An OAI-based Digital Library Framework for Biodiversity Information Systems

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

[1] Institute of Computing, University of Campinas,
13081-970 Campinas SP Brazil
{rtorres, cmbm}@ic.unicamp.br

[2] Department of Computer Science, Virginia Tech,
Blacksburg, VA, USA, 24061
{mgoncalv,fox}@vt.edu

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 such data in an integrated fashion, and integration is often based on geographic coordinates alone. Furthermore, such systems do not fully support image content management (e.g., photos of landscapes or living organisms), 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 distinct biodiversity and image information systems to combine information extracted from them. This cumbersome operational procedure is forced on users by lack of interoperability among these systems. 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 (DL) innovations to integrate networked collections of heterogeneous data. It focuses on creating the basis for a biodiversity information system under the digital library perspective, combining new techniques of content-based image retrieval and database query processing mechanisms. This approach solves the problem of system switching, and provides users with a flexible platform from which to tailor a BIS to their needs.

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 to 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) under homogeneous management.

A wide variety of environmental information systems - EISs - ([18, 12]) is being developed to answer this demand. However, they can seldom meet all end-user requirements: not only are the data sources involved very disparate, and hard to integrate, but user needs (and, indeed, users) cover a very broad spectrum. Therefore, scientists concerned with environmental issues must seek support from a large set of EIS, as well as from other systems. This, of course, brings about all kinds of interoperability problems due to system mismatch, data diversity, and variety of user profiles.

One good 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 (BIS) and image information systems. The latter are software that allow users to manage images' content (e.g., patterns, color, texture). In the biodiversity context, they are adopted by scientists for their image archives and to help them identify species.

A BIS (e.g., [3, 4, 1]) 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 biodiversity information systems 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. Besides being heterogeneous in nature (encompassing flora and fauna and the geophysical description of their habitats), these data also are heterogeneous in other aspects – such as spatio-temporal granularity or storage format. An example of a standard spatial query in a biodiversity system is "Show the areas in Brazil where the plant species *Acacia polyphylla* has been observed".

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, being 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 plant species *Acacia polyphylla*".

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

The goal of the research presented in this paper is to combine research on image processing, databases, and digital libraries to provide biodiversity researchers with a BIS that provides seamless integration of queries involving both image content and textual data. In such a context, users will just need to provide an image as input (e.g., the photo of a plant leaf) and request the system to "Show the areas in Brazil where the plant species *Acacia polyphylla* coexists with plants whose leaves are shaped like those in the photo".

Our previous work in this direction has concentrated on image metadata, and image processing and analysis techniques for extracting appropriate descriptors from species' images [9, 10, 11], with a prototype implemented. Our present focus is to combine these with digital library (DL) facilities.

To this purpose, we present a generic digital library architecture for managing heterogeneous data on living beings and their ecosystems. These data involve not only textual and location features, but also images of these beings and of their environments. A key notion considered is that of *DL component*, a specially designed software module that encapsulates specific functionality, thereby supporting modularity, flexibility, and reuse in constructing DL infrastructure. Due to its DL component-based design, this architecture circumvents the interoperability and system-switching problems discussed. To illustrate this architecture, it has been instantiated to support the creation of a BIS for fish species.

The main contributions of this paper are the following: (a) a generic architecture for managing heterogeneous collections, based on digital library components, to access heterogeneous biodiversity data sources (text and images), that allows combining text-based and content-based queries in a seamless way; and (b) a new component, for content-based image search, 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 paper proposes a generic architecture for managing heterogeneous biodiversity data in an integrated fashion. The starting point for the solution is the assumption that the source data are stored in a network of heterogeneous collections organized in a digital library. For interoperability and performance reasons, data are primarily accessed via their descriptors. This architecture takes into account two kinds of collections: image-related database and domain-specific database.

2.1 Main Modules

Figure 1 shows the digital library architecture proposed for managing these heterogeneous collections. This architecture includes a set of search services (service providers) which are executed over heterogeneous data collections (data providers).

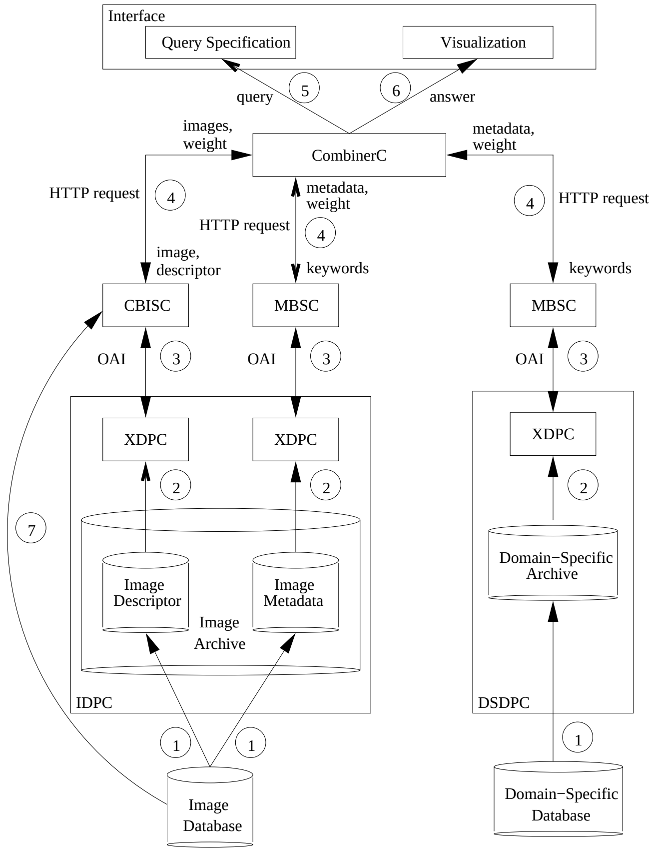

Figure 1: System Architecture.

This architecture has been implemented by using a set of digital library components which have been developed at Virginia Tech. It uses the Open Archives Initiative (OAI) protocol [25, 21] 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 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 the metadata from data providers, to process it, 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 (e.g., ecosystem) 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 *Domain-Specific Archive* for the domain-specific collection. Archives contain metadata and image descriptors, which speed up retrieval of the original data sources. This process is performed by batch programs that convert the original data sources into XML files. *XML Data Provider Components (XDPC)* are used to publish these XML files as OAI archives (arrows 2), so that these XML files can be accessed through OAI requests (arrows 3). An archive and its respective data provider can be seen as a complex data provider component. The complex data provider for the image collection is named *Image Data Provider Component (IDPC)*. The *Domain-*

Specific Data Pro-vider 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 *Content-Based Image Search Component (CBISC)* handles image content descriptors. A *Metadata-Based Search Component (MBSC)* handles both image metadata and domain-specific information.

These search components are activated by the *Combiner Component (CombinerC)*. The Combiner receives a query as input (arrow 5), dispatches this query to the search components, combines their results in a suitable way, and then returns a final answer to the interface layer (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 [37]. The following is 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 by 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 foreseen in the proposed 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* [38, 34] is an OAI-based component which is being used as the XML Data Provider in the proposed architecture. Basically, *XMLFile* is a Perl module that creates an OAI-compliant repository (data provider) to publish a system of XML files as an OAI archive. Its layout and configuration provides a clean separation between the data provider engine, the configuration data, and the data being published.

2.3 Search Components

The proposed 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* search engine [13] is being used as our metadata-based search component. *ESSEX* is a componentized vector-space search engine optimized for digital libraries. *ESSEX* acts as the core portion of an Open Digital Library (ODL [34]) search component, answering requests transmitted through an extended OAI (XOAI) protocol. *ESSEX* was primarily developed for the CITIDEL (Computing and Information Technology Interactive Digital Educational Library) project [7], and now also is being used in the *PlanetMath* project [27]. 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 of both 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 support queries based on image content. This component can be used for building special image information systems called *Content-Based Image Retrieval (CBIR) systems*. These systems can be characterized as follows: assume that we have an image database containing a large number of images. Given a user-defined query pattern (e.g., a query image), retrieve a list of the images from the database which are most "similar" to the query pattern according to the image content (i.e., the objects represented therein and their properties, such as shape, color, and texture).

A typical *CBIR* solution requires 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* (distance metric) to compute the similarity between features of two images by computing the distance between the corresponding vectors. The similarity measure is a *matching function*, often based on the Euclidean distance, that gives the degree of similarity for a given pair of images represented by their feature vectors. Usually, the degree of similarity between two images is defined as an inverse function of the distance metric, that is, the larger the distance value, the less similar the images.

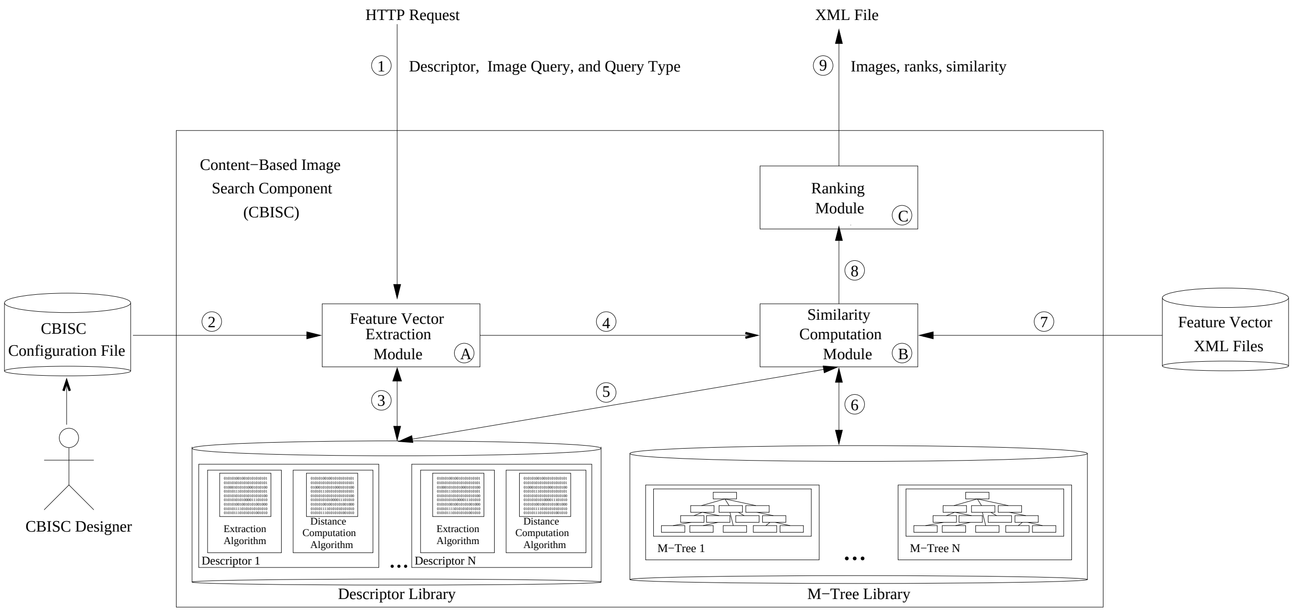

Figure 2: *CBISC* architecture.

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). In the following, the query image feature vector is used to rank the database images according to their similarity (based on a metric distance) to the query image (module B). This step relies on either performing 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 [6] index structure to speed up retrieval and distance computation. The *M-Tree Library* in Figure 2 is a repository of M-Trees. Finally, the most similar 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 [25, 21], queries are given by way of HTTP requests. However, we generalize to have an extended OAI (XOAI) protocol for image search, that fits into the ODL framework [35]. As is typical with XOAI protocols, this request specifies the Internet host of the HTTP server and gives a list of key-value pairs. Two different 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 descriptor 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
   - in a *range query (RQ)*, the user defines a search radius r to retrieve all database images 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. This process requires three preliminary phases: Image Descriptor Identification, Feature Vector Extraction, and *CBISC* XML Configuration.

- **Image Descriptor Identification**

  The identification of appropriate image descriptors (used in extraction and distance computation algorithms) requires experts to perform a set of off-line experiments. 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*, instead, allows progressive extension of the descriptor base.

- **Feature Vectors Extraction**

  Once suitable descriptors have been identified, their extraction algorithms are executed, generating a set of XML files containing the feature vectors for each image. Again, this step is performed off-line. Figure 3 presents an XML schema for the feature vector information. Basically, a feature vector XML file contains information related to the image name, descriptor name, type of feature vector (1D or 2D curve), the feature vectors themselves represented in terms of a curve (double vectors), and their location. 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 via the *Image Data Provider Component* (arrow 3 in Figure 1).

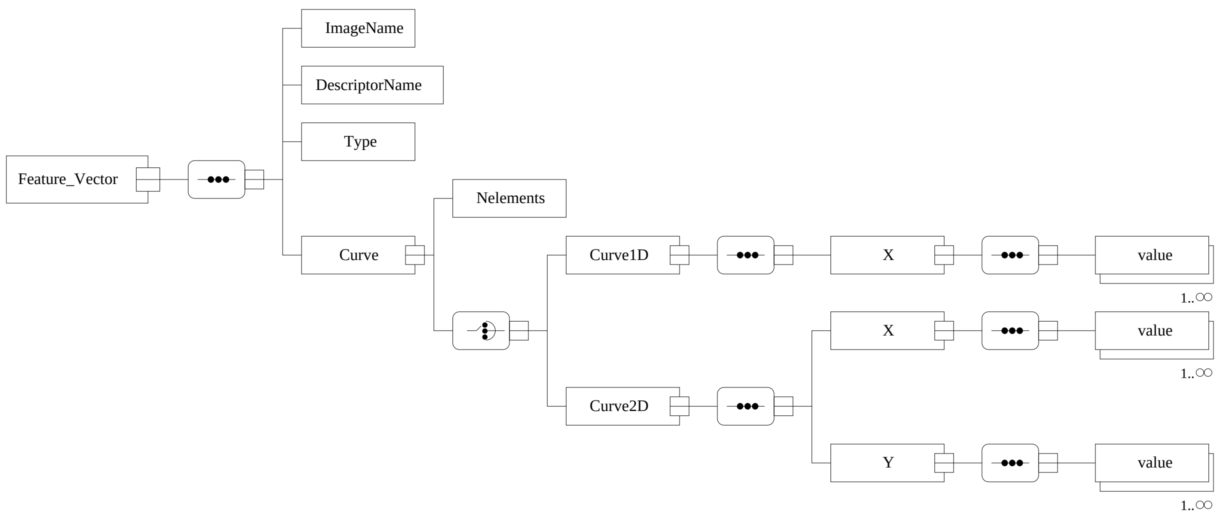

Figure 3: Feature vector XML schema.

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. 1D feature vectors can be generated by, for example, image descriptors like the *Color Histogram* [36] and the *Contour Multiscale Fractal Dimension* [10] shape descriptor. 2D feature vectors can be extracted by, for example, the *Contour Saliences* [11] or the *Curvature Scale Space* [23] 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" [10] 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>
```


Figure 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 concerns the creation of a configuration XML file describing 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 descriptors (extraction and distance computation algorithms) is available in 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)* [9, 10, 11, 2] and color descriptors, such as the *BIC* [33], and the *Color Histogram* [36]. Common metrics like L1 and L2 (Euclidean distance) also are supported.

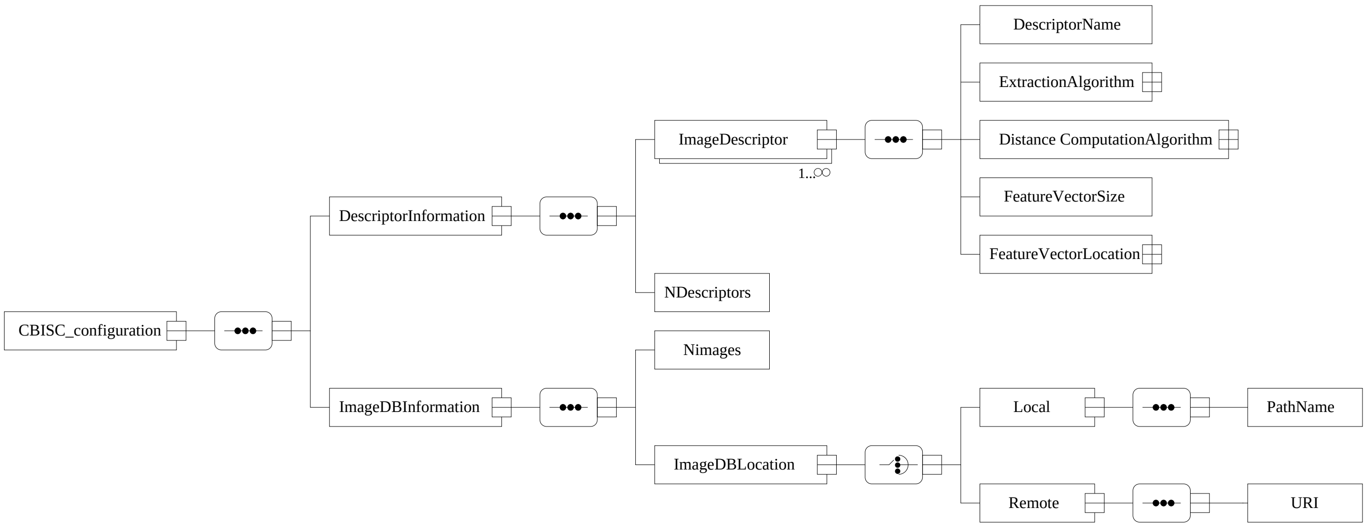

Figure 5: XML schema for the CBISC Configuration file.

Figure 6 presents a screen shot showing the *CBISC Configuration Tool* developed to support *CBISC* designers in the configuration process.

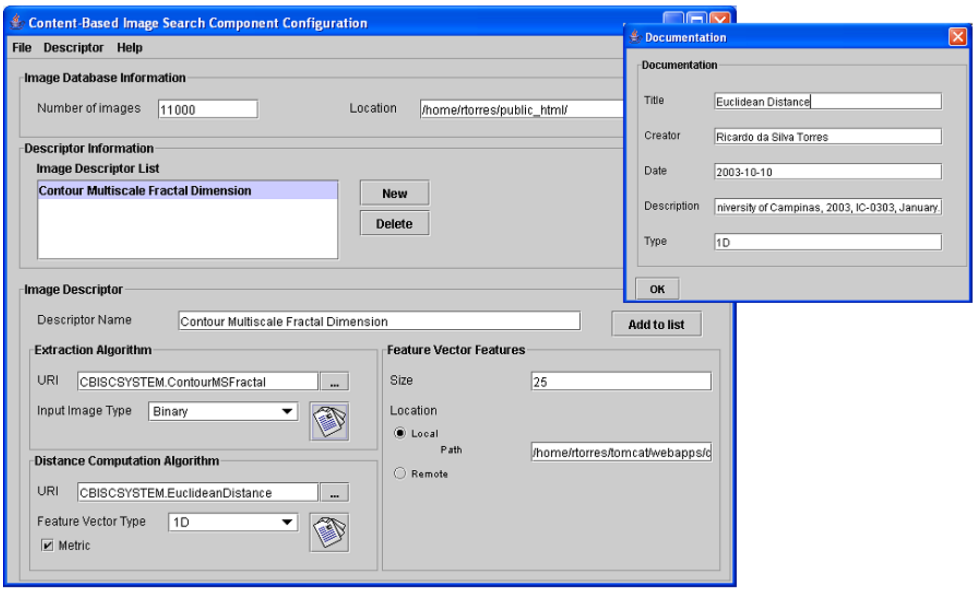

Figure 6: *CBISC Configuration Tool* screen shot.

After the above 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 is related to copying feature vectors and algorithms (extraction and distance computation algorithm) either from local directories or from remote sites (by using OAI requests) to *CBISC* main directories. Both the feature vector and algorithms location are defined in the configuration step.

In fact, the *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, allow BIS designers to easily combine distinct kinds of query features into the system, thereby creating user-tailored BIS for the same underlying archive base.

2.4 The Combiner Component

The Combiner component is responsible for combining three different kinds of evidences: content-based retrieved images, image metadata, and domain-specific metadata. Basically, it receives as input a specification in terms of a query pattern (query image) and/or query terms, forwards these queries to the appropriate search component (*CBISC* and/or *ESSEX*), 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* [17]. *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). An example of a query might start by providing an image as input (e.g., a photo of an observed fish) and then asking 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 a weighted set containing records which match their respective queries.

6. These weighted sets 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 have implemented a Biodiversity Information System concerning fish species. The image data consists of fish photos, and the domain-specific data concerns fish and associated habitat descriptions. With this system, we have carried out experiments to demonstrate the utility of our approach.

3.1 Data Sources

The fish related data were obtained from FishBase [15], an information system available on CD-ROMs, as well as on-line at www.fishbase.org/search.cfm. 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 131 species and 159 images, was used in this work. The following sections describe the archives managed in this biodiversity information system.

3.1.1 Domain-Specific Archive

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

3.1.2 Image Archive

The image archive contains metadata on fish images, and image descriptors. The main challenge of 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.

Current experiments configured the *CBISC* to use the *Beam Angle Statistics (BAS)* [2] shape descriptor.

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

3.2 Experimental Setup

The experiments were intended to evaluate the effectiveness of performance 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 1.02. 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 different combination strategies was evaluated: the *maximization union* and the *summative union* (see Section 2.4). Only the best results are presented in Section 3.3.

3.3 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 now using the summative union approach (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, all curves present a similar behavior. The summative union related curve is better than maximization union one until recall is equal to 0.8. From this point, this 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.

Figure 8 shows the Precision versus Recall graphs for queries involving the *CBISC*. Seven different kinds of queries are evaluated: queries 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 +*

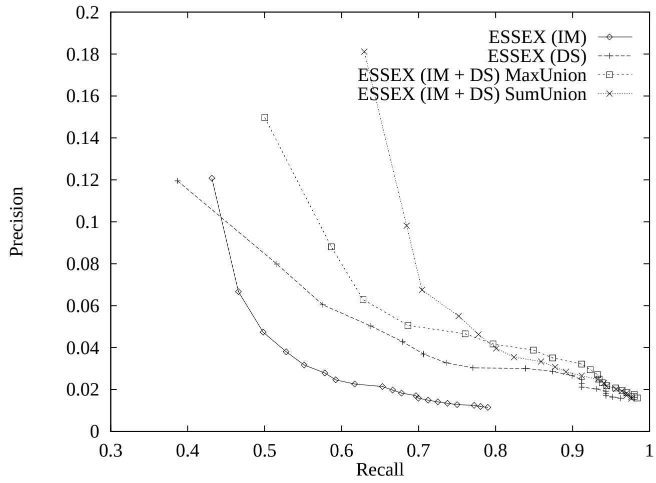

Figure 7: Precision versus Recall curves for text based queries.

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*. Note that the queries which used the summative union approach yield the best results. In fact, the best result (curve *CBISC + ESSEX (IM + DS) SumUnion*) concerns the combination of the three available sources of evidence, using summative-based method. The combination strategies involving the maximization union strategy only presents a better behavior than the curve which considers the use of *CBISC* separately, for recall values between 0.90 and 0.95.

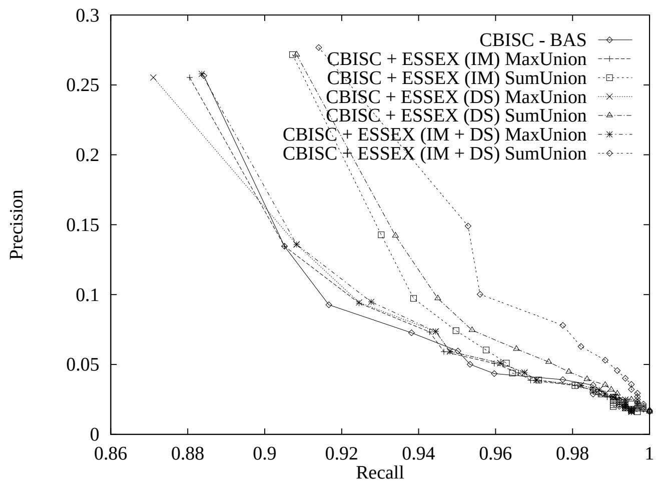

Figure 8: Precision versus Recall curves for queries involving the *CBISC*.

The better performance of the summative union method with the three sources further validates our assumption that a combination of several and 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.

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. There are some other DL initiatives for the same domain. One example concerns floristic digital libraries (FDL) [29, 28, 30]. 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. FDL uses 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 [19]. 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 an 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.* [41], 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 [32] and its successor (the Alexandria Digital Earth Prototype System (ADEPT) [20]) 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.

In the video retrieval domain, Christel *et al.* [5] 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 the 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 [22, 24, 39, 40, 31]. Refs. [31, 39] combine textual information with visual contents by using *Latent Semantic Indexing (LSI)* and *Singular*

Value Decomposition (SVD) to support image retrivel on the WWW. The combination strategy of Nakagawa *et al.* [24] 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 [40]. 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 determined keyword appearing for a given image. An algorithm for the learning of word similarities during a relevance feedback process also is presented. Finally, Lu *et al.* [22] propose a strategy based on semantic networks and relevance feedback to deduce and utilize the images' content for retrieval.

The aforementioned solutions either are too complex [31, 39] to be easily configured for a new domain, or rely on search process techniques (relevance feedback, word similarity learning, content semantic definitions) [24, 40, 22], which are not available in the proposed architecture. Note that, new combination strategies easily can be adopted in the proposed architecture. New combiners just have to follow the HTTP-based communication protocol presented here.

5 Conclusions

Interoperability has been a central research area in the digital library domain [26]. The OAI protocol has been used to promote interoperability solutions for different digital libraries initiatives [21, 16]. 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 a considerable degree of flexibility in data management. To illustrate our claim that this architecture can be applied to several domains, this paper describes its application in building a biodiversity information system on fish species. 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 the 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 trained designer, who carries out a clearly defined set of pilot experiments. It supports the use of different image descriptors (metric and non-metric; color, texture and shape descriptors; with 1D or 2D feature vectors), which can be chosen from the pilot experiment, and then easily combined to yield improved effectiveness. Besides, it encapsulates a multidimensional index structure to speed up the search process, that also can be easily configured for different image collections.

In order to validate the proposed architecture, we performed experiments concerning the combination of textual and image content information. 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. Furthermore, on average, better results were found by using the summative union combination strategy.

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 archeology domain [14]. In this case, the image collection comprises photos of archaeological artifacts and the domain-specific collection corresponds to both archaeological site information and artifact descriptions. Future work 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 [36, 33, 10, 23] in the combination process. New combination strategies based on belief networks [8] will be investigated too.

6 Acknowledgments

This work was supported by FAPESP (Proc. 01/02788-7), CAPES (Proc. BEX844/03-9), the Brazilian PRONEX/SAI and WebMaps projects, CNPq, and NSF (DUE-0121679 and ITR-0325579). Marcos Andre Gonçalves is supported by a fellowship from AOL.

References

[1] S. Analyst. Species Analyst. http://www.speciesanalyst.net (as of December 2003).

[2] 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, 2003.

[3] BioGIS. BioGIS - Israel Biodiversity Information System. http://www.eti.uva.nl/ (as of December 2003).

[4] BIOTA/FAPESP. SinBiota (Sao Paulo State's Biodiversity Information System). http ://www.biota.org.br/sia (as of December 2003).

[5] M. G. Christel, A. M. Olligschlaeger, and C. Huang. Interactive Maps for a Digital Video Library. *IEEE Multimedia*, 7(1):60–67, 2000.

[6] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. *VLBD Journal*, pages 426–435, 1997.

[7] CITIDEL. Computing and Information Technology Interactive Digital Educational Library, 2003. http://www.citidel.org (as of December 2003).

[8] T. A. S. Coelho, P. P. Calado, L. V. Souza, B. Ribeiro-Neto, and R. Muntz. Image Retrieval Using Multiple Evidence Ranking. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):1–10, July/August 2003.

[9] R. da S. Torres, A. X. Falcão, and L. da F. Costa. Shape Description by Image Foresting Transform. In *14th International Conference on Digital Signal Processing*, volume 2, pages 1089–1092, Santorini, Greece, July 2002.

[10] R. da S. Torres, A. X. Falcão, and L. da F. Costa. A Graph-based Approach for Multiscale Shape Analysis. *Pattern Recognition*, 2004. To appear.

[11] R. da S. Torres, E. M. Picado, A. X. Falcão, and L. da F. Costa. Effective Image Retrieval by Shape Saliences. In *Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI03)*, pages 49–55, São Carlos, São Paulo, Brazil, 12-15 October 2003.

[12] ERIN. Environmental Resources Information Network. http://www.tap.org/erin/ (as of December 2003).

[13] ESSEX, 2003. http://br.endernet.org/~akrowne/elaine/essex/index.html (as of December 2003).

[14] ETANA. Managing Complex Information Applications: An archaeology Digital Library, 2003. http://feathers.dlib.vt.edu (as of December 2003).

[15] FishBase. www.fishbase.org (as of December 2003).

[16] M. A. Goncalves, R. K. France, and E. A. Fox. MARIAN: Flexible Interoperability for Federated Digital Libraries. In *5th European Conference on Research and Advanced Technology for Digital Libraries, ECDL-01*, pages 173–186, Germany, 2001.

[17] 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, September 2002.

[18] O. Gunther. *Environmental Information Systems.* Springer Verlag, Berlin/Heidelberg/New York 1998.

[19] 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.

[20] 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.

[21] C. Lagoze and H. V. de Sompel. The Open Archives Initiative: Building a Low-Barrier Interoperability Framework. In *Joint Conference on Digital Libraries*, pages 54–62, Roanoke, VA, USA, June 2001.

[22] 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.

[23] F. Mokhtarian and S. Abbasi. Shape Similarity Retrieval under a Affine Transforms. *Pattern Recognition*, 35(31-41), 2002.

[24] A. Nakagawa, A. Kutics, K. Tanaka, and M. Nakajima. Combining Words and Object-based Visual Features in Image Retrieval. In *12th International Conference on Image Analysis and Processing (ICIAP'03)*, pages 354–359, September 2003.

[25] OAI. Open Archives Initiative, 2003. http://www.openarchives.org (as of December 2003).

[26] 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.

[27] PlanetMath. http://planetmath.org/ (as of December 2003).

[28] 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 (DL'98*, pages 210–216, Pittsburgh, PA, July 1998. ACM Press.

[29] J. A. Sanchez, L. Fernandez, and J. L. Schnase. Agora: Enhancing Group Awareness and Collaboration in Floristic Digital Libraries. In *Fourth International Workshop on Groupware*, pages 85–95, Rio de Janeiro, 1998.

[30] 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, United States, 1999. ACM Press.

[31] 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.

[32] T. R. Smith. A Digital Library for Geographically Referenced Materials. *IEEE Computer*, 29(5):54–60, May 1996.

[33] R. Stehling, M. Nascimento, and A. Falcão. A Compact and Efficient Image Retrieval Approach Based on Border/Interior Pixel Classification. In *ACM Intl. Conf. on Information and Knowledge Management - CIKM*, pages 102–109, November 2002.

[34] 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.

[35] H. Suleman, E. Fox, A. Krowne, and M. Luo. Building digital libraries from simple building blocks. Technical report, Computer Science, Virginia Tech, 2003. Technical Report TR-03-09.

[36] M. Swain and D. Ballard. Color Indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[37] R. 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 (CIKM03)*, pages 167–174, New Orleans, Lousiana, USA, November 2003. ACM Press.

[38] XMLFile. http://www.dlib.vt.edu/projects/OAI/software/oai-file/oai-file.html (as of December 2003).

[39] 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.

[40] X. S. Zhou and T. S. Huang. Unifying Keywords and Visual Contents in Image Retrieval. *IEEE Multimedia*, 4(2):23–33, June 2002.

[41] 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.