International Journal on Digital Libraries manuscript No.
(will be inserted by the editor)

A Digital Library Framework for Biodiversity Information Systems

Ricardo da S. Torres[1]*, Claudia Bauzer Medeiros[1], Marcos André Gonçalves[2] **, and Edward A. Fox[2]

[1]Institute of Computing, University of Campinas, Campinas, SP, Brazil
{rtorres,cmbm}@ic.unicamp
[2]Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA
{mgoncalv,fox}@vt.edu

Received: date / Revised version: date

Abstract. Biodiversity Information Systems (BISs) involve all kinds of heterogeneous data, which include ecological and geographical features. However, available information systems offer very limited support for managing these kinds of data in an integrated fashion. Furthermore, such systems do not fully support image content (e.g., photos of landscapes or living organisms) management, a requirement of many BIS end-users. In order to meet their needs, these users – e.g., biologists, environmental experts – often have to alternate between separate biodiversity and image information systems to combine information extracted from them. This hampers the addition of new data sources, as well as cooperation among scientists. The approach provided in this paper to meet these issues is based on taking advantage of advances in digital library innovations to integrate networked collections of heterogeneous data. It focuses on creating the basis for a next-generation BIS, combining new techniques of content-based image retrieval and database query processing mechanisms. This paper shows the use of this component-based architecture to support the creation of two tailored BIS systems dealing with fish specimen identification using search techniques. Experimental results suggest that this new approach improves the effectiveness of the fish identification process, when compared to the traditional key-based method.

Key words: Biodiversity Information System, Content-Based Image Retrieval, OAI

Send offprint requests to: Ricardo da S. Torres, rtorres@ic.unicamp.br

* *Communicating Author:* Ricardo da S. Torres, Institute of Computing – University of Campinas. Av. Albert Eistein, 1251, CEP 13084-851, Campinas, SP, Brazil. Phone: +55-19-3788-5840, FAX: +55-19-3788-5847, e-mail: rtorres@ic.unicamp.br

** Marcos André Gonçalves is currently with Federal University of Minas Gerais (UFMG), Brazil.

1 Introduction

Environmental changes have emerged as an important question in the global agenda. In order to support the design of policies for environmental management and ecosystem balance, it is necessary to get an accurate view of existing conditions, and to understand the complex changes that occur at all levels in the planet. One essential step toward creating appropriate scenarios is to collect relevant data about the environment and to develop information systems to manage and derive knowledge from these data. These systems must furthermore combine newly gathered data with historical and legacy information (e.g., from distinct kinds of archives), through unified management. Therefore, scientists concerned with environmental issues must seek support from a large set of systems. This, of course, brings about all kinds of interoperability problems due to system mismatch, data diversity, and variety of user profiles.

One representative example of such problems appears in the context of biodiversity, where expert end-users must contend with at least two kinds of unrelated systems: Biodiversity Information Systems (BISs) and image information systems. The latter involve software that allow users to manage images' content (e.g., patterns, shape, color, texture). In the biodiversity context, they are adopted by scientists for their image archives and to help them identify species.

A Biodiversity Information Systems (BISs) (e.g., [1, 6, 7]) is an environmental information system that manages huge sets of geographic data as well as large databases concerning species (e.g., natural history collections, field observation records, experimental data). Geo-related data concern all kinds of geophysical information, provided both by ground surveys and by remote sensing. Most BISs are concerned with determining the spatial distribution of one or more living species, and the spatio-temporal correlations and trends of these distributions.

This requires combining data on species (when and where they are observed, by whom and how) with geographic data that characterize the ecosystems where the species are observed. An example of a standard spatial query in a biodiversity system is "Show the areas in the 'Amazon basin' where the fish species *Cichla Ocellaris* has been observed". Besides being heterogeneous in nature (encompassing flora and fauna and the geophysical descriptions of their habitats), these data also are heterogeneous in other aspects – such as regarding spatio-temporal granularity or storage format.

Drawings and photos of species also may be used in this context. They are stored apart in the system's data files, and treated as auxiliary documentation, usually retrieved by species' name. Generally, images are accessed only via textual (metadata) queries, without support for content-based image retrieval, e.g., "Show all photos of fish species *Cichla Ocellaris*".

If, on the other hand, a scientist starts from incomplete pictorial information – e.g., just a photo of a fish specimen – he/she will have to resort to an image information system to request "Retrieve all database images containing fish with contour shaped like that in the photo". Once likely candidates are identified, the scientist then can continue work by turning to a BIS. Complex biodiversity queries actually may require switching several times across systems.

The goal of the work presented in this paper is to combine research on image processing, databases, and digital libraries to provide biodiversity researchers with a BIS that seamlessly integrates queries involving both image content and textual data. In such a context, users just will need to provide an image as input (e.g., the photo of a fish) and request the system to "Retrieve all database images obtained from 'Randall's tank photos' containing fish with contour shaped like that in the photo, and that are found in the 'Amazon basin' ".

To this end, we present a generic digital library (DL) architecture for managing heterogeneous data about living beings and their ecosystems. These data involve not only textual and location features, but also images. A key notion considered is that of a *DL component*, a specially designed software module that encapsulates specific functionality, thereby supporting modularity, flexibility, and reuse in constructing the DL infrastructure. Due to its component-based design, our architecture circumvents the interoperability and system-switching problems discussed. To illustrate the use of this architecture in a real application, it has been instantiated to support the creation of a BIS for fish species. The goal of that BIS is to help researchers on ichthyology to identify fish specimen by using search techniques offered by the architecture.

The two main contributions of this paper are: (a) a generic architecture for managing heterogeneous collections, based on digital library components, to access biodiversity data sources (text and images), that allows combining text-based and content-based queries in a seamless way; and (b) a new configurable component, for content-based image search, which has been integrated into that architecture.

The rest of this paper is organized as follows. Section 2 characterizes the proposed architecture, including its search components. Section 3 describes preliminary experiments conducted to validate the architecture. Section 4 briefly comments on related research. Section 5 presents conclusions and summarizes ongoing and future work.

2 Architecture

This section presents our generic architecture for managing heterogeneous biodiversity data in an integrated fashion. Our starting point is the assumption that the source data are stored in a network of heterogeneous collections organized in a digital library. This architecture takes into account two kinds of collections: domain-specific and image databases. Clearly, this architecture can be instantiated for managing data of different domains. For example, for an information system dealing with data on fish, the image-related data might include fish photos, while the domain-specific collection might contain data about fish taxonomy, morphologic descriptions, and habitats. For another system handling medicinal plant data, the image-related collection might contain plant photos, while its domain-specific collection might include descriptions of plant medicinal properties and known side effects.

Furthermore, other relevant domains, such as art and cultural imaging or biomedical information systems, can take advantage of the proposed generic architecture to combine text-based and content-based queries in a seamless way. In the former domain, image-related data would comprise art images (photos) while domain-specific collections might include data about authors, historic context, painting technique descriptions, etc. In the second case, medical images (such as X-rays or tomography scans) can be combined with textual data on patients, plus statistical and diagnostic information to help physicians in a clinical decision-making process.

2.1 Main Modules

Figure 1 shows our digital library architecture. It includes a set of search services (service providers) which are executed over heterogeneous data collections (data providers).

This architecture has been instantiated using digital library components developed at Virginia Tech. It uses the Open Archives Initiative (OAI) protocol [35, 43] as a basis for interoperability. OAI is an HTTP- and XML-based protocol for metadata harvesting. It supports digital library interoperability via a two-party model. At

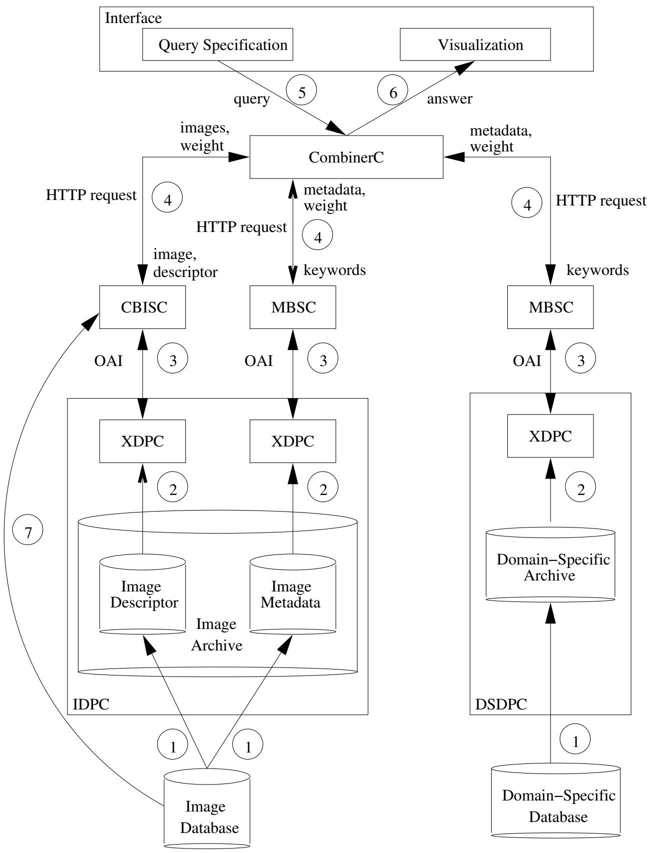

Fig. 1. System Architecture.

one end, data providers use the OAI protocol to publish structured data and metadata, in various forms. At the other end, service providers use the OAI protocol to harvest and process the metadata delivered by data providers, and to add value in the form of services.

The main modules interact as follows. The original data sources are stored in image and domain-specific databases. These collections must be pre-processed in order to generate their respective (open) archives (arrows labeled 1 in Figure 1): the *Image Archive* for the image collection and the *Domain-Specific Archive* for the domain-specific collection. Archives contain metadata and content descriptors, which speed up retrieval of the original data sources. Pre-processing is performed by batch programs that convert the original data sources into XML files. *XML Data Provider Components (XDPC)* are used to disseminate these XML files from OAI archives (arrows labeled 2), so that these XML files can be accessed through OAI requests (arrows labeled 3). An archive and its respective data provider can be seen as a data provider component. The data provider for the image collection is named *Image Data Provider Component (IDPC)*. The *Domain-Specific Data Provider Component (DSDPC)* is the "complex data provider" for the domain-specific collection.

Search components process queries against these archives. Queries are specified in terms of HTTP requests (arrows 4). A *Metadata-Based Search Component (MBSC)* handles both image metadata and domain-specific information. A *Content-Based Image Search Component (CBISC)* handles image content descriptors expressed in terms of feature vectors, which can be accessed either locally or remotely. In the former case, the *CBISC* can directly access these files (arrow 7). In the latter, they are accessed via the *Image Data Provider Component* (arrow 3).

These search components are activated by the *Combiner Component (CombinerC)*. The Combiner receives a query as input (arrow 5), decomposes it into subqueries, dispatches them to the search components, combines their results in a suitable way, and then returns a final answer to the interface layer for result presentation (arrow 6).

The interface layer is not discussed in this paper. An initial effort to provide users with semantically meaningful result presentations in *CBIR* systems is described in [17]. Here follows a description of the other modules.

2.2 Data Providers

The *Image Data Provider Component (IDPC)* and the *Domain-Specific Data Provider Component (DSDPC)* are "complex components" responsible for managing archives, using OAI-compliant XML data providers.

Archives: In this paper, the term "archive" is used to denote a repository of well-structured stored information; these repositories contain sets of XML files. Two different archives are contemplated in the architecture: *Image Archive* and *Domain-Specific Archive*. The *Image Archive* comprises image metadata and image content descriptors (feature vectors), while the *Domain-Specific Archive* concerns metadata related to a specific domain.

XML Data Provider Component (XDPC): *XMLFile* [59, 64] is an OAI-based component which is used for each XML Data Provider in the architecture. Basically, *XMLFile* is a Perl module that creates an OAI-compliant repository (data provider) to publish a set of XML files as an OAI archive. Its layout and configuration afford a clean separation between the data provider engine, the configuration data, and the data being published. This component does not require any specific metadata format to encode the XML files.

2.3 Search Components

The architecture uses two different search components: a metadata based search component called *ESSEX* (Section 2.3.1) and a content-based image search component (Section 2.3.2).

2.3.1 Metadata-Based Search Component (MBSC)

The *ESSEX* vector-space search engine [24] optimized for digital libraries is being used with our metadata. *ESSEX* is a componentized search engine. *ESSEX* acts as the core portion of an Open Digital Library (ODL [59]) search component, answering requests transmitted through an extended OAI (XOAI) protocol. *ESSEX*, available as open source software, was primarily developed for the CITIDEL (Computing and Information Technology Interactive Digital Educational Library) project [16], and now also is being used in the *PlanetMath* project [47]. In *ESSEX*, all information is indexed in "chunks" associated with field names, where chunks may correspond to XML elements in a metadata record. Its high speed is the result both of keeping index structures in memory and using a background daemon model based on socket communication with the DL application.

2.3.2 Content-Based Image Search Component (CBISC)

The *CBISC* is a new search component we created to handle queries based on image content. It supports collections of image information as a *Content-Based Image Retrieval (CBIR) system*. These systems can be characterized as follows. Assume that we have a database containing a large number of images. Given a user-defined query pattern (e.g., a query image), the system must retrieve a list of the images from the database that are most "similar," according to the image content (i.e., the objects represented therein and their properties, such as shape, color, and texture). Even though many other content-based retrieval systems exist [5, 28, 44], they do not take advantage of the component philosophy. Thus, they are not amenable to easy reuse in distinct situations. Our proposal has the advantage of encapsulating CBIR functionality into a DL component, thereby ensuring its reusability and integration in other DL-based systems. In fact, as will be seen, the *CBISC* can be configured easily by experts, as they adapt it to distinct domains and requirements.

A typical *CBIR* solution requires the construction of **image descriptors**, which are characterized by: (i) an *extraction algorithm* to encode image features into a *feature vector*; and (ii) a *similarity measure* to compare two images based on the distance between the corresponding feature vectors. The similarity measure is a *matching function* (e.g., using Euclidean distance), which gives the degree of similarity for a given pair of images represented by their feature vectors, often defined as an inverse function of the distance, that is, the larger the distance value, the less similar the images.

Figure 2 shows an overview of our *CBISC* component. It receives as input an HTTP request (arrow labeled 1 in Figure 2) which specifies a query in terms of the query pattern (query image), chosen descriptor, and kind of query (see Section 2.3.3). The *CBISC* starts processing a query by extracting a feature vector from the query image (module labeled A in Figure 2). This extraction process requires validating the proposed query against the *CBISC* configuration file (arrow 2) and searching for the appropriate *Extraction Algorithm* in the *Descriptor Library* (arrow 3). The validation process involves checking the input query parameters accordingly to the *CBISC* configuration. For example, it checks if a descriptor defined in the HTTP request is supported by the *CBISC* or if the input image matches the image type (colorful or binary) expected by the image descriptor used in the query.

Next, the query image feature vector is used to rank the database images according to their similarity to the query image (module B). This step relies on either a *Distance Computation Algorithm* (arrow 5) taking into account the feature vectors of all images in the database (arrow 7), or using an appropriate index structure (arrow 6). Images are indexed in the *CBISC* according to their feature vectors by using the M-tree [15] index structure to speed up retrieval and distance computation. The *M-Tree Library* in Figure 2 is a repository of M-Trees. Its implementation is based on the eXtensible and fleXible Library (XXL) [10, 22]. Finally, the most similar database images are ranked (module C) and the *CBISC* returns an XML file containing this ranked list (arrow 9).

The following sections present the kinds of queries *CBISC* supports (Section 2.3.3) and the steps necessary to configure and install the *CBISC* (Section 2.3.4).

2.3.3 *CBISC* Requests

Our *CBISC ODL component* is an OAI-like search component which aims at supporting queries on image content. As in the OAI protocol [35, 43], queries are submitted via HTTP requests. However, we generalized this to an extended OAI (XOAI) protocol for image search, that fits into the ODL framework [60, 61]. As is typical with XOAI protocols, each request specifies the Internet host of the HTTP server and gives a list of key-value pairs. Two special requests ("verbs") are supported by this image search component:

1. **ListDescriptors:** This verb is used to retrieve the list of image descriptors supported by our *CBISC*. No arguments are required for this verb.
2. **GetImages:** This verb is used to retrieve a set of images by taking into account their contents. Required arguments specify the query image, the descriptors to be used, and the kind of query. The *CBISC* supports two kinds of queries:
   - in a *K-nearest neighbor query (KNNQ)*, the user specifies the number k of images to be retrieved closest to the query pattern; and

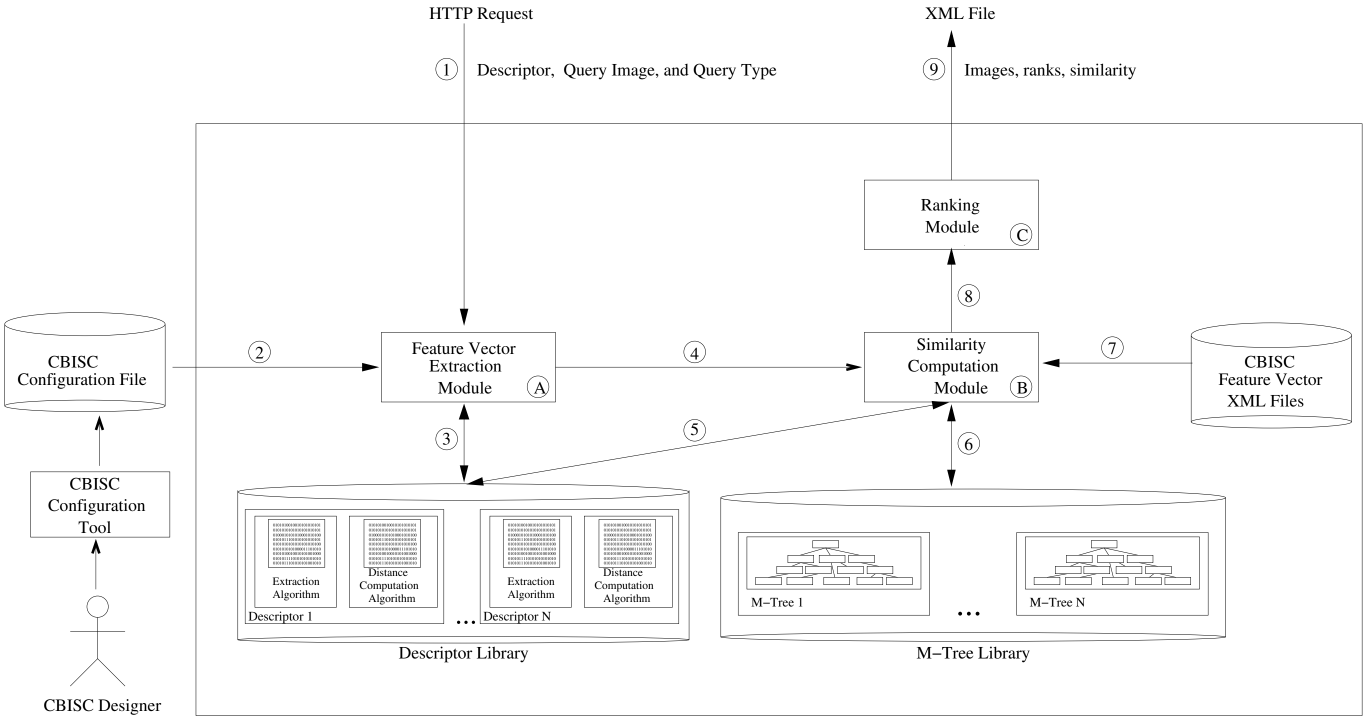

Fig. 2. *CBISC* architecture.

– in a *range query (RQ)*, the user defines a search radius r, retrieving each database image whose distance to the query pattern is less than r.

The responses to these verbs are encoded in XML.

2.3.4 *CBISC* Installation

The *CBISC* installation is performed by the so-called *CBISC Designer*, shown in Figure 2. The designer is an expert in the application domain, who is responsible for tuning the *CBISC* parameters to specific needs. This fine tuning ensures that the *CBISC* can be coupled seamlessly to distinct systems that require content-based retrieval and that take advantage of DLs. This process requires three preliminary phases: Image Descriptor Identification, Feature Vector Extraction, and *CBISC* XML Configuration.

– **Image Descriptor Identification**
Image descriptors vary with the application domain and expert requirements. Thus, in order to identify appropriate image descriptors (used in extraction and distance computation algorithms), experts must perform a set of small experiments, prior to installation. The experimental results are analyzed to evaluate image descriptors in terms of efficiency and effectiveness for a given collection of images.
Descriptors are typically domain and usage-dependent. Thus, a given image can be associated with very many descriptors. Many *CBIR* methods only support a fixed set of descriptors. *CBISC*, on the other hand, allows progressive extension of the descriptor base.

– **Feature Vectors Extraction**
Once suitable descriptors have been identified, their extraction algorithms are executed against the image database, generating a set of XML files containing the feature vectors for each image. Again, this step is performed prior to component configuration. Figure 3 presents an XML schema for the feature vector information, using the XMLSpy notation [65]. Basically, a feature vector XML file contains information related to: the image name, descriptor name, type of feature vector (1D or 2D curve), and feature vectors themselves (represented in terms of a curve – double vectors). A feature vector can be accessed either locally or remotely. In the former case, the *CBISC* can access directly these files (arrow 7 in Figure 1). In the latter, they are accessed through the *Image Data Provider Component* (arrow 3 in Figure 1).
Note that our approach relies on the use of image descriptors whose representation is encoded in feature vectors. We plan to redefine the feature vector XML schema so that it can encode other kinds of representations (e.g., graphs [54]).
One of the most important features of the *CBISC* is its flexibility in supporting different kinds of image descriptors. Firstly, the *CBISC* can be configured to perform queries involving different image properties (color, texture, or shape). In this case, it is just required that the extraction algorithm defined in an image descriptor generates a feature vector XML file as specified in Figure 3. Secondly, the *CBISC* supports extraction algorithms which create either 1D or 2D feature vectors. Thus, 1D feature vectors can

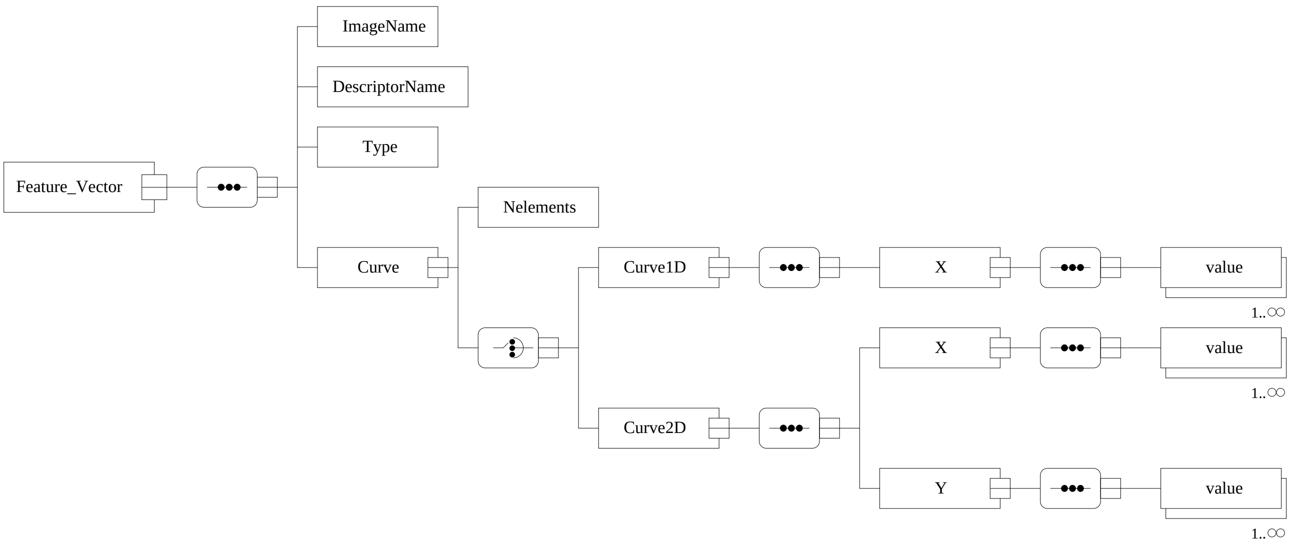

Fig. 3. Feature vector XML schema.

be generated for image descriptors like the *Color Histogram* [62] and the *Contour Multiscale Fractal Dimension* [19] shape descriptor. Similarly, 2D feature vectors can be extracted by, for example, the *Contour Saliences* [20] or the *Curvature Scale Space* [39] shape descriptors.

Figure 4 presents an example of a feature vector XML file. In this case, the feature vectors were obtained by applying the image descriptor "Contour Multiscale Fractal Dimension" [19] on image "fish0.pgm". Note that this feature vector is encoded in a 1D curve.


```
<?xml version="1.0" encoding="UTF−8"?>
−<feature_vector:Feature_Vector xmlns:feature_vector="http://feathers.dlib.vt.edu/~rtorres/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema−instance"
    xsi:schemaLocation="http://feathers.dlib.vt.edu/~rtorres/
    http://feathers.dlib.vt.edu/~rtorres/feature_vector.xsd">
    <feature_vector::ImageName>fish0.pgm</feature_vector:ImageName>
    <feature_vector:DescriptorName> ContourMSFractalDimension <feature_vector:DescriptorName>
    <feature_vector:Type> 1 <feature_vector:Type>
    −<feature_vector:Curve>
        <feature_vector:Nelements> 25 <feature_vector:Nelements>
        −<feature_vector:Curve1D>
            −<feature_vector:X>
                <feature_vector:value> 0.95105259594482394192 <feature_vector:value>
                <feature_vector:value> 0.98551214588154611995 <feature_vector:value>
                <feature_vector:value> 1.00415492765507829986 <feature_vector:value>
                <feature_vector:value> 1.00931032237937512441 <feature_vector:value>
                <feature_vector:value> 1.00583781572741104426 <feature_vector:value>
                <feature_vector:value> 0.99965178734378001835 <feature_vector:value>
                <feature_vector:value> 0.99641700001218280747 <feature_vector:value>
                <feature_vector:value> 1.00053413846216399108 <feature_vector:value>
                <feature_vector:value> 1.01448051045546439042 <feature_vector:value>
                <feature_vector:value> 1.03852447143279436048 <feature_vector:value>
                <feature_vector:value> 1.07079326852664902248 <feature_vector:value>
                <feature_vector:value> 1.10764282015553083838 <feature_vector:value>
                <feature_vector:value> 1.14425445370911771370 <feature_vector:value>
                <feature_vector:value> 1.17536781601217832360 <feature_vector:value>
                <feature_vector:value> 1.19605104931866845774 <feature_vector:value>
                <feature_vector:value> 1.20240888953449820420 <feature_vector:value>
                <feature_vector:value> 1.19213659320168563482 <feature_vector:value>
                <feature_vector:value> 1.16484253548940630552 <feature_vector:value>
                <feature_vector:value> 1.12208494304478412218 <feature_vector:value>
                <feature_vector:value> 1.06709853303495583177 <feature_vector:value>
                <feature_vector:value> 1.00422482309135441270 <feature_vector:value>
                <feature_vector:value> 0.93810555611087775851 <feature_vector:value>
                <feature_vector:value> 0.87275204902189629230 <feature_vector:value>
                <feature_vector:value> 0.81066432563100665476 <feature_vector:value>
                <feature_vector:value> 0.75224263059381879515 <feature_vector:value>
            </feature_vector:X>
        </feature_vector:Curve1D>
    </feature_vector:Curve>
</feature vector:Feature Vector>
```


Fig. 4. Example of a feature vector XML file.

– **CBISC XML Configuration**
  Once the feature vector XML files have been created, the *CBISC* can be configured. Basically, this process involves the creation of an XML configuration file detailing which descriptors are available and the image database related to this component. Figure 5 shows the XML schema that defines the *CBISC Configuration XML file*. *DescriptorInformation* includes a list of descriptors that are supported by the *CBISC*. Each descriptor is given in terms of its: name, extraction algorithm, distance computation algorithm, related feature vector size, and location of corresponding feature vector files. Image database information includes the number of images and their location.
  A list of predefined image descriptors (extraction and distance computation algorithms) is available in a tool we developed to configure the *CBISC*, called the *CBISC Configuration Tool*, allowing a quick *CBISC* instantiation for a new image collection. Examples include new shape descriptors like the *Contour Multiscale Fractal Dimension* and *Shape Saliences*, *Beam Angle Statistics - BAS)* [3, 18–20], and color descriptors, such as the *BIC* [58], and the *Color Histogram* [62]. Common metrics like *L1* and *L2* (Euclidean distance) also are supported.
  Figure 6 presents a screen shot showing the *CBISC Configuration Tool* developed to support *CBISC* designers in the configuration process.

After the previous preliminary steps are performed, the *CBISC Designer* is able to install the *CBISC*. This task also is supported by the *CBISC Configuration Tool*. Basically, this process involves copying feature vectors and algorithms (extraction and distance computation algorithms) either from local directories or from remote sites (by using OAI requests) to *CBISC* main directories. The location of both the feature vectors and algorithms are defined in the configuration step.

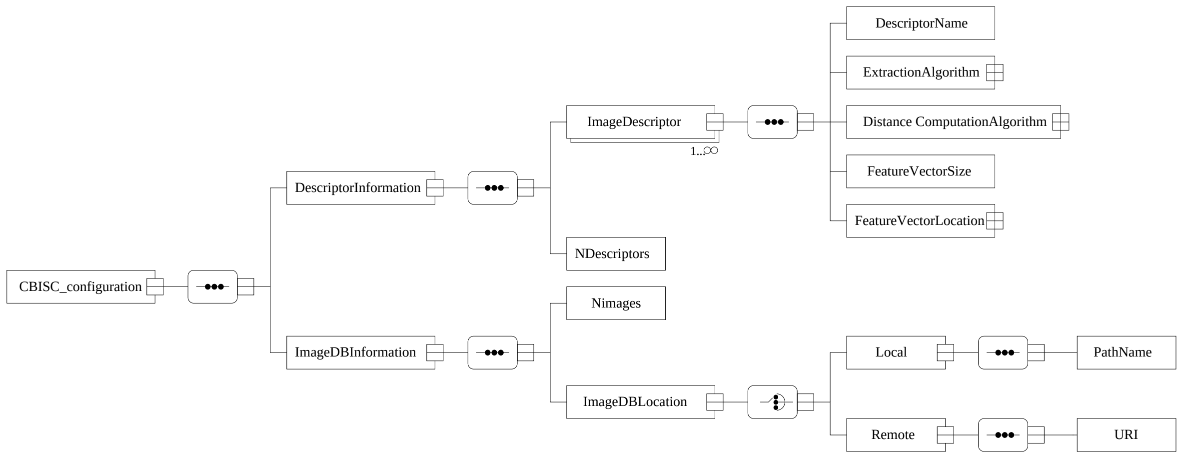

Fig. 5. XML schema for the *CBISC* Configuration file.

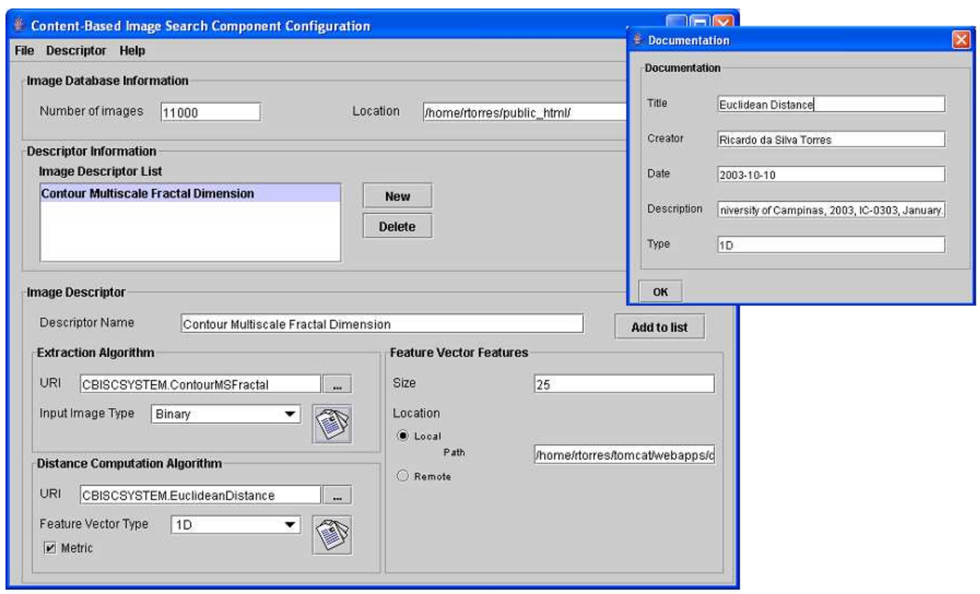

Fig. 6. *CBISC Configuration Tool* screen shot.

In fact, *CBISC* flexibility also relies on the support of both locally and remotely defined feature vectors and algorithms. In this sense, a *CBISC Designer* is able to configure a *CBISC* even without having previous knowledge about the algorithms (descriptors) code. This ease in configuration, and the DL component philosophy, allows BIS designers to easily combine distinct kinds of query features into the system, thereby allowing different user-tailored BISs for the same underlying archive base.

Note that the pre-processing of images into the image descriptors repositories ensures scalability and promotes a new, generic way of exposing image archives for creating image-based services.

2.4 The Combiner Component

The Combiner component is responsible for combining three different kinds of evidence: content-based retrieved images, image metadata, and domain-specific metadata. Basically, it receives as input a specification of a query pattern (query image) or query terms, decomposes and regroups the input into sub-queries, and forwards these resulting sub-queries to the appropriate search component (*CBISC* or *ESSEX*). Next, it combines the obtained results (weighted sets) by using an appropriate combination scheme, and returns a ranked list containing the "most" similar objects matching the original specification.

The combiner component has been implemented using search modules found in the *Java MARIAN system* [30]. *MARIAN* is an indexing, search, and retrieval system optimized for digital libraries which has been developed at Virginia Tech. Its search module is based on mapping abstract object descriptions to weighted sets of objects. In this case, the weight of each object in the set serves as a measure of how well that object matches the description.

Given a collection of weighted sets, different searching approaches can be used in the *MARIAN* system to combine them. The most commonly used types of combination include the maximization union and the summative union. The maximization union keeps only the maximum value of weighted objects that occur in incoming sets. The summative approach, on the other hand, calculates an average of the sums of incoming object sets. Other weighting schemes such as Euclidean distance or sum-of-squares also can be used.

Consider for example, a Biodiversity Information System which manages fish descriptions (images and textual information) and recall the query discussed in the introduction. In this query, a user provides an image as input (e.g., a photo of an observed fish) and then asks the system to *"Retrieve all database images obtained from 'Randall's tank photos' containing fish with contour shaped like that in the photo, and that are found in the 'Amazon basin' "*. This query deals with three different kinds of evidence: content-based image descriptors (image containing objects shaped like that in the input photo), image metadata (images from "Randall's tank photos"), and domain-specific metadata (species from "Amazon basin").

Given that query, the combiner component proceeds as follows:

1. Parse the original query. This process identifies which search component will be activated and its parameters.
2. Dispatch the query image to the *CBISC* module.
3. Dispatch the expression "Randall's tank photos" to the *ESSEX* search engine which manages image metadata.
4. Dispatch the term "Amazon basin" to the *ESSEX* search engine that manages domain-specific metadata.
5. Each search engine returns XML files containing records which match their respective queries.
6. These XML files are converted into weighted sets, which are combined, by using, for example, the summative union approach.
7. An XML file containing the final answer is returned to the interface layer.

3 Experiments

As an illustration of how this generic architecture can be instantiated, we implemented two Biodiversity Information Systems concerning fish species. The image data consisted of fish photos, and the domain-specific data concerned fish and associated habitat descriptions. With these systems, we carried out experiments to demonstrate the utility of our approach.

3.1 Combination of Evidences

The first experiment aimed at evaluating different strategies to combine textual and image content descriptors, to support exploratory searches, like the ones described in our motivating examples.

3.1.1 Data Sources

The fish related data were obtained from FishBase [27], available on CD-ROMs, as well as on-line [27]. FishBase covers over 25,000 species of fish from all over the world, including data about taxonomic classification, common names, population dynamics, fish morphology, metabolism, diet composition, trophic levels, food consumption, and predators.

A subset of these data, including 703 species and 932 images, was used in this work. *CBISC* was configured to use the *Beam Angle Statistics (BAS)* [3] shape descriptor. The following describes the archives managed in this Biodiversity Information System.

Domain-Specific Archive: The domain-specific archive contained biodiversity metadata on fish and their ecosystems. It included data about fish taxonomic classification (order, family, genus, and species names), common names, synonyms, ecological features (food items, diet remarks, etc.), morphological descriptions (type of mouth and teeth, sexual attributes, etc.), and a list of occurrences around the world.

Image Archive: The image archive contained metadata on fish images, and image descriptors. The main challenge in processing the images has been finding appropriate descriptors for the images, since species' photos are not "well behaved", because they are often taken using live (moving) species instead of more controlled specimens (that are dead and preserved). Therefore, photos that must be used present many irregularities – such as shape distortions – not found in more traditional image databases (e.g., landscapes or artwork). These distortions complicate content-based retrieval. This required a preprocessing step consisting of: image segmentation, reducing image noise, and image binarization.

The image metadata includes the picture name, related species code (fish ID), image format, color type, picture type, when the picture was obtained, author name, when the picture data were entered into the FishBase database, general comments, and last modification date (concerning the image).

3.1.2 Experimental Setup

The experiments were intended to evaluate the effectiveness achieved through the combined use of visual and textual features. In this case, we considered each available image as a query image. All images which depict fish belonging to the same species were grouped into the same relevant set. The average number of images in the relevant sets was 1.33. In order to simulate the presence of users, textual search terms were defined randomly for each query. A random attribute was determined, and then a random textual term was extracted from it. This process was performed for both image metadata and domain-specific descriptions.

Two combination strategies were evaluated: the *maximization union* and the *summative union* (see Section 2.4). The best results are presented in Section 3.1.3.

3.1.3 Results

Figure 7 shows the precision versus recall graphs concerning the use of textual evidence considering: only image metadata (curve named *ESSEX (IM)* in Figure 7), only domain-specific information (curve *ESSEX (DS)*), the combination of the textual evidence using the maximization union strategy (curve *ESSEX (IM + DS) MaxUnion*), and finally the combination of textual evidence using summative union (curve *ESSEX (IM + DS) SumUnion*). Note that both combination-based curves present the best results for recall values less than 0.9. From this point on, all curves present a similar behavior. The summative union curve is better than the maximization union one until recall is equal to 0.8. From this point on, the situation is slightly inverted. Note also the low values found for precision. This behavior is due to the low number of elements in the relevant sets.

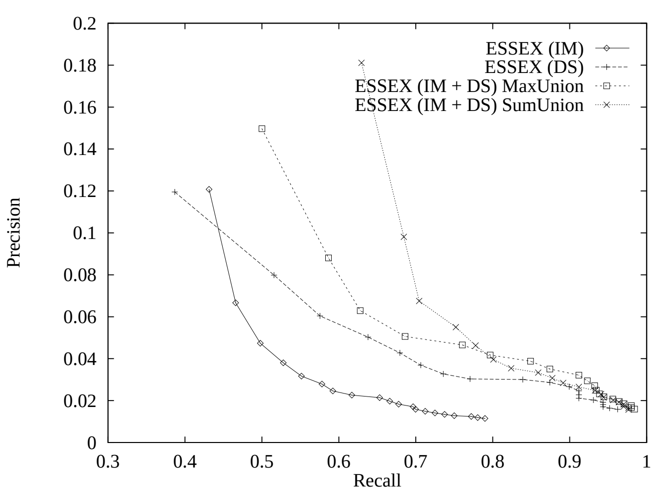

Fig. 7. Precision versus Recall curves for text based queries.

Figure 8 shows the precision versus recall graphs for queries involving both the *MBSC* and the *CBISC* search components. Seven different kinds of queries are evaluated, considering: only the *CBISC* search engine (curve named *CBISC - BAS* in Figure 8); the combination of queries on image content and textual information using the maximization union strategy (curves *CBISC + ESSEX (IM) MaxUnion*, *CBISC + ESSEX (DS) MaxUnion*, and *CBISC + ESSEX (IM + DS) MaxUnion* for image metadata, domain-specific information, and both together, respectively); and the same combination, now using the summative union strategy (curves *CBISC + ESSEX (IM) SumUnion*, *CBISC + ESSEX (DS) SumUnion*, and *CBISC + ESSEX (IM + DS) SumUnion*. The best result (curve *CBISC + ESSEX (IM + DS) SumUnion*) concerns the combination of the three available sources of evidence, using summative union. The combination strategies involving the maximization union strategy only yield a better behavior (than the curve which considers the use of *CBISC* separately), for recall values between 0.90 and 0.95.

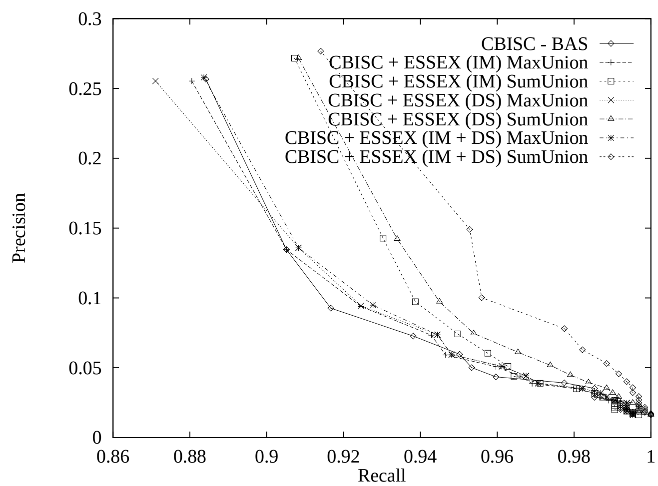

Fig. 8. Precision versus Recall curves for queries involving both the *MBSC* and the *CBISC* search engines.

The better performance of the summative union approach with the three sources further validates our assumption that a combination of several heterogeneous sources of evidence provides enhanced performance, since in this method each source contributes to some degree to the final score, while in the maximization union method only the evidence with the highest score is kept in the final result set. This result was used for tuning the system for the experiments involving domain experts (Section 3.2).

3.2 Fish Identification Process

The second experiment involved a user study concerning search retrieval techniques based on our architecture, used to support the fish identification process. The study aimed at comparing the effectiveness and quality of the proposed method versus the traditional key-based approach, using a task-oriented evaluation methodology. This research will potentially have a major impact on the development of new applications for supporting experts during the fish identification process.

3.2.1 The Problem

Given a mixed collection of specimens from a river, ichthyologists face the problem of identifying which fish species are present in that collection. Their aim is to determine which taxonomic classification (e.g., family, genus, species) is appropriate for a given specimen. The traditional approach is based on the use of *keys*, that support identification using a mechanism akin to binary decision trees [34]. Keys appear in the form of dichotomous (two-branched) couplets. Each couplet has two parts (e.g., 1a and 1b); each part of a couplet contains one or more statements. The statements give diagnostic (distinguishing) characteristics (e.g., anatomy, color). All statements in precisely one part of a couplet should fit the fish at hand [34]. Keying involves a sequential comparison of a specimen with a series of paired opposing statements (the parts of the couplets). The process continues, following the applicable statements (those that characterize the fish), until one ends at an identification [34]. For example, Figure 9 shows part of the key to families of freshwater fish of Virginia extracted from [34].

Unfortunately, the identification process based on keys suffers from many problems. A scientist knows approximately what the species is, and knows that one characteristic will separate the species, but this might not help with a static key. For a non-scientist, keys are often impossible to understand since they include many technical terms. This derives partly from the difficulty that specialist biologists sometimes have in translating their skill in identifying organisms into a written and generally easy-to-use key [34].

We try to solve these problems with the key-based approach, by creating a fish identification system based on our architecture. The main idea is to improve the fish identification process by allowing users to perform successive queries based on both fish shape information and textual descriptions. Our hypothesis is that an information system which supports different sources of evidences (textual description plus image content description) is at least as effective in the process of fish identification as the key-based method.

3.2.2 Data Sources

The fish related data were obtained from [34] and from a site recently created to help students in the fish identification process [31]. The fish descriptions included data on over 200 species found in the Commonwealth of Virginia, USA, including data related to taxonomic classification, common names, fish morphology, metabolism, diet habits, etc. A subset of these data, including 183 species and 187 images, was used in this work. The following sections describe the archives managed in this Biodiversity Information System.

Domain-Specific Archive: The domain-specific archive contained biodiversity metadata on fish and their ecosystems. It included data about fish taxonomic classification (family, genus, species), common names, reproductive and food habits, habitat description, information about similar species, and morphological descriptions.

Image Archive: In this experiment, we considered only image content descriptors as the image archive. Current experiments configured the *CBISC* to use the *Beam Angle Statistics (BAS)* [3] shape descriptor. It was chosen after a set of preliminary tests with end-users showed that it would be a good descriptor for this collection.

3.2.3 Experimental Setup

Seven subjects from the Department of Fisheries and Wildlife Sciences at Virginia Tech were recruited – three active researchers, one doctoral, two MSc and one undergraduate student. The key selection requirement was expertise in the ichthyology domain. Subjects of any age (over 18) or gender were accepted in the study.

Task: Given a fish specimen, users were asked to identify its corresponding family, genus, and species using both the traditional key-based method, and by performing queries on our system.

Procedure:

- 4 users tried to identify 10 specimens: 5 using the key-based approach (first group of fishes) and 5 using the computer system (second group).
- The other 3 users tried to identify the same fish specimens, but used the approaches in the reverse order, concerning the two groups of fish.

Opening Questionnaire: Users were asked to fill out a questionnaire concerning their familiarity with computers and search engines, as well as their expertise in the ichthyology domain and, more specifically, in identifying fish species.

Measured indicators:

- effectiveness: number of correctly identified fish;
- "usability": based on subjective grades (from 1=low to 10=high). With regard to effectiveness and ease of use of the proposed tasks, users were asked to grade both methods. They also were asked to rate their understanding of the computer-assisted fish identification process, both before and after using the tool.
- performance: time spent during the process.

3.2.4 Results

This section presents the experimental results concerning the use of the key-based approach and the computer system for identifying fish species.

On average, users performed 2.5 queries when using the computer system to correctly identify a specimen.

Key to Families of Freshwater Fishes of Virginia

1a Paired fins absent; jaws absent, mouth in an oral disk (the disk mostly surrounded by a fleshy hood in larvae); 7 external gill openings present in row behind eye .... **Lampreys - Petromyzontidae**
1b Paired fins present (at least 1 set); jaws present; 1 external gill opening per side 2
2a Caudal fin heterocercal or abbreviate heterocercal (Figure 5) 3
2b Caudal fin protocercal (Figure 13, Part 2, upper left) or homocercal (Figure 5) 6
3a Snout having a long paddle-like structure; operculum long, flexible, and pointed posteriorly .... **Paddlefishes - Polyodontidae**
3b Snout lacking a long paddle-like structure; operculum short 4
(...)

Fig. 9. Part of the key to families of freshwater fish of Virginia [34].

90 queries were performed: 22.2% including only textual terms, 30.0% based only on image content description, and 47.8% using both sources of evidences. An example of a query including both textual and image descriptor information was: *"retrieve fish descriptions of all fish whose shape is similar to that shown in Figure 10, which belong to genus 'notropis', which have 'large eyes' and 'dorsal stripe', and have been observed in both the 'New' and 'Tennessee' rivers"*.

	Fish Identification Method	
Taxonomy Level	Key	Computer System
Family	100.0%	100.0%
Genus	82.9%	91.4%
Species	51.4%	62.9%

Table 1. Number of correctly identified specimens.

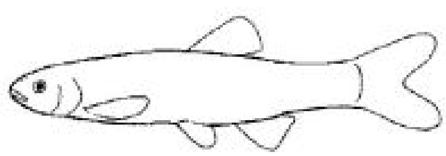

Fig. 10. Example of shape outline used to define a query.

Figure 11 presents a screen shot showing the interface used to define queries by using the fish identification tool. Here, the user can formulate the previous query by selecting on the screen the fish outline that is closest to the request. In addition, text parameters can be entered at the bottom.

Questionnaire: Users were, in general, familiar with computers (five out of seven were "fairly familiar" or "very familiar", while two were "somewhat familiar"). A similar result was found when the users were asked to grade their familiarity with search engines (e.g., Google). Six out of seven users were "fairly familiar" (four) or "very familiar" (two) with search engines. Only one user was "somewhat familiar".

With respect to their expertise in identifying fish species, users were asked to rate (from 1=low to 10=high) their ability in this task. They rated themselves fairly expert in this task (on average, 7.44). All students had taken courses where fish species were studied (six out of seven have studied this two or more times).

Effectiveness: Table 1 shows the percentage of correctly classified specimens. All users were able to assign correctly the family name of a specimen, using either method. However, when using the computer system, a higher number of genus and species names were identified (91.4% against 82.9% for genus and 62.9% against 51.4% for species). This result suggests that the proposed information system is more effective than the key-based approach to support the fish identification process.

Usability: After performing the tasks, users were asked to rate (from 0=low to 10=high) how effective and how easy both approaches are. Table 2 presents the average results. According to these results, the key-based approach was perceived by the subjects to be slightly more effective (8.1 against 7.7), while the computer-assisted method was judged easiest (6.9 against 4.9). The results obtained for the effectiveness subjective metric can be justified based on the familiarity of the users in identifying fish species by using keys. Furthermore, even though this method was felt to be more effective, the number of correctly identified specimen (see Table 1) shows the opposite result.

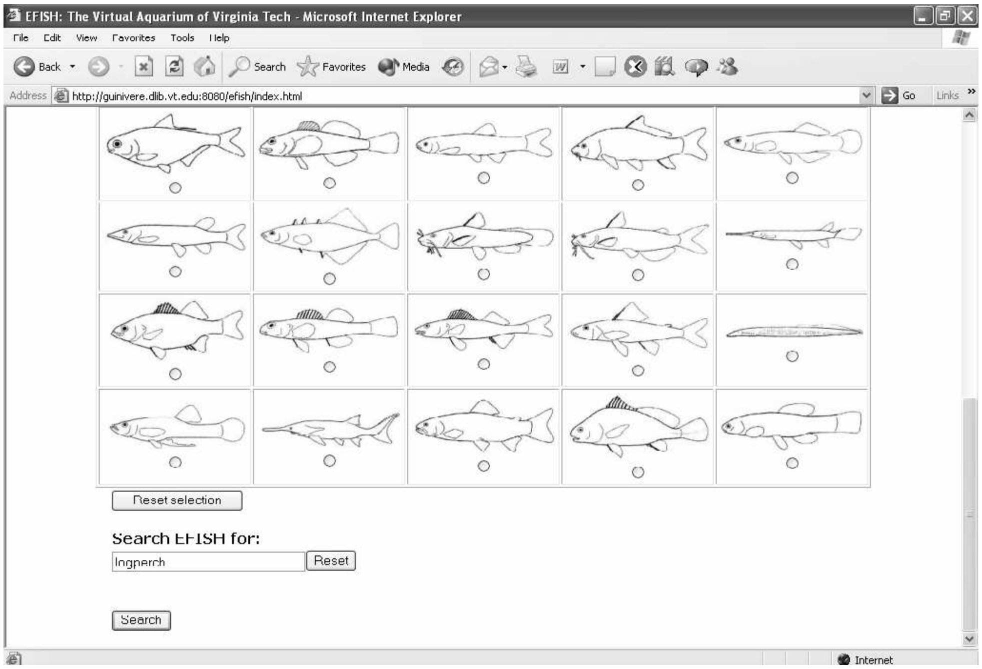

Fig. 11. Screen shot of the fish identification tool.

	Fish Identification Method	
Subjective Measure	Key	Computer System
how effective	8.1	7.7
how easy	4.9	6.9

Table 2. Average grades for subjective measures.

Fish Identification Method	
Key	Computer System
6.1	4.1

Table 4. Average time in minutes required to identify correctly a specimen.

Understanding	
before	4.4
after	8.0

Table 3. Average grade for user understanding of the computer-assisted fish identification process.

Users also were asked to rate their understanding of the computer-assisted fish identification process, both before and after this experiment. Table 3 shows the average grades. This result (improvement from 4.4 to 8.0) confirms that users were able to learn how to use our information system for identifying fish species.

Performance: Table 4 shows the average time required to correctly identify a specimen. By using the computer-based approach, users can identify species more quickly than by using the key-based approach (4.1 minutes against 6.1 minutes). Users of the computer system spent most of this time browsing the results. The system response time was less than 2 seconds, on average.

General Comments: In general, users believe that the computer-assisted approach can be very useful to help them identify fish specimens. We list below some of their comments (that relate to our evaluation, and that also can help guide future refinements of our methods):

– "Pictures on computer-based approach were much more helpful than diagrams in key-based approach";
– "The key-based approach is fine to the family level, sometimes, to genus. But it requires dissections, too many subjective judgments to identify species for large families – e.g., Percidae and Cyprinidae. Computer approach (is) thus much more convenient. (It) certainly can get you to the genus and sometimes to species";
– "The best approach is a mix. Have the computer help you through the key by providing lots of pictures, including pictures of fish showing key features in the key...";
– "For some of these species, a key is required. I think that many misidentification could result from the computer-based approach. The computer-based approach could work well with live specimens";
– "If I had more practice with the computer, I may come to prefer that";
– "(The computer system) uses multiple terms for the same thing (...), but the computer does not recognize these as identical. Maybe it would help to create a glossary so user knows which terms to choose";
– "The program is great and can be useful for experts and beginners. I suggest leaving the family names

with the form outlines so that people with experience can narrow search results more quickly".

4 Related Work

The research described in this paper differs from related research in the sense that it takes advantage of tailored DL protocols to seamlessly combine textual and content-based retrieval for biodiversity applications. Furthermore, the use of the software engineering notion of *component* ensures appropriate encapsulation of data and procedures, which allows reuse of the components developed in other DL initiatives involving content-based retrieval.

There are some other DL initiatives for the biodiversity domain. One example concerns floristic digital libraries (FDL) [50–52]. These are distributed virtual spaces comprising botanical data repositories and a variety of services offered to library patrons to facilitate the use and extension of existing knowledge about plants. FDLs use an agent-based infrastructure to manage information about taxonomic keys, distribution maps, illustrations, and treatments (morphological descriptions). Content-based retrieval, however, is not supported.

Another example is the Taiwanese digital museum of butterflies, an initiative of National Chi-Nan University and the National Museum of Natural Center [32]. This digital library contains 6 modules: XML-based information organization of digitized butterfly collections, content-based image retrieval of butterflies, a synchronized multimedia exhibition, compositional FAQ, interactive games of a butterfly ecosystem, and on-line courseware on butterflies. Even though XML documents describing butterfly species are indexed and retrieved by a search engine, this digital library does not support queries that combine image content and textual data.

DL efforts that deal with images appear in other domains. An example is the work of Zhu *et al.* [70], which presents a content-based image retrieval digital library that supports geographical image retrieval. The system manages airplane photos which can be retrieved through texture descriptors. Key goals of the Alexandria Digital Library ADL [57] and its successor (the Alexandria Digital Earth Prototype System (ADEPT) [33] are to build a distributed digital library accessible over the Internet for geographically referenced materials including maps, satellite images, etc., along with their associated metadata. The ADL system applied image-processing techniques to achieve content(texture)-based access to satellite images. Both initiatives, however, have limited support for queries simultaneously involving image content properties and textual data.

Content-Based Image Retrieval (CBIR) presents several challenges and has been subject to extensive research from many domains, such as image processing or database systems [2, 4, 12, 21, 48, 49, 56, 66]. Database researchers are concerned with indexing [9, 15] and querying [4], whereas image processing experts worry about extracting appropriate image descriptors (e.g., shape descriptors [67]). In fact, several applications requiring content-based querying and searching of images abound and can be found in a number of different domains, ranging from art and cultural imaging [36] to biomedical image databases [40]. State-of-the-art research in this CBIR area includes, for example, work on defining generic image descriptors (e.g., wavelets [36, 42]) and on applying machine learning techniques (e.g., [23, 26]) to improve image searching effectiveness.

Other important initiatives in the CBIR domain include [5, 28, 44] or more recently [11, 42, 46, 63], which also support search of images according to their content information. Even though these systems are shown to be effective, they cannot be easily customized for different domains. Firstly, most have a pre-defined and not extensible set of image descriptors. In addition, since they do not take advantage of the component philosophy, they cannot be reused and coupled to other information systems. The *CBISC* component presented in this paper overcomes these limitations.

In the video retrieval domain, Christel *et al.* [14] extract geographic references from videos aiming at improving access to the Informedia Digital Video Library. The available video retrieval process is based on date (when), word occurrences (what), and location information (where), extracted from the narrative and from the text regions in the video segments. Interactive maps are used to display places discussed in a video segment. The user can interact with these maps through toolbar icons that enable zooming in and out, panning, accessing details relevant to the video content, and selecting search areas. Content-based video retrieval is not supported.

Different strategies have been proposed, aiming at supporting the combination of textual information and visual content in the image retrieval proccess [37, 41, 53, 68, 69]. One approach [36, 53, 68] has been to combine textual information with visual contents by using *Latent Semantic Indexing (LSI)* and *Singular Value Decomposition (SVD)* to support image retrieval on the WWW. The combination strategy of Nakagawa *et al.* [41] is based on clustering image objects according to their visual features and mapping the created clusters into related words determined by psychological studies. A different approach is presented in [69]. In this system, the unification of keywords and feature contents is based on a seamless joint querying and relevance feedback scheme. Keyword annotations for each image are converted into a vector which expresses the probability of a given keyword appearing for a given image. An algorithm for the learning of word similarities during a relevance feedback process also is presented. Lu *et al.* [37] propose a strategy based on semantic networks and relevance feedback to deduce and utilize the images' content for retrieval. Lewis *et al.* [36] uses wavelet-based descriptors to encode image content information and textual descriptions encoded in RDF to support art image retrieval. They do not present how results obtained from queries that use both image content and text information are merged/combined.

In contrast to the monolithic-method adopted by the aforementioned solutions, our approach calculates the combination of textual and visual content in different and autonomous modules that can be extended independently. In addition, large systems are too complex [53, 68] to be easily configured for a new domain. Yet, some employ helpful search process techniques (relevance feedback, word similarity learning, content semantic definitions) [37, 41, 69] which are not yet available in our architecture. Fortunately, such techniques and combination strategies can be easily adopted in our architecture. New combiners just have to follow the HTTP-based communication protocol presented here.

The XML schema adopted in our Content-Based Image Search Component to encode feature vectors (see Figure 3) is similar to MPEG-7 [13, 55] solutions to describe multimedia data content. MPEG-7, for example, includes a *Description Definition Language (DDL)*, which defines representation data structures, such as matrices and arrays, to encode feature vectors of different visual features. Furthermore, the MPEG-7 initiative also standardizes a set of descriptors applied to images and/or videos [8, 38]. Current work investigates both the use of MPEG-7-based tags to define generic feature vectors (not limited to vector data structures) and the incorporation of MPEG-7 image descriptors into the *CBISC* descriptor set.

5 Conclusions

Interoperability has been a central research area in the digital library domain [45]. The OAI protocol has been used to promote interoperability solutions for different digital libraries initiatives [29, 35]. Following this trend, this paper presented an OAI-based generic digital library architecture for integrated management of image descriptors and textual information. The solution proposed is based on using DL components which are mostly new or recently developed. This architecture is easily extensible, and provides users with a considerable degree of flexibility in data management. This solution solves many current problems in this kind of system, allowing handling of images and textual information in an integrated fashion.

A new Content-Based Image Search Component was presented that supports queries on image collections. Since this component is based on OAI principles, it provides an easy-to-install search engine to query images by content. It can be readily tailored for a particular collection by a domain expert, who carries out a clearly defined set of pilot experiments. It supports the use of different types of vector-based image descriptors (metric and non-metric; color, texture, and shape descriptors; with different data structures to represent feature vectors), which can be chosen based on the pilot experiment, and then easily combined to yield improved effectiveness. Besides, it encapsulates a metric index structure to speed up the search process, that can be easily configured for different image collections.

To illustrate our claim that our architecture can be applied to several domains, this paper describes its application in building two Biodiversity Information Systems, dealing with different collections of fish species. Firstly, we performed experiments concerning the combination of textual and image content information (from FishBase). Preliminary results show that when both textual and visual information are used in the image retrieval process, results are, in general, better than those achievable using only visual or textual information. On the average, better results were found by using the summative union combination strategy. Secondly, we have evaluated the use of the proposed architecture to help experts in the process of identifying Virginia freshwater fish. Results show that the fish identification process based on our information system is more effective, easier, and less time consuming than that based on the traditional key-based approach. We have been working to provide students in ichthyology courses at the Department of Fisheries and Wildlife Sciences at Virginia Tech with an information system based on our architecture, aiming at supporting the process of learning new fish species.

Ongoing work concerns the instantiation of the proposed architecture in other domains. For instance, we are trying to combine queries on image content with textual description in the archaeology domain [25]. In this case, the image collection comprises photos of archaeological artifacts (e.g., pottery, coins, etc.) and the domain-specific collection corresponds to both archaeological site information and artifact descriptions. Preliminary experiments confirm the reusability of the components developed. Future work also includes performing user experiments to evaluate the different combination strategies which can be used by the *Combiner Component*. We also intend to evaluate other image descriptors [19, 39, 58, 62] in the combination process. Finally, experiments aiming at evaluating the performance/scalability of our architecture considering datasets with different sizes and by taking into account the execution of different queries simultaneously will be performed as well.

6 Acknowledgments

This work was supported by FAPESP (Proc. 01/02788-7), CAPES (Proc. BEX844/03-9), CNPq, the Brazilian WebMaps, PRONEX/SAI and AgroFlow projects, and NSF (DUE-0121679 and ITR-0325579). Marcos André

Gonçalves was supported by a fellowship from AOL and is currently supported by CNPq.

The authors also thank Dr. Eric Hallerman from the Department of Fisheries and Wildlife Sciences at Virginia Tech for his support during experiment design and execution.

References

1. S. Analyst, 2004. http://www.speciesanalyst.net (as of October 2004).
2. A. Antani, R. Kasturi, and R. Jain. A Survey on the Use of Pattern Recognition Methods for Abstraction, Indexing and Retrieval of Images and Video. *Pattern Recognition*, 35(4):945–965, April 2002.
3. N. Arica and F. T. Y. Vural. BAS: A Perceptual Shape Descriptor Based on the Beam Angle Statistics. *Pattern Recognition Letters*, 24(9-10):1627–1639, June 2003.
4. Y. A. Aslandogan and C. T. Yu. Techniques and Systems for Image and Video Retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):56–63, January/February 1999.
5. J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, and C.-F. S. R. Jain. Virage Image Search Engine: An Open Framework for Image Management. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 76–87, Bellingham, WA, USA, 1996.
6. BioGIS. BioGIS - Israel Biodiversity Information System. http://www.eti.uva.nl/ (as of October 2004).
7. BIOTA/FAPESP. SinBiota (Sao Paulo State's Biodiversity Information System). http ://www.biota.org.br/sia (as of October 2004).
8. M. Bober. MPEG-7 Visual Shape Descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):716–719, June 2001.
9. C. Bhm, S. Berchtold, and D. A. Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys (CSUR)*, 33(3):322–373, 2001.
10. M. Cammert, C. Heinz, J. Kramer, M. Schneider, and B. Seeger. A Status Report on XXL - A Software Infrastructure for Efficient Query Processing. *Data Engineering Bulletin*, 26(2):12–18, 2003.
11. C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image Segmentation Using Expectation-Maximization and its Application to Image Querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, August 2002.
12. V. Castelli and L. D. Bergman, editors. *Image Databases. Search and Retrieval of Digital Imagery.* John Wiley Sons, New York, NY, USA, 2002.
13. S. F. Chang, T. Sikora, and A. Puri. Overview of the MPEG-7 Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):688–695, June 2001.
14. M. G. Christel, A. M. Olligschlaeger, and C. Huang. Interactive Maps for a Digital Video Library. *IEEE Multimedia*, 7(1):60–67, 2000.
15. P. Ciaccia, M. Patella, and P. Zezula. M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. In *Proceedings of 23rd International Conference on Very Large Data Bases*, pages 426–435, Athens, Greece, 1997.
16. CITIDEL. Computing and Information Technology Interactive Digital Educational Library, 2003. http://www.citidel.org (as of October 2004).
17. R. da S. Torres, C. G. da Silva, C. B. Medeiros, and H. V. da Rocha. Visual Structures for Image Browsing. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, pages 167–174, New Orleans, Lousiana, USA, November 2003. ACM Press.
18. R. da S. Torres, A. X. Falcão, and L. da F. Costa. Shape Description by Image Foresting Transform. In *Proceedings of the 14th International Conference on Digital Signal Processing*, volume 2, pages 1089–1092, Santorini, Greece, July 2002.
19. R. da S. Torres, A. X. Falcão, and L. da F. Costa. A Graph-based Approach for Multiscale Shape Analysis. *Pattern Recognition*, 37(6):1163–1174, June 2004.
20. R. da S. Torres, E. M. Picado, A. X. Falcão, and L. da F. Costa. Effective Image Retrieval by Shape Saliences. In *Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing*, pages 49–55, São Carlos, São Paulo, Brazil, 12-15 October 2003.
21. A. del Bimbo. *Visual Information Retrieval.* Morgan Kaufmann Publishers, San Francisco, CA, USA, 1999.
22. J. V. den Bercken, B. Blohsfeld, J.-P. Dittrich, J. Krämer, T. Schäfer, M. Schneider, and B. Seeger. XXL - A Library Approach to Supporting Efficient Implementations of Advanced Database Queries. In *Proceedings of the 27th International Conference on Very Large Data Bases*, pages 39–48, Roma, Italy, 2001.
23. I. El-Naqa, Y. Yang, N. P. Galatsanos, R. M. Nishikawa, and M. N. Wernick. A Similarity Learning Approach to Content-Based Image Retrieval: Application to Digital Mammography. *IEEE Transactions on Medical Imaging*, 23(10):1233–1244, October 2004.
24. ESSEX, 2004. http://br.endernet.org/∼akrowne/elaine/essex/index.html (as of October 2004).
25. ETANA. Managing Complex Information Applications: An Archaeology Digital Library, 2004. http://feathers.dlib.vt.edu (as of October 2004).
26. T. Evgeniou, M. Pontil, C. Papageorgiou, and T. Poggio. Image Representations and Feature Selection for Multimedia Database Search. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):911–920, July/August 2003.
27. FishBase. www.fishbase.org (as of October 2003).
28. M. Flickner, H. Sawhney, W. Niblack, Q. H. J. Ashley, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by Image and Video Content: the QBIC System. *IEEE Computer*, 28(9):23–32, Sep 1995.
29. M. A. Goncalves, R. K. France, and E. A. Fox. MARIAN: Flexible Interoperability for Federated Digital Libraries. In *Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries*, pages 173–186, Germany, 2001.
30. M. A. Goncalves, P. Marther, J. Wang, Y. Zhou, M. Luo, R. Richardson, R. Shen, L. Xu, and E. A. Fox. Java MARIAN: From an OPAC to a Modern Digital Library

System. In *String Processing and Information Retrieval: 9th International Symposium, SPIRE 2002*, pages 194–209, Lisbon, Portugal, September 2002.

31. L. Helfrich, T. Newcomb, E. Halleman, and K. Stein. EFISH, http://www.cnr.edu/efish (as of October 2004).
32. J. S. Hong, H. Chen, and J. Hsiang. A Digital Museum of Taiwanese Butterflies. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, pages 260–261, San Antonio, Texas, United States, 2000. ACM Press.
33. G. Janee and J. Frew. The ADEPT Digital Library Architecture. In *Proceeding of the Second ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 342–350, Portland, Oregon, USA, 2002. ACM Press.
34. R. E. Jenkins and N. M. Burkhead. *Freshwater Fishes of Virginia*. American Fisheries Society, Bethesda, Maryland, 1993.
35. C. Lagoze and H. V. de Sompel. The Open Archives Initiative: Building a Low-Barrier Interoperability Framework. In *Proceedings of the Joint Conference on Digital Libraries*, pages 54–62, Roanoke, VA, USA, June 2001.
36. P. H. Lewis, K. Martinez, F. S. Abas, M. F. A. Fauzi, S. C. Y. Chan, M. J. Addis, M. J. Boniface, P. Grimwood, A. Stevenson, C. Lahanier, and J. Stevenson. An Integrated Content and Metadata Based Retrieval System for Art. *IEEE Transactions on Image Processing*, 13(3):302–313, March 2004.
37. Y. Lu, H. Zhang, L. Wenyin, and C. Hu. Joint Semantics and Feature Based Image Retrieval Using Relevance Feedback. *IEEE Transactions on Multimedia*, 5(3):339–347, September 2003.
38. B. S. Manjunath, J. R. Ohm, V. V. Vasudevan, and A. Yamada. Color and Texture Descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703–715, June 2001.
39. F. Mokhtarian and S. Abbasi. Shape Similarity Retrieval Under Affine Transforms. *Pattern Recognition*, 35(1):31–41, January 2002.
40. H. Muller, N. Michoux, D. Bandon, and A. Geissbuhler. A Review of Content-Based Image Retrieval Systems in Medical Applications – Clinical Benefits and Future Directions. *International Journal of Medical Informatics*, 73(1):1–23, Feb 2004.
41. A. Nakagawa, A. Kutics, K. Tanaka, and M. Nakajima. Combining Words and Object-based Visual Features in Image Retrieval. In *Proceedings of the 12th International Conference on Image Analysis and Processing*, pages 354–359, Mantova, Italy, September 2003.
42. A. Natsev, R. Rastogi, and K. Shim. WALRUS: A Similarity Retrieval Algorithm for Image Database. *IEEE Transactions on Knowledge and Data Engineering*, 16(3):301–316, March 2004.
43. OAI. Open Archives Initiative, 2003. http://www.openarchives.org (as of October 2004).
44. V. E. Ogle and M. Stonebraker. Chabot: Retrieval from Relational Database of Images. IEEE *Computer*, 28(9):40–48, Sep 1995.
45. A. Paepcke, C.-C. K. Chang, T. Winograd, and H. Garcia-Molina. Interoperability for Digital Libraries Worldwide. *Communications of the ACM*, 41(4):33–42, 1998.
46. I. J. PicHunter, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos. The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments. *IEEE Transactions on Image Processing*, 9(1):20–37, January 2000.
47. PlanetMath. http://planetmath.org/ (as of October 2004).
48. Y. Rui, T. S. Huang, and S. F. Chang. Image Retrieval: Current Techniques, Promising Directions, and Open Issues. *Journal of Communications and Image Representation*, 10(1):39–62, March 1999.
49. L. M. S., editor. *Principles of Visual Information Retrieval – Advances in Pattern Recognition*. Springer-Verlag, London Berlin Heidelberg, 2001.
50. J. Sanchez, C. Lopez, and J. Schnase. An Agent-Based Approach to the Construction of Floristic Digital Libraries. In *Proceedings of the 3rd ACM International Conference on Digital Libraries*, pages 210–216, Pittsburgh, PA, July 1998. ACM Press.
51. J. A. Sanchez, L. Fernandez, and J. L. Schnase. Agora: Enhancing Group Awareness and Collaboration in Floristic Digital Libraries. In *Proceedings of the Fourth International Workshop on Groupware*, pages 85–95, Rio de Janeiro, 1998.
52. J. A. Sanchez, C. A. Flores, and J. L. Schnase. Mutant: Agents as Guides for Multiple Taxonomies in the Floristic Digital Library. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, pages 244–245, Berkeley, California, USA, 1999. ACM Press.
53. S. Sclaroff, M. L. Cascia, and S. Sethi. Unifying Textual and Visual Cues for Content-Based Image Retrieval on the World Wide Web. *Computer Vision and Image Understanding*, 75(1/2):86–98, July/August 1999.
54. T. B. Sebastian and B. B. Kimia. Curves vs. Skeletons in Object Recognition. *Signal Processing*, 85(2):247–263, Feb 2005.
55. T. Sikora. The MPEG-7 Visual Standard for Content Description - An Overview. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):696–902, June 2001.
56. A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-Based Image Retrieval at the End of the Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, December 2000.
57. T. R. Smith. A Digital Library for Geographically Referenced Materials. *IEEE Computer*, 29(5):54–60, May 1996.
58. R. Stehling, M. Nascimento, and A. Falcão. A Compact and Efficient Image Retrieval Approach Based on Border/Interior Pixel Classification. In *Proceedings of the 11th ACM International. Conference on Information and Knowledge Management*, pages 102–109, McLean, Virginia, USA, November 2002.
59. H. Suleman. *Open Digital Libraries*. PhD thesis, Computer Science Department, Virginia Tech, Blacksburg, VA, 2002. http://scholar.lib.vt.edu/theses/available/etd-11222002-155624/unrestricted/odl.pdf.
60. H. Suleman, E. Fox, A. Krowne, and M. Luo. Building Digital Libraries from Simple Building Blocks. Technical Report TR-03-09, Computer Science Department, Virginia Tech, Blacksburg, VA, USA, 2003. http://eprints.cs.vt.edu:8000/archive/00000656/.

61. H. Suleman, E. A. Fox, R. Kelapure, A. Krowne, and M. Luo. Building Digital Libraries from Simple Building Blocks. *Online Information Review*, 27(5):301–310, 2003.
62. M. Swain and D. Ballard. Color Indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.
63. K. Vu, K. A. Hua, and W. Tavanapong. Image Retrieval Based on Regions of Interest. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):1045–1049, July/August 2003.
64. XMLFile. http://www.dlib.vt.edu/projects/OAI/software/oai-file/oai-file.html (as of October 2004).
65. XMLSpy. http://www.xmlspy.com/ (as of October 2004).
66. A. Yoshitaka. A Survey on Content-Based Retrieval for Multimedia Databases. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):56–63, January/February 1999.
67. D. Zhang and G. Lu. Review of Shape Representation and Description. *Pattern Recognition*, 37(1):1–19, Jan 2004.
68. R. Zhao and W. I. Grosky. Narrowing the Semantic Gap – Improved Text-Based Web Document Retrieval Using Visual Features. *IEEE Transactions on Multimedia*, 4(3):189–200, June 2002.
69. X. S. Zhou and T. S. Huang. Unifying Keywords and Visual Contents in Image Retrieval. *IEEE Multimedia*, 4(2):23–33, June 2002.
70. B. Zhu, M. Ramsey, and H. Chen. Creating a Large-Scale Content-Based Airphoto Image Digital Library. *IEEE Transactions on Image Processing*, 9(1):163–167, January 2000.