have used a variety of boundary based and region based features to investigate the suitability of these feature for free style hand drawn shape recognition and retrieval as explained in later sections.

4.2 Features Used in the Backend

Due to the system requirement of retrieving the most semantically similar image and the collection properties, we mainly focus on the shape based features. In the back end, we implemented one of the best performed content based feature and five shape based features. The back end also allows the front end to use different combination of these six features or only one single feature to achieve the best retrieval performance. The details of these features are as follows:

4.2.1 Edge Histogram Descriptor

The spatial distribution of edges in an image is a very useful descriptor for similarity search and retrieval [Manjunath, 2002]. To compute an edge histogram descriptor, the image is first divided into 4*4 pixel sub-images, and then the local-edge distribution for each sub-image is represented by a histogram. Edges in the sub-images are then grouped into five different types; these are vertical, horizontal, 45 degree, 145 diagonal and non-directional edges. In total, 80 histogram bins are required to represent each edge histogram.

Since considering the local-edge histogram alone may not be sufficient for image matching, global-edge descriptors are also implemented in addition to local edge descriptors. Additionally, edge distribution information for the whole image, horizontal and vertical semi-global-edge distributions, as well as local edge distributions are also used to improve the matching performance. The global-edge histogram and semi-global-edge histograms are estimated from the local 80 bins. The global-edge histogram is calculated by accumulating the five types of edge distributions for all sub-images. The semi-global-edge histograms are estimated from the grouped sub-images, which are grouped in the following ways: grouping of four vertical sub-images, grouping of four horizontal sub-images and grouping of four neighbour sub-images. In this case, 13 different segments are created. The corresponding edge histograms for each segment are then calculated using the local-edge histograms. After combining the local, the semi-global and the global histograms, a new histogram with 150 bins is constructed for similarity matching.

To calculate the similarity between two images in the edge domain, the following distance measure using two edge histograms A and B is adopted:

$$EdgeSim(A, B) = \sum_{i=1}^{80} |h_A(i) - h_B(i)| + 5 * \sum_{i=1}^5 |h_A^g(i) - h_B^g(i)| + \sum_{i=0}^{65} |h_A^S(i) - h_B^S(i)|$$

where h_A and h_B are the normalized histogram bin values of image A and B, and where h_A^g and h_B^g are the normalized histogram bin values for the global-edge histograms of image A and image B, respectively.

4.2.2 Contour Shape Descriptor

Object shape features provide a powerful clue to identity objects [Bober, 2001]; the contour shape extractor used in the system is implemented in two steps: (i) locate the objects in the image and extract its outer contour, followed by (ii) extract the contour features.

Given an image, it is first transformed to obtain a monochrome image where object and background are represented in contrasting colours. An edge extractor is first applied to obtain the edge information of the object, followed by a morphological open-close operation in order to smooth the computed contours and connect any breakage in the contours.

Although there are many edge detection algorithms available, we make use of very simple convolution in the spatial domain to obtain the boundaries of the object. The structuring elements (also called masks or filters) shown in Figure 4.3 are used to obtain inner and outer boundaries of objects in an image.

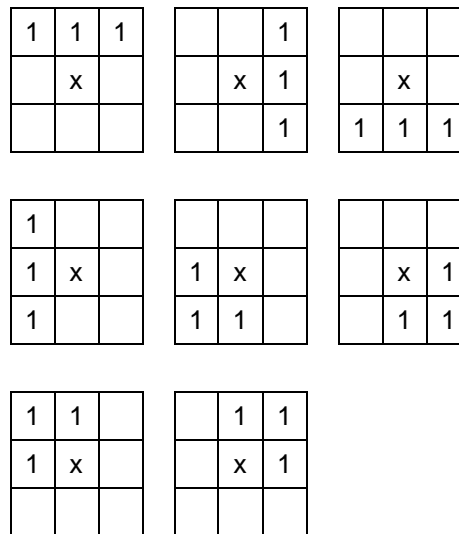


Figure 4.3: The extracted contour of apple image in Figure 4.5.

From all such generated region contours, the largest closed contour is selected and used to represent the object. Since it is usually the outer boundary of the image object, it preserves object shape. An example of the largest extracted contour of an apple image is illustrated in Figure 4.4. This is with the assumption that there is only one object in an image but may have many regions within itself.

Given the extracted contour, we follow the contour in a clockwise manner and keep track of the direction as we go from one contour pixel to the next, represented using a chain code [Kim, 2000a]. Given a contour pixel, the next contour pixel is a pixel from its 8-connected neighbours. A unique number, from 0 to 7, is used to represent each direction. Looping through all the contour pixels, any contour can be represented as a numerical array for similarity matching.

Using chain codes is efficient because of the constraints on their construction. Only a starting point is represented by its location; the other points on the shape curve can be represented by successive displacements from grid point to grid point along the curve. Since the chain code is invariant under boundary rotation, it makes the similarity matching between two contours relatively easy.

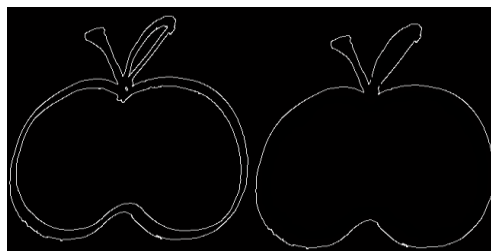


Figure 4.4: The structuring elements used for finding boundaries of an object in the image

4.2.3 Block Ratio

The fact that the shape features for irregular, free style drawings can be very complicated we have also studied the pixel distribution of the object in an image. However, depending on the pressure an artist applies on the instrument used to draw the picture, the concentration of pixels in regions may vary. It is more appropriate therefore to thin the boundaries of the object to obtain a single pixel thickness shape. This helps to generate a fair matching irrespective of the thickness due to different pressures. However, employing any thinning algorithm will distort the shape of the object, hence we extract the boundaries of the object as explained in the previous section.

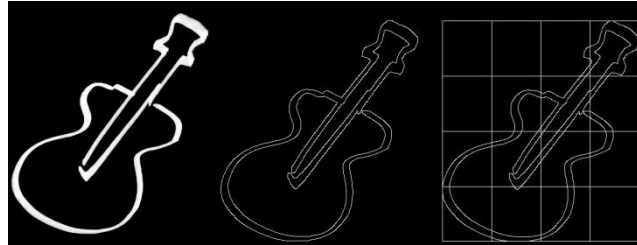


Figure 4.5: The boundaries of guitar and the minimum bounding rectangle

Once the boundaries of the object are extracted, a minimum bounding rectangle (MBR) is used to fit the object. An MBR is the smallest rectangle that can completely contain an object in an image. The object within the MBR can be normalized, i.e., scaled down or up, to obtain a standard sized image for easier matching. The minimum bounding rectangle is then partitioned into a number of blocks, as shown in Figure 4.5. The ratio of object pixels and the background pixels are computed for each block and are recorded as a feature descriptor.

4.2.4 Centroid Profile

The boundary pixels of the object are used to generate another feature descriptor for the image, a "Signature" for the object. Using every object pixel on the object boundary, the centre of the object is computed. With this centre, the object is scanned in a counter clockwise direction from 0 degree to 360 degrees with a certain interval. The distance of the pixel at, say, d degrees, from the object centre is computed and recorded as another feature descriptor. Figure 4.6, shows an instance during the generation of the centroid profile for a guitar object, showing the direction angles at intervals of 45 degrees.

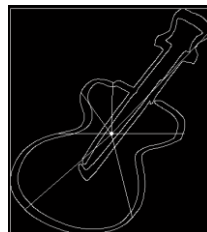


Figure 4.6: Centroid profile of guitar, showing the direction angles at intervals of 45 degrees, used in the Object Signature feature

4.2.5 Object Signature with reference to Axis of least inertia

The centroid profile feature computed in the previous section, does not take care of geometric transformations like rotation and flipping. Hence, we try to extract some features with a reference axis that is invariant to transformations. For this purpose we first compute the axis of least inertia. Theoretically, the axis of least inertia passes through an object in a way to pierce through the points which balance the object and for any object there can be only one such axis. This property promises to yield the same axis for an object irrespective of the angles they are transformed into.

Let $B = \{b_1, b_2, b_3, \dots, b_m\}$, be a set of m points on the boundary of an object. Let (x_i, y_i) be the coordinates of a point b_i for $i = 1, 2, 3, \dots, m$.

The centre (\bar{x}_B, \bar{y}_B) of the object is computed as given in (1)

$$\bar{x}_B = \frac{1}{m} \sum_{i=1}^m x_i; \quad \bar{y}_B = \frac{1}{m} \sum_{i=1}^m y_i \quad \dots (1)$$

The slope angle θ of the axis of least inertia is estimated as follows. Let α be the angle between the x-axis and the axis of least inertia. The axis of least inertia is defined by the line for which the integral of the square of the distance to points on the object boundary is minimum [Tsai, 1995]. This integral is given by

$$E = \frac{1}{2} \left((a + c) - (a - c)\cos 2\alpha - b'\sin 2\alpha \right) \dots (2)$$

Where,

$$a = \sum_{i=1}^m (x_i - \bar{x})^2$$

$$b' = 2 \sum_{i=1}^m (x_i - \bar{x})(y_i - \bar{y})$$

$$c = \sum_{i=1}^m (y_i - \bar{y})^2$$

The derivatives,

$$\frac{dE}{d\alpha} = (a - c)\sin 2\alpha - b'\cos 2\alpha; \quad \frac{d^2E}{d\alpha^2} = 2(a - c)\cos 2\alpha + 2b'\sin 2\alpha$$

$$\text{If } \frac{dE}{d\alpha} = 0, \text{ then}$$

$$\alpha = \frac{1}{2} \tan^{-1} \left(\frac{b}{a - c} \right), \quad -\frac{\pi}{2} < \alpha < \frac{\pi}{2}$$

Finally the slope of the axis of least inertia is computed as,

$$\theta = \begin{cases} \alpha + \frac{\pi}{2} & \text{if } \frac{d^2E}{d\alpha^2} < 0 \\ \alpha & \text{otherwise.} \end{cases}$$

Once the axis of least inertia of an object is computed, an axis perpendicular to the axis of least inertia passing through the centre of the object is computed. This divides the object into four co-ordinates with the axis of least inertia aligned to the 0 axis. This enables to extract a reference axis, that is, the axis of least inertia irrespective of the orientation of the object. The block ratio, and the signature features explained in previous sections are then extracted with respect to this reference axis.

4.2.6 Horizontal and Vertical projection profile

As mentioned earlier with reference to Figure 4.2, the objects drawn by a user do not necessarily preserve the exact size, but they have the same structure, such as the rectangularity, elongations, etc. Hence, after finding the axis of least inertia we, try to capture the structure by projecting the frequency of the pixels on to the x- and y- axes. The x-axis is aligned along the axis of least inertia and the axis perpendicular to the axis of least inertia is assumed to be the aligned with the y-axis. Figures 4.7(b) and (c) show the horizontal and vertical projection of the example image shown in Figure 4.7(a).

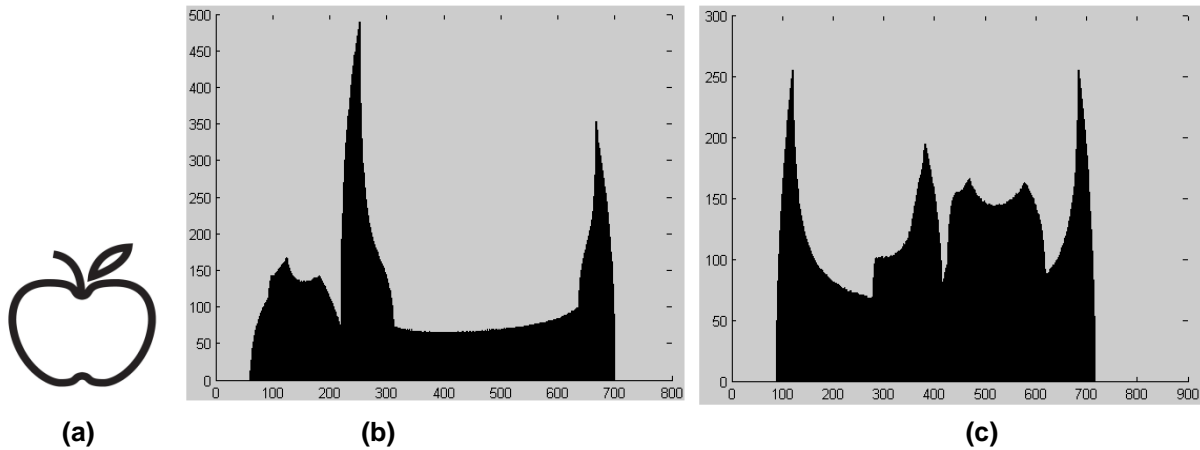


Figure 4.7: (a) An example image (b) Projection along the x- axis (c) Projection along the y-axis

4.3 Fusion techniques

As reported in D.5.5.3, depending on the query submitted by the user, many modalities can be used to retrieve the relevant data. This is diagrammatically represented in Figure 4.8. When the user issues a multimedia data, many modalities like audio features, temporal video features, textual features and image features are retrieved. Since each of these modalities have many features from their own domain, such as ASR script, metadata and concept labels as in textual features; voice, music or instrument detection features as in Audio etc,. The second level of fusion would be fusing the results originating from various modalities used for retrieval. Since we deal with only video and images, we limit to describing result fusion mechanisms from various visual features in this section. However, the result of fusion from different systems will be presented in Chapter 7.

In general the fusion of results from different features, and also of the features from the different image examples, is a major problem. What should be the criteria to merge the results and at what levels to merge the results is an open issue. On one hand, the fusion can be done at feature level and on the other hand it can be evaluated at semantic level; moreover it could be application dependent.

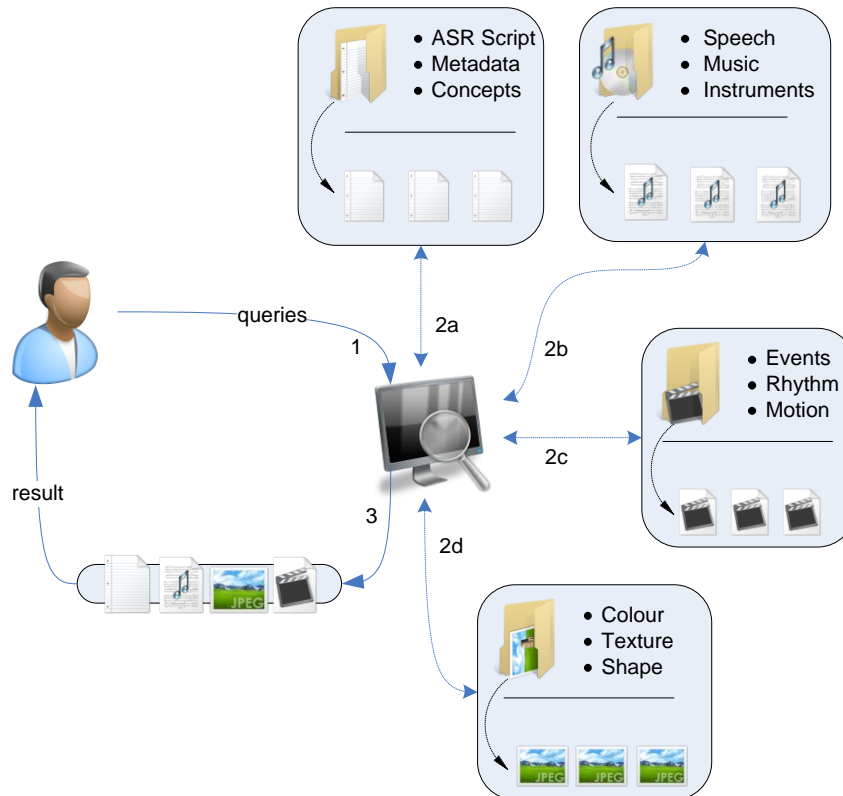


Figure 4.8: An architectural view of the various level of result fusion (2a) Fused result from various textual features (b) Fused result from Audio features (2c) Fused results from video features (2d) Fused results from Image features.

Each of the image features previously described in this chapter (and also in D.5.5.3) generates a separate result list for a given query image. We have implemented and tested six different fusion methods:

Reciprocal Rank

Reciprocal rank is a simple summing of the reciprocal of the rank of each result [Bober, 2001]. For an image i and rank list j , the reciprocal rank is defined as:

$$r(image_i) = \frac{1}{\sum_j 1/position(image_{ij})}$$

Where $position(image_{ij})$ is the rank position of image i in rank list j .

Borda Count

The Borda count method gives a number of points to each image at a given rank, and then sums the points to determine the final ranks. We use the system described in [Bober, 2001], where if there are n possible image results across all ranked lists being merged, an image at rank i will be given $n-i$ points.

Borda is a voting method in which voters rank the candidates in an order of preference. Points are given for the position of a candidate in a voter's rank order. The candidate with the most points wins.

The number of points given to a candidate for each ranking is determined by the number of candidates standing in the election. Thus, under the simplest form of the Borda count, if there are five candidates in an election then a candidate will receive five points each time they are ranked first, four for being ranked second, and so on, with a candidate receiving 1 point for being ranked last. In other words, where there are n candidates a candidate will receive n points for a first preference, $n - 1$ points for a second preference, $n - 2$ for a third, and so on.

Condorcet Method

The Condorcet method takes into account the relative positions of the ranked images, building up a matrix of which image results come above, below, or are tied in rank with other images described in [Bober, 2001].

The Condorcet method is based on democratic election strategies. The document which beats each of the other documents in a pair wise comparison wins. For example, there are three candidate documents, a, b, and c, in five systems, A, B, C, D, and E. They have following relations: A: $a > b > c$ - B: $a > c > b$ - C: $a > b = c$ - D: $b > a$ - E: $c > a$. Then, the pair-wise winners can be represented in following manner:

	Win	Lose	Tie
a	2	0	0
b	0	1	1
c	0	1	1

Considering the rules, the final ranking of documents is $a > b = c$

Weighted Voting

The features are first weighted for its suitability to retrieve most effective results for a topic. A voting method then counts the number of ranked lists an image result is part of, and ranks primarily on this number and the feature weights.

CombSum, CombMin, CombMax

These methods use the minimum, maximum or sum of the similarities generated by the different features for each result image. In this scheme, the similarity scores for an image result across all ranked lists are used as input, where a new score is generated by taking the minimum, maximum or the sum of the scores [Zahn, 1972].

The first four methods described in this section are based solely on the ranks of the results, whereas the final three utilise the similarity scores generated by the similarity matching function used in retrieval.

4.4 Evaluation of the Retrieval System with the TAIK Collection

Image retrieval plays a central role in the multimedia art productions (Chapter 9) that consist of two individual parts: Alan01 – a physical installation, and AlanOnline – the installation's online counterpart. In both cases the image retrieval system triggers the associational story in motion. The main input is a canvas where the user can create line drawings which are then used as a reference for the image retrieval. However, for the specific needs of the installation, the retrieval is made from a limited set of about 50 symbolic images. The graphical presentation and selection of these has been made bearing in mind the nature of the interface. The symbols are similar to what you would expect a user might draw in a short time of five to ten seconds. Another requirement for the selection of the symbols is that they are connected to the context of Alan Turing's life.

The idea of the drawing interface is therefore to enable a non-textual input to an art piece, which can still be translated to symbols and their textual meanings. From there on, the associational narrative structure script of the art piece can start to function.

In order to evaluate which of the combinations of features and fusion techniques produced the best results, a small evaluation was carried out which simulated a series of known item searches. Two image collections were used, the first being the collection of target images, the other a set of hand-drawn query images.

As a first step, a set of relevance judgements was manually created, where each query image was matched to its ideal target image (an example is given in Figure 4.9). In addition to an ideal "target", we also defined for each query zero or more alternative images which were deemed to be acceptable results for the query, but not ideal (Figure 4.10). This latter list was defined to enable us to consider the matching between query and target(s) as a fuzzy mapping, in order to model what was thought to be "acceptable" in the implementation of *Alan01* and *AlanOnline*, where the output generated need not always be exact. Indeed, the collection may not necessarily contain any images similar to the query.



Hand drawn query



Target image from

Figure 4.9: Example query image and associated target image**Figure 4.10: Alternative matches for the query in Figure 4.9**

Once the manual relevance judgements were constructed, each query image was matched to 47 collection images, for each combination of feature, fusion technique, and ten different rank list depths. In addition to the three features defined in Sections 4.2.1, 4.2.2, and 4.2.3 which consisted of Edge Histogram (Eh), Contour Shape (Cs), and Signature (S), two other colour based features were used for comparison purposes: Colour Histogram (Ch) and Colour Layout (Cl). Both of these features were as defined by the MPEG-7 standard **Fehler! Verweisquelle konnte nicht gefunden werden..** While these features were not expected to perform as well as the others, they are commonly used in a number of other retrieval situations, and provide an interesting comparison to the features described in this chapter.

The fusion techniques were as described in Section 4.3, consisting of the reciprocal rank (Rrank), voting (Vote), Borda, Condorcet, Min, Max, and Sum methods. Since each of these fusion methods operate on ranked lists, we also altered the depth of the ranked lists fused to between one and ten, in order to investigate the impact of fusion depth. By increasing or decreasing the depth, we introduce more or fewer image results for fusion. In the situation where only a single feature is used to perform the retrieval, the fusion is of course not required.

Table 4.1 shows the ten feature, fusion and rank list depth combinations which identified the greatest number of correct target images (left), or the greatest number of alternative target images (right). Combinations are coded as a list of features, the fusion technique, followed by the ranked list size. Note that it is only if the top result of the ranked list matches either the target image or one of the alternative targets will a correct result be counted – if the correct result appears at rank position two or below, the combination will be counted as a fail. While absolute, this reflects the needs of the *Alan01* and *AlanOnline* systems. Additionally it should be noted that when searching for an alternative target image, the ideal target is also considered as a correct response to the query.

% of ideal targets		% of alternative targets	
Combination	%Corr	Combination	%Corr
Eh/Cs/S-Vote-4	32	Cl/Cs/S-Vote-10	47
Cs/S-Vote-7	32	Cl/S-Vote-9	42
Cs/S-Vote-4	32	Cl/Eh/Cs/S-Vote-10	42
Cs/S-Vote-3	32	Cl/Cs/S-Vote-9	42
Cl/Eh/Cs/S-Vote-10	32	Cl/Ch/Cs/S-Vote-9	42
Ch/Cs/S-Vote-3	32	Cl/Ch/Cs/S-Vote-10	42
Eh/S-Rrank-9	26	Cl/S-Vote-8	37
Eh/S-Rank-10	26	Cl/S-Vote-10	37
Eh/Cs/S-Vote-3	26	Cl/Eh/Cs/S-Vote-9	37
Eh/CS/S-Min-10	26	Eh/S-Vote-4	32

Table 4.1: The percentage of target images and alternative images (including target) identified by the top ten feature, fusion and result list combinations

It can be seen in Table 4.1 that the best techniques for identifying the target image are correct for 32% of the queries; when alternative targets are also included; this increases to over 40% of queries being correctly recognized. The best combinations for both situations are, however, different. For detecting a single ideal target image, the top 6 combinations include the Contour Shape and Signature features at various sizes of rank list. When also allowing for the alternative target images, the Colour Layout is present in 9 out of the top 10 best combinations, while a combination of Colour Layout and Signature performs as well as other combinations which include Contour Shape.

Looking at the fusion techniques, the voting method dominates the best performing combinations in both retrieval situations. Rank list size does vary, however, with there being a trend for smaller ranked lists when retrieval aims to only return the single ideal target image. The best performing combination which is common to both situations combines Colour Layout, Edge Histogram, Contour Shape, and Signature together with voting fusion and a ranked list size of 10.

To expand on Table 4.1, Figures 4.11 and 4.12 show the distribution of the different retrieval combinations for different performance levels on the x-axis. Figure 4.11 shows that there are a significant number of possible combinations (over 200) which can return the correct image 21% of the times. On Figure 4.12, to retrieve any of the images classed as acceptable, it can be seen that the distribution is skewed to the right – only very few of the combinations performed better than 26% correct, although there are again many combinations that were able to perform at a level of 21% correct results.

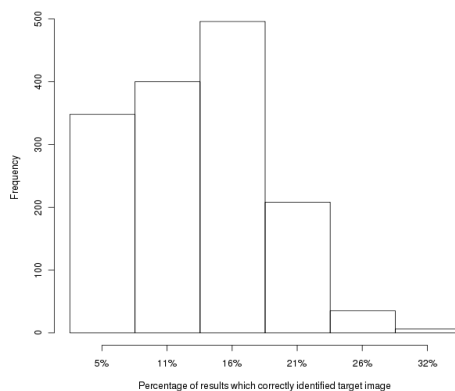


Figure 4.11: Number of retrieval combinations for different target performance levels (retrieval of single target image)

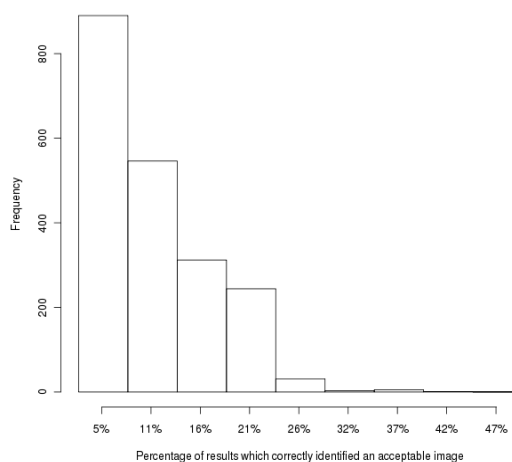


Figure 4.12: Number of retrieval combinations for different performance levels (retrieval of any acceptable target image)

4.5 Summary

The *Alan01* and *AlanOnline* concept and production present a very demanding challenge to an image retrieval system, where a visual retrieval is required which is consistent with the behaviour of the human visual perceptive system, and which can present high precision results for input queries. The level of accuracy of the results has to be high to ensure that the retrieval within the production doesn't just become a technical gimmick. Instead it should be an integral part of the artwork.

In the present production, one of the central issues has become the *predictability* of the retrieval results, an aspect which is not typically considered important in the field of image retrieval. In an interface that uses this technology, the user is tempted to start to test the system or even play against it. Seeing the results which the system has delivered previously affects the imagery that a user starts to draw henceforth. If a user tries to replicate the images he/she has seen in previous results, the following results need to be consistent in order to avoid the feeling of randomness in the system, and to ensure that the communicated illusion of the installation is not broken.

When the resulting image set is limited and preselected, the level of graphic detail also needs to be relative to the system's ability to recognize details. In the case of the experimental art productions, an added factor is limiting the number of search results. In a conventional image retrieval system, if the user is presented with the top ten retrieval results, it is typically considered sufficient if a significant ratio of those results is relevant – the fact that the highest ranked result isn't relevant doesn't render the whole result list unusable, which is the case here.

From the point of view of the retrieval, the implementation of an image retrieval system tailored to the needs of the *Alan01* / *AlanOnline* production has been a very challenging endeavour. While the image collection to be searched is small, the reality has been that creating a content based image retrieval system which is acceptable for production use has been extremely difficult, and is likely to continue to

be difficult. Problems include the lack of training data, and the difficulty in the judgment of relevance within the context of the work. While initially the aim was to concentrate purely on visual similarity, in practice this is difficult: we naturally think in terms of the semantics of the image. I.e. the knowledge that an image is an “apple” or “computer” can override particular visual similarities which may be present between two images. Working on the problems raised by the *Alan01* and *AlanOnline* productions has resulted in a rethinking of the needs and roles of the content based image retrieval.

The ability to convert non-textual input into textual concepts or symbols is fascinating and opens up endless opportunities in the context of interactive art. Image retrieval technology shows great promise as a tool for creating sophisticated installation and artwork interfaces, potentially allowing the creation of rich user interfaces, but ones which can still be used immediately by a visitor, whether adult or child.

The preliminary tests on the *Alan01* / *AlanOnline* delivery systems’ ability to produce engaging mini-narratives are promising. With the logic of the associational narrative structure script working, future concentration is on the rhetoric of physical space, user body movement, and the use of moving image and sound to support the logic. The final success of the production will be defined on how successfully we are able to tie up the rhetoric with logic, actual user interaction modes and tempo with the nonlinear narrative elements.

5 The AspectBrowser Interface

This Chapter provides a description of the AspectBrowser search interface, which is designed to allow users to search video and image databases and external Web search engines, and the Indexing interface, which is used to create and index image and video collections. Both interfaces are complementary, allowing users to provide new data collections which can then be searched by others, which also providing a search interface which can be search both local and remote resources.

5.1 The AspectBrowser Search interface

The AspectBrowser is the content-based search engine which has been designed to search both SALERO-sourced data (typically image and video) and also provide a meta-search for the Web. It is designed around the idea of a search being split into a number of “aspects”, each aspect is represented explicitly on the interface, and allows the user to search using text or visual queries on a range of databases.

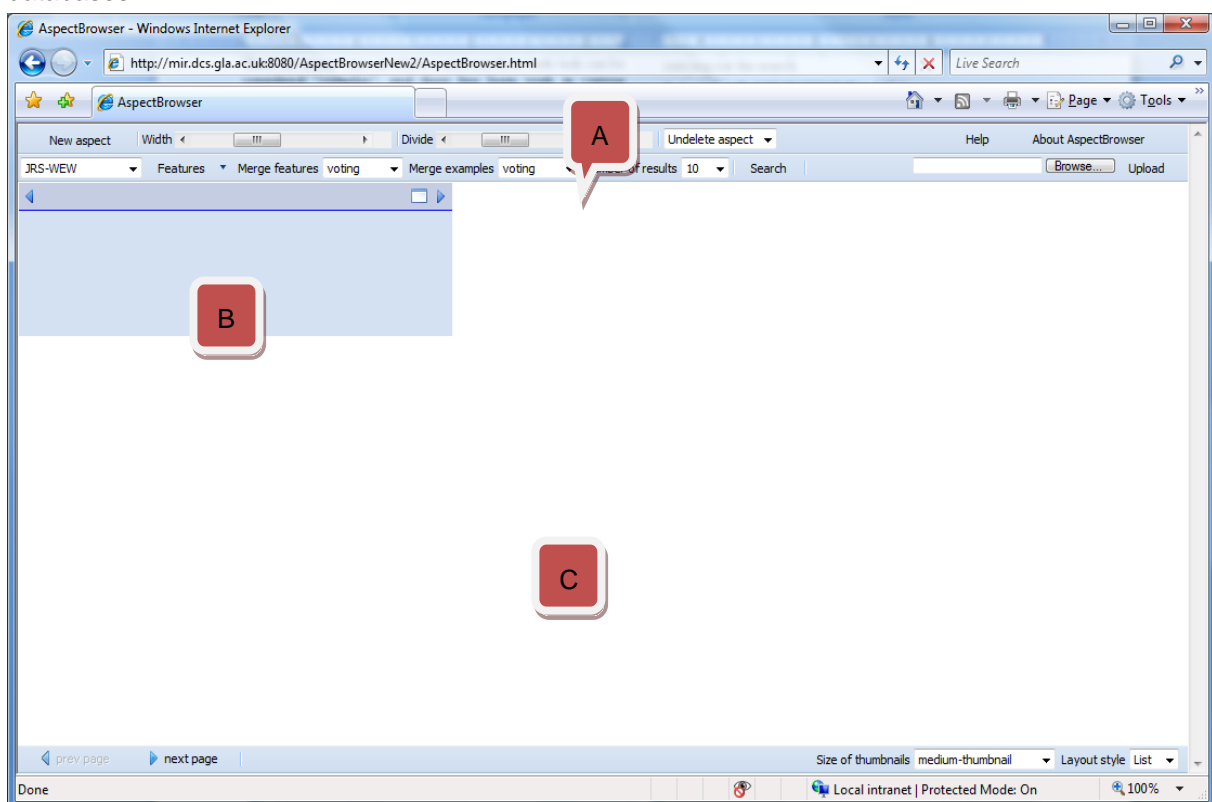


Figure 5.1: The AspectBrowser interface at startup

The aspectual interface is a development of the system described in [Villa, 2008], a screenshots of the interface at startup is shown in Figure 5.1. The interface is built around the concept of search aspects, where each aspect contains the following main elements:

1. a name, which is by default set to the last query executed, but which can be explicitly set by the user when desired
2. a list of selected documents, i.e. the Web pages which the user judges as being relevant to the aspect
3. the current search query entered by the user
4. the list of search results for the current query, where clicking on the document's title will display the corresponding Web page in a pop-up window
5. a history of the searches carried out in the aspect, and a list of any deleted documents; and finally
6. the position of the aspect within the overall sequence of aspects.

Each aspect is a self-contained entity containing all of the above states: each has its own query history, undo history, current query, search results, etc. The “new aspect” button at the top left of the interface creates new aspects, adding them to the far right of the display. At startup, a default (empty) aspect is created and shown, and the Web page will look as shown in Figure 5.1.

The interface has three main sections, indicated by the labels in Figure 5.1:

- A. The toolbars, which provide various search and other functions
- B. The tab area, where each aspect is shown as a blue area where relevant images or other documents can be placed
- C. The search results area

The toolbar area is made up of two rows of switches, the first consists of:

1. The “new aspect” button
2. Slider which adjusts the width of each aspect (tab) at the top of the screen (E)
3. Slider which adjusts the division between the tabs and search results area
4. Pull down list of deleted aspects. Clicking an aspect will undetete it, placing it at the end of the list of tabs
5. Help and about information

The second line of the toolbar area is made up of:

1. A text query box, which is not displayed if the current collection does not have a text index (and therefore cannot be searched by text)
2. List of collections pull-down menu, which allows the user to select a collection to search
3. Features menu, which allows one or more features to be selected. The features available varies depending on the collection. These will be extracted from images which have been selected in the aspect, and used in generating results, if checked. You can select combinations of features
4. Merge voting method, which is the technique used to merge different features when searching
5. Merge examples method, which defines the technique used to merge features from different examples. Both this and the previous merge features option can be left as defaults for most purposes, but are included to enable easy experimentation with different feature and merge combinations. The best methods to use are likely to change depending on the collection searched
6. The number of results required
7. Search button, which starts a search on the selected collection. A search can use both the text query, plus the visual features from any selected images of the aspect
8. The browse and upload image function: clicking the “Browse” button will bring up a file selection dialog box to your local computer. After selecting an image, and clicking ok to dismiss the file dialog box, you can then click the “upload” button. This will upload your selected image, which will appear in the aspect’s tab (area D), and which can then be used in searching

It should be noted that not all collections can be searched by visual feature (e.g. we cannot search youtube or the Web via yahoo in this way), while other collections cannot be searched by text (since this image collection contains no text information). The collections available for searching will vary over time, but as of writing include the following:

- symbolitUGlle: The set of images used in TAIK’s Alan01 and AlanOnline experimental productions
- AMPE3: Activia Multimedia videos from their 3rd experimental production (only searchable by image example).
- Test-TAIK: prototype drawings by TAIK which were used in the development of Alan01 and AlanOnline
- TAIKWeb: a downloaded set of Web images for the Turing machine opera (can only be searched by image example)
- Flickr: searches the Flickr image Web site via it’s Web service (only searchable by text)
- Yahoo-Boss: a text yahoo Web search, executed via Yahoo’s BOSS service (only searchable by text)
- YouTube: YouTube search, via their Web service (only searchable by text)

- Trecvid-2008: the TRECVID-2008 video collection, searchable by text or image
- Trecvid-2007: the TRECVID-2007 video collection, searchable by text or image
- Tinyplanets: PGP image asset collection (only searchable by image example)
- Bingandbong: PGP's little cartoon blighters, all the videos (only searchable by image example)



At the very bottom of the interface is one final toolbar which is used to interact with search results. This contains:

1. A next and previous results buttons
2. Size of thumbnails: alters the size of the displayed thumbnail images.
3. Layout style: selects between a standard list format or a grid of images

The interface makes extensive use of drag and drop. Using the crosshair icon to the left of each search result, documents can be added to an aspect's marked list by dragging them onto the corresponding screen area. There is no restriction on which aspect a result can be dragged onto, therefore it is possible to drag a search result from one aspect directly onto the relevance list of a different aspect. Documents can also be dragged and dropped between different aspects, allowing the reorganization of material across the aspects, and can also be re-ranked within each aspect. Some of the more common actions are described in the following sections.

5.1.1 Executing a text query

Executing a text query is very simple, just enter the text and press search, noting that only some collections are text-searchable, e.g. after starting the interface, and creating a new aspect, enter "Sky" in the text box, and then select the "Flickr" collection.

Search results appear as a list, most relevant first. Each search result has (where appropriate) a thumbnail image on the left hand side, followed by a document title underlined (in this case the titles are shot numbers from videos) with a URL underneath. A textual document summary will also appear where appropriate. At the far left of each result is a green arrowed icon  which allows that result to be dragged and dropped. For example, by clicking the  icon next to the first result, and dragging that result into an aspect in the tabbed area of the interface, that result will be marked as belonging to the corresponding aspect. There is not explicit limit to the number of shots which can be added to an aspect in this way. When you execute a search, the aspect's tab will also change to reflect the last text query entered. Clicking on the underlined title from a search result will either display the corresponding Web page or image in a new window.

Each search result has another icon to it's left, a right arrow () which when pressed opens a context menu. This provides three options:

- Delete: if the result is part of an aspect, this will delete the result from the aspect
- Copy: if the result is part of an aspect, this will copy the result, creating a new result which can then be copied to another aspect as required
- Info: displays some basic information about the result, such as it's ID, URL, Title, etc.

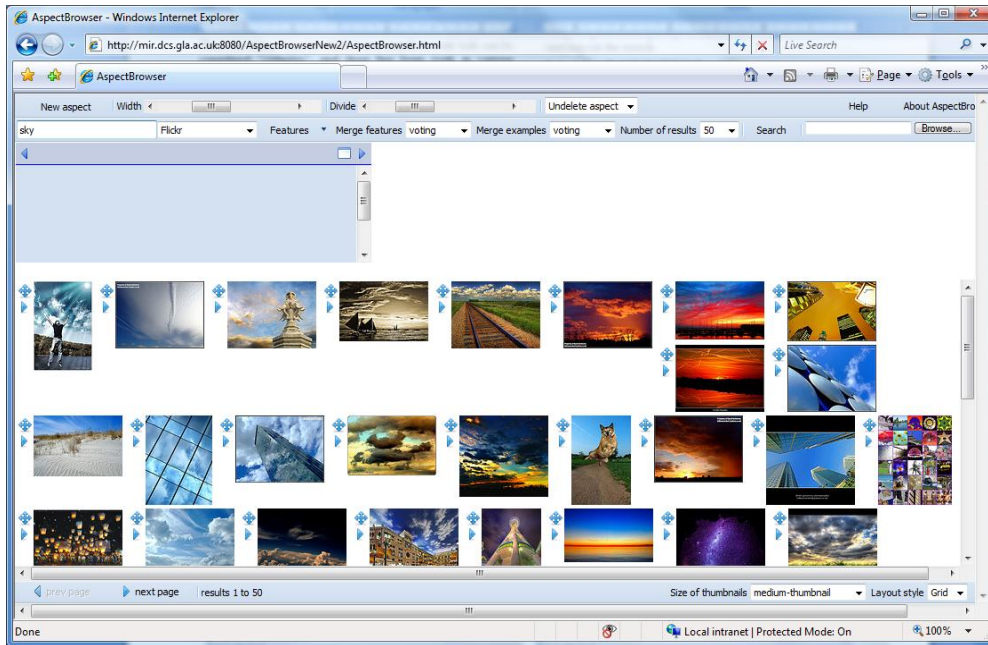


Figure 5.2: Search results after executing the query "Sky" on Flickr

5.1.2 Executing an image query

Any images which you have marked in the tab area of an aspect can be used to carry out visual searches on an appropriate collection. For example, after carrying out the text search in Section 6.2, we can now carry out a visual search using an image from Flickr. For example, select "pgp-test" from the collections list, and add one of the Flickr sky's to the aspect and press the "Search" button again. Results similar to the following may be seen, depending on the sky selected:

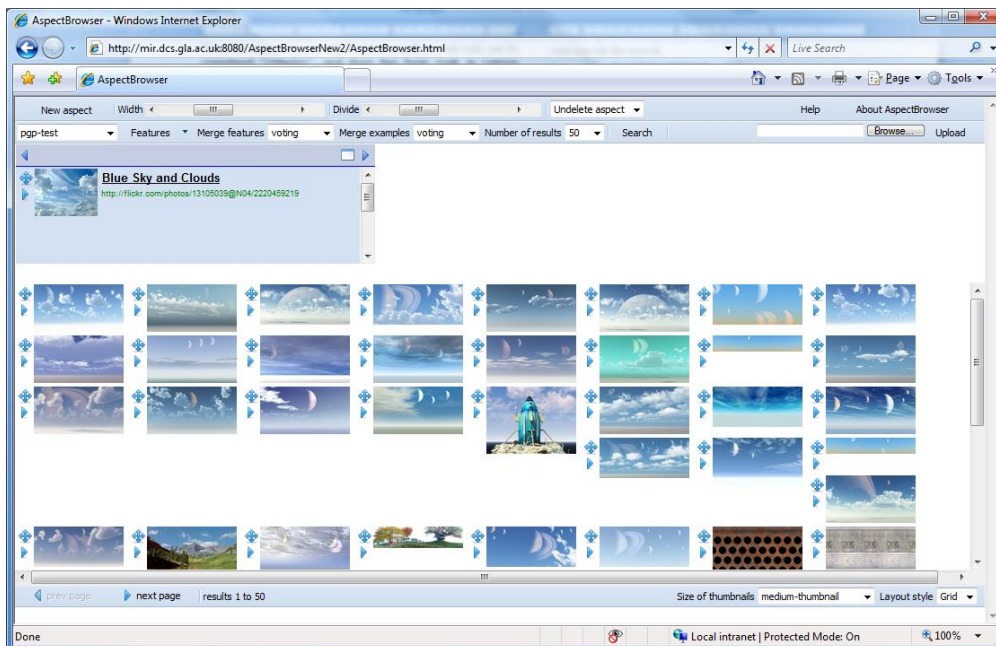


Figure 5.3: Interface after carrying out a content-based search

This particular collection has no text, and so our text query will be ignored (the text box will disappear). Instead the selected image will be used as the query, used to generate the search results. By removing features, different results will be returned.

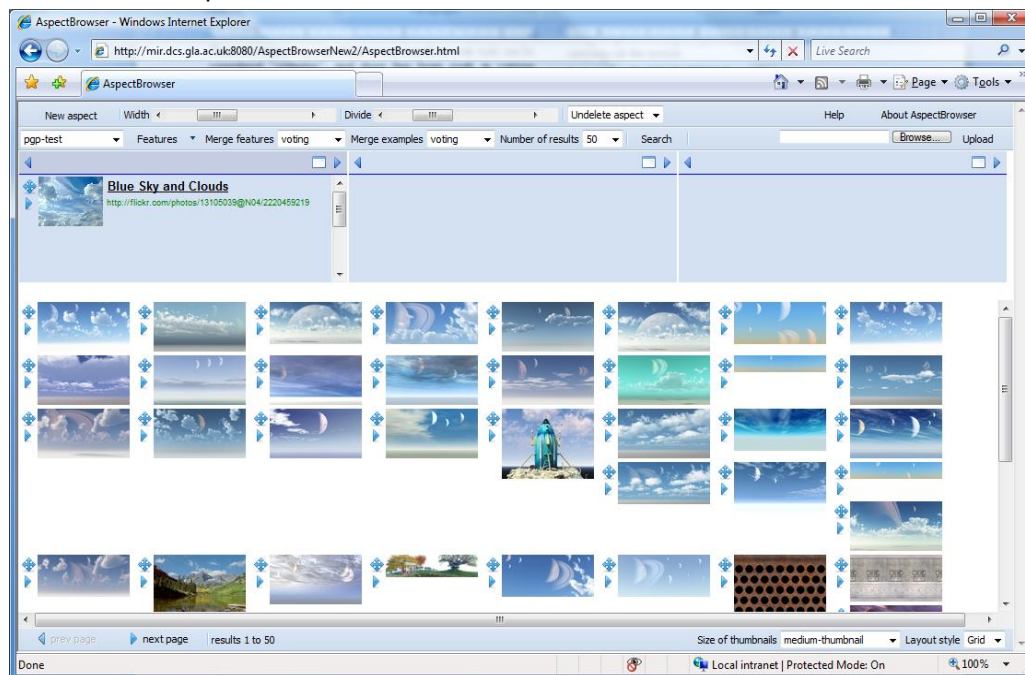
Carrying out a query on an uploaded image

As well as selecting search results, you can also upload images into the interface using the "Browse" and "Upload" buttons. Clicking "Browse" will open a file selection dialog box, allowing you to select an image from your local computer. For example, after pressing Browse we select an example image, in


this case a drawing. After pressing “open” on the dialog box, you must then press “upload” to upload the image to the search interface, after which it will appear on the aspect’s tab.





5.1.3 Using multiple aspects

The interface supports the creation of multiple search “aspects”. Up until this point we have searched in only a single aspect, although by pressing the “New aspect” button you can create as many aspects as you require. The screenshot below shows the result of pressing the “new aspect” button a further 2 times, to create 2 new aspects:



Each aspect is represented by a single tab at the top of the interface, the currently selected aspect being highlighted in a darker blue colour. By clicking on a tab, you select that aspect, which will update the search results and search panel at the bottom of the interface. As can be seen above, this third new aspect has no query, and no search results. Clicking on the first tab switches back to the first aspect.

Using drag and drop, it is possible to drag results or marked images between aspects, for example, we can drag the apple image onto the second aspect by dragging it with the  icon, and dropping it onto the second tab. Clicking on the second tab then highlights that aspect, with its associated empty result list, after which a search can be carried out as before. By clicking the tabs, you can now switch between the two result lists. Search results can also be dragged and used as queries in a similar way.

At the top of each tab of each aspect there is a bar with left  and right  arrows, and a third “menu” icon . Clicking the left or right arrows will move the aspect left or right in the sequence, allowing you to reorder the aspects. Clicking the Menu icon  will bring up a menu with three options:

- Rename allows you to give a name for the aspect, which will turn red.
- Delete will delete the entire aspect, including any selected images and search results. It’s name will appear on the “undelete aspect” tab, which when clicked will reinsert the aspect to the end of the list of tabs
- Copy will copy the contents of the aspect, inserting the new aspect at the end of the list of tabs

5.2 The Indexing Interface

The Indexing Interface (II) allows administrators to upload and index new multimedia collections on the search/indexing server. These collections once indexed they will be available in the search interface for users to search. The indexing process, Figure 5.4, pre-process multimedia items, such as videos or images, and creates a searchable index. In Figure 5.4 an overview of the process is illustrated. In general video files are first segmented in shots which are the unit of retrieval. From each shot one or more representative frames (key-frames) are extracted and their low-level features are indexed. For images the same process is followed excluding the shot segmentation and key-frame extraction steps.

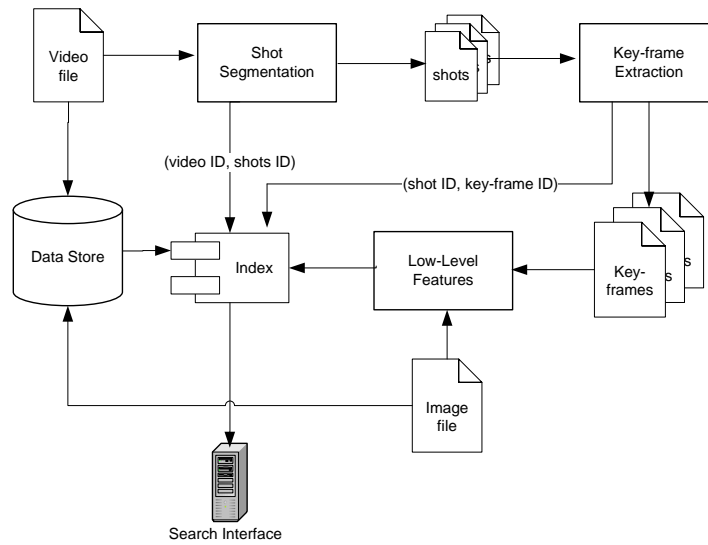


Figure 5.4: Overview of the indexing process

The indexing interface is illustrated in Figure 5.4. Creating a new multimedia collection using this interface is a four step process. Firstly the name of the collection must be entered in the collections panel. The name must be unique and the system provides feedback in case of a conflict. Once the “*Create Collection*” button is pressed the new collection name will appear in the “*Collections List*”. The system automatically creates a collection space in the SFTP server where the multimedia collection can be uploaded. The directory is the same as the name of the collection under the *FORINDEXING* directory, e.g. *FORINDEXING/new_test_collection/*. The SFTP server is located at **sibu.dcs.gla.ac.uk** and can be accessed using the Salero user name and password. The collection space contains two sub-directories, “*photos*” and “*videos*” for uploading image or video collections respectively. Once the collection is uploaded on the SFT server then the administrator can select from the “*Indexing Panel*” the low-level visual features to be used for the indexing of the collection, namely “*Color Layout*”, “*Color Histogram*” and “*Edge Histogram*”. Finally the “*Index*” button initiates the indexing process. The indexing process runs in the background and so the Web page can be closed and return back later. Once the indexing process is finished the collection will be available in the search interface.

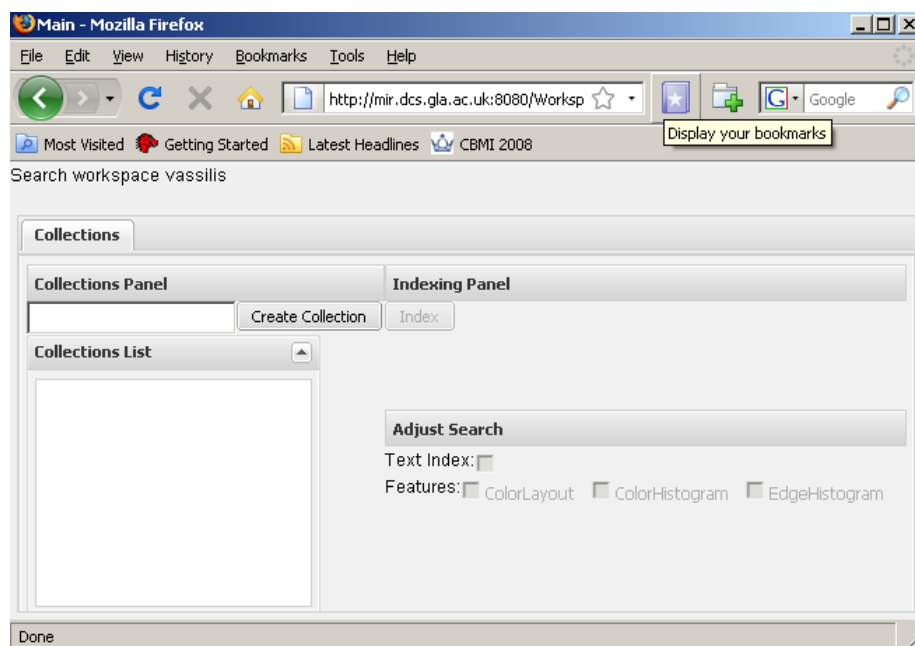


Figure 5.5: A screenshot of the Indexing Interface

5.3 Conclusions

The search interface presented here allow the flexible searching of different, mixed, collections of images, videos and text. Results found in one collection can be used as examples for searches in other collections, enabling the user to take advantage of both text and content-based searching when available. In the following chapter, we present an evaluation which used a previous version of the interface shown here.

6 User Evaluation of the AspectBrowser

A user evaluation of a subset of the AspectBrowser interface described in Chapter 5 (here referred to as the “aspectual interface”) was carried out, in order to determine the effectiveness of this type of interface for broad, complex, search tasks. In this Chapter we present this evaluation, which is based on the work also reported in the paper [Villa 2009].

6.1 Research Questions

In this evaluation, we have three principle research questions:

RQ1: Does the aspectual interface allow the user to better explore and discover relevant material when compared to a purely sequential interface?

RQ2: Does the aspectual interface aid the user in better understanding the search task?

RQ3: What features of our aspectual interface are used by the users carrying out the search tasks?

Research questions 1 and 2 are the central questions addressed in this section: can our aspectual interface, which allows the user to structure their searching, lead to the user discovering more relevant material when compared to a standard interface, and secondly, via this searching process, do users better understand the task they have carried out? Research question 3 is a more open ended, where we are interested in investigating how users take advantage of the different features of the aspectual interface. In order to test our ideas concerning the potential utility of aspectual search, and our aspectual interface, a baseline interface was created for comparison purposes, which restricted the user to carrying out one search at a time, as is common in current information retrieval interfaces.

6.2 Complex Search Tasks

There are many ways in which a user search or work task can be considered “complex”, and there has been work in various different fields which has looked into this question. From the information science literature, [Bystrom, 1995] consider the relationship between task complexity and information search and use. Five ‘task categories’ are defined: genuine decision task; known genuine decision task; normal decision task; normal information processing task; and automatic information processing task. These five categories are based on the uncertainty inherent in the tasks: in automatic information processing tasks, the task solutions and types of information required are all known in advance. At the other end of the scale, in genuine decision making tasks nothing is known about the types of information or solutions required.

The work of [Bell, 2004] also uses the uncertainty inherent in a task as a measure of complexity, and presents a study which systematically alters search tasks in order to control task complexity. More complex topics were created by removing specifics from the topic statement, providing the user with less knowledge of the task to be undertaken. A similar method is used in the study reported in [Bystrom, 1995]. The work of [Vakkari, 1999] is an attempt to synthesize a number of studies into a single model, and takes a similar approach, where task complexity is related to the lack of structure in a task, and the lack of knowledge a user has of the task.

An alternative perspective on task complexity is provided by [Campbell, 1988], who reviews approaches to task complexity in various research fields. Based on this, he outlines four basic task characteristics which define what he calls “objective complexity”, i.e. complexity which is based solely on the task, and not on the degree of the user’s knowledge of the task. These are: (a) the presence of multiple paths to a solution; (b) the presence of multiple desired outcomes; (c) the presence of conflicting interdependences among the paths to the solution(s); and (d) the presence of uncertain links among the paths and outcomes. Task complexity is therefore defined as more than uncertainty, but also involves the presence of other factors, notably the possibility that complexity may also be due to the task having multiple solutions, or multiple paths to a single solution. This model of task complexity has provided an impetus to the aspectual interface described here, and has motivated the tasks created for the user study, described in Section 6.5.

6.3 Experimental Design

In order to investigate the research questions outlined in Section 6.1, a between subjects user study was conducted. Based on the model by [Campbell, 1988], we constructed three tasks, each of which were designed to represent three different possible types of complex task. Two separate groups of users then carried out the same three tasks on two different interfaces – the aspectual interface (a simplification of the AspectBrowser interface described in Chapter 5) and a baseline interface. The same underlying Web search engine, Yahoo BOSS, was used for both interfaces.

Based on research questions 1 and 2, we created two hypotheses: first, that the aspectual interface would allow a user to investigate the tasks to a greater extent, measured by the documents marked as relevant, the results viewed, the searches carried out, and the query vocabulary size. Secondly, we hypothesized that the user perception of task complexity and difficulty would fall significantly for the aspectual interface, but not the baseline, when comparing a user's perceptions of the task before and after carrying out the search.

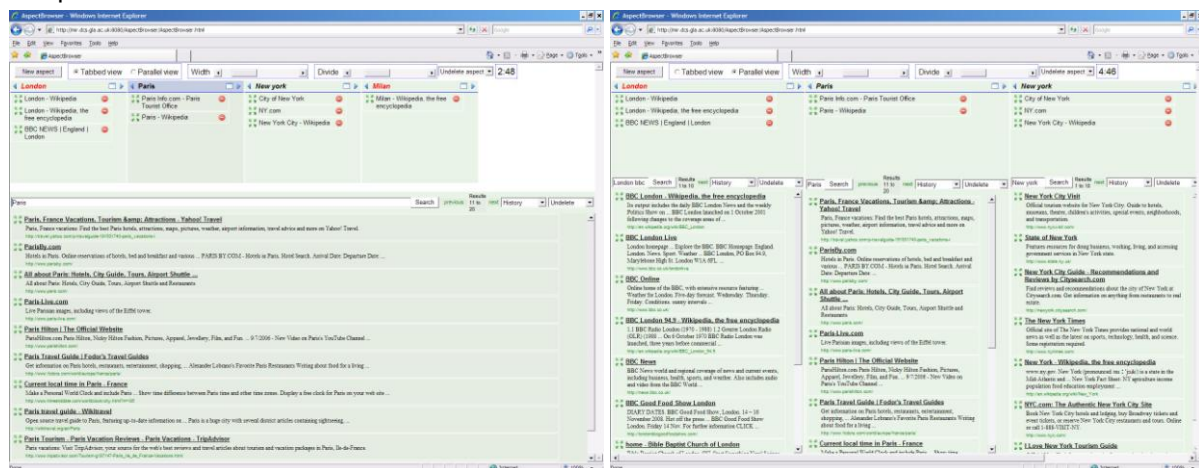
6.4 Interfaces

6.4.1 The aspectual interface

The aspectual interface used in this evaluation was a simplification of the AspectBrowser interface described in Chapter 5, a screenshot is shown below (Figure 6.1). This interface worked in the same manner as the AspectBrowser interface, with the following differences:

- A “parallel” view, as used in the previous FacetBrowser interface (described in deliverable 5.5.2) was provided, where each aspect occupied a strip of the screen (left hand screenshot in Figure 6.1)
- Searching was restricted to use only Yahoo BOSS, in order to simplify the experiment
- The interface design did not use the Ext-GWT toolkit, being an older version of the interface described in Chapter 5.

Despite some design differences, all the aspectual functionality, however, was as is described in Chapter 5.



(a) Parallel view

(b) Tabbed view

Figure 6.1: The aspectual search interface showing the same information in two different views

6.4.2 The baseline interface

The baseline interface was a simplified version of the aspectual interface, where only a single panel was used to compile relevant results obtained during a searching session, independently of being related to different aspects or topics. The purpose of this baseline version of the system was to allow us to conduct the experiment in such a way that the aspectual interface could be compared against a classic Web search interface.

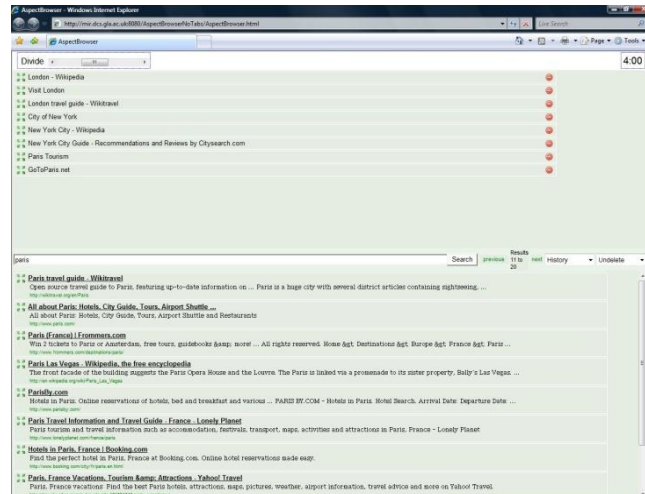


Figure 6.2: The baseline search interface used in the study

In this interface, no tabs are permitted so a user cannot organize the documents into different aspect-oriented panels. Figure 6.2 shows a screenshot of the interface. It consists of two main panels separated by a search bar. The bottom panel contains the list of documents retrieved by the query, and the top panel displaying the list of documents marked as relevant by the user. Functionally, the interface operates in exactly the same way as the aspectual interface, except there is only ever a single “aspect”: e.g. the user can make use of the drag and drop functionality to mark documents as relevant, alter the order of the documents, etc.

6.5 Tasks

Based on the categories (a) to (c) of [Campbell, 1988], summarized in Section 6.2, three different types of search task were defined:

- A. Single solution, multiple potential paths to the solution
- B. Multiple solution, where the aspects are largely independent, and are largely specified by the task
- C. Multiple solution, where the aspects are implicit in the task and may be interdependent

The first task, A, was written as a decision making task, where the user was asked to find and decide on a new digital camera. The second task, B, was setup as a report writing task, where the user had to find information about the political leaders of four specified countries. The ‘solution’ was therefore plural, and the user was given the option to search for more than the specified minimum. The third task, C, was a summarization task, where the user was given some general background information for the task, but where specific aspects were not directly provided. The aim here was to force users to find and categorize the search task as they see fit. However, unlike Task A, the solution was again plural – users were asked to find multiple aspects. Each task was written as a simulated work task [Borlund, 2003], which included a simulated situation, describing a context for the user, as shown in Table 6.1.

Our choice to carry out the evaluation on the Web was a practical one – we wanted a document collection which was large enough, and rich enough to allow a user to explore material without hindrance.

Task	Simulated situation
A	You are looking for a new digital camera to replace your old film camera before going on a safari holiday in Africa. You would like to find out which models of camera are best for you and for your holiday trip. While liking photography, you would like something small and easy to use, which can also be used by your partner who dislikes technology.
B	Your lecturer has stated that, as part of the reports requirements, you MUST provide a biography of current presidents or prime ministers of the following countries: USA, Germany, Japan and the UK. In addition to the leaders from these four required countries, you may also include any other world leader you wish. The report must be written in English.

C	You work as a researcher for a well known UK newspaper, and as part of your remit, you work on a monthly news magazine, which summarizes and reviews the main sporting events of the previous month. It has now become necessary to put together the latest magazine for August 2008, covering the Olympics in Beijing. The magazine is 25 pages long, and typically covers between 4 to 8 main stories of a page or more, and a further unknown number of much shorter stories and news pieces, often only a paragraph long. You are responsible for researching the main stories from the Olympics, putting them in order of importance, and also finding as many other shorter stories as you can which are appropriate for the magazine.
---	--

Table 6.1: The three tasks used in the study

6.6 Procedure

In total, thirty-six users were recruited through email to take part in the study, split into two groups of eighteen users: one group of eighteen performed each of the three tasks with the aspectual interface, the other group of eighteen performing the three tasks with the baseline interface. The median age was 26 (with range 22 to 39) with the majority of users being either native or near-native English speakers. Most subjects were students or post-graduates at Glasgow University, all having university degrees or higher.

After arriving at the office where the study took place, users were welcomed before being presented with an information and consent form. After these preliminaries, an entry questionnaire was administered, before the experimenter then demonstrated the search interface (either the aspectual or baseline interface). This demonstration took approximately 10 minutes for each user, and was followed by a training task, where the user was allowed up to 15 minutes to interact and use the system.

After training, each of the three tasks was administered in an order which ensured counterbalancing. Before each task, the task description was presented to the user who was then able to read it, before a pre-task questionnaire was filled in. After this, the test interface would be started and the subject commenced searching. Both interfaces contained an automatic timer, always present to the user at the top right hand corner of the screen, showing the length of time which had elapsed. After 20 minutes, an "end task" dialog box would appear indicating the end of the task, although users were informed that they could end the task before the 20 minutes if they were satisfied with their search results. After each task, a post-task questionnaire was administered. This was repeated three times, one per task.

At the end of the experiment, a final exit questionnaire was completed by all users. The procedure took up to 2 hours, and all users were paid 12 pounds on completion.

6.7 Results

The results of the study are presented in this section, split to reflect the research questions outlined in Section 6.1. Unless otherwise stated, all results are described using medians and the interquartile range, due to the often asymmetric nature of the data; likewise the non-parametric Wilcoxon rank-sum test is used unless stated otherwise, with significance level $P \leq 0.05$.

6.7.1 Search performance

The first research question looks at the search performance of users using either the baseline or aspectual search interfaces. In particular, we are interested in measuring the degree of exploration of the user, assuming that a user who is able to explore more of an information space is more likely to satisfy a task. We do not use precision and recall: the broad nature of the tasks makes defining a single set of relevance judgments difficult, and given the use of Web search, likely impossible. Instead, performance is measured using the number of Web pages marked as relevant, the number of search results viewed, the number of searches carried out, and the query vocabulary size. Table 6.2 shows the results of these four measures split by task and system.

	Task A	Task B	Task C
Marked relevant			
baseline	8 (6-9)	17 (11.3-20.5)	11 (8.5-25.3)
aspectual	14 (9.3-20.8)	22.5 (16.5-37.8)	24 (18.5-32.5)
Results viewed			
baseline	17 (11.3-21.0)	24.5 (22.3-29.5)	17.5 (14.3-24.5)
aspectual	22 (15.3-28.5)	23.5 (12.5-44.3)	29.0 (17.0-43.3)
Number of searches			
baseline	7 (5-9.8)	17 (16-20.8)	9.5 (7.3-15)
aspectual	10 (7-13.8)	18.5 (16.3-25.8)	13 (12-16.8)
Query vocabulary size			
baseline	11.5 (9-15)	21.5 (20.3-24.8)	15 (8.3-21.0)
aspectual	14.0(11.0-16.0)	25.5 (21.3-30.5)	21.0 (16.0-24.8)

Table 6.2: Number of Web pages marked relevant, search results viewed and searches carried out. The median (and interquartile range) values are reported, bold indicating relevance at $P \leq 0.05$

Considering each task individually, it can be seen that for Task C (multiple solutions, implicit aspects) there is a clear trend in all measures for an increase in performance for the aspectual interface when compared to the baseline interface: a statistically significant difference was found in all four cases at $P \leq 0.05$. Conversely, for Task B, no significant differences were found between the baseline and aspectual interfaces for all four measures. Lastly, for Task A, the picture is mixed: significant differences between the baseline and aspectual interfaces were found for two measures (number marked relevant, and number of searches), but not for the others.

One issue which is not considered in Table 6.2 is that of search time. All users in both conditions were informed before starting that they had up to 20 minutes to search for each task, but could finish early if they felt their task was finished. The instructions presented to both sets of users were the same. Table 6.3 shows the median task length for the two interfaces, and three tasks. As can be seen, there is a trend for users to spend less time searching with the baseline interface than with the aspectual interface. Out of the 54 sessions with the aspectual interface, only 7 finished before the maximum 20 minutes, with only two sessions finishing in less than 19 minutes (one session finished in 16 minutes, another 17 minutes).

All four of the measures reported in Table 6.2 consider each search session as a whole, irrespective of how long the user searched. It is also possible, as suggested by [Kaki, 2004], to take time into account, i.e. to measure the number of Web pages marked as relevant, viewed, or the number of searches carried out per minute. When this adjustment is made to the results in Table 6.2, no significant differences are found between the two systems.

	Task A	Task B	Task C
Baseline	13.8 (12.4-19.4)	17.9 (16.2-20)	17.3 (11.7-19.0)
Aspectual	20 (20-20)	20 (20-20)	20 (20-20)

Table 6.3: Median (interquartile range) for task time in minutes

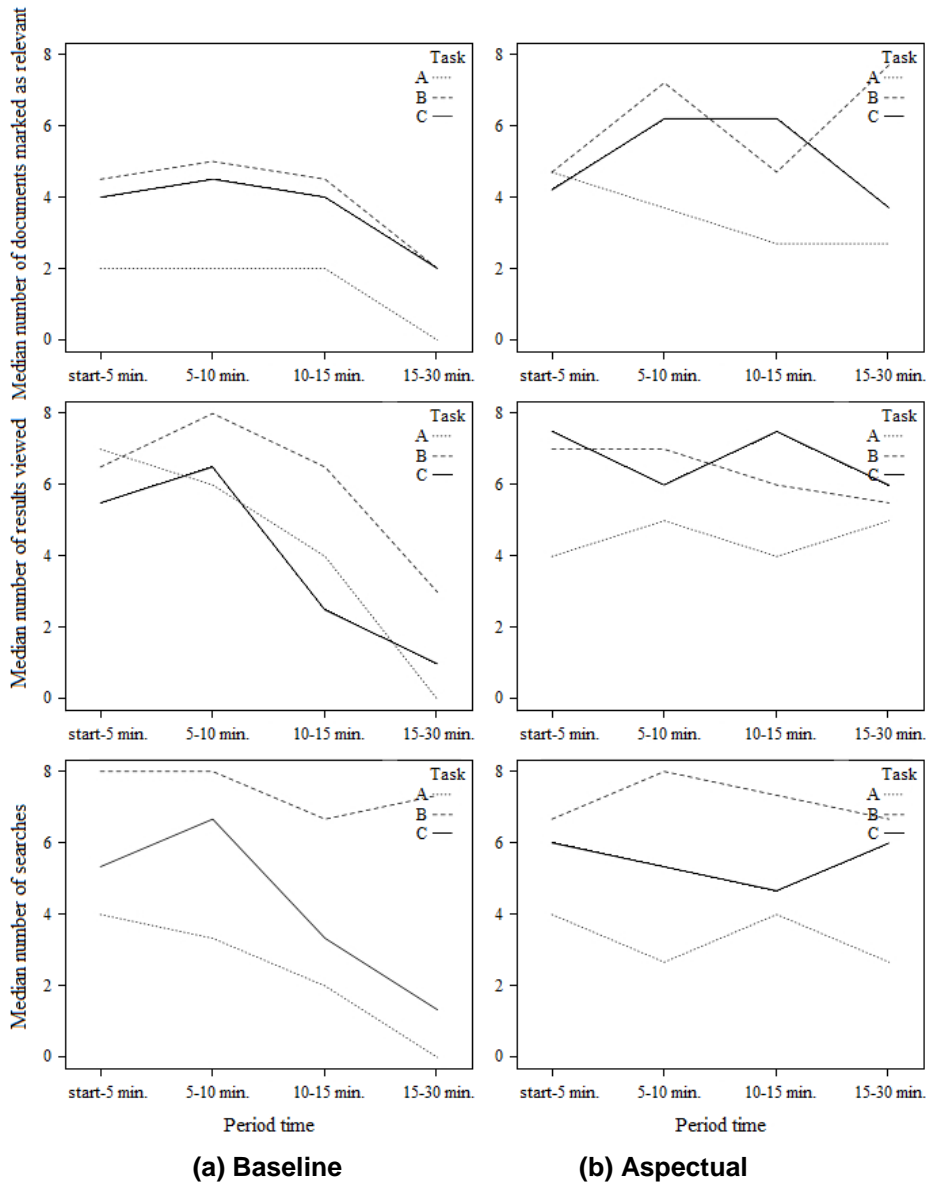


Figure 6.3: Search performance over time for number marked relevant, number documents viewed, and number of searches. Baseline on left, aspectual on right

Figures 6.3 (a) and (b) present this same information, for number of documents marked as relevant, results viewed and searches carried out, but shown grouped into four 5 minute bins over time. On the left on Figure 6.3 (a) are the graphs for the baseline interface, on the right, the aspectual interface. It should be noted that these values are not cumulative – each bin represents the activity for only that 5 minute period of time.

For the baseline results, a clear fall in user activity can be seen in the final 5 minutes of time, reflecting the number of users who stopped before the full 20 minute period was up. An exception to this is the number of searches executed on Task A. The activity over time for the aspectual interface is somewhat

more constant over the full 20 minutes, with activity roughly constant or at times increasing during the final 5 minutes. The graph for the change in query vocabulary size (not shown due to space) is similar to those shown in Figure 6.3, with the number of unique terms falling off in the final 5 minute period for the baseline interface, while remaining more stable for the aspectual interface.

6.7.2 User perceptions of search task difficulty

The second research question from Section 6.1 asks whether the aspectual interface aids the user in better understanding the search task. To attempt to measure this, as part of the pre- and post-task questionnaires, we asked each searcher to judge the complexity and difficulty of the task, before and after carrying out the search. If using the aspectual interface leads to a greater understanding of the task, we hypothesize that there will be a significant reduction in perceived task complexity and difficulty for the aspectual interface, and similarly a non-significant reduction for the baseline interface.

		Task A	Task B	Task C
The task will be / was complex				
Baseline	Pre	2.5 (2-3.75)	3.0 (2.25-4)	3.5 (2-4)
	Post	2 (1.25-3.75)	2.5 (2-3)	3.0 (2-4)
Aspectual	Pre	2 (1-4)	3.0 (2.25-4)	4.0 (3-5)
	Post	2 (1-4)	2.0 (2-3)	2.5 (2-3.75)

Table 6.4: Responses to the pre and post task question “The task [will be/was] complex], 1 = disagree, 5 = agree; median (interquartile range), bold indicates significance $P \leq 0.05$

		Task A	Task B	Task C
The task will be / was difficult				
Baseline	Pre	3 (2-3)	3 (2-4)	3 (2-4)
	Post	2 (2-3)	3 (2-3.75)	3 (2-4)
Aspectual	Pre	2 (1-4)	3 (2.25-4)	4 (3-4)
	Post	2 (1-4)	3 (2.25-4)	3 (2-4)

Table 6.5: Responses to the pre and post task question “The task [will be/was] difficult], 1 = disagree, 5 = agree; median (interquartile range)

In Tables 6.4 and 6.5, the median responses are presented for the pairs of questions “The task will be complex”/“The task was complex” and “The task will be difficult”/“The task was difficult”. Both the responses in the pre-task and post-task questionnaires are given; Wilcoxon rank-sum tests were used to compare the pre and post task judgments, with one significant result, that of users judgment of complexity with Task C ($W = 242$, $P \leq 0.05$).

6.7.3 Interface features and user perceptions

For research question 3, we present some overview statistics concerning the usage of the extra facilities of the aspectual interface (Table 6.6). It can be seen that roughly similar numbers of aspects were created by all users for all tasks (medians 5 or 6). Very few aspects were deleted – in total 39 aspects were deleted by 9 users over the 54 sessions.

Few examples were also copied between aspects: over the 54 search sessions with the aspectual interface, 69 aspect to aspect copying events were recorded, across 26 sessions. Out of the 18 users, 4 did not move bookmarked documents between aspects at all. Others, such as users 9 and 12, used this facility more, with 15 and 13 recorded aspect to aspect moves for these two users.

Similarly, documents marked relevant were also rarely deleted. Over all users, 78 document deletions were recorded, and again, 4 users did not delete any documents once marked, while others deleted numerous documents (user 14 deleted ten documents, and user 3, eight). By comparison, only 7 undelete document events were recorded over all sessions.

	Task A	Task B	Task C
Number of aspects	6 (5-7.75)	5 (5-7.75)	6 (5-7.75)
Copied between aspects	0.5 (0-2)	0 (0-2)	0.5 (0-2)
Delete relevant	0 (0-1)	1.0 (0-1)	0 (0-1)

Table 6.6: Median number of events with aspectual interface; median (interquartile range)

The aspectual interface supports two different visualization views, as shown in Figure 6.1. Table 6.7 shows the mean length of time users spent in each of these views, showing a distinct preference for the tabbed view over the parallel view. Out of the 54 sessions, 39 were carried out for a majority of the time in the tabbed view, the other 15 for a majority of the time in the parallel view.

	Task A	Task B	Task B
Tabbed	13.5 (9.0)	13.0 (9.2)	16.0 (7.5)
Parallel	6.5 (9.0)	7.0 (9.2)	3.7 (7.1)

Table 6.7: Length of time spent in the tabbed and parallel views, mean (standard deviation), in minutes

Finally, we present the results of the exit questionnaire which asked a number of usability questions based on the USE questionnaire [Lund, 2001], shown in Table 6.8. A Wilcoxon rank-sum test found that the question “The system would help me be more effective” was significantly different at the 5% level ($W = 91, P \leq 0.05$), users agreeing with this statement significantly more for the aspectual interface than the baseline. A significant difference was also found for the question “The system can be used effectively without instruction” ($W=270.5, P \leq 0.05$), this time the baseline interface being judged as being easier to use without instruction. All other comparisons were found to be not significant.

6.8 Discussion

Concerning the results for research question 1, in Section 6.1, there would appear to be strong evidence that the aspectual interface does aid the user, based on the four performance measures, for Task C (multiple solutions where the aspects are implicit). Conversely, for Task B (multiple solutions where aspects are explicitly provided), we found no significant difference between the baseline and aspectual interfaces. This is perhaps not so surprising given that in Task B the user does not have the added burden of determining aspects themselves, instead, the organization is implicit within the task description. This is consistent with the thesis that the aspectual interface aids the user in performing a classification of the task (i.e. determining the appropriate aspects): if the aspectual interface does not aid the user in specifying aspects, we may expect both tasks B and C to produce similar results, which is not the case here.

Task A, the decision making task, is less clear cut than the other two, the results shown in Table 6.2 are mixed. Given that users do mark more relevant documents, and carry out more searches, we can say there is some tentative evidence that the aspectual interface does aid the user with this type of task.

Question	Baseline	Aspectual
The system is simple to use	4.5 (4-5)	5.0 (4-5)
The system is wonderful	3 (3-4)	4 (4-5)
The system is useful	4 (3.25-5)	5 (4 – 5)
The system is flexible	3.5 (3-4)	4 (4-5)
I learned to use the system quickly	5 (5-5)	5 (5-5)
The system would make things I want to accomplish easier to get done	4 (3 – 4.75)	4 (4-5)
The system would help me be more effective	3.5 (2.25-4)	5 (4-5)

Using the system is effortless	3 (3-5)	4 (3.25 – 5)
The system can be used effectively without instruction	5 (4-5)	4 (2.25 – 4)
The system was easy to learn to use	5 (5- 5)	5 (4.25 – 5)
The system works the way I want it to work	4 (3 – 4)	4 (4 – 4.75)
The search methods I used in this study were similar to those I use when I normally search the Web	4 (3 – 5)	4 (3.25 – 5)
Overall the system is easy to use	5 (4 – 5)	5 (4 – 5)
If this system were available for use I would use it frequently	4 (3-4.75)	4 (4-5)
I am satisfied with the interface	3.5 (3-5)	4 (3.25-5)

Table 6.8: Results of the exit questionnaire, medians (interquartile range) where 5 = agree, 1 = disagree

One issue with the results presented in Table 6.2 is that of task time. As shown in Table 6.3, users tended to spend less time searching with the baseline when compared to the aspectual interface. This is also supported by the graphs of performance over time shown in Figure 6.3, where it can be seen that user activity with the aspectual interface is broadly similar to the start of the task. In contrast, there is a trend for user activity with the baseline interface to decrease over time for the three measures (although Task B shows evidence that users are still searching until the end of the task, as shown by the bottom graph in Figure 6.3(a)).

It is possible to interpret this in at least two ways: one could suggest that baseline users succeed at completing the task faster; alternatively, it can be suggested that the extra organizational facilities of the aspectual interface encourages users to search more, the shorter task times recorded by baseline users being indicative of boredom or an otherwise lack of impetuous to search more. We tend to err on the latter interpretation for a number of reasons. First, there is no performance difference between the interfaces on Task B suggesting that the baseline interface *can* perform as well as the aspectual interface, when the task explicitly specifies the aspects to search for. While not significant, it is also to be noted that the baseline task time in Table 6.3 for Task B is longer than for the other tasks, and the final five minute period of search activity graphed in Figure 6.3(a) for Task B shows more activity than the other tasks, especially for searches carried out. This extra activity at the end of the task is consistent with the extra aspect information allowing the users to search for longer with the baseline interface, therefore enabling a search performance more in line with the aspectual interface.

Turning to the second research question, and the results shown in Tables 6.4 and 6.5, no significant differences were found in user perceptions of task complexity or difficulty before and after the task, with the exception of Task C and complexity with the aspectual interface. This is consistent with the aspectual interface aiding the users understanding of the task, where in this case aspects must be identified by the user. I.e. after carrying out task C with the aspectual interface, users considered the task *less* complex, but *equally* difficult. This result backs up the previous results which suggest that users gain the most from the aspectual interface when performing Task C.

Lastly, individual features of the aspectual interface were analyzed, and the results of the exit questionnaire presented. These show mixed usage of the two different views, with a tendency for users to prefer the tabbed visualization (Figure 6.1(a)). Organization facilities were lightly used by most users (Table 6.6), most documents which are marked relevant not being moved between aspects. User feedback from the exit questionnaire suggests that the aspectual interface does require instruction when compared to the baseline interface, although user perceptions of ease of use and ease of learning do not vary significantly between systems. User perceptions of effectiveness, however, do vary significantly, with users suggesting that the aspectual interface would allow them to be more effective.

6.9 Conclusions and Future Work

The results presented in this section suggest that for tasks similar to Task C, where the user must identify aspects of a broad search task, an aspectual style interface similar to the one presented here has a clear advantage: users found more relevant Web pages, viewed more results, and searched more. There is also evidence that by enabling the user to classify and structure their searching, the perceived complexity of the task decreased after the search session. This is in contrast to Task B, where task aspects were supplied predefined to the users: in this case there is no advantage to using the aspectual interface. For decision tasks such as Task A, the advantages of an aspectual interface are less clear cut: there is evidence that task performance improves, but it is not conclusive.

7 Integration with the Semantic Search and Annotation Systems

In this Chapter, we present an overview of the integration which has taken place between the content based and semantic retrieval systems. This aim is to enable both types of retrieval to be integrated together in such a way that either part of the functionality can be used with the other.

7.1 Interface Integration

The IMAS system architecture is shown in Figure 8.1. The IMAS integrates two systems, whose functionalities are offered as a set of Web services, i.e. the Semantic Services and the Content-based Services. The main aim of the IMAS is to allow easy annotation of media assets for later retrieval and reuse by users in media production. In order to support this, it has been built based on the following design principles:

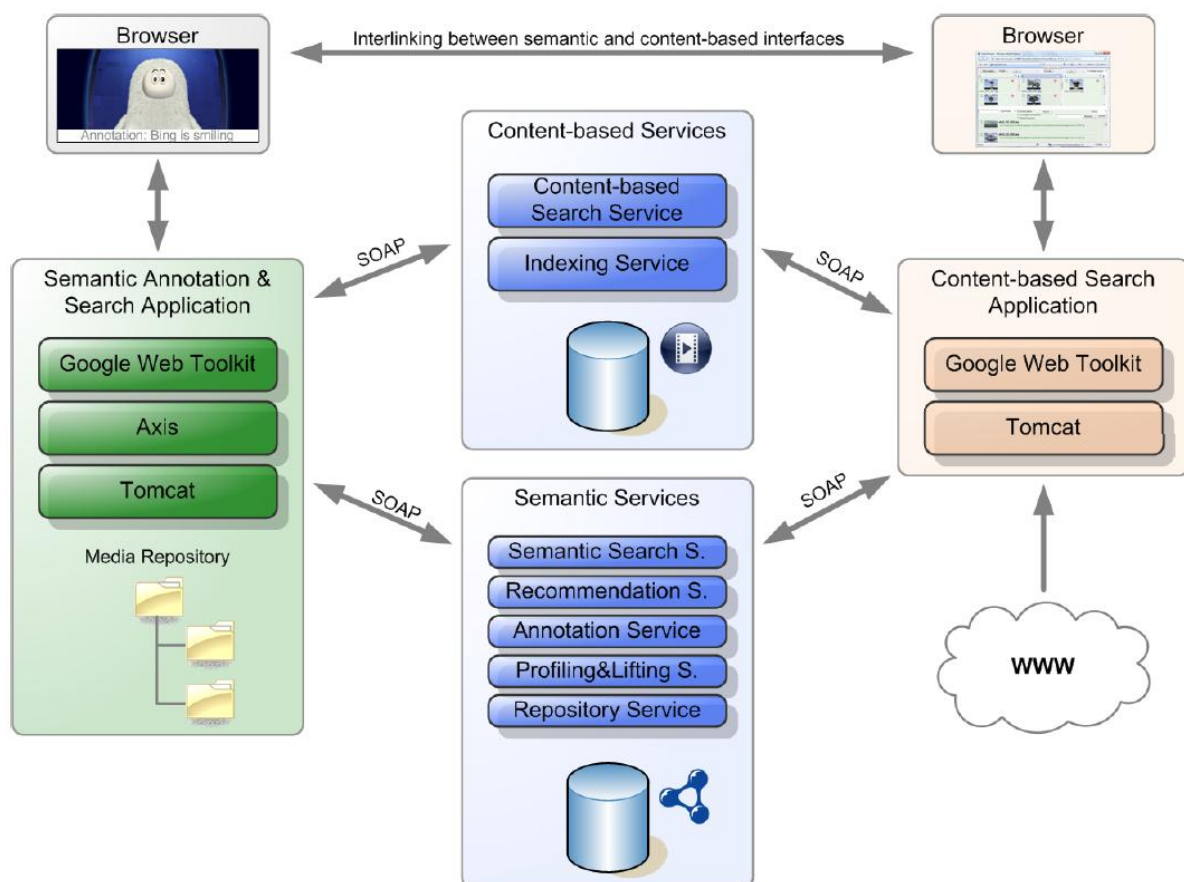


Figure 8.1: An architectural view of the Integrated System

1. **Designed for content creators.** The target users of the system are non-technically experienced content creators in the domain of media production.
2. **Easy to use.** The interface provides Web 2.0 based interaction mechanisms to make the annotation process as easy as possible.
3. **Global annotations.** To facilitate the annotation process, we only allow global annotation of media files instead of annotating parts of it.
4. **Statement-based annotation process.** We allow creating statements, which use ontological elements, to describe the content of media items.
5. **Ontology extension during use.** We allow users to easily extend the ontology during use based on principles described in Fehler! Verweisquelle konnte nicht gefunden werden..

6. **Portability of the system.** In order to port the systems to other domains, only the underlying annotation ontology has to be adapted.
7. **Integration of semantic and content-based search.** The system provides an integrative view onto results from different search engines and by that provides a fallback solution which is able to retrieve objects without annotations too.

7.2 Annotation and Search Application

The IMAS end user application is an integrated Web-based application which can be used to annotate and search for media objects. As illustrated in Figure 8.1, it consumes functionality of (i) the Semantic services which are used to add, update and delete semantic annotations as well as to search the annotation repository and (ii) the Content-based Services which are used to retrieve media items based on its intrinsic features such as colour histograms or shapes.

Annotation

The application allows annotating any arbitrary files which are stored in pre-configurable media repositories. In order to ease the annotation process for our target user group media items are annotated globally instead of region or segment-based. Additionally, media items are annotated by creating statements which contain semantic elements which are defined in ontology. The annotation statements can be seen as sentences about the content of the media item and are in the form of

< Re source is RelatedTo {Concept₁....Concept_n} > .

Using statements with semantic elements is a compromise in complexity between loose and fully semantically described annotations. Figure. 8.2 illustrates statements with an example. To create such statements, three different input options are available as shown in Figure 8.3: (1) combining concepts via drag-and-drop, (2) selecting concepts consecutively and (3) using the text box as a command line interface with auto-completion. Input option three is optimally suited for frequent users and input options one and two are ideal for users who rarely create annotations.



- Bing is related to: Bong, Alien, reading, book
- Bong is related to: smiling
- Alien is related to: surprised

Figure 8.2: An example of annotation statements

Semantic Search

IMAS is not only offers a graphical user interface to engineer ontologies but also a set of services which provide ontology management functionality to other applications. The services offered by the semantic workbench include:

- **The Repository Service** which offers an API for the persistent storage of WSMML (<http://www.wsmo.org/wsml/wsml-syntax>) ontologies and their elements. It supports management of elements and the execution of SPARQL queries. The services are realized on top of the Ontology Representation and Data Integration (ORDI) – framework (<http://www.ontotext.com/ordi>).
- **The Annotation Service** is concerned with the management of semantic annotations and provides an API to manage and validate annotations against the ontologies stored in the repository.

- **The Semantic Search Service** offers an API to search for ontology elements and additionally offers keyword-based search for annotations which is expanded into full-text queries on a generated index and SPARQL queries.
- **The Recommendation Service** offers an API for retrieval of ontology elements which are prominently used for annotation and gives recommendations of related results during search.
- **The Profiling and Lifting Service** can be used to extract structural semantic information from existing MPEG-7 (<http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>) documents and for their semantic enrichment.

Content Based Search

The Content-based Services offer functionality for the indexing and retrieval of image, video and textual information. The aim of the service is to complement the semantic search services. As such their emphasis is on automatic indexing techniques, which can be used to retrieve images, text or video without manual annotation. This provides a fall-back system in the cases where material is not indexed by the semantic services, providing an alternative method of retrieval and bootstrapping. Architecturally, the content-based services are split into a backend indexing and retrieval system which provides a SOAP interface to other software components and a frontend search interface for end users. Textual information is indexed using standard Information Retrieval techniques (the Terrier system is used **Fehler! Verweisquelle konnte nicht gefunden werden.**); image and video data is indexed by extracting low-level visual features based on the MPEG-7 standard, as currently implemented in the ACE toolbox **Fehler! Verweisquelle konnte nicht gefunden werden.** Searches can be carried out on locally uploaded and indexed data or the system can also use third party Web services to search remote collections, such as Yahoo BOSS (<http://developer.yahoo.com/search/boos/>) or YouTube.

Both local collections and remote services can be searched via the same search interface. The interface allows search based on an optional textual query and zero or more image examples which may come from already indexed collections or be serialized ad-hoc images. The FacetBrowser **Fehler! Verweisquelle konnte nicht gefunden werden.**, a flexible environment for search and search result organisation, is the end user application which makes use of the content-based services.

7.3 Search System

To search for media objects, the following input options are available to create a query: (i) free text, (ii) semantic concepts, (iii) statements, and (iv) images. Free text search is both executed in the Semantic Services and the Content-based Services. The concept-based search is expanded in the Semantic Services. Via the exemplary images a query is submitted to the Content-based Services. The results of both systems are merged based on the respective rank values being returned.

7.3.1 Semantic Search System

The Semantic Services support two different types of queries: The first options are concept-based queries which are translated into SPARQL queries and which are evaluated via the Repository Service of the Semantic Services in the repository.

The second options are keyword-based queries which are executed in a Lucene (<http://lucene.apache.org>) index which is generated based on the ontologies and instance data in the repository. The Lucene index preserves the annotation triples and also stores subsumption information to enable efficient retrieval. Keyword and concept queries can be mixed to increase the precision of keyword-based querying in the system.

Furthermore selected WordNet relations of the concepts can be expanded to increase the result set while preserving precision. Once the queries are answered, the results are ranked based on the semantics of the annotations attached to the returned results and their degree of match of the query. The ranking is based on the popular tf/idf measure which is customized for semantic annotations to take into account triples instead of just terms and the subsumption hierarchies.

7.3.2 Content Based Search System

The content based search system is capable of handling multiple query examples. Multiple query examples are generally used in two different ways: In the first case, the examples are mapped onto a single query [Kim, 2000b; Khotanzod, 1990; Teh, 1986], most often by averaging the query feature vectors. This newly framed query example is used in the matching process. This is referred to as early fusion. On the other hand, in the second case, examples are used by machine learning techniques to learn and weight the features (also called experts in the literature), such as colour, texture and shape, in

order to obtain best results for these examples. The best results for various features for a topic are then fused into a single result. This is commonly known as 'late fusion' mechanism.

An effective multimedia retrieval cannot be achieved with a single visual feature and requires possibly many features. Due to this fact, a mechanism to automatically use/select best features for a topic, or more precisely to the set of query examples, need attention in order to achieve good results. As mentioned earlier, addressing the problem of content based semantic multimedia retrieval also necessitates the utmost coverage of diversity in the visual appearance of the results retrieved as well. It is well known that the objective of an automatic multimedia retrieval system is that given an example query image/video, using a similarity measure, the system must retrieve the most semantically similar images/videos to the supplied query.

A keyword based search is employed to obtain the initial list of n results from semantic search system. The top $m \ll n$ results are then used by the content based search to retrieve more visually similar images. Since it is also important to select the best feature or re-order the features according to their preference for the query examples favoring to obtain more diverse results, the knowledge embedded in queries are exploited by computing the correlation and t-distribution in the queries. The feature which has less correlation and high t-distribution is chosen as the most useful feature.

For the prerogative visual feature selection using the query examples, we design a statistical model as follows. Let $Q = \{q_1, q_2, q_3, \dots, q_M\}$ be the set of query examples provided for a topic t . Let $F = \{CH, CL, EH, HT\}$ be the set of visual features being used for the purpose of retrieval. Here, CL, EH and HT are the standard MPEG 7 standard Colour Layout, Edge Histogram and Homogenous Texture features and CH is a simple Color Histogram feature. Let V^f represent the feature matrix of Q in feature space $f \in F$. Given V^f for all $f \in F$, we compute the following matrices and values. A distance matrix $D_{M \times M}^f$, is computed using the best distance measure for $f \in F$.

A correlation matrix $C_{M \times M}^f$ is computed using equation (1),

$$C_{ij}^f = \frac{d_f (\sum AB) - (\sum A)(\sum B)}{\sqrt{d_f (\sum A^2) - (\sum A)^2} \sqrt{d_f (\sum B^2) - (\sum B)^2}} \quad (1)$$

Here, A and B are the feature vectors of queries q_i and q_j and d_f is the dimension of feature f . From the above matrices, C^f , D^f and V^f , the statistical values, S_C^f , S_D^f , and S_V^f that is, the standard deviation of the matrices respectively are calculated. In addition to these, P_C^f , P_D^f , P_V^f , the standard deviation of the t-statistic of the correlation matrices C^f , D^f , V^f respectively is also computed. Since the basic idea behind using many query examples for retrieval is a mutual concession to retrieve more diverse results and support semantic retrieval, we argue that features having larger variance should be preferred for that topic. To be not biased to a single feature, we order the features based on the six values, P_C^f , P_D^f , P_V^f , S_C^f , S_D^f , and S_V^f . The higher the values of S_D , S_C , P_C , P_D , P_V and lower the value of S_V more useful is that feature f in fetching more distinct and diverse results.

Irrespective of the number of query examples used or the number of visual features used, a user expects one final result list covering the idiosyncrasies of the examples. The core problem of generating the one list lies in devising a mechanism of combining the results originating from each query example and each relevant feature to improve retrieval results within a search engine.

Thus, we now have $|F|$ result lists for a topic t due to M queries. In order to rank the suitability of the features for a topic, select the best feature for a topic we use the above computed values, P_C^f , P_D^f , P_V^f , S_C^f , S_D^f , and S_V^f . The results from different features are then merged in a linearly with multiple evidence.

7.3.3 Retrieval Integration

The two systems, semantic search system and the content based image retrieval system, were integrated on a late fusion fashion. The core problem of generating the final list lies in devising a mechanism of combining the results originating from different search systems Annotation and semantic search system and content based search system engine systems to produce effective results. To combine the results, any fusion technique discussed in chapter 4 can be used.

7.4 Experimental Results

We carried out an experimental study on the PGP data to verify how the integrated system would work. The collection consisted of 1675 images. The annotation tool briefed in section 8.2 was used to annotate the collection. The collection was manually scanned and a set of topics were generated and classified as easy, difficult and intermediate depending on the number of relevant documents present in the collection. A team excluding those involved in generating the topics were formed to generate the ground truth data for evaluation.

The list of topics generated were

"Medium" number of relevant images:

1. Find images containing dome topped cylindrical buildings amidst mountains
3. Find images of tree houses in forests
5. Find sketches of spacecraft
12. Find images of buildings which are made of stone but not brick [image example required example: 06_vampire.jpg]
13. Find images of buildings which are made of wood, or largely made of wood [image example required: example: 06_jungle.jpg]
14. Find images of body parts of robots which are silver or red
21. Find images of spaceships which have landed or are landing
26. Find images of dresses. Images which contain people or other characters wearing dresses are NOT relevant

Topics with many relevant images:

2. Find images of buildings which must also include a sky
4. Find sketches of images of buildings
6. Find images of modern and science fiction buildings
7. Find images of skies with clouds which also have one or more planets visible
11. Find images or mushroom or balloon shaped buildings
15. Find abstract images which are only made up of a texture, or one or two colours [image examples are required for sure: GryDirt.jpg, back06L.jpg]
16. Find images which include textual content, such as title images
19. Find images which are only gray or black and white, but which are NOT sketches
23. Find images or buildings (e.g. lighthouses) which are shown isolated in the middle of water

Topics with only few relevant images:

9. Find images of building parts which have a Gothic design, such as windows, doors, etc.
The images must not include a background, or be part of a building [image example required: example image oct05L.jpg]
10. Find images of spaceship or other types of aircraft which have helicopter style blades
17. Find images of buildings which include body parts from animals in their design, such as eyes, teeth, etc.
18. Find images showing micro chips and/or circuit boards
20. Find landscapes of buildings in mountain regions with satellite discs
22. Find images which show buildings or other objects hanging by a chain or rope
24. Find images of broken transportation machines such as cars, boats, etc.
25. Find images of shoes, boots, or other footwear. Images which contain people or other characters wearing footwear are NOT relevant

Content based retrieval system used two features, Color Structure (CS) and Edge Histogram (EH) for retrieval. The Content Based Image Retrieval (CBIR) system uses both these features and performs a late fusion using the voting method explained in Chapter 4. Semantic Search (SS) results were obtained by using the Semantic Search system which made use of the annotated data. Finally, the results originating from the CBIR system and the SS system were integrated on a voting and round robin

mechanism. Table. 7.1 tabulates the number of relevant documents retrieved for the various features and different retrieval systems.

Topic	Num_Rel	ColorStructure		EdgeHist		CBIR		SemSearch		CBIR+SS	
1	17	2	11.76%	2	11.76%	2	11.76%	17	100%	17	100%
2	89	39	43.82%	36	40.45%	39	43.82%	61	68.54%	56	62.92%
3	12	3	25%	3	25%	3	25%	1	8.333%	3	25%
4	89	82	92.13%	63	70.79%	82	92.13%	87	97.75%	82	92.13%
5	13	1	7.692%	3	23.08%	1	7.692%	5	38.46%	5	38.46%
6	241	91	37.76%	92	38.17%	91	37.76%	42	17.43%	94	39%
7	124	108	87.1%	107	86.29%	107	86.29%	91	73.39%	108	87.1%
9	4	3	75%	3	75%	3	75%	1	25%	3	75%
10	1	0	0%	0	0%	0	0%	1	100%	1	100%
11	52	37	71.15%	36	69.23%	37	71.15%	1	1.923%	37	71.15%
12	10	7	70%	6	60%	6	60%	0	0%	6	60%
13	10	4	40%	4	40%	4	40%	3	30%	4	40%
14	35	28	80%	28	80%	28	80%	25	71.43%	29	82.86%
15	124	90	72.58%	77	62.1%	77	62.1%	-	-	77	62.1%
16	96	73	76.04%	72	75%	72	75%	-	-	72	75%
17	5	3	60%	3	60%	3	60%	5	100%	5	100%
18	3	1	33.33%	1	33.33%	1	33.33%	-	-	1	33.33%
19	60	33	55%	29	48.33%	29	48.33%	6	10%	28	46.67%
20	2	1	50%	1	50%	1	50%	1	50%	1	50%
21	48	25	52.08%	32	66.67%	25	52.08%	9	18.75%	31	64.58%
22	8	5	62.5%	3	37.5%	3	37.5%	0	0%	3	37.5%
23	72	20	27.78%	25	34.72%	20	27.78%	39	54.17%	40	55.56%
24	7	5	71.43%	5	71.43%	5	71.43%	5	71.43%	5	71.43%
25	5	5	100%	5	100%	5	100%	4	80%	5	100%
26	27	23	85.19%	23	85.19%	23	85.19%	4	14.81%	23	85.19%
All	1154	689	59.71%	659	57.11%	667	57.8%	408	35.36%	736	63.78%

Table 7.1 Topicwise results showing the number of relevant documents retrieved when different feature and search system was used.

Feature	MAP	RECALL	P@20	P@100
ColorStructure	0.0335	59.71%	0.0519	0.0415
EdgeHist	0.0336	57.11%	0.0365	0.0462
CBIR	0.0256	57.80%	0.0365	0.0335
SemSearch	0.1744	43.82%	0.1630	0.1370
CBIR + SemSearch	0.0920	63.78%	0.1154	0.0988

Table 7.2 Comparison of overall performance on various visual features, content based search system, semantic search system and the integrated system

From Table 7.1, it can be seen that, for the topics (9,10,17,18,20,22,24,25), which had significantly less number of relevant documents in the collection, there is mixed performance exhibited by both content based search system and the semantic search system. For topics 9, 18, 22 and 25, the visual features happens to be more appropriate when the percentage of recall is the evaluating measure. But, for the same category of less relevant document topics, 10, 17, semantic search has 100% recall while the content based search yields relatively negligible performance. The other topics in this category has relatively same performance from both the systems.

For the topics in medium relevant document collection, the topics inferred that there were two topics for which, specifying the query examples were important, conveying that they can be best addressable by content based retrieval system. These topics are 12 and 13. Although the content based retrieval system performed good for both topics, semantic search was comparatively good for topic 13. Content based retrieval was also good for topics 3, 12, 21 and 26. CBIR and SS performed equally good for topics 13 and 14. Semantic search system outperforms for topics, 1, and 5.

For the third category of topics having many relevant documents in the collection, i.e., 2, 4, 6, 7, 11, 15, 16, 19, and 23. Both are good for 4, 7. SS is good for 2, 23. CBIR is good for 6, 11, 15, 16, 19.

When the results of both SS and CBIR are merged together, if we assume that, the merged list should have a recall which is greater than the individual lists, we have an improvement for only 6 topics, at an average of 4.13%; and there is a decrease for 3 topics at an average of 4.3%. for the other 16 topics, the best of the SS and CBIR list is retained, this is perhaps due to the suitability of the specific search system for such topics and also the MAP. Table 7.2, show the MAP, RECALL, P@10 and P@100 for

the 2 features used in CBIR individually, their fused result in CBIR, SS and finally the combination of SSandCBIR. It can be seen from the table 3, MAP, P@10 and P@100 that the semantic search is capable of retrieving relevant documents to the top of the result list, whereas CBIR can retrieve many relevant documents as can be seen from the RECALL.

7.5 Conclusion

The chapter presents an integration of Semantic and content based search within the context of SALERO. A more sophisticated fusion mechanism with the automatic computation of the preference of the semantic system or the content based system for a topic, depending on the difficulty of the query and the presence of verbs in the queries would further be explored.

8 Usage in Experimental Productions

In this Chapter, we describe two SALERO Experimental Productions (EP's) which have used the SALERO search system: in Section 8.1, from TAIK, Alan01 and AlanOnline, while in Section 8.2 we give a brief description of how search is used in the Video Disk Jockey system from AM.

8.1 TAIK: Alan01 and AlanOnline

"The difference between man and machine is not how they operate but how they are treated." [Saarinen, 2008]

Alan Turing (1912-1954), a World War II code-breaker considered one of the fathers of modern computing, made a significant and provocative contribution to the debate regarding artificial intelligence: whether it will ever be possible to say that a machine is conscious and can think. As a person he was "an ordinary English homosexual atheist mathematician" [Opera, 2008]. For his wartime achievements he was awarded the OBE, Officer of The Most Excellent Order of the British Empire. In 1952, Turing was convicted of "acts of gross indecency" after admitting to a sexual relationship with a man in Manchester. He was placed on probation and required to undergo hormone therapy to achieve temporary chemical castration. The therapy caused Turing's body to develop female forms, and the conviction resulted in his security clearance being revoked. He died after eating an apple laced with cyanide in 1954. The death was ruled a suicide.

Not surprisingly, several pieces of drama have been inspired by Turing's life, including *Breaking the Code* by Hugh Whitmore [Whitmore, 1986] and *Turing* by Miko Jaakkola [Jaakkola, 2000]. The internationally awarded *Breaking the Code* starred Derek Jacobi as Alan Turing in London's West End and New York's Broadway theatres at the end of 80s, while Jacobi's performance was immortalised by the 1996 BBC production of the play [Wise, 1996]. Jaakkola's *Turing* was originally performed by the Helsinki based Circus Maximus in 2000. It had a rebirth recently, when Opera Skaala turned the script into the critically acclaimed multimedia opera *Turing Machine* [Opera, 2008].

Both plays depict Alan Turing as man who wished to beat death by coding human consciousness. Such an interpretation is supported by the significant biographical facts of Turing's life. At the age of 18 he lost Christopher, his best friend and love – and due to that event, also his faith in religion and life beyond death. Man was needed to invent the eternal life to save loved ones.

The theme is carried on in *Alan01 / AlanOnline*, in which engagers will be able to meet a fictional Alan in dialogue, interactively, as if his consciousness had indeed been carried on by a machine's code. This time the story is experienced by an associational structure suitable to computers, and in describing behaviours of human consciousness: "...the narrative is formed by a series of moments which are linked by common elements and do not rely on chronology or episodic relationships to produce their meaning or effect." [Parker, 2002] The structural choice also reflects the ACM Multimedia 2009 art exhibition theme of "disophrenia" and the fragmentation and timelessness of consciousness that is transferred into digital representation.

The research production carries on the continuing research on associational and new forms of interaction structure and narrative. In *Alan 01 / AlanOnline*, we are particularly interested in looking into user/engager and associational system interaction logics: how will interaction modes and tempo regulate and be influenced by story tempo and mode? Previous production-based research has included, for instance, *Accidental Lovers* [Tuomola, 2006], which discovered a clear correspondence between user interaction and story tempo, though free user interaction available at all times [Ursu, 2008].

8.1.1 The Concept: Alan01 / AlanOnline

Image retrieval plays a central role in the multimedia art productions that consist of two individual parts: Alan01 – a physical installation, and AlanOnline – the installation's online counterpart. In both cases the image retrieval system triggers the associational story in motion. The main input is a canvas where the user can create line drawings which are then used as a reference for the image retrieval. However, for the specific needs of the installation, the retrieval is made from a limited set of about 50 symbolic images. The graphical presentation and selection of these has been made bearing in mind the nature of the interface. The symbols are similar to what you would expect a user might draw in a short time of five to ten seconds. Another requirement for the selection of the symbols is that they are connected to the

context of Alan Turing's life. A couple of examples of the images: bird, hand, fish and heart are shown in Figure 8.1.

The idea of the drawing interface is therefore to enable a non-textual input to an art piece, which can still be translated to symbols and their textual meanings. From there on, the associational narrative structure script of the art piece can start to function.



Figure 8.1: Examples of the symbolic images shared by the installation and its online version

8.1.2 The Alan01 Installation

The physical installation is composed of a black darkened room, in the centre of which is a glass cube that is symmetric and measures 70 centimetres on each side. The top surface is sandblasted to make it suitable for video projection, images being projected via a mirror upwards from beneath the glass. Under the table there is also a video camera that captures any movement that happens above the cube. A drawing of the set up is shown in Figure 8.2. When a user presses her finger against the glass, the finger reflects the bright white light from the video projector. The software of the installation's main computer detects these bright blobs on the surface and projects black "paint" blobs on the corresponding spots. This experience of drawing on the glass is somewhat similar to finger painting. Once the interface remains untouched for a couple of seconds, the system displays a timer icon on the screen, which functions as feedback for the user, hinting that unless they resume drawing, the system will move on. Once the timer finishes, the user's image is saved locally on the computer. This image is then used as input to the image retrieval system, described in Section 3, which will return a result symbol, which is then presented to the user. The aim of this setup is to create a sophisticated input device without utilizing traditional buttons or keyboard.

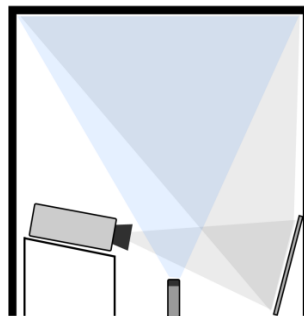


Figure 8.2: Camera and video projector set up of the Alan01 installation

A simple XML file serves as the database and structure for the content of the installation. In the XML file each symbol has a set of associative words, the number of which varies from one to five. The associative word is selected randomly from these potential matches and displayed on the surface with the resulting image. Connected to each word there is a light signalling code, structured as either short or long flashes, similar to Morse code, but not mapped to letters like the standard Morse code. The main unit sends this code sequence by blinking black and white sequences onto a TV screen.

Figure 8.3 illustrates the principle of communication between the installation units. The use of this traditional communication method is a historical reference to the work of Alan Turing as a code breaker during World War II and can also be seen as a reference to how this tradition is still amidst us. Morse code and sending messages with directional spotlights are still used by modern armies as a close range communication method, due to the difficulty of intercepting such signals. From the point of view of the

user's experience, it makes the flow of data visible and enables the user to alter the code by blocking the light that is sent from the main unit to the other receiving units.



Figure 8.3: Light signaling communication between installation units

The installation has three receiving units that monitor the TV screen of the main unit with Web cameras. These receiving units time the duration of the light sequences on the screen and the duration between them (see Figure 8.4 for a spatial sketch of the installation). Each receiving unit has an identical XML file that serves as the code key for translating the light message, the length of which varies from one to seven flashes, back to the corresponding textual meanings.

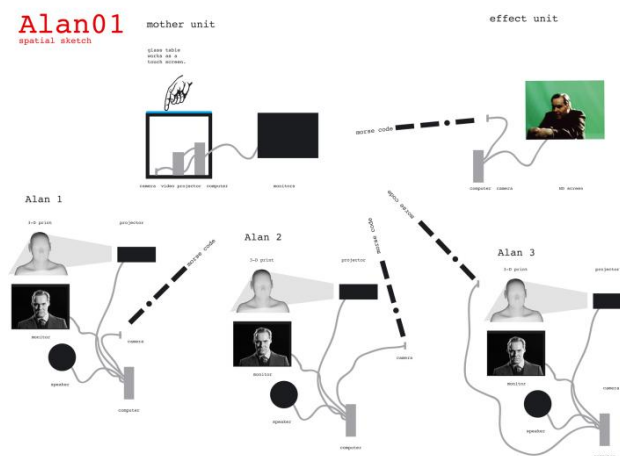


Figure 8.4: Spatial sketch for the Alan01 installation

Various different media assets are also associated with each code in this XML file. The receiving unit reacts to the message it received by showing the related content through its three output channels: video material of an actor portraying Alan Turing mixed with a 3D animation made with a model of the same actor's head are shown on a screen, as well as projected onto a physical miniature statue that is printed from the same 3D model. The emotional effect of the video projection bears resemblance to Tony Oursler's projections on balloon-heads, for example, in *System for Dramatic Feedback* [Guru, 2007], while the physicality of the 3D statue might remind the user of the uncanny robotic puppetry of Ken Feingold's *Head* [Manjunath, 2002]. However, the combination of "virtual" projection onto a statue created from the same 3D model is quite unique.

Dialog sentences are projected onto a surface and played as audio that has been produced with text-to-speech synthesis. In the XML file each machine has its own designated assets, so no two units will react in the same way. The video images are emotional reactions to the symbols and words, while the dialog is spoken by a monotonous machine voice that reveals aspects of the personal history and personality of Alan Turing.

Morse code and light signalling

The concept for the light signalling communication between computers was initially developed during a Pure Data workshop held by M. Koray Tahiroglu in Media Lab Helsinki during September 2008. The idea was to create a system where two computers could have a conversation with each other, in a way where the flow of data is visible to the viewers. The possibility of allowing errors where one machine misunderstands the other or where a viewer intervenes would then make the conversation between the two machines interesting and unpredictable.

In the system the sending and receiving units share a common code key that translates predefined words to sequences of light flashes (similar to Morse code) that vary in length from one to seven flashes. Each of the flashes is translated to be either long or short, and every burst of light within one message is followed by a short moment of darkness. Each complete message is followed by a dark period, the duration of which is at least twice the length of the short darkness. In this way the system

defines when a message is completed. Adjusting the duration of light and dark periods affects the susceptibility to interference. The longer the durations are, the less likely the system is to misunderstand the message.

Within the context of the installation, a user can manipulate the system by blocking a code sequence. The importance of a user's body for communication thus becomes articulated within the installation environment. The system can, for instance, misinterpret a long sequence such as "--...--" for a shorter one like "--...", if the user decides to block the receiving camera half way through the message. Since the code key is built in the fashion of a pyramid, shorter messages always have some meaning coded into them. Thus it is more likely that short messages are received without error and that longer messages are likely to be mistaken for shorter ones.

8.1.3 AlanOnline

The online counterpart of the physical installation shares many of its media assets with the installation. The image retrieval system is used in the same fashion and the logic of the connections of the media content in the XML file is similar. The media files are also the same, with the exception that the content is compressed and modified to be suitable for a Web browser interface. The different characteristics of physical media installations and non-material artwork, which only exist as software, are an important factor when a physical installation and online version are designed around the same theme, using shared assets. In the case of *AlanOnline*, the browser version is not an attempt to copy the physical installation, but instead to create a similar yet individual art piece that offers another interface to the same content, using the media elements and providing interaction and output in a way that is most suitable for its presentation media and technology. This is to ensure that the design choices of the online piece are valid, so that it can exist as well as if no physical installation was ever created. Lev Manovich [Manovich, 2001] describes the nature of the underlying question in the following way: "Synthetic computer-generated imagery is not an inferior representation of our reality, but a realistic representation of a different reality." Figure 8.5 shows the present version of *AlanOnline*.

In *AlanOnline* the user faces a white landscape that on closer examination reveals itself as a massive circuit diagram that disappears to a distant imaginary horizon. In the centre of this pseudo space is the main interface – a white slightly translucent cube. Where in the installation there is the main unit and its three subunits, here there is only one machine performing all the tasks, the user being figuratively placed inside the machine.

The user's tool of interaction is a black and white representation of a human hand, with its index finger extended. The top of the white cube is the canvas of the online version, which the user can draw on in much the same fashion as in the physical installation. Once the drawing is finished the Flash application sends the image back to the server, where it is used for the image retrieval that is made based on the same symbol set as in the installation.



Figure 8.5: A screen shot of the AlanOnline prototype

The result image is uploaded by the application and the result's media content is retrieved. The code that was used in the installation as a visible means of communication, is now revealed to the user as an audible series of telegraph style "dit" and "dah" sounds, which in addition to the historical reference, offer the user feedback that the system is processing data, and actions will soon follow.

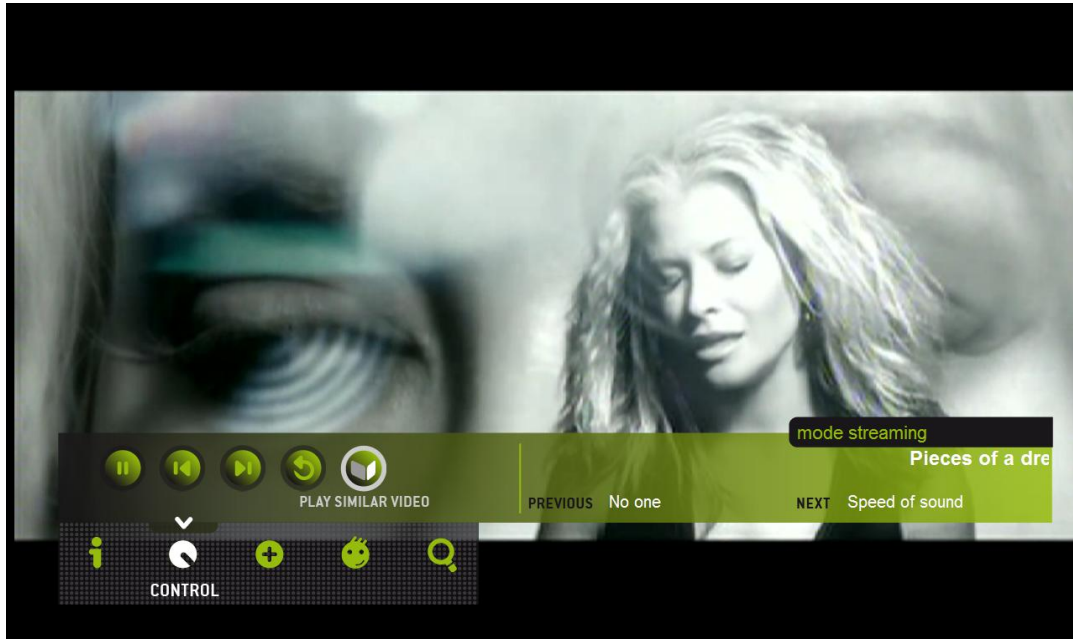
Once the associative word has been chosen, it is presented connected to the image. The dialog sentence is presented as text and played back using the synthesized voice. The voice is played using virtual surround sound so that the source of the sound seems to be behind the user, which enhances the users feeling of being in the centre of the machine. As an additional visual connection between the voice and the imaginary computer a sound wave spectrum reacts to the voice. Following the sentence, instances of the connected emotional video responses of the actor and 3D animations are then played in different parts of the virtual space.

8.2 AM: Video Disk Jockey

The 3rd experimental production developed by AM is a music video channel offering a Video on Demand (VOD) service. When the user accesses the music video channel he/she sees the videos being broadcast, but the channel offers a personalized broadcast service, allowing the user to select his/her videos of choice.

Thus, the user may, at any time, do a music video search and create a playlist with personally selected videos. The most basic way to do this is through a text search; however another option has been included, called "Play Similar Video" that can be used to do an advanced search.

This search allows us to incorporate the concepts of the Aspect Browser-Based Search System.



This search allows users to find videos similar to ones currently being broadcast. So if a user sees a music video and likes it, he/she can search for similar videos.

This search sends the request to a Web service hosted on the University of Glasgow's server, and a list of similar videos is returned back from the server.

In order to be able to do this, all of the videos had to first be sent to the UG so that they could process the videos and extract the necessary keyframes to leave the system available to do a subsequent search. To that effect, the UG created a Data Collection called "AMPE3" containing all of our indexed videos.

The system of recommendation operates as follows:

The user is watching a video clip, and when he/she asks to see similar videos, the system sends a shot of the frame of the video clip in question to the Glasgow Web service, indicating the search collection -in our case, AMPE3- and specifying the features we want included in the search: Text, ColorLayout, ColorHistogram, EdgeHistogram. Since we want the maximum number of options to appear, we indicate that we want as many features included in the search as possible, but it could also narrowed down.

Within a second or two, the application receives the list of similar videos and shows it to the user. He/she can then either play some of the recommended videos, add them to an already-existing playlist, or create a new playlist.

The following picture shows the results of a request for videos similar to Beyonce's "Single Ladies."



This function gives added value to the video on demand system because it includes an automatic recommendation system. In addition, this recommendation system can be automatically expanded in terms of number of videos and functions without affecting the user.

While the recommendation currently uses the image retrieval backend and visual features to do the recommendation, the architecture is such that other techniques could be substituted, such as those using audio, or techniques such as collaborative filtering. As such, the current recommendation system can be considered a “proof of concept”, which may be extended in future as appropriate.

9 Conclusions

In the just concluded reporting period, substantial technical progress has been made with a focus on integrating the activities of the different Work Package 5 partners. In particular, the emphasis during this phase was to provide a better match for the requirements of the SALERO experimental productions (EPs).

In this context, the requirements of the *Alan01* and *AlanOnline* concept and productions are very challenging and different from the capabilities of a typical image retrieval system. In order to make the system an integral part of the production, the system was required to have a very high precision rate and also the highest ranked result to be relevant. Moreover, though the *predictability* of the retrieval results is a central issue here, it is seldom considered important in a typical image retrieval task. These specific requirements of *Alan01* and *AlanOnline* productions have resulted in redefining the needs and roles of a content based image retrieval. As a consequence, on the feature extraction side (back-end), several new features were introduced keeping in mind the special requirements of the Talk collection. The new features were evaluated on the Talk collection. The initial results on the *Alan01 / AlanOnline* delivery systems' ability to produce engaging mini-narratives is promising and based on these evaluations some promising research directions are identified. We will continue to investigate some of them in the next stage of the project.

This report also described our proposed context-based search and retrieval system and its various interfaces. The proposed search interface allows a flexible searching of different and mixed collections of images, videos and text. One of the strengths of the system is that the relevant results found in one collection can be used as examples for searches in other collections. This feature enables the user to take advantage of both text and content-based searching.

The current version of AspectBrowser was evaluated on a set of complex and broad tasks; the user studies showed the effectiveness of the AspectBrowser in search and browsing. It was found that the tasks in which a user needs to identify aspects of a broad search task, the proposed AspectBrowser interface holds a clear advantage. To be more specific, in such a setting, the users were able to retrieve more relevant Web pages, viewed more results, and performed more search. Based on our initial findings, we will be running some more user evaluation studies to answer the questions which were not adequately covered in the current study.

Another important investigation carried out in this period is the integration of semantic and content based search for the retrieval task, and evaluating the integrated system on a standard dataset. It was observed that though the semantic search retrieved relevant documents to the top of the result list, the content based search was able to retrieve more number of relevant documents. Based on the evaluation results, it is proposed that more sophisticated fusion algorithms will be explored in the future to realize the potential of integrating semantic and content based systems.

In yet another activity, the AspectBrowser setup was used for searching relevant videos in response to a video query in Video Disk Jockey EP. The use of AspectBrowser as a video recommender systems opens up some new and interesting applications of a context based search and retrieval system.

The development of AspectBrowser is an ongoing work and is expected to go beyond the lifetime of the SALERO project.

10 References

- [Ballard, 1981] Ballard, D.H. 1981. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13, 111-122.
- [Bell, 2004] Bell, D. J., and Ruthven, I. 2004. Searchers' Assessments of Task Complexity for Web Searching. In *ECIR 2004*, Sunderland, UK, 57-71.
- [Bober, 2001] Bober, M. 2001, MPEG-7 visual shape descriptors, *IEEE Transactions on Circuits and Systems for Video Technology*, 11, 6, 716-719.
- [Borlund, 2003] Borlund, P. 2003. The IIR Evaluation Model: a Framework for Evaluation of Interactive Information Retrieval Systems. In *Information Research*, 8, 3, paper no. 152.
- [Burger, 2008] Burger, T., and Ammendola, C. 2008. A user centered annotation methodology for multimedia content. In *Poster Proceedings of ESWC*.
- [Bystrom, 1995] Bystrom, K. and Kalervo, J. 1995. Task Complexity Affects Information Seeking and Use. In *Information Processing and Management*, 2, 191-213.
- [Campbell, 1988] Campbell, D. J. 1988 Task Complexity: A Review and Analysis. In *Academy of Management Review*, 13, 1, 40-52.
- [Cutting, 1992] Cutting, D., Karger, D., Pedersen, J., and Tukey, J. W. 1992. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. In *SIGIR 1992*. Copenhagen, Denmark.
- [Fu, 1991] Fu, A.M.N., and Yan, H. 1997. Effective classification of planar shapes based on curve segment properties. In *Pattern Recognition Letters*, 18, 55-61.
- [Fujima, 2004] Fujima, J., Lunzer, A., Hornbæk, K., and Tanaka, Y. 2004. Clip, Connect, Clone: Combining Application Elements to Build Custom Interfaces for Information Access. In *UIST 2004*, New York, NY, USA.
- [Gary, 1993] Gary, J E., and Mehrotra R. 1993. Similar shape retrieval using a structural feature index. In *Information Systems*, 18, 7, 523-537.
- [Gonzalez, 2002] Gonzalez, R.C., and Woods, R. E. 2002. *Digital Image Processing*, Prentice Hall, ISBN 0201180758.
- [Guru, 2007] Guru, D. S., and Nagendraswamy H. S. 2007. Symbolic representation of two dimensional shapes. In *Pattern Recognition Letters*, 28, 144-155.
- [Han, 1994] Han, D., Bien, Z., and You, B.J. 1994. A theory of generalized incremental circle transform and its application for recognition of two-dimensional objects. In *Pattern Recognition Letters*, 15, 769-780.
- [Hearst, 2000] Hearst, M. 2000. Next Generation Web Search: Setting our Sites. In *IEEE Data Engineering Bulletin*, 23, 3, 38-48.
- [Hersh, 2000] Hersh, W., and Over, P. 2000. TREC-8 Interactive Track Report. In the 18th Text REtrieval Conference (TREC 8), Gaithersburg, MD, USA.
- [Ingwersen, 1992] Ingwersen, P. 1992 *Information Retrieval Interaction*. London: Taylor Graham.
- [Jaakkola, 2000] Jaakkola, Miko; Lehtonen, Jussi. 2000. Turing. Circus Maximus theatre, Helsinki
- [Kaki, 2004] Kaki, M. 2004. Proportional Search Interface Usability Measures. In *NordiCHI'04*, Tampere, Finland.
- [Khotanzod, 1990] Khotanzod, A., and Hong, Y.H. 1990. Rotation invariant image recognition using features selected via a systematic method. *Pattern Recognition*, 23, 1089-1101.
- [Kim, 2000a] Kim, H.K., and Kim, J.D. 2000a. Region-based shape descriptor invariant to rotation, scale and translation. *Signal Process.: Image Comm.*, 16, 87-93.
- [Kim, 2000b] Kim, W.Y., and Kim, Y.S. 2000b. A region-based shape descriptor using Zernike moments. *Signal Process.: Image Comm.* 16, 95-102.
- [Lakoff, 1985] Lakoff, G. 1985. *Women, Fire and Dangerous Things: What Categories Reveal about the Mind*. The University of Chicago Press.

- [Lund, 2001] Lund A. M. 2001. Measuring Usability with the USE Questionnaire. In Usability and User Experience, 8, 2.
- [Lunzer, 2008] Lunzer, A., and Hornbæk, K. 2008. Subjunctive Interfaces: Extending Applications to Support Parallel Setup, Viewing and Control of Alternative Scenarios. In ACM Transactions on Computer Human Interaction 14, 4, 1-44.
- [Manjunath, 2002] Manjunath, B. S., Salembier, P., and Sikora, T. 2002. Introduction to MPEG-7, WILEY, ISBN 0 471 48678 7.
- [Manovich, 2001] Manovich, L. 2001. The Language of New Media. Massachusetts Institute of Technology
- [Marchionini, 2003] Marchionini, G., and Brunk, B. 2003. Toward a General Relation Browser: A GUI for Information Architects. In Journal of Digital Information, 4, 1.
- [Marchionini, 2006] Marchionini, G. 2006. Exploratory Search: From Finding to Understanding. In Communications of the ACM, 49, 4, 41-46.
- [Mehltre, 1997] Mehltre B. M., and Kanakanahalli M. S., and Lee W. F. 1997. Shape measures for content based image retrieval: A comparison. In Information Processing and Management, 33, 3, 319-337.
- [O'Connor, 2005] O'Connor, N., Cooke, E., le Borgne, H., Blighe, M., and Adamek, T. 2005. The ace-toolbox: Low-level audiovisual feature extraction for retrieval and classification. In 2nd IEE European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies, 55-60.
- [Opera, 2008] Turing Machine (Opera). 2008. Dir. Janne Lehmusvuo, libretto (based on Jaakkola's play Turing) by Taina Seitovirta, composition by Eeppi Ursin & Visa Oscar. Opera Skaala & Crucible Studio, Media Lab, University of Art and Design, Helsinki.
- [Ounis, 2006] Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., and Lioma, C. 2006. Terrier: A High Performance and Scalable Information Retrieval Platform. In Proc. of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR).
- [Over, 1999] Over, P. 1999. TREC-5 Interactive Track Report. In the 7th Text REtrieval Conference (TREC 7), Gaithersburg, MD, USA.
- [Ozugur, 1997] Ozugur, T., Denizhan, Y., and Panayirci, E. 1997. Feature extraction in shape recognition using segmentation of the boundary curve. Pattern Recognition Letters, 18, 1049-1056.
- [Pao, 1992] Pao, D.C.W., Li, H.F., and Jayakumar, R. 1992. Shapes recognition using the straight line Hough transform: Theory and generalization. In IEEE Transactions on Pattern Analysis and Machine Intelligence, 14, 11, 1076-1089.
- [Parker, 2002] Parker, Philip. 2002. The Art and Science of Screenwriting. Intellect, London
- [Persoon, 1977] Persoon, E., and Fu, K-S. 1977. Shape discrimination using Fourier descriptors. IEEE Trans. Syst. Man Cybernet., 7, 170-179.
- [Prasad, 2004] Prasad, B. G., Biswas, K. K., and Gupta, S. K. 2004. Region-based image retrieval using integrated color, shape, and location index. In Computer Vision and Image Understanding, 94, 1-3, 193-233.
- [Raskin, 2008] Raskin, A. 2008. The Linguistic Command Line. In ACM Interactions. (January and February 2008), 19-22.
- [Saarinen, 2008] Saarinen, L. 2008. Turing Enigma. Script for the experimental production Turing Enigma (Cruible Studio / Media Lab, University of Art and Design Helsinki 2008) at <http://fullhouse.uiah.fi/turingenigma/> [the site last visited April 10, 2009].
- [Schraefel, 2006] Schraefel, M. C., Wilson, M., Russell, A., and Smith, D. A. 2006. mSpace: Improving Information Access to Multimedia Domains with Multimodal Exploratory Search. In Communications of the ACM, 49, 4, 47-49.
- [Taubin, 1991] Taubin, G., and Cooper, D.B. 1991. Recognition and positioning of rigid objects using algebraic moment invariants. In SPIE Conference on Geometric Methods in Computer Vision, 1570, 175-186.
- [The, 1986] Teh, C.-H., and Chin, R.T. 1986. On digital approximation of moment invariants. Computer Vision, Graphics, and Image Processing, 33, 318-326.

- [Tsai, 1995] Tsai, D. M., and Chen, M. F. 1995. Object recognition by a linear weight classifier. In *Pattern Recognition Letters*, 16, 6, 591-600.
- [Tuomola, 2006] *Accidental Lovers*. 2006. Dir. Mika Lumi Tuomola. Finnish Broadcasting Company YLE Channel 1 & Crucible Studio, Media Lab, University of Art and Design, Helsinki
- [Ursu, 2008] Ursu, M. F., Thomas, M., Kegel, I. C., Williams, D., Tuomola, M. L., Lindstedt, I., Wright, T., Leurdijk, A., Zsombori, V., Sussner, J., Myrestam, U., Hall, N.. 2008. *Interactive TV Narratives: Opportunities, Progress and Challenges*. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 4, 4 (Oct 2008)
- [Vakkari, 1999] Vakkari, P. 1999. Task Complexity, Problem Structure and Information Actions: Integrating Studies on Information Seeking and Retrieval Information. In *Processing and Management*, 35, 819-837..
- [Villa, 2008] Villa, R., Gildea, N. and Jose, J. M. 2008 *FacetBrowser: a user interface for complex search tasks*. In *proceedings of ACM Multimedia*, 2008, 489-498.
- [Villa 2009] Villa, R., Cantador, I., Joho, H., and Jose, J. M. 2009. *An aspectual interface for supporting complex search tasks*". In: *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009)*, to appear, Boston, USA: ACM
- [Wallace, 1980] Wallace, T.P., and Wintz, P. A. 1980. An efficient three-dimensional aircraft recognition algorithm using normalized Fourier descriptors. *Computer Graphics and Image Processing*, 13, 99-126.
- [White, 2006] White, R. W., Kules, B., Drucker, S. M. and Schraefel, M. C. 2006. Supporting Exploratory Search. In *Communications of the ACM*, 49, 4, 36-39.
- [Whitemore, 1986] Whitemore, Hugh. 1986. *Breaking the Code*. Theatre Royal, Bath & London. Also Neil Simon Theatre 1987, New York.
- [Wise, 1996] Wise, H (director) 1996. *Breaking the Code*. British Broadcasting Company (BBC), UK
- [Zahn, 1972] Zahn, C.T., and Roshies, R.Z. 1972. Fourier descriptors for plane closed curves. *IEEE Trans. Comput.*, 21, 269-281.
- [Zhu, 2002] Zhu, Y., De Silva, L.C., and Ko, C.C. 2002. Using moment invariants and HMM in facial expression recognition. *Pattern Recognition Lett.*, 23, 83-91.

11 Glossary

Partner Acronyms

AM	Activa Multimedia, ES
BLITZ	Blitz Games, UK
DFT	Digital Film Technology, DE
DIT	Dublin Institute of Technology, IE
DLLNI	DTS Licensing Ltd. Northern Ireland, UK
FBM-UPF	Fundació Universitat Pompeu Fabra, ES
JRS	JOANNEUM RESEARCH Forschungsgesellschaft mbH, AT
LFUI	Leopold-Franzens Universität Innsbruck, AT
PGP	Pepper's Ghost Productions Ltd., UK
TAIK	Taideteollinen Korkeakoulu, FI
UG	University of Glasgow, UK
UPF	Universitat Pompeu Fabra, ES
URL	Universitat Ramon Llull, ES