ADAM — A System for Jointly Providing IR and Database Queries in Large-Scale Multimedia Retrieval

Ivan Giangreco Ihab Al Kabary Heiko Schuldt
Department of Mathematics and Computer Science
University of Basel, Switzerland
{firstname.lastname}@unibas.ch

ABSTRACT

The tremendous increase of multimedia data in recent years has heightened the need for systems that not only allow to search with keywords, but that also support content-based retrieval in order to effectively and efficiently query large collections. In this demo, we present ADAM, a system that is able to store and retrieve multimedia objects by seamlessly combining aspects from databases and information retrieval. ADAM is able to work with both structured and unstructured data and to jointly provide Boolean retrieval and similarity search. To efficiently handle large volumes of data it makes use of a signature-based indexing and the distribution of the collection to multiple shards that are queried in a MapReduce style. We present ADAM in the setting of a sketch-based image retrieval application using the ImageNet collection containing 14 million images.

Categories and Subject Descriptors

H.3.4 [**Information Search and Retrieval**]: Systems and Software

General Terms

Multimedia, databases, Boolean retrieval, similarity search

1. INTRODUCTION

With the proliferation of ubiquitous devices for digitally capturing and recording various forms of multimedia, content-based multimedia retrieval has proven to be an important means to search within large collections of images, videos, and/or audio. To address this task, multimedia retrieval systems build on the notion of features that are extracted from the multimedia objects and used to perform the search task.

To date, storage systems for feature data are mainly built in monolithic, file-based ways, largely tailored to the application using the data. Databases leverage these drawbacks, but, in turn, have only limited support for unstructured data or search paradigms from the field of information retrieval.

In this demo, we present ADAM, an innovative approach to seamlessly combine database technology and information retrieval for large collections of multimedia data.

We motivate the functionality of ADAM in the setting of a content-based image retrieval application that uses sketches for retrieving images from a large collection and present the steps necessary for building this application using ADAM. However, ADAM is not bound to this specific retrieval problem or content type, but provides a generic solution for a variety of search-based applications in multimedia retrieval.

A researcher building such an application generally has the choice to either use a database or an IR system. A traditional database usually does not provide a useful solution for similarity retrieval, not only because a DB system has no notion of querying unstructured data, but particularly because it only supports Boolean search and would not be able to handle a partial match search on the multimedia object. On the other hand, an IR system lacks the valuable database features for searching in meta data, such as index structures, query optimization, etc. With ADAM we aim at bridging the gap between both kind of systems and introduce an integrated database and information retrieval system.

2. ADAM

ADAM is based both on the relational database model and the vector space model. Both models allow in combination the representation of structured and unstructured data and the retrieval paradigms inherent to both models.

Schema and data definition.

First, to create the schema underyling the relation that stores the feature vectors, a `CREATE TABLE` statement, as known from SQL, is used. To store the numerical feature vectors, ADAM introduces a `FEATURE` data type. The use of multiple attributes that store various feature types allows a very generic data definition. The table used in the motivating example is created as follows (where `keyword` references a Wordnet id used to tag the multimedia object).

```
CREATE TABLE mm_tbl (
  id INT,
  keyword VARCHAR(10),
  carp FEATURE, ...
);
```

The data is inserted into the table using the `INSERT INTO`

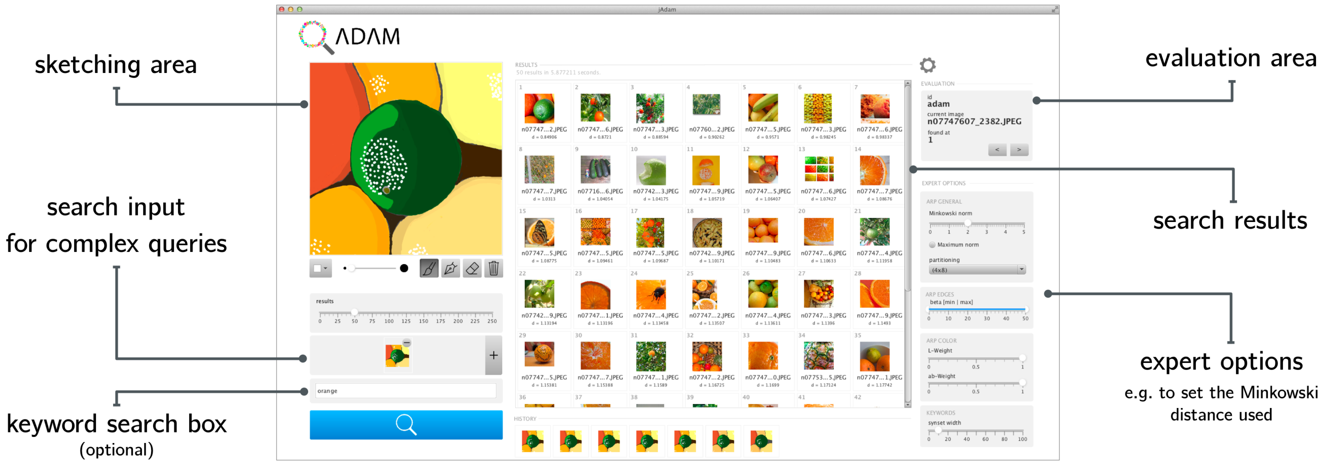

Figure 1: Screenshot of the front-end of the demo.

statement. In ADAM features are represented as `<x1,...,xn>`.

```
INSERT INTO mm_tbl VALUES
(1, toWordnetID('orange'), '<0.78,-0.22,...>', ...);
```

Indexing.

For improving query efficiency, ADAM supports an adapted version of Vector Approximation (VA) File indexing [4]. The idea of VA indexing is to compress the feature vectors using a quantization approach to a short signature and later query the signatures in a sequential manner. ADAM supports the creation of the index using equidistant marks, or marks that build quantization cells with approximately the same number of elements (equifrequent marks).

```
CREATE VA ON mm_tbl(carp) USING EQUIDISTANT MARKS;
```

Retrieval.

For structured data, ADAM applies Boolean filtering predicates (using the WHERE statement). For a similarity-based retrieval, the result objects can be ranked according to a similarity score in the context of a k nearest-neighbour search. Furthermore, ADAM allows the combination of Boolean and similarity retrieval, as we show in the following example.

```
SELECT * FROM mm_tbl
WHERE keyword = toWordnetID('orange')
USING DISTANCE MINKOWSKI(1)(carp, '<0.78,...>')
ORDER USING DISTANCE
LIMIT 10;
```

In this example, the query returns all tuples that match the keyword 'orange', sorts these by a comparison to a given feature vector ('<...>') using the L_1-Minkowski distance and limits to 10 elements returned.

As the example shows, ADAM provides the user with the functions for Minkowski distances. However, it also gives the user the flexibility to specify distance functions herself.

For complex queries, ADAM uses fuzzy theory as a basis to combine distance measures. To model complex retrieval paradigms, such as single-/multi-feature single-/multi-object queries, fuzzy union, intersect and except operations can be used. To allow generality of the system, the fuzzy union and the fuzzy intersect operations can take various functions from the class of triangular conorms and norms, respectively.

Distribution.

To meet the requirements of scalability and efficiency, we deploy ADAM to a distributed environment. For this purpose, the data is sharded over multiple ADAM instances that, at query time, are searched by an orchestrating component in a MapReduce-like way. The orchestrator combines the results to a final result set that is sent back to the querier.

3. DEMOSTRATION OF ADAM

We have implemented ADAM in PostgreSQL and deployed it to 28 small Microsoft Windows Azure instances (1 core, 1.75 GB RAM). To allow the distribution of queries, we have implemented an orchestrating component in Java that executes MapReduce tasks on the ADAM shards to answer the queries.

We present ADAM in an image retrieval use case. The sketch-based retrieval application uses angular radial partitioning (ARP) [1] to constitute 4×4 partitions in an image. A feature vector is constructed containing the first two moments and the joint moment computed in the CIELAB color space for each partition [3]. Thus, for each image, a feature vector of 144 dimensions is stored in ADAM.

In this demo, users will be able to sketch provided images and attempt to find these in the ImageNet collection of 14 million images [2], possibly also by using keywords. On the other hand, users will also get to see the SQL code that is run on the ADAM shards. The system will demonstrate highly efficient querying in large image collections.

Acknowledgments

This work has been supported by the Swiss National Science Foundation, project iMotion, and the Microsoft Windows Azure research grant, project ADAM$^+$.

4. REFERENCES

[1] A. Chalechale, A. Mertins, and G. Naghdy. Edge image description using angular radial partitioning. *IEE Proceedings on Vision, Image and Signal Processing*, 151(2):93–101, April 2004.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image

database. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, CVPR '09, pages 248–255, June 2009.

[3] I. Giangreco, M. Springmann, I. A. Kabary, and H. Schuldt. A user interface for query-by-sketch based image retrieval with color sketches. In *Proceedings of the 34th European Conference on Advances in Information Retrieval*, ECIR '12, pages 571–572, 2012.

[4] R. Weber, H.-J. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proceedings of the 24rd International Conference on Very Large Data Bases*, VLDB '98, pages 194–205, San Francisco, USA, 1998.

APPENDIX

A. DEMO REQUIREMENTS

- table
- poster stand
- electric supply
- if possible: internet connection (otherwise demo can be presented offline with a smaller collection)