Combining Content-based Image Retrieval with Textual Information Retrieval

Research project by

Tobias Weyand

supervised by

Thomas Deselaers

at the

Chair of Computer Science 6, RWTH Aachen

October 2005

1 Introduction

Among the applications of computer science to the field of medicine, the processing of medical image data is playing an increasingly important role. With medical imaging techniques such as X-Ray, computer tomography, magnetic resonance imaging, and ultrasound, the amount of digital images that are produced in hospitals is increasing incredibly fast. Thus, the tasks of efficiently storing, processing and retrieving medical image data have become important research topics. Many hospitals use *picture archiving and communication systems* (*PACS*), which are basically computer networks that are used for storage, retrieval, and distribution of medical image data. In such a network, a central server provides access to an image database from which clients such as medical staff can retrieve images by using metadata like the name of the patient, the date, the imaging method, the body part, etc. Metadata-based retrieval is done via standard database tasks that are relatively easy to implement. If the metadata for an image is not sufficient to formulate a precise enough query, a textual query can be given. This query can for example be a set of keywords or a full textual description of the desired images. Then, the PACS searches for database images with similar descriptions. This retrieval task is more complicated and involves techniques from the field of text retrieval. If for example a doctor wants to compare X-ray images of his current patient with images from similar cases, he could also use these images as queries and let the PACS find the most similar entries in the database. This kind of searching for images is called *content-based image retrieval* (*CBIR*) and is currently part of the research of many computer science groups, who are trying to find models for the similarity of digital images. Several content based image retrieval systems are currently being developed. Some are more specialized on medical data like IRMA[Lehmann & Güld^{+} 04][1] and some are designed to be as general as possible, like FIRE[Deselaers 03][2], which can handle diffent kinds of medical data as well as non-medical data like photographic databases.

But to be able to formulate more precise queries, it would be desirable to combine these three kinds of image retrieval (metadata, textual descriptions and content-based image retrieval), which was the aim of this study project. By doing this, we (the image processing group of i6) hope to achieve better results, not only because more complex queries are possible, but also because the results of simple (i.e. text-only or image-only) queries can be improved by using both kinds of data for retrieval.

1.1 Overview

In the remainder of this section, related work is described and image retrieval, especially content-based image retrieval, is explained. Then, in section 2, FIRE is introduced. Section 3 discusses meta information for images and how we included them in FIRE's retrieval. A short overview of text-retrieval and a description of the used text retrieval engine is given in section 4. Section 5 explains the way in which text retrieval was incorporated in image retrieval. Experimental results on two corpora are be discussed in section 6. This work is concluded in section 7.

1.2 Related work

Several groups have proposed approaches to incorporate image annotations in the image retrieval task.

[Müller & Geissbühler^{+} 05] describe their submission for the ImageCLEF 2004 evaluation in. They use their Image Retrieval System GIFT[3][Squire & Müller^{+} 99] in combination with the information retrieval engine easyIR[4]. GIFT is a feature-based approach, using techniques from information retrieval for image retrieval. The combination is done by query expansion: The first query is done image-based only. Then, the first k results are used for text-based retrieval. The resulting scores are weighted and combined to the final score.

[1] http://www.irma-project.org

[2] http://www-i6.informatik.rwth-aachen.de/ deselaers/fire.html

[3] http://www.gnu.org/software/gift/

[4] http://lithwww.epfl.ch/ruch/softs/softs.html

Lin and Chang define a meta-language for representing text and images [Lin & Chang$^+$ 04]. To transform images to this language, they segment all images into regions and cluster the set of all regions into blobs[Carson & Belongie$^+$ 02]. Then, they create a co-occurrence model of blobs and words in the textual image descriptions. The indices of the blobs are fed into an information retrieval engine. Now, image- and text-information can be used for retrieval.

Alvarez et al calculate feature vectors for certain properties (texture, shape and edge) of each image [Alvarez & Oumohmed$^+$ 04]. These features are used for image retrieval, together with a model that assigns each description term a kind of feature that it may refer to. For example "zebra" will likely refer to the texture properties of the image. These correlations are incorporated in the retrieval process.

Van Zaanen and de Croon also use a feature-based approach in their engine called FINT [van Zaanen & de Croon 04]. Text information is converted to an "infomap" [5], which is used as an additional image feature.

1.3 What is image retrieval?

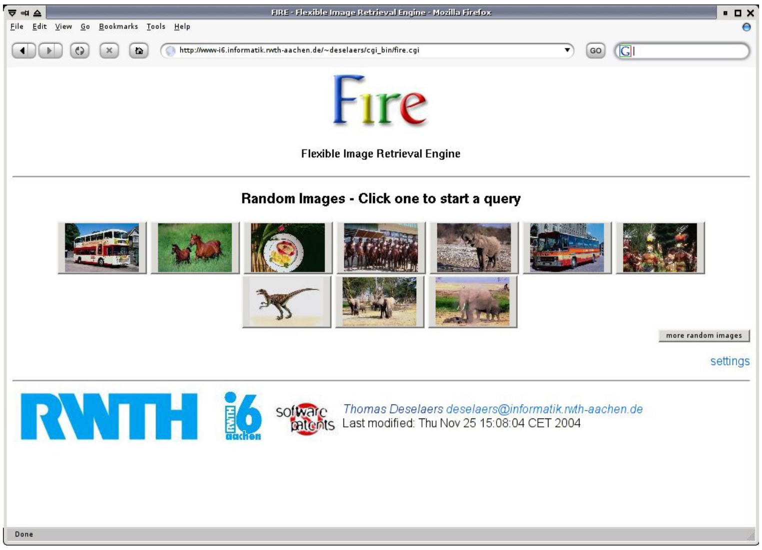
Image retrieval is the task of retrieving digital images from a database. As mentioned before, an image retrieval system in medical applications is often part of or has to interact with a PACS-System. Image retrieval systems differ in the way in which querying and retrieval is done. The possible kinds of queries were already introduced in the introduction:

- Meta information, like the patient's name
- A textual description, like "An X-ray image showing a fracture in the lower left arm."
- Visual information

Querying by visual information is called content-based image retrieval or CBIR. CBIR is the application of computer vision to the image retrieval problem. The visual information can be an image (query by example), but it can also be a sketch of the desired result or a description of the image's properties like the proportion of the desired colors (50% red, 30% green, ...).

Many image retrieval systems allow refinement of the search results by *relevance feedback*. This means that the user can rate the resulting images as relevant or irrelevant to the query and then repeat the search with this additional information.

[5] http://infomap.stanford.edu/


FIRE - Flexible Image Retrieval Engine - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www-i6.informatik.rwth-aachen.de/~deselaers/cgi_bin/fire.cgi
GO
Fire
more random images
settings
Done

Flexible Image Retrieval Engine

Random Images - Click one to start a query

Thomas Deselaers deselaers@informatik.rwth-aachen.de
Last modified: Thu Nov 25 15:08:04 CET 2004

Figure 1: The FIRE web interface

2 Fire

FIRE (Flexible Image Retrieval Engine) is a content-based image retrieval system that is being developed at the Chair for Human Language Technology and Pattern Recognition (i6) at the RWTH Aachen University. More precisely, FIRE implements query by example. The query images can come from the database itself or can be external images specified by the user. FIRE implements a large variety of different image features that can be combined and weighted individually to adapt the system to a specific task. Relevance feedback can be used to refine the result. Figure 1 shows a screenshot of FIRE's web interface.

2.1 Structure and retrieval process

Fire consists of a server and a client. The client implements FIRE's web-based user interface. It sends queries to the server and displays the result to the user. The server is the main part of FIRE that performs retrieval on the image database. The two parts are connected via a network socket, so they can (but do not have to) run on different machines. We will now describe the retrieval process in detail:

In FIRE's definition, an image X is a set of *features* $X_1, ..., X_M$. Each feature represents a ceratin aspect of the image, for example the histogram, color or texture information.

A selection of the individual features that are available in FIRE are described in the next paragraph. Features can have different data types. For example, a histogram is represented as a vector of scalars. More complex features are for example a scaled-down version of the image, which is represented as a large vector of color- or gray values. But a feature can also just contain the information whether the image is black and white or color. In this case, only one binary value is needed. Now, in order to model the similarity of two images, we have to calculate the distances of the individual image features, and then sum them up to the final image *score*. To get a reasonable value for the individual feature distances, we have to use an appropriate distance measure d_i for each of the features X_i of image X. For example, for the binary feature it is sufficient to set the distance to one, if the binary values of the two compared images are different and to set it to zero, if they are equal. For vectors like the scaled-down images, other distance measures like the euclidean distance are used. There are also more complicated distances, but I will omit them here. A detailed description of all the features is given in [Deselaers 03]. As mentioned before, FIRE allows for the adaptation of the retrieval process to different tasks. This is done by defining a set of features that is appropriate for the task and then assigning weights w_i to the individual features X_i. These weights determine the influence of the weighted feature on the final distance.

Given a query image Q, FIRE calculates the distance from this image to every image X in the database. This distance consists of the weighted sum of the respective feature distances:

$$d(Q, X) = \sum_{m=1}^{M} w_m \cdot d_m(Q_m, X_m)$$

Here, M is the number of features used. Because the ranges of the distance values can vary strongly for different distance measures, they are normalized such that

$$\sum_{X} d_m(Q_m, X_m) = 1$$

holds for each d_m. Now, the final image score is calculated from the normalized and weighted distances as follows:

$$S(Q, X) = \exp(-d(Q, X)) = \exp\left(-\sum_{m=1}^{M} w_m \cdot d_m(Q_m, X_m)\right)$$

After calculating the scores for all database images, the k images with the highest scores are returned and displayed to the user.

2.2 Image features

Now, some of the image features that are available in the FIRE framework are described. A more complete overview can be found in [Deselaers 03]:

Appearance-based image features In some applications like optical character recognition, it is a successful and easy approach to scale the image down to a very small size (e.g. 32x32 pixels) and compare these versions via the euclidean distance.

Color histograms Information about the general brightness and/or color distribution of an image can be easily captured in a histogram. Especially in medical retrieval, images created by a certain method have characteristic histograms. Consider for example X-Ray images or ultrasound images. Distances between histograms are calculated using the Jeffrey Divergence.

Tamura Features Tamura et al have proposed a set of six image features that are corresponding to human visual perception: coarseness, contrast, directionality, line-likeness, regularity and roughness [Tamura & Mori$^+$ 78]. FIRE uses the first three of them as they were found to be most important to model human perception.

Global Texture Descriptor Terhorst has proposed a model for describing texture properties of images [Terhorst 03] consisting of four parts:

- Fractal dimension: measures the roughness of a surface.
- Coarseensss: measures the grain size of the image.
- Entropy: describes the level of unorderedness.
- Spatial grey-level difference statistics: describes the brightness relationship of pixels within neighbourhoods.

Invariant Feature Histograms Invariant Feature Histograms are different from color histograms, because they are invariant to certain transformations, i.e. the histogram does not change when the image is translated, scaled or rotated.

3 Meta Information

This section is about the incorporation of meta information in FIRE. After a definition of meta information, the kinds of metadata for images, especially in the medical field, are described. Then, the implementation of the metafeature will be explained in detail.

3.1 What is meta information?

In general, meta information is "information about information". A typical example would be a library catalogue, which contains names, authors, publishers, etc. There are also many kinds of metadata for digital media, for example the meta-tags of web-pages which give common search terms for search engines. ID3-Tags for mp3-files contain information about the song, the artist, the genre, etc. Metadata for images produced by digital cameras is stored in the *Exchangeable image file format* (*Exif*) which is part of the image file. These Exif tags can store information about the time the picture was made, information about the model and settings of the camera, loaction information (if a GPS-device was used) and author and copyright information.

It can be seen that metadata provides an easy and effective means of searching and sorting any kind of data collection, no matter if it is the books of a library, an mp3 collection or an image corpus.

3.2 Meta information in medicine

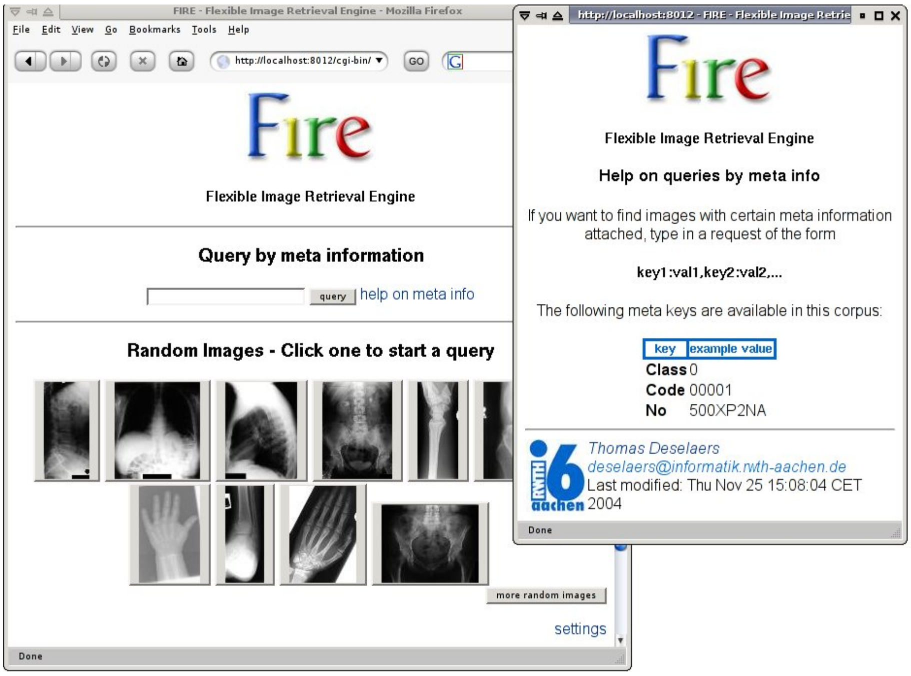
In medicine, images come from various sources, are produced using various imaging techniques and are often, but not always, associated to cases. They can be X-rays of a broken arm or microscopic images of bacteria cultures. Exif information is by far not enough to store the necessary information associated to medical images. A format is necessary that can at least store information about the patient (or other image subject), recording device and organization (if the image was made during examination or a study).

Therefore, in medicine, the *Digital Imaging and Communications in Medicine* (*DICOM*)[6] standard is widely used. It not only provides a much extended set of data fields for image metadata, but also includes a standard for sending and recieving image data. Many medical imaging devices and -software products conform to DICOM. In this standard, real-world objects (like the patient) are modelled as *information objects*. *Service classes* model different kinds of services that can be applied to information objects. Images are also treated as information objects and are associated to the corresponding patient and service. But due to the lack of complete specification of the implementation, several DICOM dialects have developed that may cause compatibility problems. Nevertheless, DICOM is widely accepted as a standard for storage and exchange of image data.

Another standard for metadata of medical images is the *IRMA-Code*[Lehmann & Schubert$^+$ 03], which is part of the *Image retrieval in Medical Applicatons* (*IRMA*) project[7]. The IRMA-Code is a 13-digit code that stores information about the following imaging modalities:

- Imaging Technique
- Imaging Direction
- Body Region
- Biological System

The IRMA-code is intended for classification of medical images and does not store patient- or treatment information, so it can only store a subset of the information that can be stored with DICOM, but is on the other hand more complete and less ambiguous.


FIRE - Flexible Image Retrieval Engine - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost:8012/cgi-bin/
GO
Fire
Flexible Image Retrieval Engine
Query by meta information
query
help on meta info
Random Images - Click one to start a query
more random images
settings
Done
http://localhost:8012 - FIRE - Flexible Image Retrie
Fire
Flexible Image Retrieval Engine
Help on queries by meta info
Done

If you want to find images with certain meta information attached, type in a request of the form

key1:val1,key2:val2,...

The following meta keys are available in this corpus:

key	example value
Class	0
Code	00001
No	500XP2NA

Thomas Deselaers
deselaers@informatik.rwth-aachen.de
Last modified: Thu Nov 25 15:08:04 CET 2004

Figure 2: The extended interface with metafeature support

3.3 Implementation of the metafeature

For the design of the metafeature in FIRE, we did not choose a certain standard for meta information, but tried to make the set of meta information as flexible as possible. Different kinds of image meta information had to be able to be easily adapted to work with the metafeature. A problem was the way in which the user should formulate a query. All features that FIRE used so far were direct features of the images, and so it was only possible to provide images as queries. With the metafeature, this should still be possible but one should also be able to search for images by only entering the desired meta information. So, we added a text field to the FIRE interface and defined a query sytax for the meta information. Via a help-link a popup-window appears and explains this syntax and lists the available meta tags. Pressing the search-button then starts a metafeature-based retrieval. That means all other image features are ignored and only meta information is used. To do a normal image-based query with metafeature-support, the user can just provide an image with meta information (for example by clicking on a random image on the bottom of the screen) and a FIRE retrieval that also uses the metafeature-distance as a distance measure is performed. A screenshot of the extended interface can be seen in figure 2.

In the following, the structure of the metafeature and the calculation of the metafeature distance, are described.

Data structure In general, meta information consists of several key-value pairs like:

```
patient Name: Hans, patient age: 50, ...
```

The choice for the basic data structure was a map of key-value pairs. Usually, within an image corpus that provides meta information, all images have the same set of keys, but with the metafeature it is also possible that an image only has a subset of the possible keys.

FIRE provides an abstract class for features, the *BaseFeature* class that defines the minimum functionality of a feature object. So, we derived the abstract feature class and created a new

[6] http://medical.nema.org/
[7] http://www.irma-project.org

class called *MetaFeature*. Feature classes have to provide a loading function that loads the feature information from a file and a value-function that returns the value of the feature. The loading function is called when FIRE loads the database, the value function is called upon a query, when the distance between two features must be calculated.

The file format for metafeatures is described in appendix A.

The creation of these files must be done beforehand. Converting existing meta-information to the described file format is very easy and can be automated using a scripting language. We made a small python script that converterts the IRMA-Tags to the metafeature format, so we could test the functionality of the metafeature on the IRMA image corpus.

Metafeature-distance The measurement of the dissimilarity of two metafeatures has to be very general and flexible, as the query can be an image or a set of user-specified key-value pairs. Both do not necessarily have to use all possible keys. The method we use iterates through all keys of the query image and counts the number of values that are different in the database image that is compared to it. This number is returned as the distance.

For the implementation, we derived the abstract class *BaseDistance* and called the new class *MetaFeatureDistance*. The only function that needed to be implemented was the distance function, that gets the query- and the database image as arguments and returns a distance value, that is calculated as described above.

Discussion With the metafeature, it is now possible to search in an image corpus more effectively by either providing an image with meta information or by entering the desired meta information by hand. The metafeature supports the image retrieval process by providing semantic information that can not be extracted from the image alone, or at least not by using today's methods of computer vision. In medicine, meta information is added by experts. The metafeature now makes this expert knowledge usable as a feature for image retrieval. This naturally improves the quality of the search results and makes it easier to find the needed images quickly and with a minimum of user input.

The implementation of the metafeature is very basic, but can be easily extended to allow for more complex queries. For example regular expressions for textual entries or comparative queries for numerical fields like "patients who are older than 60" could be implemented.

4 Text Information

This chapter and the following describe the main part of this work: The integration of information retrieval techniques in FIRE. First, the motivation for this is explained the use of textual information in image databases is described. Then, a short definition of information- and text retrieval is given and the retrieval techniques used in this work are described. The next chapter then describes how they were incorporated in FIRE.

4.1 Textual information in medical image databases

Medical image databases are not only intended for archiving patient data, but also for research and teaching in the field of medicine. Every image in such a database is associated to a certain case and patient or to a research project. Meta information suffices to store references to these, as well as basic additional information like date, imaging method, body part, etc. But there is information that neither the image nor its meta information contains, like:

- What exactly does the image show? This may sometimes be obvious from the image itself, but there are many things that are invisible to the untrained eye or that can not be noticed without knowing it. For example slight fractures in bones can be very hard to see on an X-ray image as they are often very thin.

- What are the symptoms? Which kind of pain or other problem does the patient have? What has the medical expert found to be unusual?

- What is the diagnosis? The diagnosis is made based on an anamnesis and examination by the medical expert which sometimes includes the use of medical imaging. So, the diagnosis cannot be seen from an image alone but needs further knowledge. Also, there may be symptoms that look exactly the same but have completely different causes.

- Comments from the medical expert. No disease is exactly like it's descibed in the literature. There are always special things and anomalies. These are of course of great interest for research

- Treatment history. Often, a number of images are made throughout a treatment to monitor the symptoms. Information about the development of the symptoms is therefore very important to record.

We see that textual information is inseparable from image data and that a large fraction of medical image data becomes meaningless without its associated textual descriptions. This is why more and more image retrieval groups are now starting to incorporate textual information in image retrieval. Even if an image retrieval engine may return the best possible set of images for a given query image, the results are still only selected by visual image similarity. Additionally using text retrieval helps finding images which are more suitable for the query even though they do not look like the query. For example, if an X-ray and a photo were made showing symptoms of the same disease, then pure visual image retrieval will only find other X-rays of this symptom, because a photo looks completely different from an X-ray image. Text retrieval, like metafeature supported retrieval, gives image retrieval a semantic aspect which can strongly improve the retrieval results.

4.2 What is text retrieval?

Text retrieval is a subfield of *information retrieval*, which is the art and science of searching for information in documents or for documents themselves. Text retrieval is focused on text documents. Commonly, the user places a query in form of some keywords and the text retrieval engine returns the documents that match his query best. Well-known examples of text retrieval engines are search engines on the world wide web. Text retrieval is an active field of research and there are many different approaches to this problem. The approach that we use is described in the following.

4.3 The text retrieval engine

The engine that we use has also been developed at i6. Further information can be found in [Macherey & Viechtbauer+ 03]. It implements the Smart-2 retrieval metric which is a derivation of the *term-frequency-inverse-document-frequency* (*tf-idf*) metric. It operates on a database of documents and accepts textual user queries for which it returns a list of the documents that are most suitable for the query according to the used metric. The exact procedure works as follows:

In an initialization step, the document database is preprocessed. This is done by first removing unimportant words, that means words that have a low semantic value. This step is called *stopping* and uses a list of so-called *stopwords* which are removed from the documents. This list contains the 319 most frequent words in the english language (lists for other languages also exist). In the second step, the remaining words are *stemmed*, that means they are reduced to their word stems, using an algorithm proposed by Porter[Porter 80]. The resulting terms in the document are the *index terms*, i.e. the terms by which a document is indexed. The query undergoes the same preprocessing.

Now, we define a weight that indicates the importance of a term t in a document $\mathbf{d}$. This weight is defined as follows:

$$g(t,\mathbf{d}) = \begin{cases} [1+\log n(t,\mathbf{d})]/[1+\log \overline{n}(\mathbf{d})] & if\ t \in \mathbf{d} \\ 0 & if\ t \notin \mathbf{d} \end{cases}$$

Here, $n(t,\mathbf{d})$ is the frequency of term t in document $\mathbf{d}$ and $\overline{n}(\mathbf{d})$ is the average term frequency in $\mathbf{d}$. For this equation, the logarithm is defined slightly different, because we set $\log(0) := 0$. We see that this weight is 1 if the frequency $n(t,\mathbf{d})$ is the same as the average term frequency $\overline{n}(\mathbf{d})$ and that it increases for higher frequencies of t. So, it emphasizes words that are used more frequently than others.

The final weights $w(t,\mathbf{d})$ are then obtained by the following normalization:

$$w(t,\mathbf{d}) = \frac{g(t,\mathbf{d})}{(1-\lambda)\cdot c + \lambda \cdot n_1(d)}$$

where $\lambda = 0.2$, $n_1(d)$ is the number of singletons in the document, i.e. the number of words that appear only once, and c is the average number of singletons in all documents.

For weighting the terms in the query, their *inverse document frequency* is first calculated. This value is lower the more often the word appears. The inverse document frequency is defined as:

$$idf(t) = \log\lfloor\frac{K}{n(t)}\rfloor$$

where t is a term, K is the number of documents and $n(t)$ is the number of documents in which t occurs. The term weights in the query are then calculated using:

$$w(t,\mathbf{q}) = [1\ +\ log\ n(t,\mathbf{q})] \cdot idf(t)$$

Now that we have weights for terms in both the query and the documents, we can calculate a ranking value that indicates the relevance of a document $\mathbf{d}$ for a query $\mathbf{q}$. This ranking is called *retrieval status value* (RSV) and is calculated by summing up the products of term-document weight and term-query weight for each term:

$$RSV(\mathbf{q},\mathbf{d}) = \sum_{t\in\mathbf{T}} w(t,\mathbf{q}) \cdot w(t,\mathbf{d})$$

Here, $\mathbf{T}$ is the set of all documents.

When a query is given, the text retrieval engine calculates the RSV for every document in the database and returns a list of relevant documents sorted by the RSV. The next section will now describe how this engine is incorporated in the retrieval process of FIRE.

5 Implementation

5.1 Requirements

Like with the metafeature, two kinds of retrieval should be possible with the textfeature. Text-based and image-based retrieval. For text-based retrieval, the textbox that was inserted into the web interface for the metafeature should be re-used. The user should be able to input a search string and the retrieval should only use textual information. For image-based retrieval the user should be able to specify an image that has textual information and FIRE should be able to do text-supported image retrieval.

As there are image corpora that provide image annotations in multiple languages, another requirement was the abilty to pose multilingual queries. Again, this should be possible by entering a set of queries, consisting of one query for each language, or by providing an image with multilingual textual annotations.

5.2 Connection

The first task was to connect FIRE to the text retrieval engine. Clearly, this should again be realized via a new feature for FIRE, the *textfeature*. The most flexible way to communicate with the text retrieval engine is the use of a network socket. This allows for decentralization of the text retrieval by executing the engine on a different machine and communicating over a LAN, or even the internet. As the current text retrieval engine only had a command-line user interface and lacked networking capabilities, we implemented a simple line-based protocol for network communication. Using this protocol, the client (that means FIRE, but other clients are of course also possible) sends a query, which can either be a query string or the name of a file that contains the query. The text retrieval engine then returns a list of documents, sorted descending by their RSV. This list then has to be processed by FIRE to incorporate it in the image weighting.

5.3 Feature and distance

As with every feature in FIRE, the information for the textfeature is given as a file that accompanies the according image file, so every image file has a textfeature file, or *TextID-file*. This file does not directly contain the textual description of the image, but the path to the file containing the description. The reason for this is that, especially in medical databases, multiple images can belong to one case, that means they share the same textual description. So, in order not to store duplicate information, we decided to use this approach. This also means that the only information that a TextFeature object for a particular image has to store is the filename of the TextID-file, as it is a unique identifier of the textual information of the image.

When FIRE is now queried, the TextDistance class is first called to do a text-based retrieval. If the query is text-based only, the text-query is forwarded to the text retrieval engine via the network socket. If it is an image-based query, the TextID-Information of the query, i.e. the filename containing the text information of the query image is sent to the text retriever, so the query text is loaded from this file. This is of course a much more precise query, as the complete annotation of an image has much more words than a typical user-made text query. A list of RSVs and TextIDs is returned by the text retriever and is saved in an internal map in the TextDistance object of the query image.

Now, FIRE compares every image with the query image by querying the distance of each feature. That means that the RSV has to be converted to a distance-value that is high for dissimilar documents and low for similar ones. This conversion is simply done by using the maximum RSV that was returned by the text retriever and subtracting every other RSV from it. That means the new distance of a document $\mathbf{X}$'s text-feature $\mathbf{X}_m$ to the query's text-feature $\mathbf{Q}_m$ is:

$$\mathbf{d}_{\text{text}}(\mathbf{Q}_m, \mathbf{X}_m) = \begin{cases} RSV_{\text{max}} - RSV_X & \text{if } \mathbf{X} \text{ is in the list of relevant documents} \\ \rho & \text{else} \end{cases}$$

Setting $\rho := RSV_{\max}$ has proven to be a reasonable choice. The fact that this weights all irrelevant documents as much as the least relevant document is a slight inaccuracy, but is practically not noticable. The retrieval result even deteriorates when setting it any higher.

This distance value is returned to FIRE and used in the retrieval process as described in 2.1.

5.4 Multilingual querying

Various corpora provide multilingual text annotations for their images. Throwing all texts together and using them in one text retrieval engine is certainly not the right choice for doing multilingual retrieval as the preprocessing of the textretriever is specially adjusted to a single language, because the stopword list and the stemming algorithm are language specific. Also words that occur with different meanings in different languages may disturb the search results. And even if no such word existed, a query in one language would only produce results in one language. A better choice was using multiple text retrievers, one for each language and using the same number of textfeatures. So, every image has multiple TextID-files that each contain the filename of the textual information in the corresponding language. A text query is sent to all text retrievers and their results are weighted according to the weighting settings of FIRE. For being able to do multilingual text-based queries, we intruduced a syntax of the form:

```
Ge:"Schädel" Fr:"crâne" En:"skull"
```

This syntax can be used to give queries in different languages, but not all languages have to be given. Omitted languages will be ignored. If the image database has only monolingual annotations, then the query can still be entered without using this syntax.

5.5 Discussion

We have implemented a way to use text retrieval techniques in FIRE. Queries can be done by either entering a search request into a text field or by providing an image with textual information. Our solution allows for parallelization and easily scales to multiple languages.

We have already given several reasons why text retrieval is a very valuable add-on for content-based image retrieval. The next section will show that also in practice, retrieval accuracy is greatly improved by the textfeature.

6 Databases & Results

Evaluation of image retrieval systems is a very hard task. Practice-oriented image corpora and tasks must be provided along with realistic sample queries, and an evaluation of the relevance of the returned results must be done. The *Cross-language evaluation forum* (*CLEF*) is a campaign that evaluates information retrieval systems as well as image retrieval systems. The part of the CLEF that evaluates image retrieval systems is called *ImageCLEF*. Every year, several tasks from different fields are provided. The combination of information retrieval techniques with image retrieval is particularly focused on.

The image retrieval group of i6 had successfully participated[Deselaers & Keysers$^+$ 04] in two tasks of the ImageCLEF evaluation of 2004[Clough & Müller 04]: An ad-hoc image retrieval task using the St. Andrews photographic database[8] and a medical retrieval task on the Casimage medical image database[9]. Both databases include textual image annotations. In the Casimage database, each image has English or French annotations, the images of the St.Andrews database all have English annotations. Even though at this time, FIRE had only visual retrieval capabilities, the results were comparable to the results of other groups. In 2005, we participated again[Deselaers & Weyand$^+$ 05], now using the results of this work to combine content-based image retrieval with text retrieval. We took part in two tasks, which were an automatic annotation task and a medical image retrieval task. While in the automatic annotation task, no textual information was given, because the task was to assign each image an annotation automatically, the medical image retrieval task included textual information. We will now describe the task in detail, explain the experimental settings we used and discuss the results of the evaluation. After that, we will describe my re-runs of the St. Andrews bilingual retrieval task from ImageCLEF 2004 using text retrieval and compare the results to the original submission to ImageCLEF 2004.

6.1 ImageCLEF 2005 - medical retrieval task

For the medical retrieval task task, a large database of about 50,000 medical images from four different databases was provided. Textual annotations are mostly English, but German and French annotations also exist. The task was to automatically retrieve the most relevant images in the database for a given set of 25 queries. A query consists of one to three images and a textual request like "Show me hand-drawn images of people". Figure 3 shows three of the provided queries. The list of relevant images for each query had to be returned by each of the participants and was evaluated using reference results created by medical experts. The measure for the overall score of a submission was the *mean average precision* or MAP.

Mean average precision As the name suggests, the MAP is calculated by averaging twice. First, the precisions within one result are averaged and then these vales are averaged over all results. The precision of the retrieval result for one query is calculated taking into account that the result is sorted with respect to the relevance of the returned images, that means that the most relevant image comes first, the second most relevant image comes second and so on. So, the precision which is defined as:

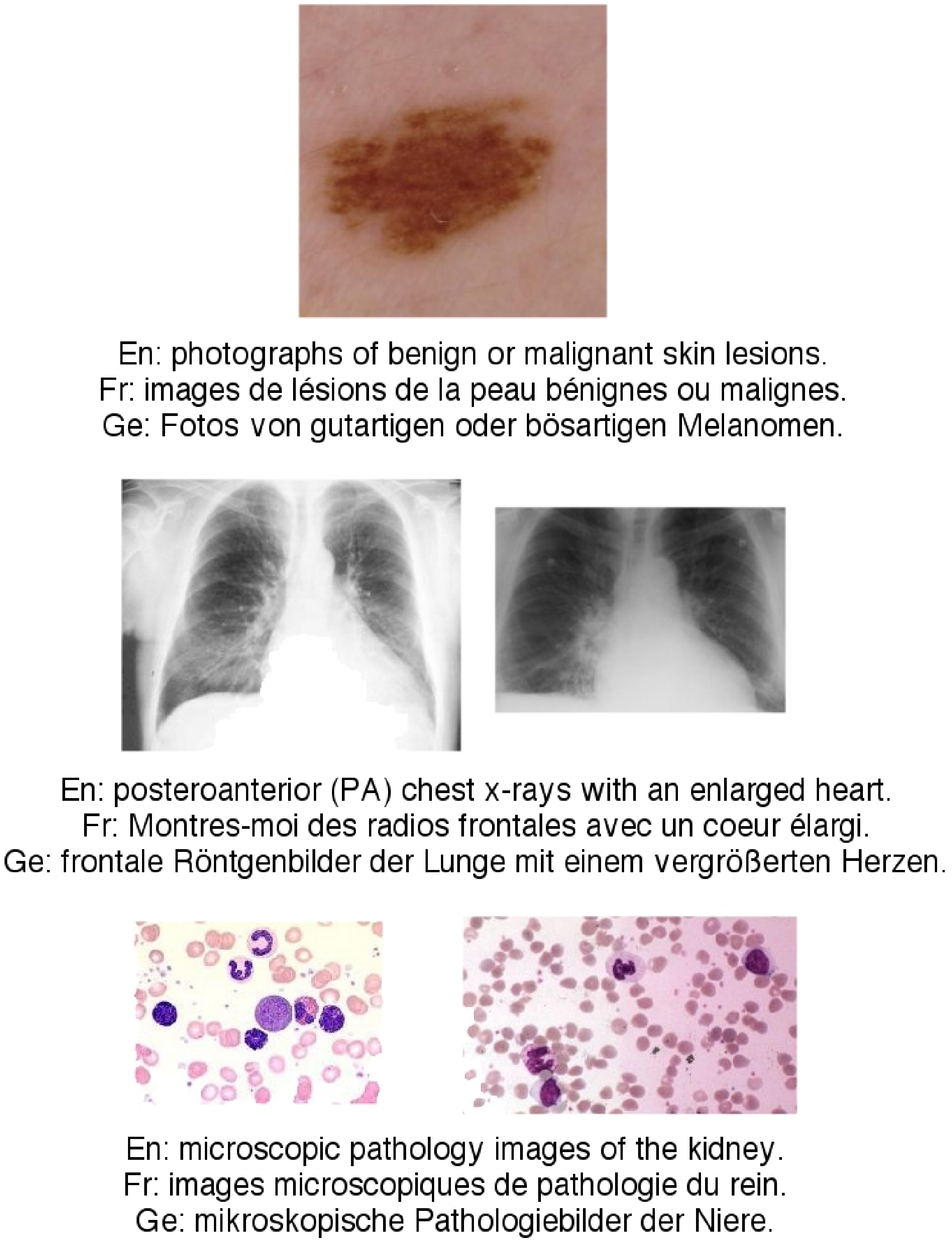
$$p = \frac{\text{number of relevant documents retrieved}}{\text{number of documents retrieved}}$$

is not sufficient to measure, how good a system can estimate the relevance of images for a query, because all retrieved documents are equally weighted. The average precision uses a modified version of the precision, $p(r)$ which is the precision that considers only the first r retrieved documents. The average precision is defined as:

$$AvgP = \frac{\sum_{r=1}^{N} p(r) \cdot rel(r)}{\text{number of relevant documents}}$$

[8] http://specialcollections.st-and.ac.uk/photo/controller

[9] http://www.casimage.com/



En: photographs of benign or malignant skin lesions.
Fr: images de lésions de la peau bénignes ou malignes.
Ge: Fotos von gutartigen oder bösartigen Melanomen.

En: posteroanterior (PA) chest x-rays with an enlarged heart.
Fr: Montres-moi des radios frontales avec un coeur élargi.
Ge: frontale Röntgenbilder der Lunge mit einem vergrößerten Herzen.

En: microscopic pathology images of the kidney.
Fr: images microscopiques de pathologie du rein.
Ge: mikroskopische Pathologiebilder der Niere.

Figure 3: Some example queries from the ImageCLEF 2004 medical retriecal task

Here, $rel(r)$ is a binary function that is 1, if the document at rank r is relevant, 0 else. For example, let there be only one relevant image and let the number of retrieved images be two. If the relevant document has rank 1, then:

$$AvgP = \frac{\frac{1}{1} * 1 + \frac{1}{2} * 0}{1} = 1$$

If the relevant image has rank 2, then:

$$AvgP = \frac{\frac{0}{1} * 0 + \frac{1}{2} * 1}{1} = 0.5$$

We see that, in contrast to the precision (which is 0.5 in both cases), the average precision also uses the relevance information given by the rank of the image. The average precision is 1, if the retrieval result only contains relevant images and all relevant images were found, and it is 0, if no relevant image was found. The mean average precision is just the average of the average precisions of all query results. It is an important measure for all kinds of retrieval tasks and is used as the performace criterion in ImageCLEF and in TREC[10]. The performance values we discuss later are also mean average presisions.

The setup for our submission The setup of FIRE for the medical retrieval task contained the features described in section 2.1 plus three textfeatures for the three languages. Totally, we submitted ten runs with different feature sets and -weightings:

Textual information only First experiments with multilingual textual information showed, that the retrieval precision when using all three languages was worse than the retrieval precision when only the English annotations were used. This was probably due to the fact that most of the annotations were in English and only few images had German or French annotations. The query text was given in all three languages, but the most results were of course returned from the English retriever. The retrievers for the other languages had a database of very few images, and thus less relevant documents for the respective queries. The problem is, that the RSVs are not absolute values, but that they depend on the given text corpus. This means, the RSVs and thus the scores, which are calculated as described in 5.3, are not directly comparable. Mixing the scores of different retrievers led to noisy overall scores which produced bad results. A way around this problem was to combine the distances of the textfeatures by using the minimum distance for each image. This way, the problem with the uncomparable scores still has effects, but they are less drastic. Also, when the description of an image in one language did not match the query, but another one did, the image was still considered relevant. This run was called `EnDeFr-min`.

For second run that only used textual information, we chose to only use the English annotations, since they were the by far biggest part of the annotations and we wanted to test whether the use of the other languages would support the results of the English retriever or if they would disturb them.

Visual information only Three runs using only visual information (`5000215`, `0010003`, and `1010111`) were submitted, which differed in the weightings of the features. These weightings were determined based on experiments and previous experiences.

Textual and visual information For the combination of textual and visual information, we again decided to choose the minimum of the three retrieved text distances and treat this as a single distance that is then combined with the visual distance information. Two different runs were submitted, again with different weightings. In the run `3(1010111-min(111))`, all visual distances and the combined text distance had a weighting of three and in the run `3(3030333)-min(111)`, the image distances had a weighting of nine and the combined text distance was weighted with one.

[10] http://trec.nist.gov/

Table 1: Runs submitted to the medical retrieval task together with feature weightings and achieved MAP with wrongly chosen ρ and with properly chosen ρ. * means that the minimum among all lines marked with * in this column was taken and weighted by 1.

	textual information only		visual information only			visual and textual information				+relevance feedback	
run	`En`	`EnDeFr-min`	`1010111`	`5000215`	`0010003`	`3010210111`	`3(3030333)-min(111)`	`3(1010111)-min(111))`	-	`vistex-rfb1`	`vistex-rfb2`
X×32 image features	-	-	1	5	3	3	9	3	1	1	1
32×32 image features	-	-	1	0	0	0	0	3	1	1	1
color histograms	-	-	1	0	1	1	9	3	1	1	1
tamura features	-	-	1	2	0	2	9	3	1	1	1
invariant feat. histo.	-	-	1	1	0	1	9	3	1	1	1
English text	1	*	-	-	-	1	*	*	2	*	*
German text	0	*	-	-	-	1	*	*	0	*	*
French text	0	*	-	-	-	1	*	*	0	*	*
relevance feedback	-	-	-	-	-	-	-	-	-	+	+
score w/ wrong ρ	0.21	0.05	0.07	0.06	0.05	0.07	0.07	0.06	-	0.09	0.08
score w/ properly chosen ρ	0.21	0.15	0.07	0.06	0.05	0.22		0.20	0.25		

For testing, we also submitted a run where the text features were combined in the usual way (`3010210111`).

Because manual feedback was allowed, we submitted two runs with manual relevance feedback for each query (`vistex-rfb1` and `vistex-rfb2`).

Results Unfortunately, we made a mistake in our submission. Because we did not test the combination of textual and visual features properly enough, we set an important parameter to a very bad value, namely the parameter ρ, descibed in section 5.3, which defines the distance of a query text to a text that was not considered relevant by the text retrieval engine. By setting $\rho = 10,000$ we wanted to ensure that irrelevant documents had a much greatuer distance than relevant ones. But this led to the effect that by the normalization of the feature distances (cf. section 2.1), the distances of relevant documents were normalized to a value near zero, because the overall sum of the distances became very high as there were much more irrelevant documents than relevant ones.

The results of the evaluation are given in table 1.

The effects of this error can be seen from the table, especially on the runs `3010210111` and `3(1010111)-min(111))`. This made our system look worse in the competition than it should have, because the results with $\rho = RSV_{\max}$ would have achieved a higher rank. Nevertheless, the results we produced afterwards show that pure text retrieval already outperforms pure image retrieval by a MAP factor of about three. Another interesing thing to note is that the multilingual text retrieval performs signficantly worse than the English-only run. This is probably due to the extremely different amounts of annotations of the given languages, the resulting uncomparable RSVs and the quality of the annotations. So, the German and French textfeatures added more noise than information to the images.

Using the combination of visual features and text features, the best score was achieved using a uniform weighting for the image features and again excluding German and French from the text retrieval. This resulted in a MAP of 0.25. The other combined runs that used all languages and different image feature weightings achieved scores of 0.22 or 0.20, which is comparable to pure text retrieval with English only. Again, these scores can be explained by the distortion that is caused by mixing the languages.

As expected, manual relevance feedback helps to improve the scores, but does not increase

Table 2: Results on the St. Andrews image corpus using the textfeature

run	MAP
textual information only	0.0757
visual information only	0.0861
textual and visual information	0.1564

the score too much, because the feedback was only given on the first 20 results and was done by computer scientists with only little knowledge of medicine.

In the overall ranking, our text-only runs performed quite well. The English-only run achieved rank three with a MAP of 0.2065. The first and second ranks were slightly better with MAPs of 0.2084 and 0.2075, respectively. The Multi-language run ranked 5th out of 9.

Our visual-only retrieval got only average scores, with our best result being 0.0713. The best MAP in the competition was 0.1455.

The most interesting results were of course the results for the combination of textal and visual retrieval. Without our fatal error, our best score would have been 0.25, which would have achieved place three in the competetion.

6.2 ImageCLEF 2004 - St. Andrews bilingual retrieval task

To further evaluate our approach of combining text retrieval and image retrieval, we ran further experiments on the St. Andrews historic photographic collection, a database of 28,133 historic images, most of which are landscape photographs, pictures of monuments or portraits. This database was provided for the ad-hoc retrieval tasks of ImageCLEF 2004 and 2005. After the evaluation of ImageCLEF 2004, a file with the relevant images for the given queries was released by the organisers, so groups could evaluate their systems themselves after the competition. The original task was to find the most relevant images for a given textual topic. This means that the first retrieval step had to be a purely textual one and CBIR techniques could only be used in the following step(s). For each topic, a sample image was given which could be used for query expansion. Since the kind of mixed retrieval that was necessary for the task was currently not possible with FIRE, we decided to combine the topics with the sample images and use these combinations as queries. This of course makes our results uncomparable to the results of the participants of the evaluation, but the provided list of relevant images is still valid and we could use it to compare pure text retrieval, pure image retrieval and the combination of the two.

Since the image retrieval group of i6 participated in the task of 2004 with visual information only (the example images were used as queries), the visual features for this database already existed and only the textual information had to be prepared to be compatible with the text retrieval engine. The results of our experiments as well as results of other groups are given in table 2.

The score for the run that only uses visual information is the same as the score that was achieved last year by our group. We see here that, in contrast to the results of ImageCLEF 2005, the use of textual information only performs worse than the run with visual information only. This shows the large difference between the two retrieval tasks. Retrieval on the St. Andrews database is much easier for image retrieval systems. For example if, on a medical corpus, an X-ray image that shows a human torso with a slight fracture in a rib is used as the query, then the retrieved images are only relevant if they show a similar fracture. But because such a fracture is very small and can hardly be noticed by untrained humans, an image retrieval system will perhaps find other X-rays taken from the same perspective, but the slight fracture which is the criterion of significance will not be ”noticed” by the system and the retrieved images will only accidentially show rib fractures. This is because the features which are used for image retrieval are not appropriate for the task. With histogram features, an image retrieval system will easily find other X-ray images. Apprearance based features and texture features will help finding images taken from a similar perspective. But unless a highly complex medical image analysis feature is implemented, retrieval of coarsely similar images will be the best that pure general image retrieval can do in medicine.

While in medical imaging, the only ”motive” is the human body, in a photographic database, the variety of different images is much larger. There is a much greater variety in histograms, different kinds of textures, contrasts, etc. This already makes image retrieval much easier. But the most important point is, that relevant images are easier to find, because the motives are far more obvious. If the query is a picture of a church, then the relevant images will be pictures of churches and not only pictures of churches with the same kind of crack in a certain stone. This is why a general image retrieval system like FIRE performs much better on a photographic corpus.

The retrieval score of the combination of text retrieval and image retrieval is nearly twice as high as the scores for each of the retrieval methods. This again shows the potential of this combination, even on a completely different database. Apparently, the mistakes that one of the methods makes can be corrected by the other method. For example, if an image of a house is the query, then image retrieval will probably find other houses. But if the textual annotation of the query says that this house is, say, the buckingham palace, then text retrieval will help finding other images with "buckingham palace" in the description. On the other hand, if the query is a portrait of the queen and the textual annotation of the query says "Queen Elizabeth", then text retrieval will find pictures of the queen herself, but also of several things related to the queen like her house, her car, etc. Image retrieval can now help to find only images which are visually similar to the query image, that means showing the the portrait of a person, which will likely be the queen.

Generally, we see that combining image retrieval and text retrieval is a large step in image retrieval and that the respective techniques can help each other to find the most relevant images in an image database.

7 Summary and review

For this study project, two new features for FIRE were implemented that enable it to retrieve images based on meta information and textual annotations. The metafeature implements a very general key-value concept, so that any meta information format should be easily converted to be used in FIRE. Retrieval based on meta information can be done using a text field in the GUI of FIRE. Also, the metafeature can be used as an addition for conventional image retrieval.

The textfeature creates a link between FIRE and a text retrieval system. Now, an image database can be searched by using textual annotations of images. Again, this can be done in two ways. The user can either enter a query text into the web interface of FIRE and do a search that is only based only on text retrieval or the user can provide an image with textual annotations and start a normal image retrieval with text support. Comunication between FIRE and the text retrieval engine was realized via a network socket, which makes it possible to run the text retrieval engine on a remote system. Also, it would be possible to connect other text retrieval engines to FIRE by implementing this protocol in them. Because multiple features of the same kind can be used in FIRE, multi-lingual text retrieval can be realized by using multiple textfeatures that each connect to a text retriever that uses the same corpus in a different language.

Evaluations of the retrieval using both image and text information showed that very good retrieval results can be achieved and that the combination of both media produces much higher MAPs than the respective retrieval techniques alone. We could achieve results that would have ranked 3rd in the medical retrieval task of the ImageCLEF 2005 evaluation, which shows that our system is comparable to other state-of-the-art retrieval engines.

Though the work on the features is finished, a problem that still requires further research was found in our ImageCLEF 2005 results. We still have not found a suitable way to combine the results of text retrievers that use multiple languages, especially when the amount of annotations differs strongly between the languages. This led to a distorsion of the retrieval results and to a multilingual score that was worse than the monolingual one.

Nevertheless, the incorporation of information retrieval in FIRE was a success. Cross-media retrieval is an important trend that is supported by organizations like ImageCLEF and that is currently followed by many groups who are investigating techniques for image retrieval as well as text retrieval. Though image retrieval techniques are still evolving, the combination with text retrieval is a necessary step, especially regarding practical applications like searching for images in clinical PACS systems, where pure image-queries are often not enough to find the desired information.

A Metafeature file format

The file format for the metafeature follows a simple key-value scheme:

```
<key1>: <value1>
<key2>: <value2>
(...)
```

For each image, a metafeature-file must be provided. In an image database, the metafeature files do not have to use same set of keys. So, tags that do not apply to certain image can be omited.

B Text retriever network protocol

The protocol for communication between FIRE and the text retriever is very simple. There are four kinds of commands that FIRE can send to the textretriever. Except for the simple query, all commands start with a hyphen.

B.1 Retrieval commands

A simple text retrieval is started by just sending the query text without any further commands. An example would be:

```
cat sitting in front of a house
```

Another way of querying is by specifying a query file. The contents of this file will then be used as query text, after XML tags have been removed. An example query would be:

```
:qfile Query.xml
```

The argument to `qfile` can either be a full path or just the filename. In the latter case, the full path is searched in the file list of the textretriever. So, if the query file is not in the database, the full path has to be given. The textretriever returns an ordered list of relevant documents with their respective RSVs. Before the actual list, it sends the number of found documents, so FIRE knows when to stop listening. An example would be:

```
26
Doc3567.xml 0.0528
Doc7347.xml 0.0519
Doc3668.xml 0.0417
(...)
```

B.2 Other commands

```
:bye
```

This ends the communication between FIRE and the textretriever, but the textretriever stays alive and waits for new connections.

```
:kill
```

This ends the communication and then kills the textretriever. `:kill` should only be called when the textretriever is no longer needed.

C Extensions to the FIRE prtocol

Several commands were added to the FIRE protocol that are used by the extensions of the web interface.

C.1 Metafeature commands

Retrieval only based on the metafeature (image features are ignored) is started with the `metaretrieve` command:

```
metaretrieve <metaquery>
```

`metaquery` has the format described in 3.3. FIRE then returns the n images that best match the query, where n is given in the settings of FIRE. To display the next n images, i.e. the next page of results, the `metaexpand` command is sent to FIRE.

```
metaexpand <metaquery> <resultsstep>
```

`resultsstep` specifies which page is requested.

```
metafeatureinfo
```

For showing the user the available keys and example values for these keys in the metafeature help window, the webinterface needs to get this information from FIRE with the `metafeatureinfo` command which has no arguments. FIRE then searches all the metatags in the image database to find the available tags together with example values and returns them as a string of the following form:

```
<key1>:<value1> <key2>:<value2> (...)
```

C.2 Textfeature commands

The commands for text retrieval are analogous to the commands for metadata-based retrieval.

```
textretrieve <textquery>
```

This starts a pure text-based retrieval on the database. If the textual information for the images is given in only one language, `textquery` is just the query string itself. If there is textual information in several languages, the syntax from 5.4 is used.

```
textexpand <textquery> <resultsstep>
```

Like described above, the next page of results can be retrieved with `textexpand` which takes a text query and the index of the page to be displayed as arguments.

References

[Alvarez & Oumohmed$^+$ 04] C. Alvarez, A. I. Oumohmed, M. Mignotte, J.-Y. Nie. Toward Cross-Language and Cross-Media Image Retrieval. Proc. *Multilingual Information Access for Text, Speech and Images. Proceedings of the 5th Workshop of the Cross-Language Evaluation Forum. CLEF 2004*, Vol. 3491 of *LNCS*, pp. 676–687, Bath, UK, September 2004. Springer.

[Carson & Belongie$^+$ 02] C. Carson, S. Belongie, H. Greenspan, J. M. k. Blobworld: Image Segmentation Using Expectation-Maximization and its Application to Image Querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 8, pp. 1026–1038, Aug. 2002.

[Clough & Müller 04] P. Clough, H. Müller. The CLEF Cross Language Image Retrieval Track (ImageCLEF) 2004. 2004.

[Deselaers & Keysers$^+$ 04] T. Deselaers, D. Keysers, H. Ney. FIRE – Flexible Image Retrieval Engine: ImageCLEF 2004 Evaluation. Proc. *Working notes of the CLEF 2004 Workshop*, pp. 535–544, Bath, UK, Sept. 2004.

[Deselaers & Weyand$^+$ 05] T. Deselaers, T. Weyand, D. Keysers, W. Macherey, H. Ney. FIRE in ImageCLEF 2005: Combining Content-based Image Retrieval with Textual Information Retrieval. Proc. *Working Notes of the CLEF 2005 Workshop*, 2005.

[Deselaers 03] T. Deselaers. Features for Image Retrieval. Diploma thesis, Lehrstuhl für Informatik VI, RWTH Aachen University, Aachen, Germany, Dec. 2003.

[Lehmann & Güld$^+$ 04] T. Lehmann, M. Güld, C. Thies, B. Fischer, K. Spitzer, D. Keysers, H. Ney, M. Kohnen, H. Schubert, B. Wein. Content-based image retrieval in medical applications. *Methods of Information in Medicine*, Vol. 43, pp. 354–361, 2004.

[Lehmann & Schubert$^+$ 03] T. Lehmann, H. Schubert, D. Keysers, M. Kohnen, B. Wein. The IRMA code for unique classification of medical images. Proc. *Proceedings SPIE 2003*, pp. 109–117, 2003.

[Lin & Chang$^+$ 04] W.-C. Lin, Y.-C. Chang, H.-H. Chen. From Text to Image: Generating Visual Query for Image Retrieval. Proc. *Multilingual Information Access for Text, Speech and Images. Proceedings of the 5th Workshop of the Cross-Language Evaluation Forum. CLEF 2004*, Vol. 3491 of *LNCS*, pp. 664–675, Bath, UK, September 2004. Springer.

[Macherey & Viechtbauer$^+$ 03] W. Macherey, H.-J. Viechtbauer, H. Ney. Probabilistic Aspects in Spoken Document Retrieval. *EURASIP Journal on Applied Signal Processing*, Vol. Special Issue on "Unstructured Information Management from Multimedia Data Sources", No. 2, pp. 1–12, Feb. 2003.

[Müller & Geissbühler$^+$ 05] H. Müller, A. Geissbühler, P. Ruch. Report on the CLEF Experiment: Combining Image and Multi-lingual Search for Medical Image Retrieval. Proc. *CLEF Proceedings - Springer Lecture Notes in Computer Science*, 2005.

[Porter 80] M. F. Porter. An algorithm for suffix stripping, July 1980. Programm.

[Squire & Müller$^+$ 99] D. M. Squire, W. Müller, H. Müller, J. Raki. Content-Based Query of Image Databases, Inspirations from Text Retrieval: Inverted Files, Frequency-Based Weights and Relevance Feedback. Proc. *Scandinavian Conference on Image Analysis*, pp. 143–149, Kangerlussuaq, Greenland, June 1999.

[Tamura & Mori$^+$ 78] H. Tamura, S. Mori, T. Yamawaki. Textural Features Corresponding to Visual Perception. *IEEE Transaction on Systems, Man, and Cybernetics*, Vol. 8, No. 6, pp. 460–472, June 1978.

[Terhorst 03] B. Terhorst. Texturanalyse zur globalen Bildinhaltsbeschreibung radiologischer Aufnahmen. Research project, RWTH Aachen, Institut für Medizinische Informatik, Aachen, Germany, June 2003.

[van Zaanen & de Croon 04] M. van Zaanen, G. de Croon. Multi-model Information Retrieval Using FINT. Proc. *Multilingual Information Access for Text, Speech and Images. Proceedings of the 5th Workshop of the Cross-Language Evaluation Forum. CLEF 2004*, Vol. 3491 of *LNCS*, pp. 728–739, Bath, UK, September 2004. Springer.