Project no. 34721

TAGora

Semiotic Dynamics in Online Social Communities

http://www.tagora-project.eu

Sixth Framework Programme (FP6)

Future and Emerging Technologies of the Information Society Technologies (IST-FET Priority)

D2.4 Final version of Tag-based navigation systems for images and music

---

Period covered: from 01/06/2008 to 31/08/2009
Start date of project: June 1st, 2006
Due date of deliverable: October 15th, 2009
Distribution: Public

Date of preparation: 15/09/2009
Duration: 39 months
Actual submission date: September 15th, 2009
Status: Final

Project coordinator: Vittorio Loreto
Project coordinator organisation name: Sapienza Università di Roma
Lead contractor for this deliverable: Sony Computer Science Laboratory

Executive Summary

Sony CSL implemented a tagging system that handles two types of media: images and music. It is designed as an open platform to experiment with new analysis techniques. In particular, we are evaluating the combination of tags and data analysis to improve the tagging systems. The development of this new platform gives us a good understanding of the internals of tagging systems and the available Web-based technologies to make them accessible on-line. The system, called Ikoru, is available online and the source code has been released under an Open Source license.

Contents

Chapter 1

Motivation

The Ikoru system, developed at SONY-CSL, is primarily used to experiment with collaborative tagging and content-based analysis. The reasons why we developed Ikoru are as follows:

Content-based tools: The initial motivation of the Ikoru project was to explore the combination of content-based analysis and tagging. A similarity-based search tool for images that takes advantage of both tags and content analysis is part of Ikoru.

Data Gathering: Ikoru is used as a platform to gather tagging data. To this end, a web site was developed that is open to the public. The current web interface, viewable at `http://www.ikoru.net`, handles both image and music files. We also designed the art installation *Phenotypes / Limited Forms*, which uses Ikoru as the underlying technology, with the aim to gather tagging data.

Small, reusable server: Ikoru consists of two main components: the web site and a web server. We built a new stand-alone HTTP server for this project. We made this choice to enable its integration in third-party projects, in particular in embedded devices or peer-to-peer applications.

In the next chapter we start with an overview of the main elements of Ikoru, while in chapter 3 we take a deeper look into the features of Ikoru. Finally, in the last chapter, we try to assess what we have learned from our work on Ikoru and look at some of interesting new developments.

Chapter 2

A Brief Overview of Ikoru

In this chapter we shall introduce the main features of Ikoru and give a general introduction. In the next chapter, you will find more details about Ikoru's implementation.

2.1 Web Interface

An Ikoru web page displays a set of resources – photos or music titles – and their associated tags people (Fig. 2.1.2 B and D). The set of resources is called a *context* in Ikoru and is defined by a unique URL. For example, the URL `http://www.ikoru.net/ikoru/hanappe/photo/erice/vittorio` represents all the photos of the person "hanappe", tagged with "erice" and "vittorio". All the navigation links on the web page are taken from the existing relations between tags, people, and resources.

People who register have a personal home page that becomes her default page for the Ikoru web site. This page lists the icons of most recently uploaded photos and their associated tags.

The web site can be seen seen at `http://www.ikoru.net`. A demonstration Web site with data taken from Flickr.com and Last.fm can be seen at `http://demo.ikoru.net`.

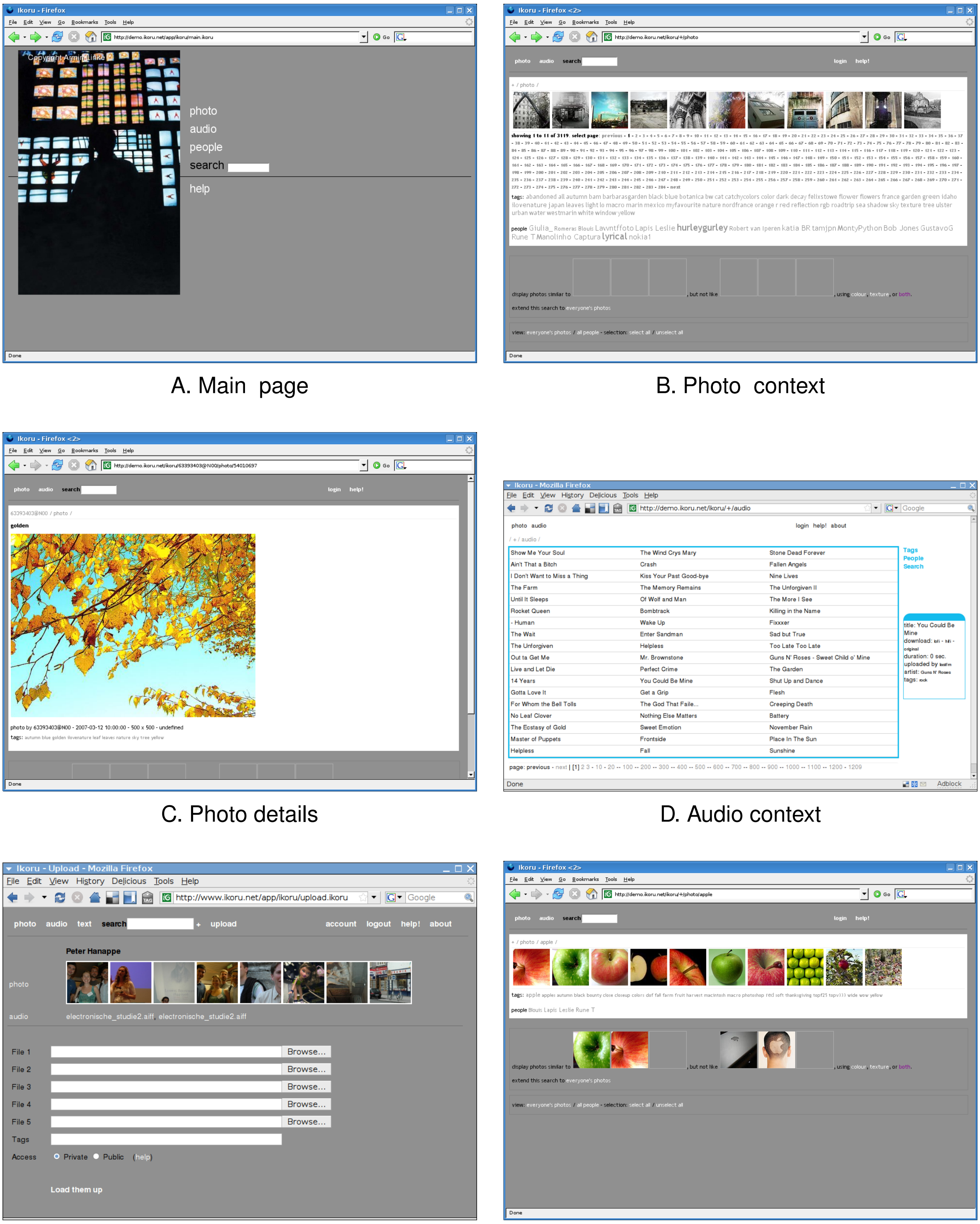
2.1.1 Browsing Photo's

The photos are shown as icons. When a visitor clicks on an icon, a larger size version of the photo is shown, including the details of the photo (title, capture date, etc.) (Fig. 2.1.2 B and C). By selecting a tag link, only the photos in the current context that are annotated with the selected tag are shown.

While navigating her photos and music, a visitor may at any time select one or more resources and add new tags to the selection by typing a space-separated list of words in a dedicated text field. Not only can she tag her own photos but we allowed the tagging of other people's public photos. When someone tags someone else's photo, it will become visible in her own photo collection.

2.1.2 Browsing Audio Files

The audio files are shown as a list of titles (Fig. 2.1.2 D). Clicking a title will start the playback of the audio, if the audio file is available. After holding the mouse over the title for several seconds, the detailed information of the audio file will be shown.



A. Main page

B. Photo context

C. Photo details

D. Audio context

E. Upload page

F. Photo context with classifier

Figure 2.1: Screenshots taken from http://demo.ikoru.net .

2.1.3 Uploading Files

With the upload page, people can transfer up to five media files to the server (Fig. 2.1.2 E). The user can assign tags to the new resources and specify the visibility of the media (public or private).

2.2 Similarity-Based Search Tool for Images

Every web page associated with a photo context has a similarity-based search panel. By drag-and-dropping a photo icon on the search tool, the photos of the context will be sorted according to the visual similarity with the example image. Up to three positive and negative example images can be given. The similarity measure can be based on colour, texture, or both. To enable the similarity search, the Ikoru server computes the colour and texture features vectors when a photo is uploaded (see also Section 3.2).

The search for visually similar images starts with one or more images selected by the user. These initial images can be found through tags. In our implementation, we focussed on a totally user defined process: Not only is the number of selected images left to the user, he is also free in all further actions to take. When the results of the similarity search are displayed, the user can either (1) exclude images, (2) select images for refinement, (3) combine (1) and (2), or (4) simply not take any action. This distinguishes our approach from methods suggested for *relevance feedback* in image retrieval (see e.g. (Rui et al., 1998)), where the user is forced to take certain actions, such as giving feedback to every retrieved image, or where he has to follow a strict order of interaction.

2.3 Phenotypes / Limited Forms

In 2006 SONY-CSL started working with photographer Armin Linke. He is an internationally recognized photographer who regularly displays his work at prestigious contemporary art gatherings. The art installation *Phenotypes / Limited Forms* evolved from the discussions with a large group of people, including designer Alex Rich, architect Wilfried Kühn, curator Doreen Mende, and students from the Staatliche Hochschule für Gestaltung in Karlsruhe.

From a research perspective, the installation has several aims. First, it explores the use of tagging in the off-line world. This implies working with physical objects, away from computer displays, with people who don't necessarily work with computers. Second, we wanted to investigate whether a tagging system can evolve a diverse and meaningful layer on top of an existing archive. Thirdly, the installation provided a good testing ground for the Ikoru technology.

In several aspects, the set up is different than standard tagging systems, as we will see. This closed set up and a tagging process that was in a sense designed for the installation, allow us to study tagging from a new angle. It consists of a thousand printed photos that are exposed on three 12 meter long shelves. Visitors are requested to browse the archive, select photos of their liking, and let a topic for their selection emerge. They can then lay out eight photos they have picked out on a specially created table. Using this table, and the embedded touch screen, they enter a single tag for the eight photos and print out a small booklet with their selection. The tag of the newly created book is projected on the wall together with the associated tag-cloud (Fig. 2.3.E).

The installation uses RFID technology to identify the photos that are layed on the table. The table and the projector communicate with the Ikoru server to store the tag assignments and retrieve the tag co-occurences.

A web site was developed, using Ikoru technology, as a virtual counterpart of the installation (see Fig. 2.3.F). It is accessible at `http://armin.hfg.ikoru.net`.

The installation has proven the be a more successful avenue to gather tagging data than the online web site. The installation was used by more than 22000 people and has been shown at the following exhibitions (more are being planned):

- "You_ser", Zentrum für Kunst und Medien (ZKM), Karlsruhe, Germany. 10/2007 - present.
- Institute for Contemporary Art and Thought (ITYS), Athens, Greece. 03/2008 - 07/2008
- Arts Bienal (Fundação Bienal de São Paulo), Brazil. 10/2008 - 11/2008

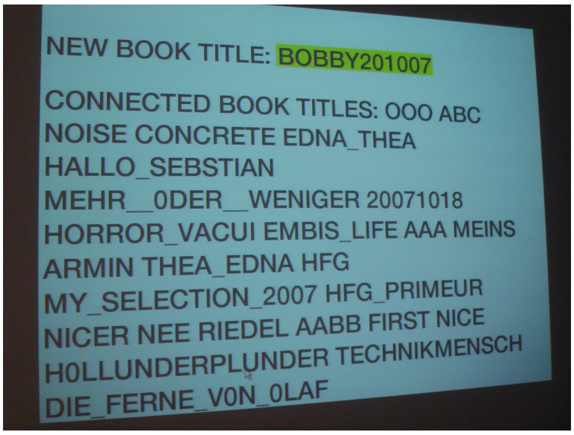
A. The installation at the ZKM, Karlsruhe, Germany.

B. Visitors at the ZKM

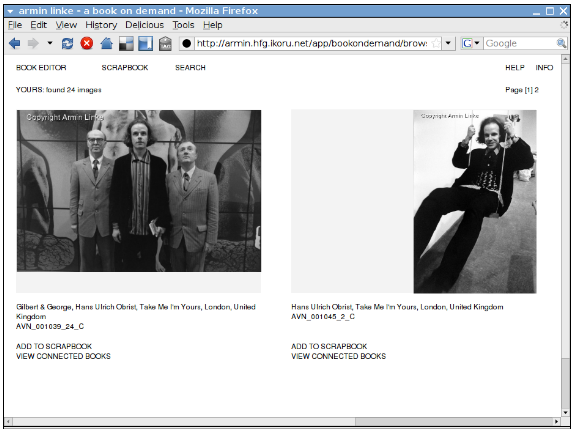
C. The installation at the Bienal of Sao Paulo

1) Put on the white gloves
2) Browse through the archive
3) Let a topic emerge
4) Choose eight photographs
5) Arrange your selection on the table
6) Input the title of your topic
7) Print
8) Return photographs to the archive
9) Take your book

D. The instructions given to the visitors



E. The projected tag cloud



F. The Phenotypes / Limited Forms web site

- Museum of Contemporary Art, Siegen, Germany. 05/2009 - present.

Figure 2.3 shows the growth of the Phenotypes dataset since the beginning of the installation, in number of tag assignments. In the deliverable D4.6 you can see the results of the ongoing analysis of this data set.

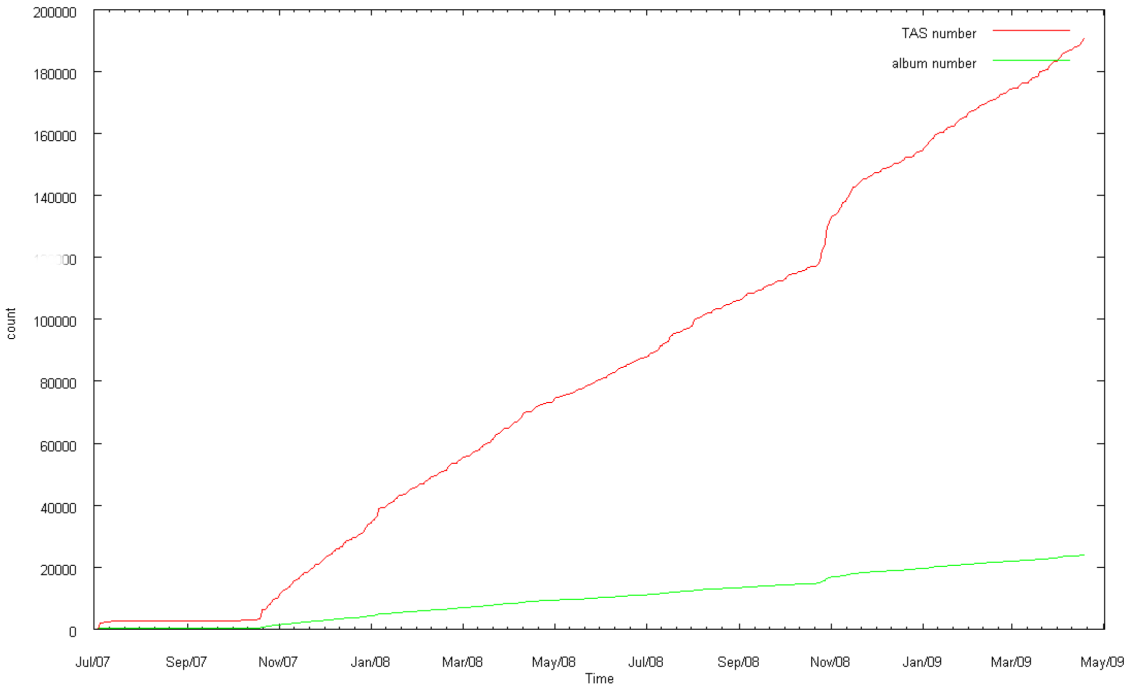


Figure 2.2: The growth of the Phenotypes data set.

Chapter 3

Implementation Details

3.1 Ikoru's Context Description

The notion of *context* is central in the structure of the Web site and the low-level API. A context defines a set of resources and is identified by a unique string that is similar to the format of a file path: a sequence of names separated by a slashes ('/'). The first element of the sequence defines the owner of the context, i.e. the name of the user. The plus sign ('+') can be used instead of a user name to indicate that the context covers all users known to the system. The second element defines the media type. It is a literal that should be equal to "photo" or "music". Following the media type, a list of zero or more tags can be given. It is possible to give the identifier of a resource as the last element of the context ID. In that case, the context contains only a single element, namely the specified resource. More formally, a context identifier has the following form:

$$ContextID : /\{PersonID|+\}/\{\text{photo}|\text{music}|...|+\}[/Tag]^*[/ResourceID]$$

Some examples of valid context identifiers are:

- $/hanappe/photo$: all the photos of user "hanappe".
- $/+/photo/tagora/2007$: the photos of all users tagged with "tagora" **and** "2007".
- $/hanappe/+/1331ba371dd3$: the resource of user "hanappe" with the unique id "1331ba371dd3".

There is a direct mapping between the context identifier and the URL of the Web page to view the context. For example, the URL `http://www.ikoru.net/ikoru/hanappe/photo` will display all photos of user "hanappe".

3.2 Image Analysis

In the following, we describe the visual features and the implementation of our content-based image retrieval process. Most of the techniques we used reflect either the state-of-the-art in image retrieval or are well-established standards in image analysis and pattern recognition. The idea of our system is to advance neither of these fields but to use the available tools in a new, intuitive, and creative way.

We tested our system with photographs downloaded from Flickr. Currently, we use about 3000 photographs from 12 randomly chosen users.

3.2.1 Image Features

We intentionally employ simple global features in our system. Rather than trying to recognise objects or even explain the meaning of an image, we seek to measure a certain "atmosphere", or a vague visual pattern, which we believe is possible to capture by low-level image features.

The visual features we used are colour and texture, i.e.

$$F = \{f_i\} = \{\text{colour,texture}\}$$

Colour Features

Comparison of colour histograms is known to be sensitive to small colour variations caused e.g. by lighting conditions. In order to obtain a more robust and simpler measure of the colour distribution, we calculate the first two moments (mean and standard deviation) in RGB colour space. In addition, we use the standard deviation between the means of the three colour channels. Intuitively, this yields a measure for the "colourfulness" of an image. The feature has a value of zero for grey-scale images and increases for images with stronger colours. We map the values to a logarithmic scale in order to distribute them more equally. In total, the colour feature vector has thus seven dimensions.

Texture Features

Texture refers to the properties that represent the surface or structure of an object. In our work, we seek to employ texture features that give a rough measure of the structural properties, such as linearity, periodicity, or directivity of an image. In experiments, we found *oriented gaussian derivatives* (OGD) to be well-suited for our purposes (Alvarado et al., 2001). This feature descriptor uses the steerable property of the OGD to generate rotation invariant feature vectors. It is based on the idea of computing the "energy" of an image as a steerable function.

The features are extracted by a 2nd order dyadic pyramid of OGDs with four levels and a kernel size of 13x13. The generated feature vector has 24 dimensions. The first order OGD can be seen as a measure of "edge energy", and the second order OGD as a measure of the "line energy" of an image.

3.2.2 Feature Integration

The distance between a query image and an image in the database is calculated according to the $l2$ norm (Euclidean distance). We use a linear combination of the distances in the colour and texture spaces to combine both features. In order to give the same initial weight to all features, the values are normalised linearly before calculating the distance. The joint distance d between a database image x_l and a query image s_k over all features spaces f_i is thus

$$d(x_l, s_k) = \sum_{i=1}^{N} w_i d_i, \qquad \text{with} \quad \sum_{i=1}^{N} w_i = 1$$

where N is the number of features in the set F and w is a weighting factor. In our implementation, w was set to $\frac{1}{N}$.

3.2.3 Image Selection

In case the user selects several images for his query (multi-image query), we think of these images as representing different classes. Thus, we accept images for retrieval that are similar to one of

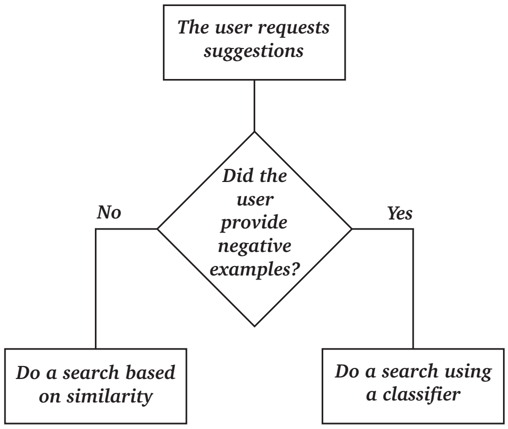


Figure 3.1: Based on the user selection, Ikoru either performs a similarity search, or a classification.

the query images. An alternative approach would be to average over the selected images which is, however, rarely relevant because the user might select visually distinct images. To give a simple example, a user selection of a yellow and a blue image should not yield green images as a result, but images that are either yellow or blue. Selection of the retrieved images is performed according to the following equation. Let X denote the archive and let x_l denote the l-th image in the archive. Let S denote a set of query images selected by the user. The distance D of x_l to S is then defined by

$$D(x_l, S) = \min_k d(x_l, s_k) \tag{3.1}$$

where d represents the distance of x_l to an image s_k contained in S, and k denotes the number of query images in S.

3.2.4 Refinement of Results

If the user is not entirely satisfied with the retrieved images, he has the possibility to refine the results. He can choose (1) one or more images as positive examples, or (2) one or more images as negative examples, or (3) combine (1) and (2). In case only positive examples are chosen, these are added to the initial query images and the query is started anew by evaluating Equation 3.1 and selecting the n closest images. If the user chooses to provide the system with one or more negative examples, the retrieval process becomes a classification problem. (see Figure 3.1). The set of all user-selected images can then be seen as prototypes labelled either “positive” or “negative”.

It is important to note that the user might choose very different examples for the same label, i.e. he might choose for example, a red image with a very smooth texture, and a green image showing high contrast leaves both as positive examples. Therefore, a parametric classification method is not suited since it assumes the distribution of the underlying density function to be unimodal. In our case, it is a much better choice to employ a non-parametric approach that can be applied for arbitrary distributions and without the assumption that the forms of the underlying densities are known. Furthermore, it is important to ensure a smooth transition between retrieval and classification in order to avoid a drastic change of the results as soon as negative examples are selected.

A method that fulfils these requirements is a simple nearest neighbour classifier. Equation 3.1 basically defines the distance of an image in the database to a set of query images to be the distance between the test image and its nearest neighbour in the query set. For this reason, nearest neighbour classification is the natural choice to follow similarity retrieval. Let $P^n = \{x_1, \ldots, x_n\}$

denote a set of n labelled prototypes and let $x' \in P^n$ be the prototype nearest to a test point x. Then the nearest neighbour rule for classifying x is to assign it the label associated with x'.

3.3 Music Analysis

For images, SONY-CSL successfully used a combination of tags and content-based similarity to navigate large archives. For music, we studied whether this approach yields the same advantages as for images.

Both the similarity and classification of audio recordings have been studied at SONY-CSL using state-of-the-art techniques. The results that were obtained were among the best in this field. A SONY-CSL study that pre-dates the TAGora project but that is relevant to the current question, investigates similarity-based search for audio (Aucouturier, 2006). For short and monophonic sounds, the existing techniques yields satisfying results. But when they are applied to longer recordings of polyphonic music, these techniques are no longer able to discriminate between the different samples. The approach that was taken at SONY-CSL was to represent audio recordings were a Gaussian Mixture Model (GMM) of the Mel-Frequency Cepstral Coefficients (MFCC). However, using this technique it is not possible to directly measure the distance between two GMM's, and thus between two audio recordings. An intermediate step is necessary in which data points are generated using the probability distribution of the GMM. The distance is then calculates between the two sets of generated data points.

SONY-CSL also did an in-depth study of automatic classification of audio recordings. This study is very relevant for automatic tag suggestion. The results of this study are discussed in detail in (Aucouturier et al., 2007; Pachet and Roy, 2007).

The integration of content-based search tools in Ikoru faces several difficulties. First, the content-based search tools for images are very different from content-based search techniques for audio and a direct translation of the search for one media to the other is not possible. Secondly, these techniques are very computation intensive. The successful integration of these techniques in a public, interactive web site is an enourmous engineering task. Third, although the results that were obtained for the similarity and the classification are better than previous results found in the literature, they are still unsatisfactory for practical use. These results do not yet justify the immense development task that is required. Fourth, the classification study uses a database that was established internally by another division in the Sony group, not by SONY-CSL. We do not have the rights to make this database available online, even for demonstration purposes. Lastly, such a development makes sense only for a long-term commercial project, which is outside the scope of the TAGora project and the mission of SONY-CSL.

For these reason we did not pursue the integration of the music content-based search into Ikoru.

3.4 The Ikoru Server

The Ikoru server was developed as a stand-alone server written in C++. We aimed to make the server compliant with Web standards. It is developed and maintained on Linux but is easily portable to Windows and UNIX variants. It consists of about 40000 lines of code and handles the following tasks:

- manage network connections and thread pool,
- reply to HTTP requests (SOAP and REST),
- manage persistent sessions (using HTTP cookies),

Figure 3.2: The steps in the handling of a basic query.

- assure the security (login, validate user input, permissions),
- store/retrieve tagging and user data in a persistent storage,
- receive the uploaded files (testing file type, conversion to Web formats, feature extraction),
- search data based on the text and content (for images),
- generate pages dynamically using key/value substitutions, and

The choice of writing a complete Web server was motivated by several goals. We mainly wanted to make it easy to deploy Ikoru on a wide range of platforms. In particular, we wanted to keep the option that users install Ikoru for personal use on their computers to manage their local media files. This, in the long run, may allow Ikoru to be deployed in peer-to-peer systems. We wanted to retain the possibility to use Ikoru on embedded devices, such as Network Attached Storage (NAS) devices, game consoles, or even mobile devices. For this reason, the amount of resources used by Ikoru had to be controllable. Finally, we wanted to facilitate the integration of the system in larger projects. As a result, Ikoru was developed with a self-contained "Web component".

Ikoru publishes the API of its services on the Web using the Web Service Description Language (WSDL). Client libraries can use the WSDL description to build interface classes that allow third party applications to communicate with Ikoru. Client libraries for JavaScript, Java, and PHP are available, in addition to a partially implemented C++ client library. During the course of the TAGora project, Web practices have evolved and we have implemented a RESTful interface to the Ikoru server.

3.4.1 Basic Query Handling

At its core, the Ikoru server mainly processes tag assignments, as defined in (Hotho et al., 2006). In our implementation, every tag assignment has also an associated resource type (photo, audio, . . .) and a resource index. People can use this index to manually change the order of the resources associated with a tag.

Most queries to the Ikoru server start with obtaining all the tag assignments for the requested context. It then checks the access permissions (public/private) associated with the resources. In the next step, the assignments are filtered using the active content-based classifier, if there is one. Depending on whether the client requested the list of people, tags, or resources associated with the context, only the distinct tag assignments for that data field are retained. Next, the tag assignments are sorted according to a client-defined criterion (post date, tag size, alphabetic order, . . .). We designed the comparison function between tag assignments such that it introduced a total order on the tag assignments. Finally, the client can request to obtain a specific interval from this ordered set, such as the first ten tag assignments. This sorting and slicing can be applied several times. A client can request, for example, to pick the 50 biggest tags (i.e. with the most tag assignments) and then sort this list alphabetically.

3.4.2 The SOAP API

SOAP is a object-oriented communication protocol used to build Web services. The messages are encoded in XML and are, in most cases, exchanged over an HTTP communication layer.

In Ikoru, all interactions between the client and the server follow a request-response pattern.

The API is formally described using the Web Service Definition Language (WSDL). Ikoru's service definition can be obtained at `http://www.ikoru.net/wsdl/v3`. This description is normally not read by humans but by middleware that glue web services together. Support for WSDL and SOAP exists in many languages, including Java and PHP. Despite the advantages of SOAP, the need for middleware and the verboseness of the XML messages make the integration of SOAP/WSDL based services into Web applications more difficult. As a result, this approach has been superseded by the simpler and more losely defined REST API's in the Web development community.

3.4.3 Meta-Data for Music

The Ikoru system was initially developed for images but has been extended to handle music files as well. Although Ikoru is targeted at user-made contents, most music tagging sites (e.g. Last.fm) focus on music produced for commercial purposes. The latter already has a fair amount of associated meta-data, such as the artist name or the album title. The challenge was then how Ikoru could maintain a simple tagging concept yet include the existing meta-data. Instead of adopting the MusicBrainz database schema (a often used schema to store music information) we choose to add this information using "machine tags". Machine tags were introduced to add, for example, geographical information to photos. Machine tags have the following syntax: $namespace : predicate = value$. We subsequently adapted the existing meta-data to this format. For example, "music:artist=the_beatles" indicates that the music file is a recording of "The Beatles". Similarly, "music:album=the_white_album" specifies that the music files was is part from "The White Album". The Ikoru server does not make a distinction between "normal" tags and machine tags. Ikoru clients, such as the Ikoru Web site, detect machine tags and display the information appropriately.

3.4.4 Source Code

The source code of Ikoru is available at `http://ikoru.svn.sourceforge.net`. The code is licensed under the GNU Library General Public License (LGPL).

Chapter 4

Conclusions and Future Work

Ikoru is a fairly finalised product that has been actively used for the past two years. It is available online at `www.ikoru.net` and, in source code, at `ikoru.svn.sourceforge.net`.

One way to evaluate Ikoru is to look at its acceptance as a Web service or as a software component in other systems.

As a web service, Ikoru is online and will continue to be available after the end of the TAGora project as a demonstration of the achievements of the project. However, in 2007 we have decided not to promote the Ikoru web site to attract a large audience. On the one hand, many competing, high-quality photo sharing services are on offer today and their service constantly improves. On the other hand, SONY-CSL simply cannot guarantee such a minimum quality of service in the long run.

Contacts within the Sony group have triggered a wide interest in our work, both in Europe and Japan. In particular, we maintain contacts with Sony Digital Imaging (based in the UK) and SoNet, Sony's internet devision in Japan. However, these contact have not resulted in a transfer of the Ikoru technology.

Ikoru has been successfully employed in two smaller projects that were elaborated together with photographer Armin Linke. The first project ran in collaboration with the university IUAV, in Venice, Italy. The results were presented in 2006 during the "Intensive Science" exhibition organised by SONY-CSL at "La Maison Rouge" in Paris. The second project, the art installation *Phenotypes / Limited Forms* was successful both for the gathering of data and for the promotion of the TAGora project.

As a tool for content-based analysis in combination with tagging, Ikoru has integrated novel ideas in the field of images. These results were presented at international conferences and have received positive reviews (Aurnhammer et al., 2006a,b). SONY-CSL performed an extensive study on the use of content-based tools for tagging systems for music (Aucouturier et al., 2007; Pachet and Roy, 2007). However, the integrated of these audio analysis techniques in Ikoru of proved to be much more challenging than for images.

As a technology, Ikoru has proven to be a very capable building block. The choice to develop a stand-alone server in C++ was motivated mainly to be able to make the Ikoru web technology available offline on a personal machine. The debate on how to make web applications available offline has not settled, yet, but it can be expected that support for the functionnality will eventually be built into the browsers using standard Javascript API's. As a result, it is probably wiser to write a web application using the popular web technologies.

Ikoru has been most successful in small-scale projects that explore an alternative use of tagging. In particular, the Phenotypes / Limited Forms installation has gathered more tagging than we expected.

The lessons learned from Ikoru and from the zexe.net project, i.e. to use standard web technology and to explore alternative uses of tagging interacting with small but motivated communities, are some of the main motivations for the **NoiseTube** project.

Bibliography

P. Alvarado, P. Doerfler, and J. Wickel. Axon2 – A visual object recognition system for non-rigid objects. In *Proceedings International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA)*, July 2001.

J.-J. Aucouturier, F. Pachet, P. Roy, and A. Beurivé. Signal + Context = Better Classification, 2007. Submitted to the International Conference on Music Information Retrieval (ISMIR).

Jean-Julien Aucouturier. *Ten Experiments on the Modelling of Polyphonic Timbre*. PhD thesis, University of Paris 6, Paris, France, May 2006. URL http://www.jj-aucouturier.info/papers/PHD-2006.pdf.

M. Aurnhammer, P. Hanappe, and L. Steels. Integrating Collaborative Tagging and Emergent Semantics for Image Retrieval. In *Proceedings WWW2006, Collaborative Web Tagging Workshop*, May 2006a.

M. Aurnhammer, P. Hanappe, and L. Steels. Augmenting Navigation for Collaborative Tagging with Emergent Semantics. In *International Semantic Web Conference*, 2006b. http://www.csl.sony.fr/downloads/papers/2006/aurnhammer-06a.pdf.

Andreas Hotho, Robert Jaeschke, Christoph Schmitz, and Gerd Stumme. BibSonomy: A Social Bookmark and Publication Sharing System. In *Proceedings of the Conceptual Structures Tool Interoperability Workshop at the 14th International Conference on Conceptual Structures*, 2006. to appear.

F. Pachet and P. Roy. Exploring billions of audio features. In Eurasip, editor, *Proceedings of CBMI 07*, 2007. http://www.csl.sony.fr/downloads/papers/2007/pachet-07a.pdf.

Yong Rui, Thomas S. Huang, Michael Ortega, and Sharad Mehrotra. Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval, 1998.