PIDALION: A Reconfigurable Agent-based Multimedia Search Engine Platform

ANASTASIOS DOULAMIS¹, and KLIMIS NTALIANIS²
1: Technical University of Crete, Decision Support Lab, Chania, Polytechnioupolis, GREECE,
2: National Technical University of Athens, Image, Video and Multimedia Lab. Zografou, 157 73, Athens, GREECE

Abstract: - Fuelled by the rapid expansion of broadband connectivity and increasing interest in online multimedia-rich applications, the growth of digital multimedia content has skyrocketed. Among others, this growth is compounding the need for more effective methods for searching multimedia information. The automated web search engines that are currently used rely only on text descriptions and as a result provide matches of poor quality in case of multimedia content. The services of a multimedia search engine are therefore a possibility that the internet users still lack. The scope of this paper is thus to present an implementation approach for a personalised web-based multimedia search engine in the Java programming language. This approach combines the characteristics of the current search engines as well as new innovative features which guarantee at the same time the system’s quick response and better search results. In this paper, the reader can find an analytical presentation of all the components required to form a multimedia search engine, as well as indications on how to implement key algorithms and functions.

Key-Words: - Leave multimedia content, queries, content-based retrieval, multimedia crawler, metadata, image histogram, hierarchical presentation

1 Introduction
The web creates new challenges for information retrieval. The amount of information on the web is growing rapidly and so is the number of new users inexperienced in the art of web research. It is estimated that 1-2 Exa-Bytes (millions of Tera-Bytes) new information are created each year over the Web. This huge amount of information is anticipated to grow by a factor of 10 in the following two years.

Automated search engines that rely on keyword matching usually return too many low quality matches. The situation is worse as far as multimedia content is concerned. The most popular search engine, Google relies only on keywords to search for images and does not contain any information on semantic content. Content-based image retrieval systems (CBIR) try to solve this problem. Many CBIR systems have been recently proposed and implemented in the literature, such as the QBIC system [1], the PicToSeek [2], and the Virage [3]. In [4], 3D searching is discussed, while applications of content-based retrieval systems are examined in [5]. Finally, personalized retrieval is examined in the work presented in [6].

Marvel is the latest and more intelligent content-based search engine. It was developed by the research centre of IBM in USA in 2004 [7]. It tries to increase the retrieval precision accuracy by incorporating semantic annotation in the media volumes. However, all the adopted approaches have static and local access only to the system’s database and thus cannot retrieve content from the web. Furthermore, the aforementioned works focus on the algorithms for efficient content-based retrieval and not on the practical issues regarding the implementation of a large scale multimedia search engine over the Web.

So far, several different techniques for making distributed multimedia content searchable have been proposed. In [8] there is information on the techniques of checking the outgoing links, analyzing the referring page, mining for textual information in the media file and utilizing metadata using the Dublin Core metadata model or the MPEG-7 standard. Among the before-mentioned methods only the MPEG-7 standard was designed for multimedia search.

The effort in this paper focuses on describing a multimedia search engine that combines features from existing search engines and enhances their functionalities through innovative algorithms and mechanisms. The goal is not only to describe the system’s architecture and interconnectivity, but also
to explain how the algorithms can be implemented in Java code. This system, called PIDALION, runs on Windows environment, while the Java and Java Servlets technologies are adopted to ensure the system's interoperability and dynamic behaviour. The PIDALION search engine constitutes a fully functional and commercial solution.

One of the key features of the proposed search engine is the provision of fully personalised retrieval services; users of PIDALION may have the possibility to share their personal content either with all web users or within the frame of groups, as well as maintain a personal profile, where their preferences are stored. Personalised retrieval can be achieved through the creation of social groups and the use of dynamic relevance feedback mechanisms, which tailor the system performance to the current user’s preferences.

This paper is organised as follows: Section 2 presents the system’s architecture, explaining the role of each main component. Sections 3 to 7 present the functionality, architecture and key features-innovations of each component. Key algorithms are depicted in the form of pseudo-code. Finally, in Section 8 the issues covered in this paper are summarised, final conclusions are made and future expansions are proposed.

2. System Overview
The web is dominated by various search engines, such as Google, Yahoo and AltaVista. The main disadvantage of these systems is that they rely only on text descriptions and provide results which are only characterised by given keywords. This means that when the user wishes to find images or video files, the results hardly ever satisfy his/her search criteria. In other words, there are no mechanisms that enable users to look inside the data.

In Google, the web crawling is carried out by several distributed crawlers. A URL server sends to the crawler lists of URLs to be fetched. The web pages that are fetched are afterwards sent to the store-server which then compresses and stores them into a repository. In order to scale to hundreds of millions of web pages, a fast distributed crawling system is required. Still problems arise when apart from web pages, multimedia content has to be fetched, since the volume of transmitted data is drastically increased. This is addressed by implementing intelligent crawlers which process multimedia volumes in compressed domain and compact multimedia content through the dynamic extraction of thumbnails. In [9], the authors present a model for intelligent web crawlers, which use a probabilistic model to enhance their crawling behaviour. Moreover, as proposed in [10], ontologies could be used to determine the relevance between links and referenced web pages. However, these approaches do not focus on multimedia crawling.

Similar problems exist in the area of content presentation (visualization). All current search engines display serially results organised in teams of 10-100 elements. This however makes it too difficult and time-consuming for the user to detect the content of his interest. In case of multimedia content, such an approach would be not just unsatisfactory but also unacceptable, since the user would have to browse through hundred of thumbnails in order to find the content of his interest. Instead, a hierarchical approach is needed, to ensure that content is grouped based on its similarities and help the user limit his search range, given that he will seek further only in those groups that appear to contain content of his liking.

The PIDALION system improves the functionality of the above components in the directions of 1) applying a crawling mechanism so as to locate and draw multimedia content, 2) enabling similarity-based and real-time retrieval, 3) appropriately organizing the system's database in order to support multimedia metadata storing and real time response to multimedia queries, and 4) adjusting the system's user interface so as to support hierarchical multimedia display (easy and non-linear multimedia content navigation).

The described platform consists of the following subsystems:

The multimedia crawling subsystem, whose role is to index multimedia content and handle the function of updating the indexing process

The multimedia metadata subsystem, which extracts metadata from multimedia content, according to the MPEG-7 descriptors achieving in this way interoperability

The retrieval and display subsystem, which is responsible for searching multimedia content that matches specific criteria in the database and forwarding it to the interface subsystem.

The interface subsystem, which enables the interaction and communication between the user and the system, provides a functional projection of retrieved content and allows the composition of complex queries

The multimedia database subsystem, which covers the necessity of storing large amounts of metadata and thumbnails, as well as of keeping
additional information which derives from the previous subsystems.

Interconnection between the subsystems and the multimedia database is achieved through JDBC. The way in which the above subsystems interact and cooperate with each other is shown in Fig. 1.

![Fig. 1: Interconnection between subsystems.](image)

### 3. Multimedia Crawling

This subsystem is responsible for locating and indexing multimedia content. The architecture analysis of the multimedia crawling subsystem owes to cover the issues of detecting new web pages and storing the relevant information in the system’s database. In the frame of PIDALION, multimedia crawling is achieved by using Java sockets to transfer web pages and multimedia content. The architecture and functionality of the multimedia crawling subsystem is shown in Fig. 2. As it is observed, we can discriminate two different applications scenarios: the content indexing from Web servers and the content indexing from home personal computers.

![Fig. 2: Architecture of the multimedia crawling subsystem](image)

#### 3.1 Content indexing from web servers

Content indexing from web servers concerns the case of multimedia information that is distributed and available on-line throughout the Internet. In this case, we can furthermore distinguish two different approaches: the automatic and the manual multimedia crawling.

##### 3.1.1 Automatic Multimedia Crawling

From the moment that a new web page is detected and registered, according to the architecture proposed, it is up to the multimedia crawling subsystem to access every new website and fetch the content that lies there. The tasks performed by the system’s agent once new multimedia content has been located include scanning of all the relevant records in the database, defining among the latter the ones not checked, extracting through sockets the necessary material and storing it in the appropriate form in the database. The accomplishment of the above tasks provokes also a change in the value of the before mentioned indicator.

The whole process described above is translated in java code using the following guidelines: Firstly, a class defines which web pages are to be examined. Secondly, the web page is downloaded via Java sockets and is then parsed. The parsing process provides two kinds of useful information: multimedia content links and links to other web pages. For each image link found, a socket is afterwards opened and the image is downloaded, processed and (optionally) stored, while the database registrations are properly updated. Links to other web pages are also stored as unchecked locations, meaning that they will be examined in the future by the system agent.

##### 3.1.2 Manual Multimedia Crawling

In this case, the user is allowed to declare the location of his multimedia content. This approach is based on the activation of the system’s indexing mechanism from the web pages’ creators themselves.

#### 3.2 Content Indexing from Home PCs - Social Group Scenario

In this case, the multimedia content is not on-line at all times but it is located in the users’ home personal computers. This indexing approach is really convenient in case of social groups, which constitute an interesting web application. It allows to home users to share multimedia content among friends and groups of common interest.

Content indexing in this case is accomplished through the use of a Java application, which permits to the home users to scan and retrieve personal multimedia content and send it to the system server. In the PIDALION search engine, we have designed separate sector in the database to indicate the
respective social group to which the submitted content belongs. The Java application allows to home users to a) create a social group, by for example, declaring a list of other users that will constitute the social group or b) to register to an existing group.

As soon as the user downloads and runs the application, he/she is prompted to specify the remote directories where the multimedia content to be indexed lies. The latter is then automatically located, sent to the system server, processed and stored. A personalized multimedia index is created for this purpose. Of course downloading and utilizing this application is possible after it is validated that the user is registered and thus possesses the right to submit content.

### 4. Multimedia Metadata

#### 4.1 Metadata Extraction

This task aims at processing multimedia content that has been crawled, so that the proper metadata are extracted and stored in the database, being available for future searches.

As far as image processing is concerned, initially a new Image object is created. Once the pixels have been acquired, a mask is applied so as to isolate the R, G and B values. Afterwards, the RGB values of the extracted pixels are transformed into HSV. Based on the HSV values calculated, the image histogram is then constructed. Since full independency of the image size is required, the histogram’s values are normalized, being divided with the total number of pixels in the original image. Other types of metadata can be supported by this architecture.

![Fig. 3: Query process.](image)

#### 4.2 Query Process

This task is related to the utilization of the search engine by internet users and lies in the extraction of metadata from multimedia content submitted by the latter through the system’s interface within the frame of composing a query. Of course in this case there is no need for the metadata to be stored but just to be forwarded to the retrieval and display subsystem which is going to perform the searching. The whole process is shown schematically in Fig. 3.

### 4.3 Indexing Process

#### Image Content:

Fig. 4 shows which actions take place every time a new image has to be added in the system’s database. The reader must note that, in case of video files, key frames are extracted in the first place, enabling the process of metadata extraction to be repeated for each frame, which therefore is being treated as an independent image.

![Fig. 4: Add new image to database.](image)

#### Video Content:

In case that the submitted content is a video file, a video processor is activated which analyze the video content and extract key-frames based on the video summarization algorithms of [11]. Selection of this algorithm is due to the fact that a) it is extremely fast (real time processing) and b) it is not required to be known the number of key-frames extracted (the number of key frames are automatically estimated based on the video content).

The idea is presented in Fig. 5.

The Video Processor has been implemented by using the Java Media Framework (JMF) API. Firstly, a media locator is created as well as a processor that will be used as a player to playback the media. The Video Indexer accesses individual video frames by using a "pass-through" codec which is inserted into the data flow path. As data pass through this codec, a callback is invoked for each frame of video data. During the processor's configured state, two codecs, PreAccessCodec and...
PostAccessCodec, are set on the video track. These codecs are used to get access to individual video frames of the media file.

4.4 Type of Metadata
The visual descriptors of the MPEG-7 standard are supported by the proposed search engine. In particular, we estimate color and textural visual descriptors. As far as colour information is concerned, the scalable colour descriptor (SCD) and the dominant colour descriptor (DCD) are adopted. For texture information, the homogenous and the non-homogeneous texture descriptors are used.

Other types of metadata such as the filetype, category and textual information are used to further improve the precision of multimedia retrieval. Moreover, the system supports metadata for maintaining social groups and personalized indices. Multimedia thematic category is estimated manually in the framework of this search engine. In this way the number of erroneous categorizations is minimized. A more complicated approach is proposed in [12], where semantic representations form different abstraction levels are used create hierarchical groups.

5. The Retrieval and Display
This section describes the functionality and architecture of the specific subsystem, distinguishing at the same time between its two basic functions: retrieval and display. The subsystem's functionality is summarized in the following steps:

5.1 The Retrieval Functionality
Once the registrations that match the search criteria are determined, the Retrieval and Display subsystem draws the corresponding thumbnails, organizes their projection and presents them to the user. The processes described previously are depicted in Fig. 6, while the pseudo-code in Fig. 7 explains the query composition.

---

**Lead user search criteria:**

- determine which criteria are active and their corresponding parameters

**Criterion 1:** File type -> select only registrations with the file type indicated

**Criterion 2:** Category -> select only registrations that belong to one of the categories indicated

**Criterion 3:** Text -> select only registrations with relative text similar to at least one of the keywords provided

**Criterion 4:** Dominant color -> select only registrations

---

with |Reg_value-Ref_value|<10 for each R,G,B value

**Criterion 5:** Color Histogram -> select only registrations with |Reg_value-Ref_value|<5 for each one of the 10 vector elements for each one of the H,S,V vectors

**Criterion 6:** Texture Histogram -> select only registrations with |Reg_value-Ref_value|<5 for each one of the texture elements

Compose and execute sql query (the following is the most complex form)

select registrations that satisfy criterion 6 and are included in ( select registrations that satisfy criterion 5 and are included in ( select registrations that satisfy criterion 4 and are included in ( select registrations that satisfy criterion 3 and are included in ( select registrations that satisfy criterion 2 and are included in ( select registrations that satisfy criterion 1 )))

**Fig. 6:** Query composition pseudo-code

---

---

**Fig. 7:** Search and retrieval

The retrieval process is more complex, and thus requires further analysis, when the user provides more than one search criteria. An approach for handling multiple search criteria would be to detect the groups of registrations that satisfy each one of the given criteria and afterwards find their section, thus detect the registrations that satisfy all the criteria. This approach constitutes though the least efficient strategy, since a lot of time is required and consequently the system's efficiency is significantly reduced. So, instead multiple search criteria are confronted by being incorporated in complex queries, which are composed following a hierarchical structure. This structure ranges from the most easy-to-check criterion, and thus the criterion that requires less computational load (bottom) to the most complex and time consuming (top). As a result, the colour histogram criterion is found on top of the hierarchical structure, since for each registration that possibly matches this criterion, the system has to examine three histogram vectors each one composing of ten values, which makes a total of thirty parameters for each registration. In a few words, this retrieval approach enables to reject as
many registrations as possible using easy-to-check criteria, so that the most complex ones are left to be applied on a limited group of registrations.

So far only the simplest search scenario, which involves submission of an image and searching in image content registrations, has been analyzed. However, the PIDALION system offers also the possibility to just search for video files or even submit a video as a prototype for getting back images or other video files. So at this point, it is clear that there are finally four different possible search modes 1) submit image and search for images (one on one), 2) submit image and search for videos (one on many), 3) submit video and search for images (many on one), and 4) submit video and search for videos (many on many).

The adopted approach for handling queries for video files is even more complex. When the fourth scenario takes place, a video file uploaded by a user is split in frames and each frame is treated as an independent image. The registration of a video file in the database is finally considered a match only if at least one frame similar to those of the prototype is encountered. As far as scenarios 2 and 3 are concerned, it is obvious that the latter derive from the combination of cases 1 and 4.

5. 2 The Display Functionality

In the frame of enabling non-linear access to the retrieved content, so as to reduce the required access time for the user, the latter is organized in 8 groups, each one of them being determined by a particular set of algebraic relations involving dominant color or histogram values which are used as reference exactly for the purpose of categorizing the results. The reference values are either the corresponding dominant color or histogram values of the file submitted by the user (in case of query by example), or the calculated mean values of the retrieved results in case of simple queries.

The presented to the user projection of the retrieved content will include just one element of each one of the before-mentioned groups. Each one of these elements will serve as a link to the rest of the results of the corresponding group, forming a hierarchical structure and enabling the user to browse the content of his interest.

Non-linear projection of the retrieved content constitutes one of the most important and innovative features of PIDALION system, since almost all current search engines have adopted the alternative of serial projection.

Moreover, features such as the opportunity of defining the number of presented results per page or selecting a particular results page are also incorporated in the PIDALION system.

6. The Interface Subsystem

The user interface constitutes a dynamically changing environment and as a result it has been implemented using the JSP (Java Server Pages) and Java Servlets technologies which enable the dynamic production of html pages, through Java code.

6.1. Multimedia Content Declaration

The functionality in question covers the case of manual crawling and lies in the completion of a simple HTML application form by the user, in which the latter declares his wish to have his multimedia content indexed. The user visits the search engine’s webpage and fills in the required information. Afterwards, the interface subsystem sends this information to the database, where a suitable registration is made in the database sector which is disposed for this purpose.

6.2 Composition of Queries and Retrieval of the Results

It is obvious that in this case the basic functionality of the search engine PIDALION, thus multimedia content searching, is in question. In the system’s interface which is disposed for the composition of queries, the user is able to define the search criteria of his choice with the help of a set of text-fields, radio-buttons, check-boxes and combo-boxes. In case of queries by example, the user must firstly introduce his prototype through the system’s interface. Once the latter is submitted, a dynamic web page similar to the previous one is produced, in which the system has added the submitted model’s thumbnail.

The user’s content is uploaded to the system server by using the Commons FileUpload package. The role of the latter is to parse HTTP requests which conform to RFC 1867, enabling file uploads to servers and web applications (http://jakarta.apache.org/commons/fileupload).

The following actions take place afterwards: the multimedia metadata subsystem performs the task of extracting the appropriate descriptors from the prototype submitted by the user, while the retrieval and display subsystem commences the search process using the before mentioned descriptors as selection criteria. Once the search process is completed, the retrieval and display subsystem determines the projection of the retrieved...
results which are presented to the user through the system’s interface in the form of thumbnails.

More specifically, the system’s interface interacts with the retrieval and display subsystem and presents the retrieved results in clusters following the projection scheme that the latter provides. As it has already been explained, this projection scheme lies in the categorization of the retrieved results in groups taking into account the proximity of their dominant color or histogram values to those of the prototype that the user has submitted. In the initial results’ web page, thus in the first level of the retrieval hierarchy, each group is represented by an indicative element. This way of presenting retrieved content is a great advantage of PIDALION system, since it offers the user the possibility to decide at a first glance whether the retrieved results are satisfactory or not, navigate effectively among them and detect more easily and quickly the content of his interest. Fig. 8 provides an example of how the former approach works and how helpful it is while searching for similar images.

![Hierarchical presentation](image)

**Fig. 8: Hierarchical presentation**

An additional innovative feature of PIDALION is the possibility of evaluating the retrieved results. By clicking in the corresponding checkbox that is found below each retrieved registration, the user may provide to the system feedback regarding the type of results that are more appealing or of more interest to him. It is then up to the system to update the user’s profile and provide more personalized searches in the future. The possibility of preserving users’ profiles is examined in section 6.4.

6.3 Browsing Multimedia Content of Remote IP Address

This service offers the user the capability to access multimedia content that is located at a remote IP address. So as to facilitate browsing for the user, the multimedia content is in advance organized (either by its owner or automatically by the system) in a hierarchical structure. Once the presentation structure of the multimedia content is defined by the retrieval and display subsystem one or more dynamic pages with thumbnails are produced.

6.4 User’s Personal Profile

In order to improve the search engine’s services, thus to obtain better results through the search process, the user may enable the relevance feedback mechanism, which actually refines the returned results taking into account the selection criteria that the user chooses most times while introducing a query. Activation of this mechanism though, requires the creation of personal profiles, in which the system keeps information about them and about the users and their preferences.

7. The Multimedia Database Subsystem

7.1 Database structure

Since the database interacts with most subsystems and plays an important role in case of storing and retrieving metadata, it is analyzed as a separate component of the search engine. Although the storing space disposed for keeping the necessary for the system’s operation information is indivisible, in order to simplify the analysis it is assumed that the database is divided in the following general sectors, each one of which is named here after the type of data that it contains i) Multimedia content location, ii) Metadata, iii) Thumbnail, iv) User Profiles and v) Social Groups Info.

7.2 Index Organization for Real-time Response

Instead of storing the colour histogram as it is extracted from the multimedia content, it is preferred to store its difference from a reference vector: Initially, 256 bins are extracted for each chromatic component from each image or video frame that is being processed. Apparently, all this information is definite to provoke the system’s overloading and increase of response time. For this reason, the extracted bins are organized in 10 clusters, each one - except the last one - containing 26 original bins. Therefore, each registration is composed of 30 bins in total, enabling in this way fast search through numerous registrations.

The system’s quick response is also guaranteed through the use of SQL queries that are precompiled in the form of SQL procedures.

The search criteria provided by the user limit the range of the search especially in the case of
complex queries. The search space though can be further limited through scalable sorting of metadata: In order to achieve an efficient database organisation, proper algorithms need to be used. For example, metadata can be sorted according to the image histogram they carry. More specifically, the registrations are sorted taking into account the histogram vector’s distance from a reference vector. The sorting process is realized off-line, so as to avoid the system’s overloading. Moreover, since the sorting process should not interrupt the queries made by users, it is realized in a secondary database (which serves also as a backup). The secondary database draws the registrations from the primary database, sorts them and returns them, while the Retrieval and Display subsystem uses the primary database for handling queries.

8. CONCLUSIONS
In this paper, we describe a multimedia search engine architecture and we present implementation issues as the latter arises from the National funded project of PIDALION. The proposed search engine focuses on a) efficient organization of multimedia indices to allow real time response even for large scale multimedia content, b) hierarchical and content–based visualization of the retrieved results to allow for easy and efficient navigation of multimedia content, c) intelligent query composition to minimize response time and best match search criteria, d) fusion of text, semantic and visual descriptors to enhance content searching and interoperable representation using the XML schema of the MPEG-7 standard, e) efficient multimedia crawling and indexing either from web servers or home personal computers and f) support of personalized sharing of multimedia content through the creation of social groups.

9. ACKNOWLEDGMENTS
This work has been conducted under the support of the PIDALION, funded by the Greek Secretary for Research and Development.

References: