IADIS International Journal on Computer Science and Information Systems
Vol. 4, No. 2, pp. 120-139
ISSN: 1646-3692

SYNCHRONISING VIDEO SESSION RECORDINGS AND TEXTUAL TRANSCRIPTIONS FROM THE ANDALUSIAN PARLIAMENT

Luis M. de Campos. *Department of Computer Science and Artificial Intelligence. E.T.S.I. Informática y de Telecomunicación. CITIC-UGR. Universidad de Granada, 18071, Granada, Spain.*

lci@decsai.ugr.es

Juan M. Fernández-Luna. *Department of Computer Science and Artificial Intelligence. E.T.S.I. Informática y de Telecomunicación. CITIC-UGR. Universidad de Granada, 18071, Granada, Spain.*

jmfluna@decsai.ugr.es

Juan F. Huete. *Department of Computer Science and Artificial Intelligence. E.T.S.I. Informática y de Telecomunicación. CITIC-UGR. Universidad de Granada, 18071, Granada, Spain.*

jhg@decsai.ugr.es

Carlos J. Martín-Dancausa. *Department of Computer Science and Artificial Intelligence. E.T.S.I. Informática y de Telecomunicación. CITIC-UGR. Universidad de Granada, 18071, Granada, Spain.*

cmdanca@decsai.ugr.es

ABSTRACT

In the Andalusian Parliament, at Spain, all the matters discussed in the sessions are stored in two different media: a textual transcription containing all the speeches of the Members of the Parliament, plus a video recording all the session. The only link between these two information sources is the event that they represent. But it could be very interesting and useful for a user, that given a part of a record of parliamentary proceedings, she could access directly to the part of the video related to this text. To make this connection, a previous process of video segmentation must be performed, in order to split the video in the most appropriate units. Later, a synchronization step between the different parts of the transcriptions and the segments is needed. In this paper, approaches to give a solution to this problem are presented.

KEYWORDS

Video and text synchronization, video segmentation, structured information retrieval.

1. INTRODUCTION AND MOTIVATION

There are some situations in which the information system from a company or an institution has to deal with multimedia information in its daily operation. In some cases, the same information is represented in different but complementary multimedia formats. For example, and focussing in the context of this paper, national or regional parliaments usually generate a set of electronic documents containing the exact transcriptions of the speeches of the Members of the Parliament (MP) relating to every matter discussed in the different sessions. But also, these sessions could be recorded in video. Therefore, there are two information sources that are implicitly related as they represent the same event and could be very useful in conjunction. But sometimes this last information, the reference to the event, is the only information able to link them, and due to the fact that their formats are different, there is no possibility of linking them by their content. The consequence is then that the multimedia sources representing the same event are not able to be used in an integrated way. Therefore, new tools that try to synchronise both information sources must be developed.

A common example of this separation in terms of use is a search application for textual documents: users formulate queries in order to obtain relevant documents that satisfy their information needs. Then, once the search engine has returned the list of relevant documents for a given query, the user has to inspect one by one all of them in order to determine if each document satisfies her/his information need. If so, with the information related to the event that this document represents, if the user is interested in watching the associated video, then she/he has to look up the video and access the relevant part inside it. This is a very time consuming task, and by this reason not very useful for the user. The ideal situation would be that the user could read a text and, at the same time, have the possibility of watching the corresponding segment of the video, and with the minimum effort.

A possible solution could be implemented in a two step process:

1. If we are able to detect the internal organization of the documents, then they could be transformed to structured documents, containing explicitly the different parts, with their associated text, that compose them.
2. By a (semi-)automatic process, each component of the document is associated to its corresponding segment of the video.

Then, coming back to the search engine context, this application should be able to return only the relevant parts of the documents, instead of the whole documents, as well as direct pointers to the video segments related to these relevant parts. Therefore, the user is directed to the most specific relevant portions of documents/videos, avoiding the posterior search inside the document/video, saving a precious time.

Focusing in the case of study of this paper, the Andalusian Parliament generates a group of electronic documents in PDF format called records of parliamentary proceedings, published in the www.parlamentodeandalucia.es site. They store all the discussed matters in every session and the participations of the MPs for these matters. Moreover, the sessions are also recorded in video, so additionally to the transcriptions, the digital library of the Parliament is complemented with the session recordings.

In the records of parliamentary proceedings, and therefore, in the videos, all the participations of the members of parliament are found, and also all the agreements achieved in the plenary and specific area committee sessions passing laws or celebrating informative sessions with members of the regional Government.

These documents could be accessed through by means of a search engine that works with a representation in XML of the PDF documents where the internal structure of the records of parliamentary proceedings could be exploited. Then the user formulates a query and gets the relevant documents (Baeza-Yates, Ribeiro-Neto, 2001), or parts of them (Chiaramella, 2001). But, this is not the case of the videos, which currently are accessed separately from the text by date, basically. There is no link between the document of the record of parliamentary proceedings and the video. But it could be very useful for the user that when she/he retrieves a relevant document (the text), or a portion of it, she/he could watch the associated video at the same time. Then the structured Information Retrieval field (Chiaramella, 2001; de Campos et al., 2006) gives the possibility of making the decision of determining the type of XML element containing relevant information, so the user does not have to inspect the whole document to find the requested information. But also, the user could watch only the portion of the video in real time. This feature could be an added value for the search engine.

But before the Information Retrieval System (IRS) could use the videos to be presented to the user, there is a previous work of synchronizing the text contained in the record of parliamentary proceedings and the corresponding video. The output of this stage would be the XML elements containing text from a speech marked up with time tags, corresponding to the beginning and end points in the associated video.

From beginning of 2008, the Andalusian Parliament is working with a new video management system that is able to support the creation of smaller videos corresponding to each one of the MPs who speak in each session. Although manual, this is a relatively fast process carried out by video operators. Given a complete video of a session and its agenda, the result is a set of XML files (let us call them *XML Speech files*) containing, for each speaker, basic information about the legislature, the session, the initiative code, and temporal information about when her/his speech begins and finishes. Moreover, each XML Speech file has got associated a new video containing the speech recording. Then an automatic process of synchronisation must be performed in order to include a link to the corresponding video in the XML file containing the transcription of the session (let us call it *XML Transcription file*).

But for videos previous to that date, this new system can not be used so the approach has to be rather different: automatic video segmentation, i.e., the video is automatically partitioned in segments of inner similar content, detecting the boundaries between them, and later synchronized by hand. When there is a change of camera, a new segment is created. Finally, these segments are associated with the textual transcription of the speeches. Basically the requirements of this piece of software are: 1) Automatic segmentation of videos; 2) Manual edition of the segmentation; 3) Annotation of the XML documents containing the transcriptions, synchronizing them with the segments of the videos.

Most of the existing segmentation algorithms found in the specific literature are designed for general videos (Camastra, F. and Vinciarelli, A., 2007). This means that they are complex algorithms prepared to detect the boundaries of the segment in all conditions. But in our context, a simple algorithm based on histogram comparison could work very well, as the case is. In this paper we describe the algorithm itself, how it has been improved, as well as the main features of the segmentation and annotation tools.

Therefore, this paper describes the two approaches followed for synchronising the session videos and transcriptions, in order to facilitate the access to the video by means of a structured textual search engine, in the context of the Andalusian Parliament. Then the article is organised as follows:

In Section 2, we shall introduce the digital library of the Andalusian Parliament, as well as some basic terminology to understand the rest of the paper. In Section 3 a brief review of the available annotation tools is presented. Section 4 will describe a synchronisation strategy for transcriptions and videos generated with the new system. The segmentation algorithm as well as the synchronisation method used for older videos is presented in Section 5. The paper will end with Section 6, where some details about the video players are described, and Section 7, containing some conclusions and further research.

2. THE ANDALUSIAN PARLIAMENT AND ITS DIGITAL LIBRARY

This section presents a brief introduction to the Andalusian Parliament and its official publications in order to contextualise this paper and introduce specific terminology.

The Andalusian Parliament was established in 1982 and to date there have been eight legislatures (periods of political activity of up to four years). The Parliament edits two main official publications: the *record of parliamentary proceedings* and the *official bulletin.* The first publication contains full transcriptions of all the MPs' speeches in each parliamentary session in which laws are passed or in the informative sessions held with MPs. Additional information such as the official agenda (the matters to be discussed as agreed by all the political groups, i.e. the minutes of the meetings), results of possible votes, agreements, etc. is also included. In the second publication, the Parliament publishes any texts and documents to be made available to the public about laws passed or to be processed.

The Parliament works around the concept of *parliamentary initiative*, whereby an action taken by an MP or political party is discussed in a plenary or specific area committee session. These initiatives are included in the plenary or committee sessions and identified by means of an initiative code. Before a plenary session is held, the political parties represented in the House decide on the agenda for the session and this consists of a sequence of initiatives. These are subsequently grouped according to type and are previously published in the official bulletin. Once the agenda has been agreed, the Speaker leads discussion of each point, allowing MPs to speak on and discuss the corresponding initiative.

All official documents are published electronically on the Andalusian Parliament's website, and PDF is currently the format most widely used by organizations (including the Andalusian Parliament) for storing and publishing textual information.

Once the PDF documents have been published, the Andalusian Parliament also provides a search engine so that users can consult the legislative collection by means of a web form. As the sessions are also videoed, the Parliament's digital library with its records of parliamentary proceedings is supplemented with videos on the website which while not able to be searched can be browsed by date and viewed.

Since it was established 26 years ago, more than 6000 documents have been published with the records of parliamentary proceedings and the official bulletins. While the size of this digital library is not excessive, it is a respectable figure and is constantly growing.

To help the reader understand the rest of the paper, we will now describe how the records of the parliamentary proceedings are organized. They comprise four distinct yet well-defined parts:

1. *General information section:* this contains general information about the parliamentary session in question (e.g. type of session, legislature, date and presidency).
2. *Agenda*: a list of the initiatives grouped according to type. This is decided by the political parties before the session and follows the format of initiative code, subject and proposer. For example, in the type "oral questions", all the questions are listed specifying for each one the question and the asker.
3. *Summary*: a detailed description of the agenda which is created once the session has finished. Once again, all the initiatives are grouped according to type and in addition to a description of the initiative (code, matter and proposer) they also include a list of MPs participating in the debate and the result of any vote. New agenda items may be added or others removed.
4. *Development of the session:* For each point included in the summary section (following the initiative information), transcriptions of all the speeches are included and set out like a script for a play or a film.

Each PDF document has associated an XML file containing the same text but explicitly stating the internal structure of the document, i.e., the XML Transcription file. In the following text, an extract of the XML file related to the Development of a (imaginary) session is shown.

```
<development>
    <initiative>
        <extract> 8-08/DG-000009 . General debate about the reform of the regional financing.
        </extract>
        <participation>
            <speaker> Mr. Carlos Dancausa, President of the Andalusian Parliament </speaker>
            <speech>
                <paragraph> Good morning, ladies and gentlemen. We are going to start the debate
                of the initiative 8-08/DB-000009 about the regional financing. </paragraph>
                <paragraph>  I give the floor to Mr. Fernández. Please. </paragraph>
            </speech>
        </participacion>
        <participation>
            <speaker>Mr. Juan Manuel Fernández </speaker>
            <speech>
                <paragraph> May it please the Court.  </paragraph>
                <paragraph> Ladies and gentlemen, following with the discussion of the previous
                speeches, I would like to remark…</paragraph>
                …
        </participation>
        <participation>
            <speaker> Mr. Carlos Dancausa, President of the Andalusian Parliament </speaker>
            <speech>
                <paragraph> Thank you very much, Mr. Fernández. Now it is the turn
                for…</paragraph>
            </speech>
        </participation>
        …
    </initiative>
    …
</development>
```

In this extract, we can observe the basic structure of de Development part of a XML Transcription file. It is composed of several *initiative* tags, containing the discussion of several initiatives. Inside them, there is an extract tag, with a brief description of the subject of the initiative, plus a sequence of *participation* tags, containing the speeches of the different MPs who participate in the debate, with indication of the speakers (*speaker* tag) and the paragraphs of their speech.

3. BRIEF REVISION OF EXISTING ANNOTATION TOOLS

Having a look at the available tools for video annotation, we could find, among others, the followings:

- *IBM MPEG-7 Annotation Tool* (http://www.alphaworks.ibm.com/tech/videoannex): it allows annotating videos using the metadata associated to a MPEG-7 video. This is segmented automatically, and a key frame extracted. On this representative, the user can make the pertinent annotations about events of static scenes and key objects. For our purposes, the annotation is very restrictive, because only can be done using predefined scenes and objects. Also it is in MPEG-7, which is effectively a standard, but we are not considering this multimedia format. Another problem is that the segmentation can not be edited.
- *IBM Multimedia Annotation Tool* (www.alphaworks.ibm.com/tech/multimodalannotation) is very similar to the previous one, but also can annotate audio, background and foreground sounds. Both tools are free.
- *Virage VideoLogger* (www.virage.com) analyses the video and generates automatically a structured index about the content. It allows making manual annotations, but it is not free. This tool is really complex because of its high functionality.
- iFinder (www.imk.fhg.de/de/ifinder) is another tool which works with the MPEG-7 standard. It is able to segment videos, recognize faces, as well as to make speech recognition. It is a server client application. But it is not free and really complex.
- Ricoh MovieTool (www.ricoh.co.jp/src/multimedia/MovieTool/about/index.html) allows automatically segmenting MPEG-7 videos, but giving the possibility of refining the output, making it manually. A very interesting feature is the possibility of creating a hierarchy of segments.
- *ZGDV VideTo* (www.zgdv.de/zgdv/departments/zr4/Produkte/VIDETO) presents the interesting feature that the annotation could be stored in XML format, without any restriction (later it could be converted to MPEG-7). Patterns about specific domains could be created, facilitating the annotation. The segmentation is fixed and can not be edited.

With the analysis of all these tools we realise that there is no tool that fulfils all our requirements.

4. SYNCHRONISATION OF VIDEOS AND RECORDS OF PARLIAMENTARY PROCEEDINGS VIA XML FILES

The first approach that we present in this paper is used when, from the current video management system, the video operator generates an XML file per speaker from a session recording (we shall call this files XML Speech files), containing to a great extent the following information:

a. Session identification: date, number of legislature, type and number of session.
b. Name of the speaker.
c. Speech information: beginning and end times inside the video of the full session, speech length, name and location of the video file.

The following XML extract is an example of XML Speech file:

```
<dcm>
  <track>
    <title> Participation of  "Fernández, Juan Manuel" in "Plenary Session", 26/06/08. </title>
    <timecode>26/06/08 09:03:15</timecode>
    <length>08:05:55.160</length>
    <clip_in>00:02:05.000</clip_in>
    <clip_out>00:13:31.000</clip_out>
    <elements>
      <element name="speaker" content="Fernández, Juan Manuel"/>
      <element name="legislature.ordinal" content="VIII"/>
      <element name="session.label" content="PL-VIII/2008-JUN-26/1"/>
      <element name="session.type" content="Plenary session"/>
      <element name="session.abstract" content="Record of Parliamentary Proceeding, no. 13"/>
    </elements>
    <resourcesCo>
      <resource>
        <url>file:///./media/2.wmv</url>
      </resource>
    </resourcesCo>
  </track>
</dcm>
```

This same video operator also generates a shorter video containing only the speech associated to this file.

Obviously, there is no need for using any of the previous presented annotation tools, as the problem is quite different, so a “prêt a porter” solution has to be given.

The synchronisation process takes as input the set of XML Speech files from a session, as well as one XML Transcription file containing the complete session. The result of this stage is the same XML Transcription file completed with links to the corresponding videos as an attribute of the *speech* tag. The following XML Transcription file fragment represents the same one presented in Section 2, but with pointers to the videos.

```
<development>
  <initiative>
    <extract> 8-08/DG-000009 . General debate about the reform of the regional financing.
    </extract>
    <participation>
      <speaker> Mr. Carlos Dancausa, President of the Andalusian Parliament </speaker>
```

```
            <speech videoXML="VIII/DSPAP8013-1/1.flv">
                <paragraph> Good morning, ladies and gentlemen. We are going to start the debate
                of the initiative 8-08/DB-000009 about the regional financing. </paragraph>
                <paragraph>  I give the floor to Mr. Fernández. Please, whenever you wish.
                </paragraph>
            </speech>
        </participacion>
        <participation>
            <speaker>Mr. Juan Manuel Fernández </speaker>
            <speech videoXML="VIII/DSPAP8013-1/2.flv">
                <paragraph> May it please the Court.  </paragraph>
                <paragraph> Ladies and gentlemen, following with the discussion of the previous
                speeches, I would like to remark…</paragraph>
                …
        </participation>
        <participation>
            <speaker> Mr. Carlos Dancausa, President of the Andalusian Parliament </speaker>
            <speech videoXML="VIII/DSPAP8013-1/3.flv">
                <paragraph> Thank you very much, Mr. Fernández. Now it is the turn
                for…</paragraph>
            </speech>
        </participation>
        …
    </initiative>
    …
</development>
```

The matching process would be very simple as the algorithm should only iterate the speakers in the XML Transcription file, find the XML Speech file associated to that speaker, and include the needed data in the XML Transcription file (basically an attribute containing the path and the name of the video: <**speech videoXML**="VIII/DSPAP8013-1/1.flv">. But if we take into account that due to some limitations of the video management system, for those speeches with a length less than 5 seconds XML Speech files and videos are not generated, then the algorithm has to be more complex in order to reduce synchronisation errors. Therefore, there could be short speeches that are represented by neither an XML Speech file nor a video, as well as included inside other speech video. In any case, no XML Speech file is generated for them.

In order to determine when a transcription of speech has got a length of 5 or less seconds, we have set up empirically a threshold of 25 words. Then if we find a speech of 25 words or less, then we may consider that there is no XML Speech file associated to that speech, so we skip it.

Basically, the algorithm for synchronisation iterates all speakers in the XML Transcription file, then for each one, tries to find the same speaker in the following XML Speech file. If both speakers are the same and the speech of this speaker is greater than 25 words, then we say that the current speech has been synchronised. If it is less, then we assume that we are not going to find it, so the next speaker in the XML Transcription file is considered, and the last recognised speech is assigned to it.

In case a speaker in her/his speech is interrupted by other(s) speaker(s) with speech(es) less than 5 seconds, all the sequence of speeches is assigned to the same segment of video, as they will be contained in it.

Finally, the XML Transcription files, as well as all the speech videos, are provided to the IRS so it could index them, first, and retrieve them, later.

As it may be realised, the video formats are different in the XML Speech video and in the XML Transcription video. The reason is that for visualization reasons, explained in Section 6, the IRS will show a Flash video instead of the Windows Media Video from the original recording.

5. SYNCHRONISATION OF VIDEOS AND RECORDS OF PARLIAMENTARY PROCEEDINGS VIA VIDEO SEGMENTATION AND ANNOTATION

In this section the solution provided for those older videos not managed by the new video management system is described. In this case, the solution could be closer to the philosophy underlying in all the tools presented in Section 3, but as the features of the problem are very different, it is also more appropriate to find a solution that fits well with the Andalusian Parliament's context.

In this case, the input is the full video of the session, and the output is the XML Transcription file annotated with beginning and ending stamps in the desired tags, with respect to that original video. In this case, the synchronisation could be made not only at speech level, but also to paragraph level. The general process of this second approach is as follows: first, the video is automatically segmented in pieces (each segment will contain a high internal homogeneity); second, the segmentation could be manually modified to solve any problem, or to create a more refined segmentation; third, and finally, the XML Transcription file is manually synchronised with respect to the previously created segments.

When the annotation of a video is carried out, we have to present the information to the user in such a way that all the content is summarized in a small amount of information. An adequate way is to split the video in segments (Camastra, F. and Vinciarelli, A., 2007). A segment is a fraction of the video that has continuity. Once the video has been divided, it must be presented to the user in such a way that with only a look she/he could have an idea of the content of each segment. This is made showing a keyframe, a representative of the content.

The segmentation based on difference between shots (Jain A. and Chaudhuri, S. 2002; Hanjalic, A., Lagendijk, R. and Biemond, J., 2006; Sáez, E. et al., 2003) obtains some features from each shot and compares them with those from the following shot. If the difference is greater enough, then it is considered that there is a change of scene. The most used features are the colours (Godil, A., 2004) or the edges of the image (Gastaud, M. and Barlaud, M., 2002). To obtain the differences of the shots, the format of the video storage could be considered (Calic, J. et al., 2002): shots storing the full image, or those storing only the change with respect to the previous shot. The segmentation based on flow of movement tries to detect the object in the scene (foreground and background objects) and analyses the movements in the following shots (Kobla, V. et al, 1997; Apostoloff, N. and Fitzgibbon, A., 2006; Lu, Y., Gao, W. and Wu, F., 2002). There also exist hybrid techniques which combine both methods (Liu, L. and Fan, G., 2005).

Our objective is to find the method that is able to segment the videos of the Andalusian Parliament, in a correct and efficient way, without forgetting the complexity of the implementation, which has to be low. These videos show long scenes, with few movements,

and sudden changes. This means that the analysis of objects in the scene is not going to help, because the speakers are usually static in the talk. In the chamber of the Parliament, there are four fixed cameras, focusing to the speaker, a general view of the chamber, and two centred in the seats of the MPs. Therefore, the production of these videos is usually very static, and therefore very easy to detect the changes of cameras. Considering the segmentation using shot differences, we shall notice that in the changes of camera, these differences are large, and inside the same segment, the differences are low, because the cameras are static and the movement is almost null. Therefore, we shall use this last method, considering the colour as the feature in which the method will be based on, basically because of the simplicity of the method itself and good results.

5.1 The Segmentation Algorithm

The segmentation algorithm that we present in this section is based on detecting differences between shots. More specifically, these differences will be given by the different colours of the shots. We shall adopt a grey scale representation. From the RGB images of the video, we get that representation using a simple transformation:

$$I(i,j) = (R(i,j) + G(i,j) + B(i,j))/3,$$

where I is the matrix that represents the image in grey scale; R, G and B are the matrices that represent the levels of red, green and blue of the image, respectively, and i and j are the indexes of a pixel.

The histogram of the image represents the number of times that a specific colour occurs in it. As we are working in grey scale, and considering that 8 bits are used to represent the intensity of 256 tones, we could represent the image by means of vectors of 256 elements. If we note H the vector, and H(i) the number of occurrences of the colour i in the image, then, in order to determine if there has been a change between two shots S1 and S2, with histograms H1 and H2, respectively, we could compute the difference of both vectors:

$$(H1 - H2)[i] = | H1[i] - H2[i] |.$$

Computing the difference for each colour, and summing up all of them, we have a scalar value of the difference between both. If this value is greater than a certain threshold, then there is a change in the shots, as may be observed in Figure 1. The blue lines are the differences between shots and the red line is the threshold.

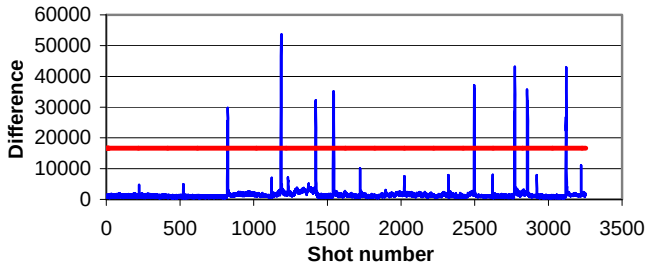

Figure 1. Differences between shot histograms.

This is a really simple algorithm as a set of shots are considered included in the same segment if the difference between their histograms is low. But experimenting with the videos

of parliamentary sessions, we realised that the changes of cameras origins that the difference of histograms is high, so they could easily be detected. But it is also high (although a little bit smaller) for shots in which there is no such change in the camera. Therefore, mistakes might be made. This is due to a minimum variation of the luminosity of the image, which considerably alters the histogram of the image. For example, a simple movement of an MP who is speaking could provoke that his/her jacket reflects the light in a different way. This could make that the colour represented in a shot is lightly different to the previous shot. Although the difference is not noticeable for the human eye, a minimal variation in the histograms increases the difference between both of them. In order to solve this problem, we have to achieve that between close histograms the difference is not so important, so we could soften the histogram, in such a way that each element depends on the neighbours, and therefore a light change does not affect the difference.

A solution is the application of a convolution filter, which makes that each element of the vector is the sum of those closest elements. We have carried out such convolution using the vector [0.1, 0.2, 0.4, 0.2, 0.1]:

$$H1'[i] = 0.1*H1[i-2] + 0.2*H1[i-1] + 0.4*H1[i] + 0.2*H1[i+1] + 0.1*H1[i+2].$$

An important decision that will clearly have a great influence in the performance of the segmentation is the selection of the threshold value. It will depend on the resolution of the video, as in shots with a higher resolution the difference of their histograms will be proportionally larger. Moreover, images with a higher number of colours will also present a higher difference among shots, so we will have to consider the number of tones contained in the images. Therefore, the threshold used in our algorithm takes into account the width of the image, its height, as well as the number of colours.

5.2 Optimization of the Algorithm Efficiency

The first parameter that must be estimated is the *threshold.* For the type of considered videos, its value has been obtained empirically studying several of them. The process which has led to get it has been the following: first of all, we have obtained the difference for each pair of contiguous shots in each video; secondly, we have localized manually the cuts (changes of scene) that the segmentation algorithm should detect; and finally, once we have studied the values of differences in the shots in which there is a cut, we have selected a value such as is sufficiently low to detect all the real cuts, and sufficiently high to not detect cuts which does not exist. With a threshold of 16,000 all the cuts are detected, and nothing except real cuts will be detected.

This basic segmentation algorithm works properly, but not efficiently. As the videos from the sessions of the Andalusian Parliament are very long (about 5 hours), it is required to improve the segmentation speed, but without worsening the effectiveness.

The first attempt is to reduce the number of shots to be considered. In the original algorithm, the histograms of each pair of shots are compared. But it is noticed that if there is a cut between the shots i and i+1, this cut also will be placed between the shots i-x and i+y+1. This means that we could detect the cut analysing the difference of histograms between two non contiguous shots surrounding shots i and i+1.

Therefore, instead of analysing all of them, we will discard x shots between each studied pair. The next step will be to refine the segmentation to locate exactly when the cut is produced, i.e. if the algorithm finds a cut between shots x and y ($x < y$), then it will visiting

each pair of shots i and i+1, with x < i < y, contained in the interval, to detect the cut. This process is much faster than to compare each single pair of shots and also offers the optimal result. For videos in which there are not an excessive number of cuts, as the case is, this is an appropriate method. Figure 2 shows a graphical example of this process.

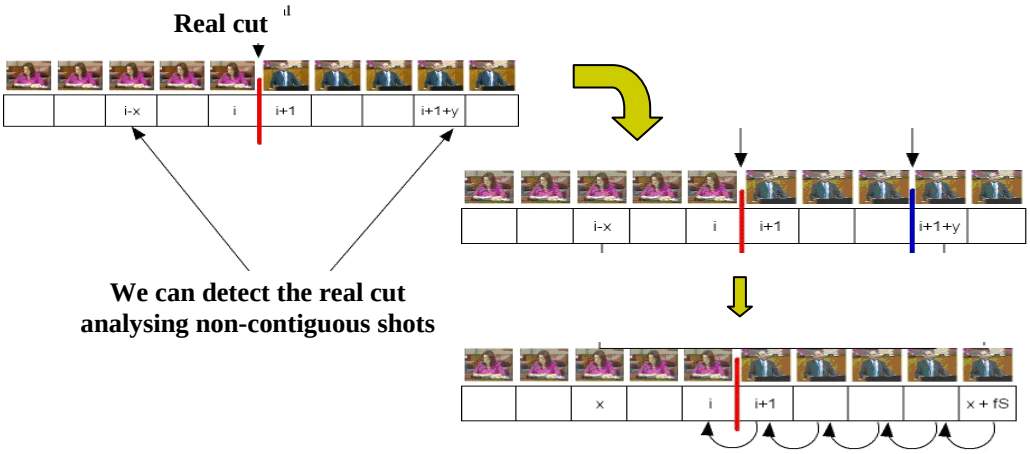

Figure 2. Comparison of non contiguous shots and posterior refinement.

A second optimization is related to the size of the image. If the difference of histograms with the full image is enough to differentiate shots, we could suppose that the histogram of only a portion of the image could offer enough information to perform this action, adjusting the threshold to the corresponding size. Then the reduction will improve the efficiency of the process, as the number of computations is lower. This is not only interesting in terms of efficiency but also in term of efficacy of the segmentation.

In most of the videos, there are movements but inside a scene without a compulsory change of shot. In this case, the algorithm could make a mistake because, in these cases, the movement originates a change in the colours of the image. In the case of the videos of the Andalusian Parliament, where the location of the cameras is known, we could know the part of the image that will suffer less interferences of this type. If we only use the pixels of this area to compute the histograms, we reduce the risk of finding cuts where there are not. In the case of study, the lower right corner of the image is selected, as empirically it is the area where fewer movements are observed. Figure 3 present a shot of the chamber divided in 9 parts (reduction factor of 3). The lower right part is selected.

Figure 3. Shot with a reduction factor of 3.

Finally, once the segmentation is finished, we have to select a representative shot of each segment (key frame), in order to present them to the user. There are several techniques to select the key frame (Avrithis Y., 1999; Calic, J. and Thomas B., 2004; Calic J. and Izquierdo, E. 2002). But in the case of the videos of the Parliament, and due to their nature, it is not required a sophisticated technique as the user is going to use it as a guide to the annotation phase. Then, the shot which is in the 10% of the total length of the segment is selected. With this selection there are no mistakes, as the segment is supposed to be accurate in terms of very similar contents of all its shots.

5.3 Experimentation and Evaluation

In this section, we shall show how the selection of the best values for the two main parameters of the segmentation algorithm has been made, so the best performance is reached. In this line, we have to decide the size of the window used to compute the histograms and later the difference between them (reduction factor), and the number of shots ignored in each stage of histogram comparison, and the posterior refinement to find the exact point of change.

We have created a set of 6 videos of different length. Their durations as well as the number of exact segments, obtained manually when there is a change in the scene are the following:

V5(5m,5), V9(9m, 7), V20(20m 26s, 27), V30(30m 13s, 36), V40(40m 8s, 29), VF(5h 46m 36s, 433).

The first experiment tries to show how important is to use a reduction factor. In Table 1 we show the accuracy of the segmentation considering the first 5 videos, with and without refinement (reduction factor of 2 and 24 shots between two compared shots). As it may be noticed, the refinement is configured as an essential technique to maintain a high accuracy.

Table 1. Percentage of correct segments detected (accuracy) with and without refinement.

Videos	Accuracy with refinement	Accuracy without
V5	100.00	27.78
V9	100.00	87.50
V20	92.59	20.00
V30	88.89	25.53
V40	100.00	76.32

With respect to the reduction factor, we have tested the same set of videos with several values, from 1 to 9. In Figure 4, we can observe the graphical representation of the reduction factor and segmentation time, as well as, the reduction factor and the percentage of correct segments. Initially, these experiments have been carried out with the number of shots jumped set to 24 and refinement applied.

The segmentation time is the sum of times of different tasks: the shot extraction process from the video, the computation of its histogram, the convolution, the comparison with the previous shot, and the refinement, in case of needing it. In affirmative case, the time would increase considerably, as the algorithm has to look for the change from the previous shot until the shot where the real change is produced. When the reduction factor is increased, the time required to compute the histogram is reduced. But, as a smaller portion of the image is used, there is less information available and therefore there will me more segmentation errors.

These two situations are found in videos V30 and V40. In the former, as there are lots of cuts, the reduction factors 2 to 7 make the algorithm to spend more time testing the differences between histograms when refining than the time saved using smaller image areas. In the latter, the cuts are less frequents, so better times are obtained with factors 2 to 6.

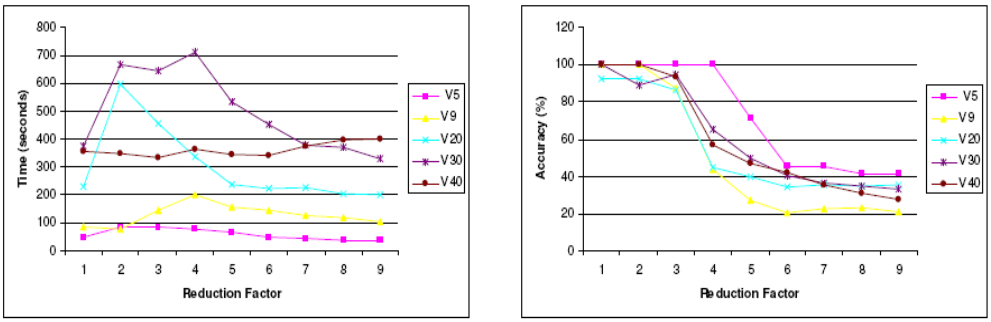

Figure 4. Time vs. Reduction Factor (left) and Accuracy vs. Reduction Factor (right).

With respect to the accuracy of the algorithm, we may notice that, in most of the cases, from a reduction factor greater than 2, it decreases considerably. This is due to the fact that there is less information about the change of images testing a small area, so the algorithm makes more mistakes. As a conclusion, we shall use a reduction factor of 2 because the accuracy is usually very close to the 100%, and the segmentation time is usually one of the shortest.

Fixed the value of this parameter, next we shall study the number of shots ignored between two shots to be compared. In this case, the analysis will be done with VF in two steps. Figure 5 shows the segmentation time for values 1, 12, 22, 32. We can observe how the value 12 gets a better performance. In a second step, we have tried with values 10, 12, 14, 16 and 18. In this case, 14 is the best. Until there the segmentation time decreases. From it, this time increases.

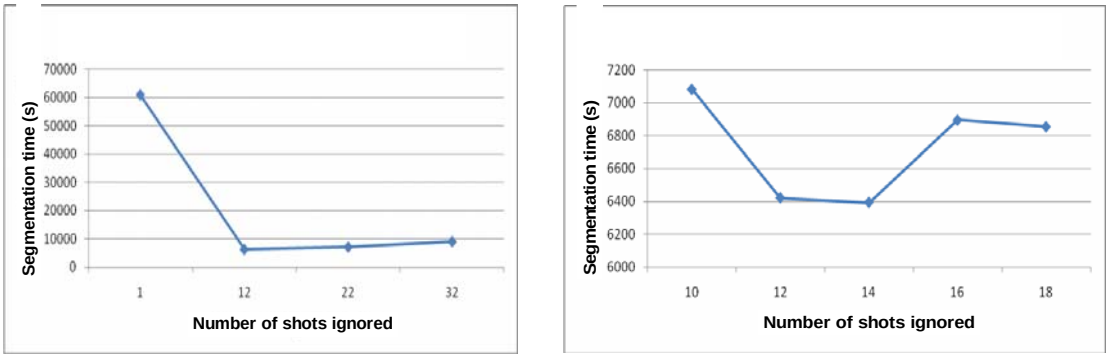

Figure 5. Segmentation time for different numbers of shots ignored.

5.4 Features of the Segmentation Application

Before starting with the main features of the segmentation tool, just to mention that it has been developed in Java, using the Java Media Framework (JMF) library to incorporate multimedia elements and tasks in our application in a comfortable and easy way.

There exists the possibility of making a batch segmentation of several videos. This facility is implemented because the segmentation time is usually long taking into account its average length. Then the user selects a group of videos to be segmented, and the application will process each one, given as an output the segments of each video.

Once the automatic segmentation of a video has been carried out, the software offers the possibility of editing the segmentation manually. The output of this process is a set of segments, represented by a key shot. The user may need to adjust the segmentation to prepare the posterior process of annotation, in order to be more accurate. An example of this need is the case in which a deputy is speaking, and a camera focusing her/her. Then there is a change of camera, and that focuses another deputy. In spite of this change of scene, the first speaker is continuing with her/his talk. The segmentation algorithm would give as an output two segments (correct), but the user may wish to join both of them, because the current speech is being given in both of them. Therefore, the user is allowed to edit the segments, combining them (two segments s1 and s2 could be joined if they are contiguous) or dividing segments in two.

In the application, all the segments found by the algorithm are shown in a window (see Figure 6). More specifically, the key shot of each segment. If we click in one of them, then all the shots contained in it are shown in a separate window. By means of submenus activated by the left bottom of the mouse, the user could edit the segments. A viewer allows playing any segment.

In order to show a shot, the application must access the video. This means a high access, the player must be placed in the correct position, the file in the hard disk must be accessed and copy the contents in main memory. For a real time usage, this would mean that there is a high response time. To avoid this, we have implemented a cache module, in which we store the shots of the video in a cache memory (those which are more probable to be shown), so their access will be much faster.

Figure 6. Screen shoot of the Segmentation tool.

5.5 The Synchronization Tool

Once the segmentation of a video has been performed, and the posterior edition, the user is ready to carry out the annotation stage. The input of this process will be the sets of segments found in the video corresponding to a parliamentary session and the transcription of the speeches given in the chamber for that video. This transcription is represented by means of an XML document, which contains the structure of the session, as well as the text itself. The output will be the XML document containing the transcription synchronized with the video by means of time stamps in the elements of the document. The segment of the video related to a specific text could be easily access.

The annotation tool (Figure 7) will consist of a manual association of segments with the corresponding elements in the XML document, so each tag will have a link to the part of the video where this text is played.

Figure 7 shows the user interface of the annotation tool. It is composed of four windows. The left window shows the tree representation of an XML document containing record of parliamentary proceedings. If a leaf node in clicked, then the text contained in it is shown in the central uppest window. The window below contains the segments found in the first part of the process. Finally, a player is included in the right part of the interface, in order to help the annotation.

The annotation process is as follows: the user selects a segment in the video, then find the node in the XML document containing the transcription of the audio of that segment, and by means of a drag and drop action, associate the former with the latter. These steps are repeated until all the segments have been assigned to a node of the document.

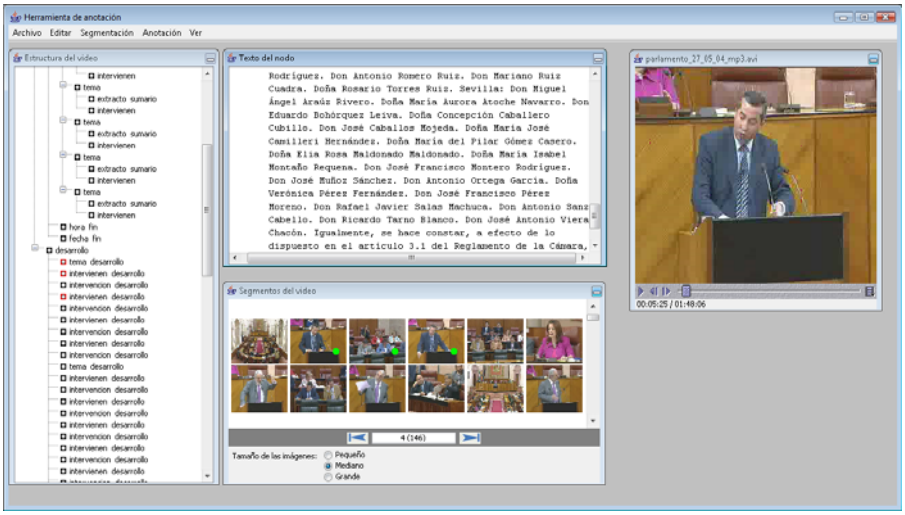

Figure 7. Screen of the Annotation tool.

Actually, with the association of a segment to an XML element of the document, we introduce a pair of attributes to the desired tag, containing the beginning and ending times of the segment (for example, <speech bt=”00:45:54”, et=”00:59:02">), plus a video tag pointing to the full video of the session at the beginning of the XML Transcription file. This information will be enough to access the portion of the video in retrieval time.

Once all the segments have been assigned to leaf nodes of the XML tree, and therefore, all the affected tags have been complemented with temporal attributes linking the text with the video, it is necessary to propagate the times to upper nodes until reaching the root node. In our context of the parliamentary sessions, the text is usually contained in ‘paragraph’ tags. These are contained in ‘speech’ tags (the speech of a deputy), which are included in the debate of a parliamentary initiative (several MPs usually take part in the discussion of a point of the agenda). Finally, several points of the agenda, plus general information of the session, are included in the ‘session’ tag, the root of the tree. Then if the search engine retrieves a whole point of the agenda, it should also allow the access to all the segments related to the elements contained in that tag. Therefore, a propagation of times is performed from leaf nodes to the root node.

6. THE VIDEO PLAYER

Finally, once the transcriptions in form of XML documents have been completed with references to the video (first approach) or time stamps (second approach), the search engine, once it has indexed the parliamentary digital library, is ready to be used. When a user query is formulated to the system, it ranks all the elements from all the documents according to their relevance with respect to the query, and presents this ranking to the user. Then she/he could click on a link, in which she/he is interested, to inspect the text or to watch the video segment, which is played by means of a player (Figure 8).

Figure 8. Screen of the search result screen with the video player.

An important element in the retrieval stage is the video player. We have to take into account two important features in order to make the decision of what type of player to use: first of all, it must be an accessible element for any type of user, independent on the platform used; secondly, in order to save broad band in the video transmission, we only have to send to the player from the server the portion of the video in which the user is interested in, not the whole video. Usually the videos we are working with are very heavy, so the user can not be waiting to download 40 Mbytes of video to watch only 30 seconds. This must be fast.

For this reason, the player is implemented using the Flash technology, which allows creating vectorial animations, easily dimensionable, as well as with low storing requirements. This space optimization together with the possibility of the feature of loading the image at the same time it is shown in the browser, allows adding visual elements that animate a web avoiding the fact of a really long load of the web page. Moreover, Flash implements a programming language called ActionScript. One of the things that this language allows, by means of its libraries, is to incorporate a multimedia player inside a Flash animation.

An important additional feature by which we made the decision of using this technology to develop the player used in the search engine for the digital library of the Parliament of Andalusia, is the possibility of playing a piece of a video in any time of it, without downloading the whole video in the computer.

7. CONCLUSION AND FURTHER RESEARCH

In this paper we have presented two approaches to tackle the problem of synchronizing video and text, corresponding to the recordings and transcriptions of parliamentary sessions. The videos record the sessions of debate, while the documents, represented in XML, store the exact transcriptions of the speeches given in the session. From the point of view of the access to this material from the user, it would be very interesting that, as the output of a query, the system returned the text associated with its corresponding portion of video.

The first approach takes into account the fact that with the new video management system, the videos corresponding to a complete session are divided in so many shorter videos as speeches. Then with the XML Transcription file and all the XML Speech files a

synchronization process is performed. The output is the former file completed with the name of the video files related to each speech.

The second approach relies on a video segmentation and annotation tool for the videos and the documents. To implement this, we need a first step in which the video is segmented in pieces, and later, each interval is assigned to a text element belonging to the XML document. Therefore, in this paper, we have presented a very simple but effective and efficient segmentation algorithm to carry out this task, as well as an annotation tool to synchronize both media. Both tools have been designed thinking in the easiness of use.

With respect to further research, we think that the human work in the process of synchronization is too much, so we would like to reduce it, changing the philosophy of the tool: instead of segmenting video and later manually annotate it, we are researching on a process in which automatically the audio of the session would be synchronized with the text of the transcriptions. We think this would improve the process, alleviating the human intervention.

ACKNOWLEDGEMENT

Work jointly supported by the Spanish 'Ministerio de Ciencia e Inovación' (TIN2008-06566.C04.01) and the Spanish research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018).

REFERENCES

Apostoloff, N. and Fitzgibbon, A., 2006. Automatic video segmentation using spatiotemporal T-junctions. *Proceedings of the British Machine Vision Conference*. On line: http://www.bmva.ac.uk/bmvc/2006/papers/357.pdf.

Avrithis Y., 1999. *A Stochastic Framework for Optimal Key Frame Extraction from MPEG Video Databases*. Computer Vision and Image Understanding, Vol. 75, No. 1-2, pp 3 – 24.

Baeza-Yates, R.; Ribeiro-Neto, B., 1999. *Modern information Retrieval*, Addison-Wesley.

Calic, J. and Thomas, B., 2004. Spatial analysis in key-frame extraction using video segmentation. *Proceedings of the 5th International Workshop on Image Analysis for Multimedia Interactive Services*. On line: http://www.cs.bris.ac.uk/Publications/Papers/2000065.pdf

Calic, J. et al., 2002. Temporal video segmentation for real-time key frame extraction. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. pp. 3632 – 3635

Calic J. and Izquierdo, E. 2002. Efficient key-frame extraction and video analysis. *Proceedings of the International Conference on Information Technology: Coding and Computing*. Las Vegas, USA, pp. 28 – 33.

Camastra, F. and Vinciarelli, A., 2007. *Video Segmentation and Keyframe Extraction*. In Advanced Information and Knowledge Processing. Springer, pp. 413 – 430.

Chiaramella, Y., 2001. *Information retrieval and structured documents*, Lectures on IR, Springer, 286-309.

de Campos, L. et al., 2006. *Garnata: An Information Retrieval System for Structured Documents based on Probabilistic Graphical Models*. Proceedings of the Information Processing and Management Uncertainty Conference. Paris, France, pp. 1024 – 1031.

Gastaud, M. and Barlaud, M., 2002.Video segmentation using active contours on a group of pictures. *Proceedings of the International Conference on Image Processing*, Vol. 2, pp. 81 – 84.

Godil, A., 2004. ViSA: Video Segmentation and Annotation. *Proceedings of the UPA Conference*. On line: http://zing.ncsl.nist.gov/godil/UPApaper.pdf.

Hanjalic, A., Lagendijk, R. and Biemond, J., 2006. *Automated Segmentation of Movies into Logical Story Units*. Information Systems, Vol. 31, No. 7, pp. 638 – 658.

Jain A. and Chaudhuri, S. 2002. *A* Fast Method for Textual Annotation of Compressed Video. Proceedings of the Third Indian Conference on Computer Vision, Graphics & Image Processing, Ahmadabad, India. On line: http://www.ee.iitb.ac.in/~icvgip/PAPERS/103.pdf.

Kobla, V., Doermann, D., Rosenfeld, A. 1996. Compressed *Domain Video Segmentation*. Technical Report CAR-TR-839 (CS-TR-3688).

Kobla, V. et al., 1997. Compressed domain video indexing techniques using DCT and motion vector information in MPEG video. *SPIE conference on Storage and Retrieval for Image and Video*, pp. 200-211.

Liu, L. and Fan, G., 2005. *Combined Key-frame Extraction and Object-based Video Segmentation.* IEEE Transactions On Circuits And Systems For Video Technology, Vol. 15, No. 7, pp. 869 – 884.

Lu, Y., Gao, W. and Wu, F., 2002. *Automatic video segmentation using a novel background model*. ISCAS.

Sáez, E. et al., 2003. New edge-based feature extraction algorithm for video segmentation. Proceedings of the 15th Annual Symposium on Electronic Imaging, Science and Technology. Vol. 5020, pp. 861 – 872.