See, Hear or Read the Film[⋆]

Carlos Teixeira[1] and Ana Respicio[2]

[1] Lasige/DI/Universidade de Lisboa, Portugal
cjct@di.fc.ul.pt
[2] CIO/DI/Universidade de Lisboa, Portugal
respicio@di.fc.ul.pt

Abstract. Films have been the most entertaining art form during the past century. Sometimes they were inspired in written novels; sometimes they have inspired new written novels. Film scripts are halfway between the film in the screen and the pure world of written imagination. Real time is one of the dimensions lost in the script, breaking the anchors to the time signals of what films are made. This paper presents a full approach for merging these two worlds in the real time dimension. Using subtitles time stamping and a new parallel text alignment algorithm, a time stamped script is produced. This is used to create new enriched narrative films, also presented in the paper.

1 Introduction

A film script is usually available as a separated text document which can be read apart from the film visioning. It is structured into scenes and includes information from both the dialogue and the action. The dialogue lines are usually very close from what can be found in the subtitles, sometimes even closer to what was actually spoken in the audio signal. Each dialogue line begins with the identification of the corresponding character. Usually a description of the character behavior (visual expression and type of speech, etc.), for specific parts of the discourse, is described within the dialogue lines. Action information includes scene and shots identification, and several descriptions of the visual and audio scenario as well as the position of the characters. Often camera directions as well as shot boundaries are also included. Most of the above mentioned information is not explicitly available in the film itself. One aim of the present study is to explore contexts where this information will be helpful to be presented synchronously. However, there is no explicit link between the script and the video, as the first one does not include references to the precise time of the signal.

Nowadays, together with the video and audio signals, many programs and films are distributed including subtitles. Not matching exactly the speech transcription, as it would be done with an idealistic automatic speech recognizer (ASR), these are manual transcriptions often done by human transcriber, which main concern is to keep the subtitle semantic value according to the available slot of time and visual space. Including subtitles serves mainly as a suitable form of presenting language translation or as a

[⋆] This work was partially supported by FCT through the Large-Scale Informatic Systems Laboratory and Centro de Investigação Operacional – POCTI/ISFL/152.

substitute of the audio signal in any conditions where audio is not available. These conditions include problems with the availability of the audio signal, channel transmission, environment of the listener (i.e. noisy or bad sound propagation) or the listener himself (i.e. deaf). On the other hand, if for some reason the access to visual information is not allowed, translated subtitles could be used by a text to speech (TTS) system to replace those subtitles by audio. Additional real-time information can also be useful for the above mentioned restrictions in visual or audio access. If time synchronization is made possible between the available text streams, script extra information can be used revealing advantages comparable to the use of subtitles.

Obtaining a time synchronous script allow us to envisage two main types of applications. The first type will rely on the audio as a substitute for the video signal, in any conditions where video is not physically available. Similarly, as it was mentioned for audio, these conditions include problems with the availability of the video signal, channel transmission, environment of the viewer (i.e. driving) or the viewer himself (i.e. blind). In such cases, information from the script could be helpful if transmitted synchronously with the use of TTS technology. The second type of applications aims to provide an integrated view of the process of film production, namely about what was originally in the mind of the scriptwriter. Current technology allows foreseeing alternative views of the script as well as other texts, namely the possibility to access these documents synchronously with the visioning of the concerning film.

The present work proposes an approach for the automatic alignment of the script with the video signal, producing a synchronized script, and its integration in the visual domain together with extended browsing and querying facilities over the film. The paper is organized as follows. Next section summarizes previous related work. Section 3 describes the general approach for obtaining the time stamped script, and the methodology used for parallel text alignment, focusing on the preprocessing of the text pieces and on the alignment algorithm. Preliminary experimental results are given. Section 4 presents the tool developed for an enriched visualization of video. Finally, some conclusions and perspectives of future work are presented.

2 Previous Related Work

In the past some approaches have been published for aligning texts close related with video. TV programs are sometimes transcribed for documentation on web sites. Gibbon [2] proposes an approach to use this type of transcriptions aligned with the video signal to create improved hypermedia web pages for those programs. The time alignment is found by doing parallel text alignment of the transcripts with the subtitles of the same program. Gibbon refers difficulties arising at many-to-one sentence mappings. Text alignment is first done at the word level using a dynamic programming (DP) procedure. At a second stage, an alignment at the sentence level is searched for.

A framework for aligning and indexing movies with their scripts is proposed by Ronfard and Thuong [6]. They propose a grammar for the script of a given movie. Structural entities, such as shots, scenes, actions and dialogs are extracted. Subtitles are extracted from the video stream using optical character recognition, producing a stream of time-stamped short texts. The alignment is performed by searching for the longest increasing subsequence of matched shots and subtitles.

Aligning the script dialogues with closed captions is the basis to address character/speaker identification in [8]. A DP procedure finding the ”best path” across a similarity matrix allows for defining an alignment at the word level. The approach combines the text alignment with audio segmentation to accomplish audio speaker identification.

Martone et. al. [4] propose the off-line generation of closed captions by synchronizing program transcripts with the texts (subtitles) produced by an ASR system. A DP procedure similar to the one used by Gibbon [2] is applied to perform text alignment at the word level. More recently, Martone and Delp presented two systems based on the alignment of program transcripts with ASR results [3]. One of them allows for locating speakers in news program, while the other one generates DVD chapter titles using the information of the closed captions.

MUSA IST Project(2002-2004) aimed the creation of a multimodal multilingual system that converts audio streams into text transcriptions, generates subtitles from these transcriptions and then translates the subtitles in other languages. MUSA operated in English, French and Greek. When the audio transcripts are available, the ASR MUSA module aligns the audio with the transcript and provides time codes. This is effective if no large omissions are found in the audio, as is the case in many TV productions [5]. The algorithm presented in this paper can help to overcome such difficulties.

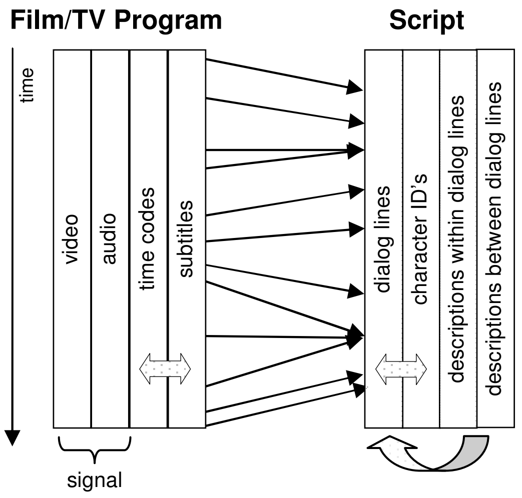


Fig. 1. Aligning a film/TV program with the script.

3 Obtaining Time Stamped Scripts

In this section an approach is proposed for obtaining the time stamped script (TSS). The description of the global approach must consider two main blocks of information which are represented in figure 1: time synchronized information (video and subtitles); and the textual information structured according to the time line but without any detailed anchors to the real time of the movie (script). The arrows intent to represent the above mentioned anchors. In the present work, these time anchors are built exclusively between the subtitles and the dialogue lines. However, we also foresee future developments that will enrich this alignment based on anchors between the video and the audio and the remaining information in the script - character identification as well as the descriptions of the scenes. Also considering future alignments with texts with different time structures, a new text alignment algorithm is proposed in this paper.

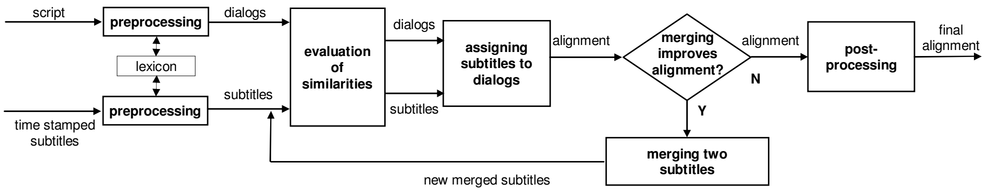


Fig. 2. System for obtaining time aligned scripts.

Figure 2 displays a pictorial description of the new proposed algorithm for parallel text alignment. The process begins by the preprocessing of two texts in order to obtain two streams of word sets: dialogs and subtitles. Then, the core alignment follows including two main phases. In first phase, the algorithm computes a heuristic assignment of subtitles to dialogs that maximizes the global similarity. The second phase searches iteratively for merging pairs of consecutive subtitles that produce local improvements. The last iteration occurs when no further improving merge can be found. In the end, due to several reasons explained in section 3.3, a small set of subtitles and dialogue lines remain unassigned or out of expected time order. A final post-processing phase (third phase) is used to assign these subtitles with unassigned dialogue lines. Each of these dialogue lines should be surrounded by already assigned dialogue lines that encompass the time order where the concerned subtitle occurs. Finally, the resulting assignment is used to link time codes to the dialogs in the script providing the TSS.

3.1 Texts Preprocessing

The alignment process begins by preparing the texts to align: script and time stamped subtitles. Both texts are converted to a common standard format. A token is a sequence of alphanumeric characters (letters or digits) occurring between white spaces or a white space and a punctuation mark. Punctuation characters, as well as non-alphanumeric characters are removed in both texts. Additionally, all the letters are converted to lowercases. This means that every word, number, acronym, etc. will be coded into a lexicon as a token. A single lexicon is build which will allow referring to every token with an unique index.

Each dialog line in the script is represented by a sequence of tokens. Descriptions of an emotion to be expressed by the character or a special setting of the scene, frequently appear as text between parentheses in the script. These may occur within and between dialog lines and are ignored in the alignment. However, they remain linked to the corresponding dialog lines as they are essential to be displayed in the final enriched browsing video. In the same way, the character ID, although not considered in the alignment, remains assigned to each dialogue line. Each subtitle is also converted to a sequence of tokens. Additionally, subtitles are labeled with their time codes (not used in the alignment) that will make possible, afterwards, to produce the synchronized script.

Thus, each subtitle or dialogue line can now be represented by the respective set of indexes. Actually, as the vector space model will be used (subsection 3.2), the order of the tokens is not considered, and this set is represented in a vector with same size as the total number of tokens in the lexicon. Accordingly, each element of this vector contains the number of repeated tokens occurring for a specific lexicon entry. Each token is weighed according to its length. In such way, in the core alignment, more weight is given to longer tokens.

3.2 Text Alignment Algorithm

To produce synchronized scripts, a correspondence between single subtitles or subsequences of consecutive subtitles, and dialog lines in the script, must be established. Furthermore, many to one assigning of subtitles to dialog lines must be produced. The algorithm proposed here uses the similarity measure of the vector space model [7] which has been adopted for many applications in the information retrieval area. This model achieved special relevance in the text retrieval area where the it proves to be relatively language independent. One of the characteristics that can be seen as a problem is the fact that this model does not take into account the order of the tokens inside a document. However this did not seem to compromise the success of this model with such linear structures such as language.

Similarities between subtitles and dialog lines are ranked by evaluating the deviation of angles between the related token vectors. This is equivalent to compute the cosine of the angle between the corresponding vectors. Consequently, the similarity between the i-th subtitle and the j-th dialog line is given by:

$$S(i,j) = \frac{st_i.dl_j}{\|st_i\|\|dl_j\|}, \tag{1}$$

where st_i is the vector representing the i-th subtitle and dl_j is the vector representing the j-th dialog line. A formal pseudo-code description of the core alignment procedure is given in figure 3. The algorithm aligns two lists: m sets of tokens from subtitles $st[i]$ and n sets of tokens from dialogue lines $dl[j]$ – typically $m > n$. The similarity of each subtitle with each dialog line is kept in a $m \times n$ similarity matrix denoted by sim.

The result of the core alignment procedure is the assignment of subtitles to dialogs given in a list denoted by $assigned$. $assigned[i]$ contains the index of the dialog to which the i-th subtitle is assigned. Some subsequences of subtitles are eventually

```
1: function $one2one\_assignment(st, dl, sim)$
2:   $pairs \leftarrow sort(sim)$ {sort pairs (i,j) in decreasing order of sim[i,j]}
3:   $k \leftarrow 1$
4:   repeat
5:     if $is\_free?(st\_in\_pair(k)) \wedge is\_free?(dl\_in\_pair(k) \wedge$
       $sim[st\_in\_pair(k), dl\_in\_pair(k)] > \alpha$ then
6:       $assigned[st\_in\_pair(k)] \leftarrow dl\_in\_pair(k)$
7:     $k \leftarrow next\ k$
8:   until $all\ pairs\ were\ analyzed$
     $\{k = m \times n \vee sim[st\_in\_pair(k), dl\_in\_pair(k)] \leq \alpha\}$
9:   return $assigned$
10: main ()
11:   $Input\ st[1..m]$ {list of subtitles }
12:   $Input\ dl[1..n]$ {list of dialogs }
13:   $assigned[1..m] \leftarrow 0$ {assignments of subtitles}
14:   {first phase}
15:   $sim[1..m, 1..n] \leftarrow compute\_similarities(st, dl)$ {matrix of similarities }
16:   $assigned \leftarrow one2one\_assignment(st, dl, sim)$
17:   {second phase}
18:   repeat
19:     $improvement \leftarrow false$
20:     $i \leftarrow 1$
21:     while $i \leq m \wedge \neg improvement$ do
22:       $max\_sim \leftarrow max\{sim[k, assigned[k]], k = i, i+1\}$
23:       $best\_s \leftarrow s : max\_sim = sim[s, assigned[s]], s = i, i+1$
24:       $merged\_s \leftarrow merge(st[i], st[i+1])$
25:       if $max\_sim < compute\_similarities(merged\_s, assigned[best\_s])$ then
26:         $st[i] \leftarrow merged\_s$ {merge two consecutive subtitles}
27:         $st \leftarrow remove\_subtitle(i+1)$ {update subtitles list}
28:         $sim \leftarrow update\_similarities(st, dl, i)$
29:         $assigned \leftarrow one2one\_assignment(st, dl, sim)$
30:         $improvement \leftarrow true$
31:       $i \leftarrow next\ i$
32:   until $\neg improvement$
```

Fig. 3. Core alignment procedure.

merged during the execution of the process and the current list st is updated accordingly.

In first stage, one-to-one assignment is performed by a greedy heuristic. The matrix of similarities sim is computed using equation (1). The values in this matrix are ranked building a list of ordered pairs (subtitle, dialog) sorted by decreasing order of the corresponding similarities.

Then, iteratively, assignments are made by looking first for the pairs with higher similarity values. The matching is done only if both elements of the pair were not previously assigned and if their similarity value is greater than a given threshold α. In that case, the $assigned$ list is updated accordingly. This cycle continues until a pair with a similarity value lower or equal to the threshold is found or all the pairs have been analyzed.

The number of subtitles is usually greater than the number of dialogs and, consequently, in the end, many subtitles remain unassigned. Giving a minimum value of similarity to accomplish a matching also contributes to this output. Also, potentially, in the result, two consecutive subtitles may be linked to two dialogs located apart.

On the other hand, the process described above does not take into account the order of subtitles and dialogs in their respective streams. Considering these order restrictions in the time neighborhood of each subtitle, and to overcome the above mentioned problems, an improvement phase is executed. Iteratively, the algorithm searches for pairs of consecutive subtitles that can be merged together leading to an increased value of local similarity. This is done by going through each subtitle i and evaluating the impact of merging it with its successor. The merging of two subtitles is considered if it leads to a local improvement. In that case, the list st is updated, replacing the two sets of tokens by the concatenation of both into a single set. Finally, a similarity matrix is updated for another one-to-one assignment run. The whole process repeats until no eligible merges can be found.

3.3 Experimental Results

The approach has been tested with episodes of a series produced by RTP[3], one of the Portuguese major television channels. The episodes have 45 minutes of duration. The script of each episode includes an average of 3500 words, comprising an average number of 200 dialogue lines. The average number of subtitles for each episode is around 300. The algorithm presented above was coded in the C programming language and preliminary experiences revealed very promising results.

Figure 4 illustrates the evolution of the alignment between subtitles and dialogs during the execution of the algorithm for a given episode. The first plot is the result of the initial one-to-one assignment. Each point represents for each subtitle, in the X axis, the dialog to which an alignment was obtained, in the Y axis. The second plot displays the alignment at iteration 50 of the second phase. Most of the subtitles analyzed in this second phase are now assigned. The third plot corresponds to the last merging iteration (107). A few subtitles are still unassigned and a very few were wrongly assigned. Two situations leading to non assignment of subtitles were found. One relates

[3] Rádio Televisão Portuguesa.

with texts that do not exist in the script as, for instance, the first 21 subtitles referring to a summary of the previous episodes as well as a flashback to a specific scene in a previous episode (164-170.) Also, it happened that a single word subtitle reinforced its predecessor, although absent in the script, as for instance, in subtitles 274, 275 – ”Anda, vamos embora.” (Come on, let’s go.); ”Vamos!”. The information about the authoring of the adaptation and edition of the subtitles also appeared at the end of the episode (297-299). Another situation concerned audio information that appeared transcribed in subtitles, as these were conceived for deaf people (subtitles 263 to 270).

The few false alignments, which occurred in less than 1% of the cases, were due to detached situations where the subtitles editor included in the same subtitle speech from different dialog lines. In these cases, the dialogs were very short, for instance subtitle 44 – ”Adeus! Adeus.” (Bye! Bye) – referring to two different characters saying goodbye to each other, was assigned to a later dialog line including the word ”adeus” (bye).

Final alignment, found after post-processing is presented in the last plot. The outlier cases described above, although specific for this episode, are similar to those found in other experiences.

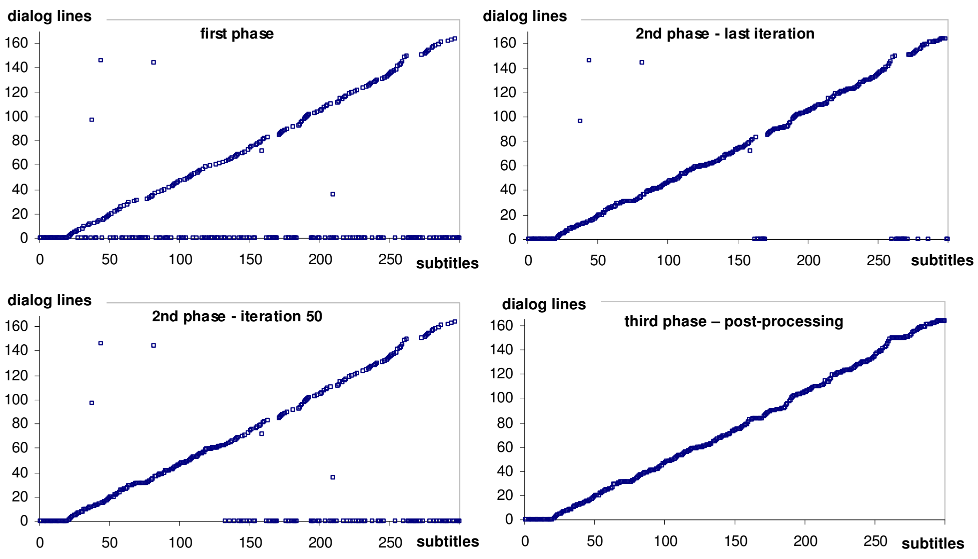


Fig. 4. Evolution of the alignment - assignment of subtitles to dialog lines.

The average number of aligned subtitles was around 87%. In this case, all the subtitles were accounted for. Excluding the situations where the subtitles texts do not exist in the script (flashbacks and extra audio information), and, therefore, that could not be solved by the algorithm, the average number of aligned subtitles was greater than 98%. The merging phase contributed with an improvement of 60%, as the initial one-to-one assignment could only find 51%. The average global similarity reached values around

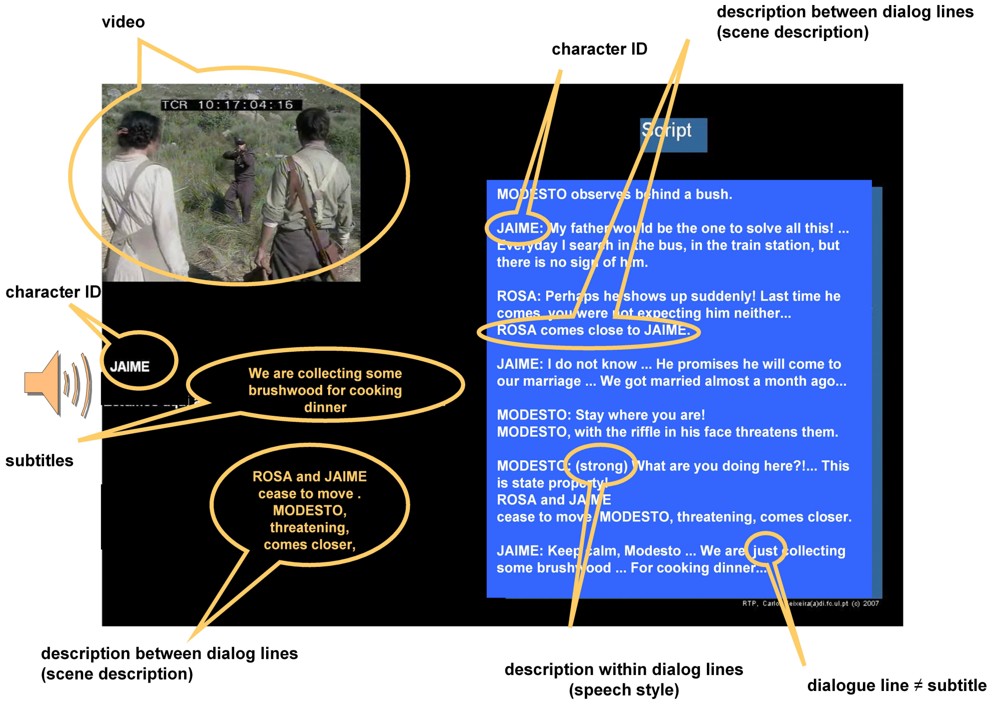
82% when considering all subtitles and 95% excluding those non-assigned. Above all, the detected unassigned subtitles and dialogue lines were found to be useless and probably confusing for the expected applications.

4 Towards an Enriched Narrative Film Player

This section describes how to play and browse the narrative films integrating each of the pieces of information contained in the TSS. These pieces can be very useful after being converted to the audio domain, specially in situations in which vision can not be used - already detailed in section 1. The use of the audio domain can be done with TTS systems. Audio synthesized dialogue lines sound necessarily very unnatural when compared to the original audio dialogues or to the traditional output from dubbing. However, the remaining information from the script (character ID, scene descriptions, etc.) is actually the main novel addition to consider and can be synthesized with lower quality requirements. Namely, emotional speech aspects which are essential for the dialogues, such as prosody, are not so crucial for this type of information. Actually, if a real speaker should pronounce this, his speech should be even less emotive than narrator. In order to be efficiently discriminated, each piece of information in the TSS can be assigned to different audio spatial channels and, preferably, to different speaker voices. However, reducing a demonstration of this approach to the audio representation was found not very convincing, since it does not allow an easy evaluation for users who are not really restricted to the audio input. Considering this, at the present stage of this work and after implementing a robust algorithm for text alignment, the second concern is actually to provide a visual integration of the TSS with the film in order to provide a convincing demonstration of the use and efficiency of the approach. The integration and navigation of video and other media is the central aspect in hypervideo, where video is not regarded as a mere illustration, but can also be structured through links defined by spatial and temporal dimensions [1].

A small video demonstration of the present work is available in:
“http://www.di.fc.ul.pt/∼cjct/ICEC2007/“.
It shows 80 seconds of a scene from an episode of the television series “Quando os Lobos Uivam” (when the wolves howl), an adaptation of the novel of Aquilino Ribeiro with the same name made by Francisco Moita Flores for the RTP. In this scene, a couple of peasants (Rosa and Jaime) collect some brushwood from an area they were recently forced to sell to a capitalist. A national guard named Modesto founds and menace them. The original titles are in Portuguese, they were translated just for a better illustration in the present paper.

Figure 5 was obtained from the same portion of this film. The video is shown in the upper left corner and the name of the speaking character is shown right below. The subtitles are also presented bellow the video just according to common standards. Further bellow, a description about the specific shot can be found. All the right side of the display area was used for the script. This is shown as a scrolling text with new dialogue lines and related comments coming from below shortly after the corresponding subtitle was shown. It seems like the subtitles feed the scrolling script but, as mentioned before, these are often different from the dialogue lines and can sometimes include extra information about the actor expected behavior. Together with the dialogue lines, the corresponding shot description is included – the same way as it appears in the script and it was previously shown the bottom left corner.


video
character ID
description between dialog lines (scene description)
TCR 10:17:04:16
Script
character ID
JAIME
We are collecting some brushwood for cooking dinner
subtitles
ROSA and JAIME cease to move . MODESTO, threatening, comes closer,
description between dialog lines (scene description)
description within dialog lines (speech style)
dialogue line ≠ subtitle
RTP, Carlos.teixeira(a)di.fc.ul.pt (c) 2007

MODESTO observes behind a bush.

JAIME: My father would be the one to solve all this! ... Everyday I search in the bus, in the train station, but there is no sign of him.

ROSA: Perhaps he shows up suddenly! Last time he comes, you were not expecting him neither...
ROSA comes close to JAIME.

JAIME: I do not know ... He promises he will come to our marriage ... We got married almost a month ago...

MODESTO: Stay where you are!
MODESTO, with the riffle in his face threatens them.

MODESTO: (strong) What are you doing here?!... This is state property!
ROSA and JAIME
cease to move. MODESTO, threatening, comes closer.

JAIME: Keep calm, Modesto ... We are just collecting some brushwood ... For cooking dinner...

Fig. 5. Enriched narrative film.

Each of these text streams have different showing times. However, all these timing sequences were built from the subtitles time codes. Table 1 shows the adopted time shifts for each piece of information taking the corresponding subtitle time code as reference.

Text unit	character ID	subtitle	shot description	dialogue line
begin	-1	0	-2	+1
end	+5	0	+6	+6

Table 1. Selected delays (in seconds) for the script units with regard to the subtitles time code.

The integration of all the text elements in a single video allows browsing capabilities where the player can do fast forward, backward or accessing directly a specific shot while all the elements appear synchronously, namely the scrolling TSS. More than that, simple query facilities will be integrated which will allow to directly access a particular shot given some words spoken by a specific character. Other than character ID and subtitles or dialogue lines, other queries can be made based on the shot description, scene names, expected character behavior and any combinations between these.

5 Conclusions

This paper proposes an approach to create enriched narrative films. The central idea is to enrich the experience of watching films by providing synchronized textual information from the script. The approach relies on aligning the script with the subtitles and a new text alignment algorithm is presented. The alignment is made at sentence level and avoids some differences between sentences, such as those resulting from word ordering as well as structural differences between documents. This is a novelty regarding other text alignment algorithms from the literature, that are mainly based on dynamic programming procedures working at the word alignment level [2, 4, 3]. This feature also provides flexibility to deal with more complex alignments allowing the integration of other related textual contents, as well as to consider flashbacks and flash forwards of scenes. Moreover the algorithm is almost language independent and robust enough to deal with other complex alignments.

Future work will concentrate on applying the proposed algorithm to a broad collection of English films and upon further testing by comparing it with other text alignments algorithms from the literature. Experiments will also consider other related texts namely behind the scene documents. An envisage application is to use a text to speech system to convert the script and produce contents specially designed to visual-impaired people. The presented approach could have a high impact on next generation multi-media retrieval, digital content management or entertainment electronics for the home environment.

6 Acknowledgments

The authors want to thanks RTP, namely João Sequeira and the RTP Multimedia Director Francisco Teutónio Pereira for providing the videos and scripts used in the present publication; as well as the anonymous reviewers for the useful comments.

References

1. T. Chambel and N. Guimarães. Context perception in video-based hypermedia spaces. In *Proc. ACM Hypertext'02*, College Park, Maryland, USA, June 2002. ACM.

2. D.C. Gibbon. Generating hypermedia documents from transcriptions of television programs using parallel text alignment. In *Proc. Eighth Int. Workshop on Research Issues In Data Engineering, Continuous-Media Databases and Applications*, pages 26–33, 1998.
3. A. Martone and E. Delp. An overview of the use of closed caption information for video indexing and searching. In *Proc. CBMI Fourth International Workshop on Content-Based Multimedia Indexing*, 2005.
4. A. Martone, C. Taskiran, and E. Delp. Automated closed-captioning using text alignment. In *Proc. SPIE International Conference on Storage and Retrieval Methods and Applications for Multimedia*, pages 108–116, 2004.
5. Demiros I. Prokopidis P. Vanroose P. Hoethker A. Daelemans W. Sklavounou E. Konstantinou M. Piperidis, S. and Karavidas Y. Multimodal multilingual resources in the subtitling process. In *Proc. of the 4th International Language Resources and Evaluation Conference*, Lisbon, Portugal, May 2004. LREC.
6. R. Ronfard and T.T. Thuong. A framework for aligning and indexing movies with their script. In *Proc. IEEE International Conference on Multimedia and Expo (ICME 2003)*, volume 1, pages I21–24. IEEE, 2003.
7. G. Salton, A. Wong, and C.S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613620, 1975.
8. R. Turetsky and N. Dimitrova. Screenplay alignment for closed-system speaker identification and analysis of feature films. In *Proc. IEEE International Conference on Multimedia and Expo (ICME 2004)*, volume 3, pages 1659– 1662. IEEE, 2004.