Reusing a Compound-Based Infrastructure for Searching and Annotating Video Stories

Nádia P. Koziervitch¹, Jurandy Almeida², Ricardo da S. Torres³, André Santanche⁴, Neucimar J. Leite ⁵
Uma Murthy⁶, Edward A. Fox ⁷

¹Department of Computer Science, Federal University of Technology
85503-390, Pato Branco, PR- Brazil
², ³, ⁴, ⁵Institute of Computing, University of Campinas – UNICAMP
13083-852, Campinas, SP- Brazil
⁶, ⁷Department of Computer Science, Virginia Polytechnic Institute and State University
Blacksburg, VA, 24061-010 – USA

¹nadiap@utfpr.edu.br; ²jurandy.almeida.ic.unicamp.br; ³torres.ic.unicamp.br; ⁴santanche.ic.unicamp.br;
⁵neucimar.ic.unicamp.br; ⁶umurthy.vt.edu; ⁷fox.vt.edu

Abstract- The fast evolution of technology has led to a growing demand for multimedia data, increasing the amount of research into efficient systems to manage those materials. Significant in those efforts is the work being done by the Content-Based Image Retrieval (CBIR) community in processing and retrieving images, along with their further combination with annotations. Nowadays, images play a key role in digital applications. Contextual integration of images with different sources is vital – it involves reusing and aggregating a large amount of information with other media types. In particular, if we consider video data, annotations can be used to summarize textual descriptions and metadata, while images can be used to summarize videos into storyboards, providing an easy way to navigate and to browse large video collections. This has been the goal of a rapidly evolving research area known as video summarization. In this paper, we present a novel approach to reuse a CBIR infrastructure for searching, annotating and publishing video stories, taking advantage of the complex object (CO) concept to integrate resources. Our approach relies on a specific component technology (Digital Content Component - DCC) to encapsulate the CBIR-annotation related tasks and integrate them with video summarization techniques. Such a strategy provides an effective way to reuse, compose, and aggregate both content and processing software.

Keywords- Complex Object; CBIR; Video Stories; Annotation; Reuse

I. INTRODUCTION

Recent advances in technology have facilitated the creation, storage, and distribution of digital resources. It led to an increase in the amount of multimedia data deployed and used in many applications, such as search engines and digital libraries. In order to deal with those data and to efficiently manage their collections, it is necessary to develop appropriate information systems.

For over two decades, significant research efforts have been spent by the Content-Based Image Retrieval (CBIR) community, focusing on image retrieval and annotation. Currently, they are the basic units of many digital applications. Annotation has been recognized as one of the most important services in digital library systems to foster cooperation among users and the integration of heterogeneous information resources [3].

Contextual integration of images with different sources is vital. In particular, if we consider video data, images can be used to summarize videos into storyboards, providing an easy way to understand a video content, without having to watch it entirely, so that a user can quickly judge whether a video story is relevant or not.

In order to reuse and aggregate different resources, complex objects (COs) have emerged, motivating solutions for integration and interoperability. Such objects are aggregations of different information combined together to shape a unique logical object [19, 36]. Particularly, Complex Image Objects (ICO) are a representative example of a data source which is generally integrated with different components, such as links and metadata.

Recently, we developed a component-based CBIR infrastructure aiming to process and encapsulate ICOs. Our contribution relies on the reuse of two basic digital library services (annotation and CBIR) in a compound-based infrastructure to integrate, annotate, search, retrieve, and publish video stories. Our solution is the Digital Content Component (DCC) [38], a self-descriptive unit, which converges the complex object and the software component models in a unified model, to encapsulate content and executable software, and to correlate both.

Those properties of DCCs are exploited by our infrastructure to encapsulate several aspects of annotations and CBIR for a further integration with video material. This framework provides an effective way to reuse, compose, and aggregate both content and processing software. We demonstrate these benefits in a case study with a sample of 50 videos.

A preliminary version of this work was presented at IEEE IRI 2011 [30]. Here, we introduce several innovations.
First, we redesign the proposed method for supporting user annotations, searching and publishing. Additionally, we present an in-depth discussion of the state-of-the-art solutions. Finally, we demonstrate the use of annotations, searching, and publishing in a case study.

The remainder of this paper is organized as follows. Section 2 introduces basic concepts and describes related work. Section 3 presents an overview of our solution. Section 4 summarizes a case study. Finally, we offer our conclusions and directions for future work in Section 5.

II. BASIC CONCEPTS AND RELATED WORK

The fast evolution of technology has aided the exponential increase in volume and variety of data sources. As a result, specialized information systems (such as digital libraries) have been developed to address the large scale requirements of integration and exchange of media. This section outlines previous works on images and CBIR, superimposed information, DCC, and videos and storyboards.

A. Images and CBIR

In general, two different approaches have been applied to allow searching on image collections: one based on image textual metadata and another based on image content. The second approach, Content-Based Image Retrieval (CBIR) [12], proposes the automatic retrieval of images, based on visual properties, such as texture, shape, or color. In this sense, query images are used to search for other images in CBIR systems.

This solution requires the construction of image descriptors, which are characterized by (i) an extraction algorithm to encode image features into feature vectors; and (ii) a similarity measure to compare two images based on the distance between the 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. The larger the distance value, the less similar the images.

Several CBIR systems have been proposed [11], such as QBIC, Chabot, and Photobook. Even though a few of them became commercial products, many CBIR systems were proposed as research prototypes, being developed in universities and research laboratories. Other specific projects compare several CBIR descriptors to enhance performance [37], visualize results [44], or even explore different domains [32]. None of these initiatives have focused on mechanisms to manage complex image objects.

There are several applications which support services based on image content, allowing integration in distinct domains [1, 33]. One example is the digital collection of images present in the Crisis, Tragedy, and Recovery Network (CTRnet) [21]. CTRnet objectives include to develop better approaches toward making technology useful for archiving information about such events, and to support analysis of rescue, relief, and recovery, from a digital library (DL) perspective. CTRnet has several modules, including crawling, filtering, a Facebook application, user visualization, metadata search, and Content-Based Image Retrieval (CBIR). Here, the CBIR module builds upon the testing and evaluation of different descriptors.

In a different context, images are used as the basis to match fingerprints [22]. In this case, algorithms used for processing images are not only related to the implementation of CBIR mechanisms, but also to specific software used to properly represent fingerprint details, such as direction, flow, contrast, and curvature. Along with these parameters, specific details (minutiae) are analyzed on the image, to compare if two different images (from the police records and the crime scene, for example) belong to the same person.

B. Superimposed Information

Superimposed information (SI) [25] refers to new information associated with existing, or base, information (for example, an annotation, term or keyword, a comment about a base document). SI is created to reference subdocuments, or contextualized parts of documents, in base information resources (e.g., annotation on part of an image). By integrating SI, a digital library can provide enhanced support to tasks that involve working with subdocuments, in our context to the task of identifying details on images. As a result, annotations (and the subimages to which they refer) become separately accessible, searchable, and manageable within an application.

Superimposed applications (SA’s) [26, 28, 30] enable working with superimposed information. SA’s allow the manipulation and creation of superimposed information, typically to highlight, annotate, draw, select, organize, connect, or reuse the information [34]. Examples include notes organization tools, such as SLIMPad [13], Mashomatic [29], RIDPad [30], and a multimedia presentation editor and player, SIMPEL [31]. However, none of these SA’s has been used to manipulate video frames to develop storyboards.

The research described in this paper differs from related research in the sense that it takes advantage of a compound-based infrastructure to provide mechanisms for image annotation and search in video stories. Furthermore, the use of a component technology ensures appropriate encapsulation of data and procedures, which allow reuse of the developed component in other Digital Library initiatives involving image annotation.

C. Videos and Storyboards

A video sequence is generally composed by a huge number of frames. For example, 1,500 images are necessary for producing one minute of video at a frame rate of 25 fps, which is the minimum speed for humans to not perceive any discontinuity in the video stream. This enormous volume of video data is a barrier to many systems for search-and-retrieval of video sequences [42].

Making efficient use of video information requires that data are accessed in a usable way. For this, it is important to provide users with a concise video representation of video content without having to watch it entirely. In this way, a user can quickly decide whether to watch the entire video or...
not. This has been the goal of a quickly evolving research area known as video summarization [14].

A video summary is a short version of a video sequence. There are two different kinds of video summaries: static video abstract, which is a collection of video frames extracted from the original video; and dynamic video skimming, which is a set of short video clips, joined in a sequence and played as a video [9].

One advantage of a video skim over a video abstract is the ability to include audio and motion elements that potentially enhance both the expressiveness and information of the summary. In addition, it is often more entertaining and interesting to watch a skim than a slide show of frames. On the other hand, since they are not restricted by any timing or synchronization issues, once video frames are extracted, there are further possibilities of organizing them for browsing and navigation purposes, rather than the strict sequential display of video skims [7].

A comprehensive review of video summarization approaches can be found in [42]. Some of the main ideas and results among the previously published methods are briefly discussed next. In this paper, we are interested in research works that manipulate video sequences using a collection of static images, also known as a storyboard.

In addition to facilitating the search-and-retrieval of video sequences in large collections [10], storyboards also help the user to interact with a single video in a nonlinear manner [16]. They enable quick access to relevant positions of a video sequence, which is particularly useful in video editing and authoring applications [40].

Moreover, storyboards also can be used to highlight events in review services, especially for sports and TV programs [17]. Since only the essential information of a video sequence is retained, they significantly improve bandwidth, storage, and viewing time. In this way, users can share, digest, and enjoy video stories.

In computational terms, the use of a concise representation of a video sequence allows for significantly reducing the computational overhead of many compound tasks, for instance, medical diagnosis [15] and content filtering [43].

D. Complex Objects and Digital Content Component

Some authors name the integration of resources into a single digital object as Aggregation [45], a Component-Based Object [38], a Complex Object [35, 46] or yielding a Compound Object [9]. We will adopt the same definition of structuring digital objects present in [9]; atomistic, compound, and complex. The atomistic approach is when the user has a single file (whether made up from a single or multiple text files) from a preferred format. The compound approach is made up from multiple content files, which may have different formats. A complex object is described using a network of digital objects within the repository.

In particular, Santanché used the idea of COs in the field of software reuse and exchange [38] in a component-based technology named Digital Content Component.

Digital Content Component (DCC) [38, 39] refers to a model and a technology that implements it. It has a capsule model, of which internal structure is organized as a complex object (CO) and an external structure is built as a software component. The internal CO has a hierarchical containment structure, able to store and relate any kind of digital content, including executable software. This structure is based on a synthesis and generalization of related CO standards: Reusable Asset Specification -RAS, MPEG-21, METS, and IMS CP. Externally any DCC acts as a software component, declaring required and provided interfaces. When a DCC encapsulates executable software (Process DCC), it behaves as a software component. However, a DCC can also encapsulate other kinds of content, without executable software included (Passive DCC).

DCC has been used in the management of sensor data for scientific applications [18] (working on the management of access, operations, processing, and metadata), and an authoring tool based on components [38]. Regarding advantages of DCC over other technologies, we mention the use of ontologies, the encapsulation of software, and the description of an interface to manipulate the information. The comparison of DCC with other CO technologies, such as OAI-ORE [24], highlight that they manage the same concepts (such as unique identifiers, components, encapsulation, how to structure their parts, etc.), therefore they can be mutually mapped.

A DCC is composed of four distinct subdivisions (Fig. 1):

- (a) content: the content itself: data or code in its original format, or another DCC;
- (b) structure: the declaration of a management structure that defines how components within a DCC relate to each other, in XML;
- (c) interfaces: a specification of the DCC interfaces using open standards for interface description - WSDL and OWL-S (semantics); and
- (d) metadata: metadata to describe version, functionality, applicability, and use restrictions - using OWL.

Digital Content Component (DCC) refers to a model and a technology that implements it. It has a capsule model, of which internal structure is organized as a complex object (CO) and an external structure is built as a software component. The internal CO has a hierarchical containment structure, able to store and relate any kind of digital content, including executable software. This structure is based on a synthesis and generalization of related CO standards: Reusable Asset Specification -RAS, MPEG-21, METS, and IMS CP. Externally any DCC acts as a software component, declaring required and provided interfaces. When a DCC encapsulates executable software (Process DCC), it behaves as a software component. However, a DCC can also encapsulate other kinds of content, without executable software included (Passive DCC).

DCC has been used in the management of sensor data for scientific applications (working on the management of access, operations, processing, and metadata), and an authoring tool based on components. Regarding advantages of DCC over other technologies, we mention the use of ontologies, the encapsulation of software, and the description of an interface to manipulate the information. The comparison of DCC with other CO technologies, such as OAI-ORE, highlight that they manage the same concepts (such as unique identifiers, components, encapsulation, how to structure their parts, etc.), therefore they can be mutually mapped.

A DCC is composed of four distinct subdivisions (Fig. 1):

- (a) content: the content itself: data or code in its original format, or another DCC;
- (b) structure: the declaration of a management structure that defines how components within a DCC relate to each other, in XML;
- (c) interfaces: a specification of the DCC interfaces using open standards for interface description - WSDL and OWL-S (semantics); and
- (d) metadata: metadata to describe version, functionality, applicability, and use restrictions - using OWL.
III. OVERVIEW OF OUR SOLUTION

From the data perspective, the objective was to integrate videos by summarizing video stories, to obtain images, and correlate them with text annotations, and CBIR information, as shown in Fig. 2.

A. The Compound-Based CBIR Infrastructure

The architecture of our solution is presented in Fig. 3. The bottom layer contains the data sources: the descriptor library, the image collection, and the database. The second layer contains DCCs which encapsulate, provide access, and manage several parts of the CBIR process: the descriptors, the descriptor library, the images, the retrieval from the database, and finally a manager, for setting up the entire process. Note that these DCCs can represent an aggregation of other DCCs, like the DescriptorLibraryDCC managing a collection of DescriptorDCCs. The third layer comprises the applications which manipulate the image COs, by accessing the CBIR process or the data sources.

In particular, the ImageDCC encapsulates one image and its respective textual metadata (such as file name and file location). ImageCODCC aggregates the ImageDCC with the visual metadata (e.g., feature vector name and similarity scores). DescriptorDCC encapsulates the descriptor library and its respective metadata. DescriptorLibraryDCC manages the collection of descriptors. Each DCC has an XML file, representing how the parts are associated with the CO.

As an example, consider retrieval visualization using the CBIR similarity scores for the image shown in Fig. 4. The image, descriptor, and similarity scores are later aggregated by an ICO.

B. Generating Stories from Videos

Humans judge more quickly the relevance of interrelated video segments. However, discovering the ideal relationship for such a judgment is non-trivial. The simplest approach is to group together video frames with similar content, so that a relevant judgment for one video frame could be applied to all near-duplicated ones.

In our approach, stories are the meaningful and manageable units for searching video segments. They consist of multiple shots and are represented by a collection of frames, as illustrated in Fig. 5 and Fig. 6.

Numerous methods of video summarization can be used to produce storyboards. Each type of feature spans a multidimensional feature space. The distance function determines the dissimilarity between images based on features within this space.

Our solution was designed to be flexible and robust, and, therefore, the feature input is not limited to a specific type. The only requirement is that the dissimilarity between
images, as represented by sets of features, must be numerically represented by an appropriate distance metric. In this work, four techniques were used: a method based on Delaunay Triangulation (DT) [27], STIll and MOving Video Storyboard (STIMO) [14], Video SUMMarization (VSUMM) [8], and Vldeo Summarization for ONline applications (VISON) [3].

The Delaunay Triangulation (DT) [27] has several phases. At the beginning, redundant information is discarded by a pre-sampling step and, hence, instead of considering all the video frames, only a subset is taken. Then, Principal Component Analysis (PCA) is applied on a matrix formed by color histograms extracted from the remaining frames, reducing its dimensionality. After that, the Delaunay diagram is built. Finally, the clusters are obtained by separating edges in the diagram.

The STIll and MOving Vi deo Storyboard (STIMO) [14], is a summarization technique designed to produce on-the-fly video summaries. Initially, a pre-sampling step is applied in order to discard a lot of redundant information, taking only a subset of video frames. The remaining frames are then converted into color histograms and stored in a feature-frame matrix. Next, similar frames are grouped together by a clustering method based on an improved version of the Furthest-Point-First (FPF) algorithm. For obtaining the number of clusters, the pairwise dissimilarity of consecutive frames is computed according to Generalized Jaccard Distance (GJD). Finally, a post-processing step is performed for removing meaningless frames from the storyboard.

The Video SUMMarization (VSUMM) [8] is a similar approach to cope with the video summarization problem in which the clustering step is achieved by an improved version of the k-means algorithm. For that, the frames are initially grouped in sequential order instead of randomly distributed between the clusters. Thereafter, the frames are grouped by the traditional k-means algorithm. Finally, one frame per cluster is selected for the summary.

Lastly, the Vldeo Summarization for ONline applications (VISON) [3] is a summarization technique that operates directly in the compressed domain, allowing for online usage. For each frame of an input sequence, visual features are extracted from the video stream for describing its visual content. After that, a simple and fast algorithm is used to detect groups of video frames with a similar content and for selecting a representative frame for each group. Finally, the selected frames are filtered in order to avoid possible redundant or meaningless frames in the video summary.

### C. Reusing the CO-Based Infrastructure for Retrieving and Annotating Stories

Considering the CO-based infrastructure presented in Fig. 3, the goal is to reuse their components, adding new components (Fig. 7): (i) the CBIR Story module, (ii) the story module, (iii) the annotation module, and (iv) the publishing module. Note that different interface services can iterate with different DCCs.

To the developer, this approach facilitates services integration since it reuses resources that are already available. For this, new DCCs were created: StoryDCC and CBIRStoryDCC (Fig. 8), the AnnotationDCC (Fig. 9), and the PublishingDCC.

The StoryDCC is a passive DCC, which is responsible for story aggregation. It aggregates metadata and the collection of video frames that resulted from video summarization. Fig. 8 presents the StoryDCC for the storyboard presented in Fig. 5. Notice that there are 15 ImageCODCCs, encapsulating the 15 images obtained by the summarization method.

The CBIRStoryDCC is a process DCC, centralizing the CBIR search for the video stories, using as source the StoryDCC and the CBIRProcessDCC. Note that the CBIRStoryDCC in Fig. 8 reuses the CBIRProcessDCC previously available, processing each image present in the StoryDCC.
The AnnotationDCC is a passive DCC, which is responsible for all the functions for manipulating the one subimage and the respective text annotation. It interacts with the RetrievalDCC (for retrieving the textual data from the DBMS), and with ImageDCC (to specify the image which is annotated), as shown in Fig. 9. Note that since one image can have several subimages and annotations, it can be associated with several AnnotationDCCs.

The PublishingDCC is a passive DCC, which is responsible for the functions for creating the XML file which will be used later for publishing the data. Note that several combinations can be available here: an XML for publishing only images, annotations, movie stories, or their combinations.

IV. CASE STUDY

First, a sample of 50 videos was randomly selected from the Open Video Project, along with their respective metadata. All videos are in MPEG-1 format (at 352 x 240 resolution and 29.97 frames per second), in color and with sound.

The selected videos are distributed among several genres (e.g., documentary, educational, ephemeral, historical, and lecture) and their duration varies from 1 to 4 minutes. These videos are the same used in [3, 8, 14, 27].

A. The Implementation

Our component-based infrastructure enables the installation of different image descriptors, but for the tests presented, the Border/Interior pixel Classification (BIC) descriptor was used. The library was implemented in C and the DCCs were implemented in Java. The functions and parameters available for each DCC are stored in a PostgreSQL database.

B. Encapsulation of Resources

This section describes two of the implemented DCCs in our case study: the StoryDCC and the CBIRStoryDCC.

The StoryDCC allows the aggregation of the frames resulting from the video summarization. Since a story can have different types of images (depending on the summarization and parameters), this DCC is particularly valuable due to the temporal organization of the frames. Operations include the number of images aggregated, and the the method getStory. DCC metadata includes the movie name. Later this DCC could also include parts of audio files for each story, for example.

The CBIRStoryDCC centralizes the process of running the content-based image retrieval for each image on the story. Operations include methods ProcessImage (with the parameters source image and the distance matrix) and ProcessStory (which interacts with method ProcessImage within an image list). DCC metadata includes the movie name.

C. The Prototype

The case study had the following phases: (i) the “discovery” and definition of each part of the CO; (ii) the identification of the complex object parts; (iii) the video summarization process; (iv) the CBIR process; (v) the encapsulation of the image and related metadata.

All the videos and the storyboards produced by DT [27], STIMO [14], and VSUMM [8] are available online. The video summaries of VISON [3] can be seen at http://www.liv.ic.unicamp.br/~jurandy/summaries/.

In phase two, we defined that the aggregation would comprise the video story concept. For the identification of the compound object parts, we used the database, which matched the images to respective video, metadata, and CBIR information.

In phase three, the storyboard was generated. The CBIR process comprises the execution of an extraction algorithm on an image collection, and later comparison of similarity scores. Other component parts related to the image become available, such as the feature vectors and similarity ranking.

For instance, consider the fifth video stories presented in Fig. 5. The CBIR processing of this image generates a feature vector, and the similarity scores for the other images on the collection. Fig. 10 shows the ranking according to the color comparison.

If we analyze the five top images retrieved, we have the following:

• the first three returned images are related to the same story;
• the fourth image is related to a story in video Drift Ice as a Geologic Agent, segment 03; and
• the fifth image is related to a story in video Exotic Terrane, segment 02.

The same story can have annotations, as shown in Fig. 11. The figure illustrates a specific subimage, along with a mark and the text annotation “Movie a New Horizon, segment 04, white cube”.

1 Http://www.open-video.org

2 Http://sites.google.com/vsummsite/
The basic CBIR and annotation services from the compound-based infrastructure where extended for multimodal retrieval (the interface is presented in Fig. 12), based on re-rank and rank-aggregation algorithms [23]. At the left side, the user selects the image which is used for CBIR retrieval. At the right side, the user inputs the text that will be searched (exemplified by the word “white cube”). The user also can define if (i) the weight percentage ranking among the image and text, (ii) the search is based on similarity or ranking, and (iii) if the text search is based on annotations or movie descriptions.

In the last phase, our CO-based infrastructure provides the encapsulation of the complex image object, aggregating the image, the annotations, the respective feature vector, the similarity scores, and an XML file (as shown in Fig. 13). This phase is particularly valuable when an application needs to be integrated with other applications, with other media, or even for publishing and harvesting purposes.

Part of this framework has been implemented and validated in a case study with a sample of 50 videos randomly selected from the Open Video Project. The benefits of our solution include:

- the uniform management of the video collection, storyboards, metadata, descriptors, annotations, publishing, searching, and processing software through DCC;
- the flexibility of using a CO-based infrastructure to combine components and services; and
- the centralization of image, annotation, publishing, searching, video processing, and packaging into a single infrastructure.

Future work includes the extension of our approach to consider other video features (e.g., video metadata, video skims [4], and motion analysis [5, 6]), algorithms for image comparison (e.g., texture, shape, and local features), performance evaluation, audio search, multimodal search integrated with audio, and comparison with other methods.

ACKNOWLEDGMENT

We thank CNPq, CAPES, and FAPESP for financial support.

REFERENCES


