RC 21903 (98529) 5 December 2000 **Computer Science**

IBM Research Report

On the Evolution of Videotext Description Scheme and Its Validation Experiments for MPEG-7

Chitra Dorai, Ruud Bolle,
IBM Research Division
Thomas J. Watson Research Center
Yorktown Heights, NY 10598

Nevenka Dimitrova, Lalitha Agnihotri, Gang Wei
Philips Research
345 Sscarborough Rd
Briarcliff Manor, NY 10510

Research Division
Almaden - Austin - Beijing - Haifa - T. J. Watson - Tokyo - Zurich

On the Evolution of Videotext Description Scheme and Its Validation Experiments for MPEG-7*

Chitra Dorai[†], Ruud Bolle[†]

Nevenka Dimitrova[‡], Lalitha Agnihotri[‡], Gang Wei[‡]

IBM T.J. Watson Research Center[†]
P.O. Box 704, Yorktown Heights
New York 10598, USA

Philips Research[‡]
345 Scarborough Road
Briarcliff Manor, NY 10510

Abstract

Videotext refers to text superimposed on still images and video frames, and can be used in many MPEG-7 applications that deal with archival and delivery of images and video. It can be used to annotate and index large video and image collections, and enables text based search, automatic video logging, and video cataloging. This paper describes the joint work of IBM and Philips Research Laboratories on designing an MPEG-7 description scheme based on videotext. It describes the elements comprising the Videotext DS and discusses validation experiments performed to demonstrate the effectiveness of the DS in video browsing and classification tasks.

*Appeared in the Proceedings of the ACM Multimedia 2000 Workshop on Standards, Interoperability and Practice, Los Angeles, California, November 2000.

1 Introduction

As content becomes readily available, archiving, searching, indexing and locating information in large volumes of multimedia data becomes an important issue. Videotext (i.e., superimposed text on images and video frames) is an important source of semantic information in the video track. Videotext can be used to annotate and index large video and image databases, and allows for searching, browsing, and random access viewing of videos and images based on the occurrences of videotext as well as on keywords obtained from its recognition. For example, extracting keyframes for constructing visual table of contents can be aided by detecting frames with videotext. These keyframes contain information such as anchor name, football score, introductory and ending credits. Further, the location and size of text in commercials can be used in conjunction with other features for reliable commercial detection and program classification.

In this paper we describe the standardization efforts related to the MPEG-7 Videotext Description Scheme (DS). We present the results of validation experiments of the proposed Videotext DS in the context of efficient video browsing and classification.

At the MPEG-7 meeting in Lancaster, UK, held in February 1999, it was recommended that proposals p183 entitled "Videotext descriptor proposal" from IBM and p639 entitled "Text detection and representation" from Philips be merged. Both proposals addressed the detection of text in videos and images and its representation to enable automated annotation and efficient browsing of videos and images. Subsequently IBM and Philips jointly proposed a videotext descriptor as documented in M4791 [1], which then evolved into the videotext description scheme [2] presented at the Melbourne meeting. At the Noordwijkerhout meeting in March 2000 we presented a work plan, as described in M5740 for core experimental validation of the Videotext DS. The results of the core experiments (CE) were reported in M6081 [3] at the Geneva meeting. Currently the Videotext DS is being integrated into the MPEG-7 working draft.

2 Definition of the Videotext DS

Within the MPEG-7 standard, there exist different description schemes such as the MovingRegion DS that cover the basic video object attributes such as bounding box, trajectory and others. A Videotext DS is proposed as a specific MovingRegion DS. In a nutshell, the Videotext DS inherits all the properties of the MovingRegion DS (attributes, decomposition, descriptors, and DSs). The Videotext DS describes a text region on a still image or a set of video frames. The text region is characterized by the superimposed characters it contains, syntactic attributes of the text such as language, font size, font style, and other temporal and visual information such as its time, color, motion, and spatial location through its derivation from the MovingRegion DS.

The syntax of the Videotext DS in XML-Schema is as follows:

```
<!- ################################### ->
<!- ''Videotext DS'': Syntactic Aspects ->
<!- ################################### ->
<simpleType name="TextDataType" base="string">
    <enumeration value="Superimposed"/>
    <enumeration value="Embedded"/>
</simpleType>
<complexType name="Videotext" base="MovingRegion"
             derivedBy ="extension">
    <element name="Text" type="TextualDescription"
             minOccurs="0" maxOccurs="1"/>
    <attribute name="TextType" type="TextDataType"
               use="optional"/>
    <attribute name="FontSize" type="positiveInteger"
               use="optional"/>
    <attribute name="FontType" type="string"
               use="optional"/>
</complexType>
```

Details of the elements and attributes used in this syntax are the following:

- *TextDataType*: Data type defining the kind of videotext. Two primary kinds of videotext are the superimposed and embedded text. Embedded videotext appears as part of and is recorded with the scene (e.g. street and shop names or text on people's clothing). On the other hand, superimposed videotext is specifically intended to carry and stress important information and is typically generated by video title machines or graphical font generators in studios.
- *Videotext*: Text region in an image or a set of video frames.
- *Text*: Textual description that contains the text string recognized in the videotext region. The textual description data type includes an attribute to specify the language of the text string.
- *TextType*: Attribute that specifies the type of the videotext.
- *FontSize*: Integer that specifies the font size of the videotext.
- *FontType*: String that specifies the font style of the videotext.

```
<!- ########################################## ->
<!- ''VideotextObject DS'': Semantic Aspects   ->
<!- ########################################## ->
<complexType name="VideotextObjectDS"  base="Object"
             derivedBy ="extension"/>
   <attribute name="id"  type="ID"/>
   <attribute name="href" type="uri"/>
   <attribute name="CharacterCode" type="string"/>
</complexType>
```

Videotext has an important role of providing auxiliary informational links to existing objects in videos. For example, when a person object is detected in a frame with a superimposed text region right below the face of the person, we usually infer that the text conveys the person's name. Similarly with scene text, the presence of text on an object shown in a frame indicates the product name, which may not be

inferred from the shape or other attributes of the object (e.g., "Starbucks coffee" on a styrofoam cup.) Therefore the need to establish a relationship between a videotext region and an object in a frame is clear. The VideotextObject DS is a specific object DS and has the additional attribute that provides the character string which relates to the object.

We have included character codes in syntactic (attribute, *Text*) and semantic (attribute, *CharacterCode*) aspects of the Videotext DS. Depending on the type of application both placeholders can be meaningful. For example, in a high level application that uses the semantic information it needs to access only the VideotextObject DS part. On the other hand, the syntactic element of the Videotext DS can be used to signal and describe the presence of videotext, which can be used in the construction of visual summaries for efficient browsing applications. There may arise issues related to inconsistencies between the syntactic and semantic elements of this DS. We propose that in the event of inconsistent data in the character codes of syntactic and semantic parts of the DS, the semantic character codes override the others. We anticipate that there may not be a single solution to address this issue, and will stimulate further discussion within the MPEG-7 Video, DDL, and DS subgroups.

The relationship between the syntactic and the semantic elements of the Videotext DS will be provided using the syntactic-SemanticLinkDS definition to be specified in the working draft.

3 Extraction of the Videotext DS

Extraction of videotext in a frame is the result of image analysis involving text character segmentation and location. This may be done automatically or manually. In this section, we discuss how videotext can be extracted automatically from digital videos. Text can appear in a video anywhere in a video frame and in different contexts. The algorithms presented here are designed to extract superimposed text and scene text which possesses typical (superimposed) text attributes. No prior

knowledge about frame resolution, text location, font styles, and text appearance modes such as normal and inverse video are assumed. Some common characteristics of text are exploited in the algorithms including monochromaticity of individual characters, size restrictions (characters cannot be too small to be read by humans or too big to occupy a large portion of the frame), and horizontal alignment of text (preferred for ease of reading).

Approaches to extracting text from videos can be broadly classified into three categories: (i) methods that use region analysis [4], (ii) methods that perform edge analysis [5, 6], and (iii) methods that use texture [7]. The algorithms available for the MPEG-7 experimental software include the region-based algorithm from IBM [4] and the edge-based algorithm from Philips [6]. The IBM algorithm [4] works by extracting and analyzing regions in a video frame. The processing stages in this system are: (i) isolating regions that may contain text characters, (ii) separating each character region from its surroundings and (iii) verifying the presence of text by consistency analysis across multiple text blocks. Optionally, if consecutive frames in videos are being processed together in a batch job, then text regions determined from say, five consecutive frames can be also analyzed together to add missing characters in frames and to delete incorrect regions posing as text. This interframe analysis used by the IBM system [8] exploits the temporal persistence of videotext in videos, and it involves examination of the similarity of text regions in terms of their positions, intensities and shape features and aids in omitting false positive regions. The Philips algorithm [6] for text detection exploits the text properties namely, the height, width and area on the connected components (CC) of the edges detected in frames. The processing stages in this system are: (i) image enhancement and edge detection, (ii) connected component analysis for character detection, and (iii) connecting multiple characters together for text line detection. Horizontal alignment is used to merge multiple CC's into a single line of text. The output is a binary image of detected text lines with text as foreground in black on a white background. This can be the input to an OCR [9] which recognizes the text characters.

4 Validation of the Videotext DS

The applications used to demonstrate the utility of the Videotext DS highlight two typical scenarios: (i) video browsing scenarios in which interesting events in the video (i.e., the presence of videotext as an indicator of information pertaining to persons, locations, product advertisements, sports scores) are detected automatically and the video content is browsed based on these events; (ii) video classification scenarios based on videotext.

The core validation experiments for the Videotext DS used a selection of video segments from the MPEG-7 content set that were encoded using the MPEG-1 compression standard. The event-based video browsing experiment used CM1002.mpg from CD 16 and news2.mpg from CD 18. The video classification scenario made use of jornaldnoite2.mpg from CD 15; news1.mpg from CD 17; news2.mpg from CD 18; and CM1002.mpg from CD 16.

4.1 Methodology

The experimental methodology involved the following steps:

1. Define DDL for the Videotext DS in XML-Schema.
2. Select audio-visual material from MPEG-7 content set for core experiments.
3. Analyze selected content by executing automatic text extraction-algorithms to extract videotext, their locations in terms of bounding boxes, and their time information. Annotate each frame with information related to superimposed text present in the frame by generating the XML description of the Videotext DS.
4. Develop visual browsing and classification applications based on the XML description of the Videotext DS. The browsing application demonstrates random access to the annotated videotext segments through a graphical user interface. The classification application demonstrates higher-level labeling of video using videotext and other visual features and its use in automatic segment classification.

5. Evaluate browsing and classification of selected content with and without the Videotext DS. Video browsing is investigated to determine its efficiency given the absence or presence of videotext annotation. The classification accuracy is determined on the basis of improvements resulting from the use of videotext.

6. Identify the missing components and add modifications to the Videotext DS.

4.2 Measurements

We derived results relating the performance of an automatic videotext event detection algorithm to video browsing efficiency from theoretical considerations. Quantitative evaluations were made on the improvement of video classification accuracy using videotext as compared to classification using other features. The classification experiment using the Philips' algorithm categorized video clips into predefined categories, e.g., commercial, news, sitcom and soap based on face and text locations. We measured the performance of the classifier using face only and using both face and text. We extracted face and text information from the video and stored them in the proposed XML format. We then parsed this information and performed the classification using a Hidden Markov Model classifier. In our experiments we have achieved 85% accuracy for video classification using both face and text location as opposed to the 65% accuracy using only faces.

5 Experimental Results

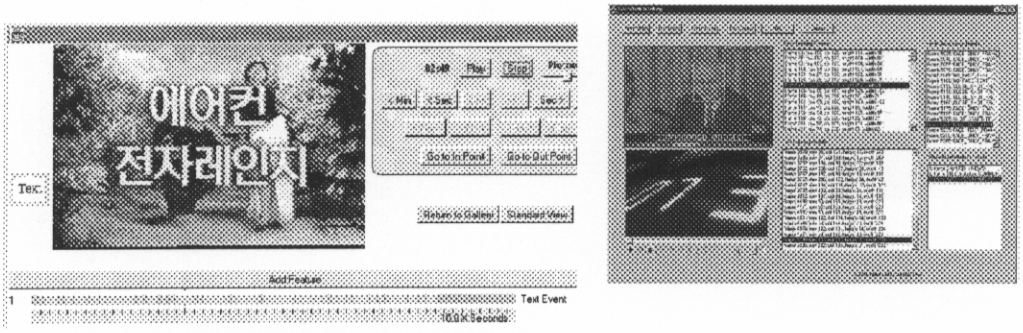
The applications demonstrating the effectiveness of the Videotext DS are outlined in this section.

5.1 Browsing Based on Videotext Events

For the first experiment, we applied our automatic videotext event detection technique to the test video streams, namely, CM1002.mpg and news2.mpg. The auto-

matic text event detection system first analyzes each frame in the stream, locates and segments characters composing the text superimposed on a frame, when present, using the technique described in [4]. It outputs the total number of text regions in a frame along with the attributes of each videotext region including the timing information of the frame, as in the Videotext DS for each frame where the text presence was detected. In the second stage, an interframe analysis [8] is performed to identify contiguous time intervals where text is persistent to result in markers for text events. This outputs a videotext event definition file in which the video name, the start and end of text events, along with pointers to their representative start and end key frames are stored. The video browsing application uses the videotext event file, whenever a video is selected for browsing to render a graphical timeline of the video along which the text events are highlighted. Figure 1 shows a snap shot from the application in which the text events detected from CM1002.mpg are highlighted along the video timeline.

The graphical interface of the application allows for efficient browsing. A user can click on a highlighted segment to mark the beginning of the video portion to be browsed and start the player to play the video from the marked time. Figure 1 (a) shows the time of the video clip being played and the frozen frame resulting from the stopping of the played video. In the figure "text" icon in the left shows that this frame has videotext feature.



(a) (b)

Figure 1: (a) A snapshot from the videotext event browsing application showing videotext events detected in CM1002.mpg; (b) a snapshot from the video classification application showing detected face and videotext.

We refer the reader to [3] for a quantitative analysis of video browsing efficiency with videotext annotation. From the detailed results reported [1] on our experiments conducted on a subset of video frames chosen from the MPEG-7 test data, we found that different automatic text detection algorithms perform with different false positive and miss rates, and therefore have an impact on the browsing efficiency. For example, the IBM videotext extraction system does well in terms of low miss rate by not missing many of the true text lines, while the Philips' system has higher miss rate. On the other hand, the system from Philips is much better in terms of detecting far fewer false positive text regions than the IBM system. The IBM system will be a very good fit for applications such as classification emphasizing low miss rates. The Philips system is more appropriate in applications where fast browsing of video is needed. Here, too many falsely retrieved videos based on falsely identified videotext can become a bottleneck. Therefore, it is more efficient to use the system from Philips since it not only results in low numbers of falsely detected text regions but also detects text in video frames quickly.

5.2 Classification Based on the Videotext DS

For the second experiment we adopted an existing application of videotext, which classifies video segments into known categories based on the location of faces and text. This video classification method is based on the observation that in different TV categories there are different face and text trajectory patterns. Face and text tracking is applied to arbitrary video clips to extract faces and text trajectories. We used two different methods: domain based method [10] and Hidden Markov Models (HMM) [11] to classify a given video clip into predefined categories, e.g., commercial, news, sitcom and soap. Our preliminary experimental results show classification accuracy of over 80% for HMM method on short video clips [11]. Originally the application extracted videotext and face positions and stored them in a specific format. This stored information was used by the video classification application. In this experiment, we used our Videotext DS to encode the extracted information from videotext and we used the bounding box DS to encode the ex-

tracted information from faces. Figure 1 (b) contains a snapshot from the user interface of the video classification application.

6 Conclusions

In this paper we presented the Videotext DS and the VideotextObject DS within the context of MPEG-7 standardization efforts. We presented the current definition of the description scheme in XML. The definition of the Videotext DS is continuously evolving over the course of the MPEG-7 standardization efforts. We have proposed two automatic videotext extraction methods: an edge-based method and a region-based method.

We have described two representative applications that were demonstrated as part of the MPEG-7 core experiments: keyframe browsing and program classification. From the wide range of scenarios that cover video indexing and annotation, and scene analysis it can be concluded that the videotext feature is quite powerful, providing rich, high-level semantic information that can be used in numerous video applications.

References

[1] C. Dorai, R. Bolle, L. Agnihotri, and N. Dimitrova, "MPEG-7 videotext descriptor proposal." ISO/MPEG/M4791, MPEG Vancouver Meeting, July 1999.

[2] C. Dorai, R. Bolle, L. Agnihotri, and N. Dimitrova, "MPEG-7 videotext description scheme." ISO/MPEG/M5206, MPEG Melbourne Meeting, October 1999.

[3] C. Dorai, R. Bolle, N. Dimitrova, L. Agnihotri, and G. Wei, "Core experiments on the videotext DS." ISO/MPEG/M6081, MPEG Geneva Meeting, May 2000.

[4] J.-C. Shim, C. Dorai, and R. Bolle, "Automatic text extraction from video for content-based annotation and retrieval," in *Proc. International Conference on Pattern Recognition*, vol. 1, (Brisbane, Australia), pp. 618–620, August 1998.

[5] T. Sato, T. Kanade, E. Hughes, M. Smith, and S. Satoh, "Video ocr: Indexing news libraries by recognition of superimposed caption," *ACM Multimedia Systems*, vol. 7, no. 5, pp. 385–395, 1999.

[6] L. Agnihotri and N. Dimitrova, "Text detection for video analysis," in *Workshop on Content Based Image and Video Libraries*, (Colorado), pp. 109–113, 1999.

[7] Y. Zhong, H. Zhang, and A. K. Jain, "Automatic caption localization in compressed video," *PAMI*, vol. 22, pp. 385–392, April 2000.

[8] J.-C. Shim and C. Dorai, "Interframe analysis for improved text extraction from video," working document, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, January 1998.

[9] J.-C. Shim and C. Dorai, "An end-to-end video character recognition system for automated video annotation and search," working document, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, August 1999.

[10] G. Wei, L. Agnihotri, and N. Dimitrova, "TV program classification based on face and text processing," in *IEEE International Conference on Multimedia and Expo*, (New York), July 2000.

[11] N. Dimitrova, L. Agnihotri, and G. Wei, "Video classification based on HMM using text and faces," in *European Conference on Signal Processing*, (Finland), September 2000.