THE APPLICATION INTERFACE FOR VIDEO DATABASES

Tomáš Volf

Doctoral Degree Programme (1), FIT BUT

E-mail: xvolft02@stud.fit.vutbr.cz

Supervised by: Jaroslav Zendulka

E-mail: zendulka@fit.vutbr.cz

Abstract: This paper discusses the possibilities of manage multimedia data in terms of databases. It briefly introduces several database systems that are addressed to process them. It also introduces application interface VTApi, which aims to simplify the use of video data and its metadata stored in the database PostgreSQL.

Keywords: VTApi, video database, video metadata.

1 INTRODUCTION

The surveillance systems are widespread more and more all around us. Increasing number of multimedia data (images and videos) requires better and especially efficient methods of storing and querying this data. Trajectories of moving objects are the most interested features in multimedia data of surveillance systems.

In this paper, I introduce an application interface called VTApi, which aims to simplify the use of multimedia data and metadata, which are stored in the database PostgreSQL. To the best of our knowledge, there is no other similar work, which can simplify storing and querying video data and metadata at the level of the implementation.

2 STATE OF THE ART

There are some interesting works, which I briefly introduce. First of them are image retrieval systems. In addition to systems working with images there are systems working with video and MPEG-7 compatible video database management system.

Image retrieval systems are able to organize multimedia semantic content, retrieve image based on similarity and detect images, which are duplicate. **Cortina** is an image search and retrieval system, which retrieves images by similarity and based on categories [1]. It also offers the face detection, image annotation and segmentation tools. At the beginning, some web crawler acquires images. The features are extracted and feature descriptors are computed for each image. They are stored in a flat file structure. Cortina stores textual informations and centroids for the feature clusters.

We can increase a performance of the retrieval system by giving a relevance feedback. Giving the relevance feedback also provides an additional advantage: the precision of results.

Another image retrieval system is an extension for PostgreSQL database system. This extension is called **PostgreSQL with Image-handling Extension (PostgreSQL-IE)**. It uses powerful tools from PostgreSQL and it is independent of the application. One can create new feature descriptors or define new feature vectors without extra cost. PostgreSQL-IE provides two similarity search procedures, specifically range search and K-nearest neighbour (KNN). It provides also twelve feature extraction procedures: one is related to the color and the rest are related to the shape [2].

The International Organization for Standardization proposes a standard for image data type in part 5, called „**Still Image**“, of **SQL/MM**. This proposal of standard does not contain retrieving images by similarity search.

BilVideo is a video database management system, which is designed to provide spatio-temporal queries. In spatio-temporal queries can figure the spatio-temporal conditions, specifically spatial, temporal, object-appearance, external-predicate, trajectory-projection, and similarlity-based object-trajectory conditions by a rule-based system built on knowledge-base [3]. There is also an object-relational database used for querying shape, color and texture features. The BilVideo knowledge-base consists of fact-base and a set of rules, which are written in Prolog. Thanks to the rules in the knowledge-base, fewer of facts are needed to store in terms spatio-temporal querying of the video data. Fact-base includes moreover Prolog facts like object trajectories, object-appearance relations and spatio-temporal relations between video objects. In this approach, they are not explicitly related to semantic units of video [3]. BilVideo provides spatio-temporal, shape, semantic, color and shape queries, so it is very complex video database management system. The query language for spatio-temporal queries is a textual SQL-like query language, which is simple but strong and powerful.

There are also **Multimedia Database Systems (MMDBMS)**, which are used for managing and storing multimedia data for content based retrieval. Most of these systems are common database system enhanced by extenders of basic processing for multimedia data. MPEG-7 Multimedia Database System (MPEG-7 MMDB) is a full-fledged multimedia database system [4]. It is based on extensibility services of Oracle 10g and allows to use, store, index and query multimedia and its content description in the form of standard MPEG-7. MPEG-7 MMDBS supports low-level and high-level multimedia data. It maps types of MPEG-7 schema to database types and sets up new indexing and querying system and a query optimizer, it also offers internal and external libraries for applications. It uses the feature vector with high dimension as an indexing structure for efficient processing of queries. MPEG-7 MMDBS contains Multimedia Indexing Framework (MIF), where we can meet Generalized Search Tree (GiST). GiST allows customize a content-based retrieval functionality to index domain specific content. The main part of MIF is GistService, MIF also partially rely on a GiST framework. The GiST framework allows build any kind of balanced index tree on any kind of data, but it is needed to implement some specific methods for insertion, search and deletion.

Another MPEG-7 compatible video database management system is called **BilVideo-7**. The name is very similar to the above described system BilVideo, but BilVideo-7 is completely new system developed from scratch, which supports multimodal queries [5]. An MPEG-7 profile is used for representing videos decomposed into shots, keyframes, still regions and moving regions. BilVideo-7 provides feature extraction and annotation tool to obtain MPEG-7 compatible XML representations of video. XML representations are stored into a native XML database. Users can make use of location, motion, shape, spatio-temporal, text-based semantic, color and texture queries. Each query may be executed in separate thread – queries are separated to subqueries at first and after they are executed, the results of subqueries are fused to obtain a final query result.

There are some query types, which are not supported by existing systems in a framework compatible with MPEG-7 [5], [6]:

- Content-based queries by example: The user may specify an image, a region or a video segment and the system returns video segments similar to the input.
- Text-based semantic queries: Queries may be specified by a set of keywords corresponding to high-level semantic concepts and relations between them.
- Spatio-temporal queries: Queries related to spatial and temporal locations of objects and video segments within the video.
- Composite queries: These queries may contain any combination of other simple queries.

3 VTAPI

VTApi is an application interface, written in C++, based on PostgreSQL database system. It aims to simplify storing and querying data, especially multimedia data and its metadata. It is oriented to processing image and video information, its categorization, searching and coparison. VTApi uses OpenCV library for computer vision, in development version it uses Postgis and GEOS libraries for managing spatial data.

VTApi is developed on Faculty of Information Technology, Brno University of Technology. The development started in year 2011.

In the next part I introduce some basic terms of VTApi, which are important and crucial for VTApi.

3.1 BASIC TERMS

Dataset is a set of multimedia data with metada, which are the description of multimedia data. Datasets are mutually disjoint, but on the other hand, some dataset can be based on another dataset.

Sequence is a basic unit of dataset, it is a continuous set of video frames or continuous set of images. We assume, that frames or images are timephased.

Interval is a subset of sequence (set of frames), which is suitably chosen so as to be able to assign the same information to this subset of sequence. The metadata may be generally various (numeric values, strings, arrays, structures), but they are always created by a process. An example of interval may be surveilled object in the video or any video scene of its.

Method defines the structure of data, it is a tool or a method, which is not contained in VTApi. One can implement some own method.

Process is a concrete instance of some method, so it inserts data.

Selection is a subset of logically related metada, which are suitably chosen so that operations over them are efficient and allows common concatenation of processes (an output of the first process is an input of the second one). Interval is special case of the selection.

KeyValues is a mechanism of pairs `<key, value>` , it is a crucial way of organizing metadata in VTApi. This data structure is independent of the implementation of VTApi, so the change in the definition of data will not cause the need for change the code of the application interface.

3.2 SIMPLIFIED OBJECT MODEL

The simplified object model is shown in Image 1. *VTApi* class is the entry class of VTApi and it holds common settings and it provides connection to the PostgreSQL database system. As you can see, almost all classes inherit attributes and operations from class *KeyValues*. This class is most general and provides a basic functionality, which ensures a principle of KeyValues pairs. There are some inner classes in class KeyValues:

- *Commons* class manages common functionality for whole VTApi. It loads a configuration file or parameters from command line, provides a connection to PostgreSQL database and handles warning and errors that occurred.
- *Query* class is a general class for queries. It may be affected by methods *whereString, whereInt, whereFloat,* which compose the WHERE condition of the query.
- *Select* class is used for constructing queries. It is required to call method *next* to obtain new row data including the first row. Data from the row can be obtained by methods *getX*, where *X* is the required data type, array types are also supported – they are marked with letter „A“ in suffix of the methods name. Some examples of methods are: *getString, getInt, getFloatA* and so on.

- *Insert* class allows to insert new data into Sequence, Interval, Process or using the class *KeyValues*. New data are added using *addX* methods, where *X* has the meaning as was mentioned above. Changes are applied using method *addExecute*.
- *Update* class allows changing values of the current data held in *KeyValues* based instances. Data may be changed by *setX* methods, where *X* has the meaning as was mentioned above. Changes are applied using method *setExecute*.

Other classes are derived from *KeyValues* and are specialized for particular purposes.

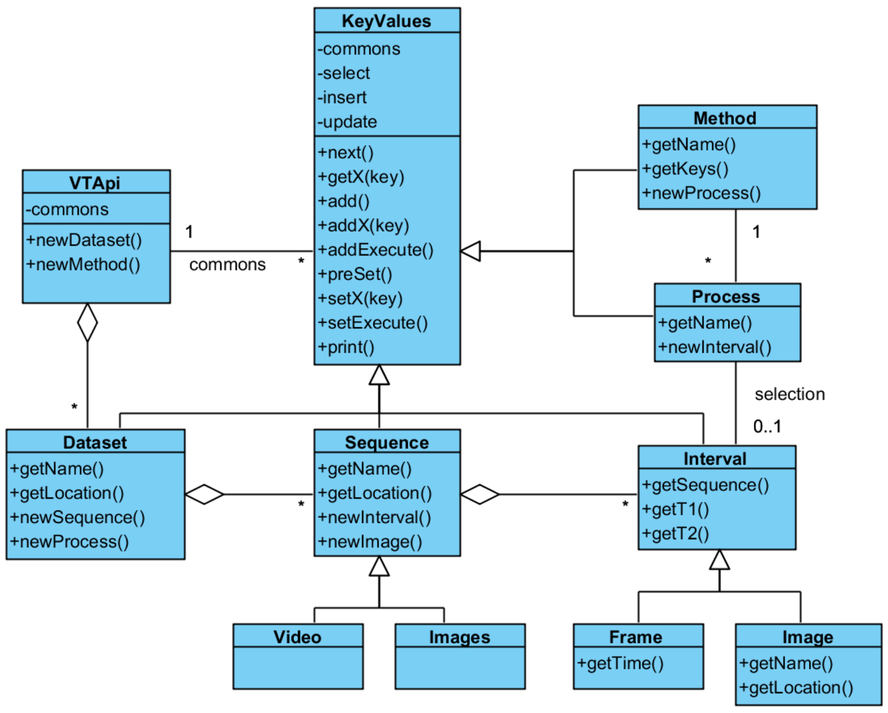

Image 1: Simplified object model of VTApi

3.3 EXAMPLE OF USE VTAPI

The following sample code shows, how can be easily selected a data for tag „dog“ and how can be some data easily changed. There is no need to compose the SQL query. One does not even know the SQL language.

```
VTApi* vtapi = new VTApi(argc, argv);
Dataset* dataset = vtapi->newDataset();
if (! dataset->next()) error("No dataset")

KeyValues* tagskv = new KeyValues(*dataset, "tags");
tagskv->select = new Select(*tagskv);
tagskv->select->whereString("real_name", "dog");
while (tagskv->next()) {
  // ... you may read some data
  cout << tagskv->getString("notes") << endl;

  // ... you may update some data
  tagskv->setString("notes", "changed note for dog tag");
  tagskv->setExecute();
}
```

Image 2: The sample code, how to work with VTApi

At the beginning of this code, an instance of VTApi is created (1), which creates new dataset (2). It is important to call method *next* to obtain first result of datasets (3). This is a general principle to execute queries and retrieve data at the *KeyValues* class and classes derived from *KeyValues*. Once the dataset is selected, the *KeyValues* class created for *tags* in the current dataset can be constructed (4). The next step is to create select query over this instance of *KeyValues* (5). Some specific criteria for this select query may be required, it can be specified by method *whereString* (6). To execute this query and to obtain first row of result from table *tags*, it must be called method *next* (7) on the instance of *KeyValues* as was mentioned above. One can read some data for the current tags, for example the read some string value *note* (8). One can update some values for current *tags* (9). Method *setExecute* confirms all changes for current tag (10). In this case calling the method *setExecute* is optional, because next call of method next confirms these changes (7).

4 CONCLUSION

VTApi aims to simplify working with data, especially spatial and spatio-temporal data and their metadata, stored in database and focuses on processing image and video information. VTApi is quite a newbie, but is still being developed and improved. We develop a second version of VTApi currently, which includes spatio and spatio-temporal support.

ACKNOWLEDGEMENT

This work was partially supported by the research project Tools and Methods for Video and Image Processing for the Fight against Terrorism (VG20102015006), the research plan MSM0021630528, the specific research grant FIT-S-11-2 and the IT4Innovations Centre of Excellence CZ.1.05/1.1.00/02.0070.

REFERENCES

[1] Gelasca, E. D., Ghosh, P., Moxley, E., Guzman, J. D., Xu, J., Bi, Z., Gauglitz, S., Rahimi, A. M, Manjunath, B. S.: Cortina: Searching a 10 Million + Images Database. VLDB ´07, September 23-28, 2007.

[2] Guliato, D., Melo, E. V. d., Rangayyan, R. M., Soares, R. C.: PostgreSQL-IE: An Image-handling Extension for PostgreSQL. Journal of Digital Imaging, vol. 22, no. 2, April 2009, p. 149 – 165.

[3] Dönderler, M. E., Şaykol, E., Arslan, U., Ulusoy, Ö., Güdükbay, U.: BilVideo: Design and Implementation of a Video Database Management System. Multimedia Tools and Applications, 2005, 27, p. 79 – 104.

[4] Döller, M., Kosch, H.: The MPEG-7 Multimedia Database System (MPEG-7 MMDB). The Journal of Systems and Software, 2008, 81, p. 1559 – 1580.

[5] Baştan, M., Çam, H., Güdükbay, U., Ulusoy, Ö.: A MPEG-7 Compatible Video Retrieval System with Integrated Support for Complex Multimodal Queries. IEEE Multimedia 2009, IEEE Computer Society, p. 1 – 30.

[6] Baştan, M., Çam, H., Güdükbay, U., Ulusoy, Ö.: BilVideo-7: An MPEG-7-Compatible Video Indexing and Retrieval System. IEEE MultiMedia, vol. 17, no. 3, September 2010, p. 62 - 73.