

A Computer-Vision-Assisted System for Videodescription Scripting

L. Gagnon, C. Chapdelaine, D. Byrns, S. Foucher, M. Héritier, V. Gupta
Computer Research Institute of Montreal (CRIM)
405 Ogilvy Avenue, Suite 101, Montreal, QC, CANADA, H3N1M3

Abstract

We present an application of video indexing/summarization to produce Videodescription (VD) for the blinds. Audio and computer vision technologies can automatically detect and recognize many elements that are pertinent to VD which can speed-up the VD production process. We have developed and integrated many of them into a first computer-assisted VD production software. The paper presents the main outcomes of this R&D activity started 5 years ago in our laboratory. Up to now, usability performance on various video and TV series types have shown a reduction of up to 50% in the VD time production process.

1. Introduction

Videodescription (VD), also known as audio description or described video, is a narration of key visual elements in films that are added on the non-speech segments of the audio track in order to help blinds and visually-impaired to make a mental picture of what is occurring on the screen.

The VD production process is a very tedious task done mostly manually. It implies pre-viewing the film 1 or 2 times to identify the audio segments where the VD narration can be added, identify the pertinent visual content to describe, and then write a VD script, make the script read by a voice talent (usually a professional actor), synchronize, remix, etc. This represents a workload ratio of about 25:1 for films and documentaries, and up to 40:1 for more “artistic” or “impressionistic” productions, e.g. those containing only music and visual elements [1]. Despite of that, the growth of the VD industry is accelerating because of the increasing demand from the blind community and regulations imposed by the national broadcasting organizations.

The VD production process is related to film indexing/summarization task since it requires identifying visual content that could be relevant to understand the story flow, such as scenes transitions, who the actors are and when they appear, where the action takes place, what the textual information says, where the silent segments are in the audio track, etc.

Adapting the indexing/summarization paradigm to VD production is necessary because the use-cases are different. For content-based video retrieval, the user is typically a video archivist who looks for a specific portion of the film containing a specific audiovisual content. In principle, there is no restriction on the visual content to be encoded (within the current state-of-the-art) as they are all potentially interesting for a video query. On the other side, visual content to be described for VD is often restricted by the content of the audio band (one cannot add an additional VD when an actor is talking or when the sound level is high like in action scenes). Also, there is no need to describe the visual information when it does not add useful insight to imagine what is going on in the scene. An automatic VD scripting generator would thus require a combination of interacting video and audio processing modules, integrated with a human-computer interaction adapted to the production process.

In this paper, we describe the first fully operational computer-assisted VD software tool (VDManager) using integration of automatic audio detection, computer-vision and synthetic voice technologies. VDManager is currently in its beta version and is used on a regular basis in our laboratory to produce professional VD.

The paper is organized as follows. Section 2 gives a brief overview of the open literature and current state-of-the-art on the various VD topics. Section 3 presents the layout of the system. Section 4 gives some usability performance results of the VDManager. Finally, Section 5 gives a working example of the usage of the various modules GUIs.

2. Literature review on VD

There are many aspects to VD: accessibility, usability, guidelines, regulations, software tools, etc. The amount of literature is variable among those topics and rather recent but some international research initiatives exist on those subjects ([5][6][21]). We give here a quick survey.

There is not yet universal standard for producing VD scripts although some guidelines and typologies exist ([6]-[8],[2],[10]-[13]). Recent studies analyze the types of VD information currently done on film productions ([11][12])

as well as the type of corpus used ([10][13]). One can generally conclude from those studies that VD is often concentrated to [7]:

- Actions and details that would confuse the audience if omitted
- Actions and details that add to the understanding of personal appearance, setting, atmosphere, etc.
- Visible emotional states, but not invisible information as mental state, reasoning, or motivation
- Titles, subtitles, credits, etc.

Those general concepts are widely accepted in the industry and establish the challenge for any technological advancement on computer-assisted VD.

Usability aspects, either on the producer and blind user side, are also started to be studied ([4], [14]-[20]). Recent interviews we have performed with producers have shown that the VD production process is also not standardized and vary from one company to another. However, there are common needs among VD producers: (1) automatic off-line video abstraction to reduce the number of viewings, (2) consistent identification of recurring actors or places to avoid confusing the listeners, (3) audio segmentation, (4) tools to speed-up the scripting process to cope with the increased regulations, (5) use of synthetic voice to reduce production cost, (6) video indexing as a by-product of VD (i.e. a same software tool that do both), etc.

On the end-users side (blinds and visually-impaired), one has found during our screening interviews that they do not appraise VD in the same way. It has been demonstrated recently that user needs are quite diverse ([15][17][19][20]). Whatever the quality or quantity of the VD track, each individual has preferences depending on his or her level of vision, tastes, and experience. Thus, a computer-assisted VD production tool should also feature an option to provide various types and levels of VD. In addition, designing and implementing a VD player that is adaptive is an important consideration to take into account. Research along this way has been engaged by our team. A beta version of an adaptive VD player has been developed and has been reported elsewhere ([15][19][20]).

No technical literature exists on the technical aspects of VD production; only standard video editing tools are currently used in the industry to assist the VD production process. Some software tools like Softel ADePT (Softtel-USA, 2001) and Magpie (NCAM, 2003) support VD production but without any automatic audio-visual processing. To our knowledge, the only works targeting the analysis and development of off-line computer-assisted VD are the one of Lakwitz and Salway [1] and Gagnon et al. ([14][15]). For on-line VD production, it is even poorer. Practical real-time VD for broadcast television does not exist yet, although a first tentative study has recently been explored for Web-based applications ([3][9]). LiveDescribe [3] is designed to allow near real-

time VD to be added to on-line content by VD describers for events like emergency broadcasts, parades, concerts, and other events which are often viewed only once, and relevant only for the time in which they appear. Our VManager system is for off-line VD scripting.

3. VManager

Figure 1 shows the main VManager GUI. There are five main build-in modules accessible through it: (1) plug-in scheduler and interactive visual detection validation, (2) interactive timeline, (3) interactive VD editing, (4) embedded video player and (5) embedded VD player with synthetic voice rendering (not shown). Almost all interactions can be done via keyboard shortcuts to speed-up the VD production process which is also compliant with the actual working habits of the VD producers.

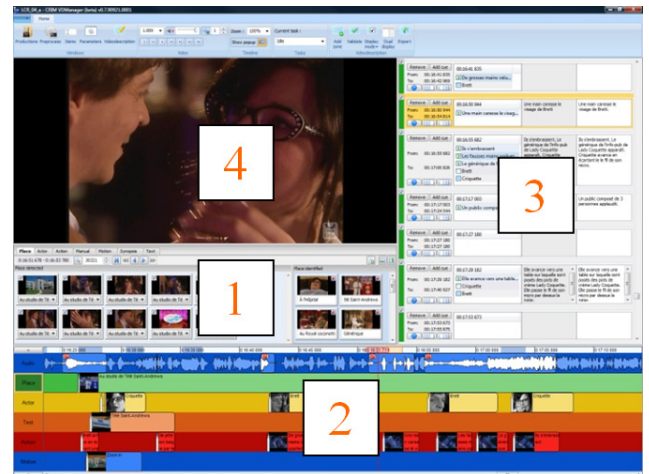


Figure 1. Overview of the VManager GUI. The main working windows are (1) plug-in scheduler and interactive audio-visual detection validation, (2) interactive timeline, (3) interactive VD editor, (4) embedded video player and (5) embedded VD player with synthetic voice rendering (not shown).

3.1. Plug-in scheduler and interaction (1)

The plug-in scheduler dynamically loads the detection plug-ins (speech/non-speech segmentation, key-places, key-faces, camera motion, and text). Each plug-in GUI provides access to features detection and validation.

The plug-in scheduler parallelization can be achieved on multi-core computers by using different threads. It automatically determines the number of processor cores to optimize the number of jobs to launch simultaneously.

The plug-in interaction panel has navigation options: start, previous shot with detection, previous shot, next shot and next shot with detection. The same navigation options are offered for each plug-ins to provide consistency of interaction for the producers.

3.2. Interactive timeline (2)

The interactive timeline displays all features detected by the plug-ins. The length of the timeline object represents the duration of the detection. This enables the producers to have an overview of the video as well as the frequency and duration of features to be included in the VD. Each object possesses a white cursor, called the feature cue time, which can be used to synchronize VD rendering; this cue time is used by the VD editor panel (3) to regroup features in VD rendering segments. Timeline cues can be modified directly by the user to change durations or group objects together. Finally, the timeline provides several ways of controlling the video player. The cursor can be used to scroll at a specific position or the header can be double-clicked to seek directly at that position. Each timeline object can be “played” by left double-clicking while a single click will seek at its beginning.

3.3. Interactive VD editor (3)

As the user identifies VD tags, those are cumulated in the interactive VD editor panel. Each tag is ordered according to the time code related to the frame where the visual element is located. A tag or a group of tags are either aligned to a non-speech segment or to a speech segment. In this later case, the zone is marked in red and indicates to the user that the tags must be either omitted from the VD or they need to be resynchronized in a non-speech segment.

This panel provides an automatic draft VD that is essentially the tags of the plug-in detection outputs. This draft version can be edited by the user to generate two additional more verbatim versions (Figure 2): one where the sentences fill the whole non-speech zones (called standard version) and one that gives a more detailed VD which requires pause the video during the VD voice rendering (called extended VD version).

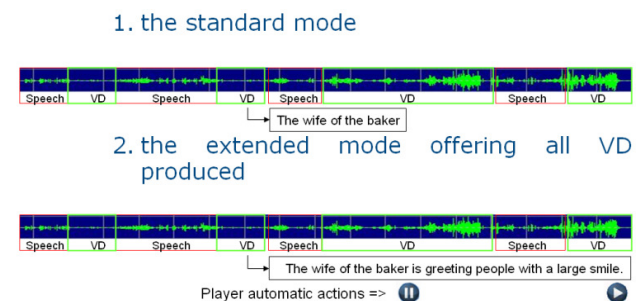


Figure 2. Sketch showing the difference between the standard and extended version of the VD synthetic voice rendering

The VD editor panel offers a full interactive scripting environment at every level. For instance, cue zones can be added in the non-speech segment in order to pace VD

rendering with the visual content. Furthermore, the duration of VD rendering needs to fit in the duration with the non-speech segment. The rendering length can be validated using the real file length from the voice synthesis which can be heard to assess its quality and understandability.

3.4. Embedded VD player

The embedded VD player is a specialized player that can read media and render the various VD types produced by the VDManager using a commercial synthetic voice. Also, it can export the VD embedded within the media to a new media file, which could be rendered on any traditional player. The architecture is based on DirectShow API and the VD rendering is made available through the DirectShow Editing Services. This high level API let the VDPlayer adds a second sound track to the media in which VD can be added. The user can select between the standard and extended VD rendering options. When the extended version is selected, the video pauses automatically to render the whole VD sentence. This pause is a still frame, inserted into the video and repeated for the required amount of time while no sound comes from the original soundtrack.

4. Core technologies

VDManager integrates various audio and video content analysis tools that we have developed and optimized especially for VD scripting. The following is an overview of the main plug-ins used by the user during a VD production process: speech/non-speech segmentation, place identification and actor face recognition.

4.1. Speech/non-speech

The audio is first divided into short homogeneous segments using acoustic change point detection (CPD) algorithm. This change point detection algorithm uses a symmetric Kullback-Leibler (KL2) metric, and a 13-dimensional feature vector (12 MFCCs + energy) with diagonal covariance matrix [29].

The homogeneous audio segments obtained by the CPD algorithm are refined by an iterative Viterbi re-segmentation step where we keep the number of segments fixed, but we change the segment boundaries based on the best path found through Viterbi decoding. In Viterbi decoding, the mean and variance are used as models for each segment, and the decoding finds the maximum likelihood sequence of segments with the given means and variances. During decoding, we impose a minimum segment length of 1 second. The Viterbi re-segmentation step is iterated until convergence or a maximum of 6 iterations.

The audio segments obtained after Viterbi re-segmentation are classified using Gaussian mixture models (GMMs) for speech, speech + music, speech + noise, music, and other non-speech sounds. The GMMs use 64 mixtures with 26 feature parameters (12 MFCCs + energy + 13 delta parameters) each.

In the final step, we use the voice activity detector to remove silent segments from audio. These silent segments are then removed from the segments classified as speech or speech+music or speech+noise by the GMM classifier. The segments that do not have the label of speech or speech+music or speech+noise are labeled as non-speech.

Initially, the GMMs were trained from broadcast news audio. These GMMs trained from broadcast news did not perform very well for audio from short films. We then adapted these GMMs to the audio from these short films by manually segmenting and labeling parts of the audio into speech, speech+noise, speech+music, music or other.

4.2. Key-places identification

The key-place identification algorithm consists in finding links between key-frames of a common key-place based on the use of a probabilistic latent space model over the possible local matches between the key-frames image set. This allows the extraction of significant groups of local matching descriptors that may represent characteristic elements of a key-place. We use the concept of “Bag of Visterms” (BOV) for representing each key-frame. It consists in quantizing local SIFT descriptors into visterms. A visterm is defined as a set of matching local descriptors between different images.

An exhaustive evaluation of this key-place clustering approach was conducted on various datasets:

- An in-house dataset consisting of 11 key-places classes (5 outdoor, 5 indoor and a “noise” class). Each outdoor and indoor class is made of 10 images with different view-points and illuminations. The noise class is composed of 10 random indoor images and 10 random outdoor images.
- Two full-length movies of about 1.75 hours each.
- Two public image datasets of places; namely the “Raglan” and “Valbonne” datasets [22].

Results revealed that our method is very efficient for near-duplicate object/background detection with weak overlap. Performance measurements on full-length movies indicate a recognition rate of about 75% on the key-places clustering with a false alarm of approximately 2%.

During the batch process, key-places are grouped to form recurring thus relevant places of the video. A key-place can be recognized over one or many shots composing a scene. The user must validate that all the shots in the scene are effectively part of the scene and name the place. The name is propagated to all similar

places in the video.

4.3. Key-faces identification

Our algorithm follows four steps: detection, tracking, normalization and clustering. The near frontal faces are detected using an improved version of a cascade of weak classifiers [23][24]. A recursive nonparametric discriminant analysis (RNDA) was added to the training to obtain more discriminant features [25][26]. The tracking is done by a particle filter where the particle weight for a given ROI depends on the face classifier response [27]. Face frames corresponding to the best frontal views are sampled along the face trajectory and used for recognition. The faces are then normalized by detecting the face center and estimating its width in order to compensate for translation and scaling variations. Face features are derived from a 2-Dimensional Principal Component Analysis (2DPCA) technique [28] which directly seeks the optimal projective vectors from face images without preliminary image-to-vector transformation.

The user interacts with the plug-in in the following way.

- For each shot containing face tracks, a list of key-faces is presented to the operator.
- The user can enter the name of a new person and indicates if this person will be seen again (e.g. main character) or not. A new entry in the face database is then created. The first face sample available is selected for the training dataset.
- When going to the next shot, the automatic face recognition is performed on the new face tracks using the faces already present in the face database.
- When the user makes a correction on a face name, the system will choose automatically the best face sample in the track that would have improved the automatic decision. This best sample is automatically added to the corresponding training set.

The performance of this semi-supervised process was evaluated on the full-length movie “Inside Man” (Figure 3). The user labeled the tracks for the main actors (33), false alarms and unknown faces. 754 face tracks were detected, 570 represented main actors on which 371 were correctly recognized (65%). The recall values fluctuate largely between actors. For the top actors, the recall becomes stable around 45% after 100 occurrences. For secondary actors, higher recall values are reached maybe because their appearance is more stable. This evaluation is probably pessimistic because we did not evaluate the system performance as a verification system (i.e. the system ability to automatically assign the “ignore” label to the track if no links are found).

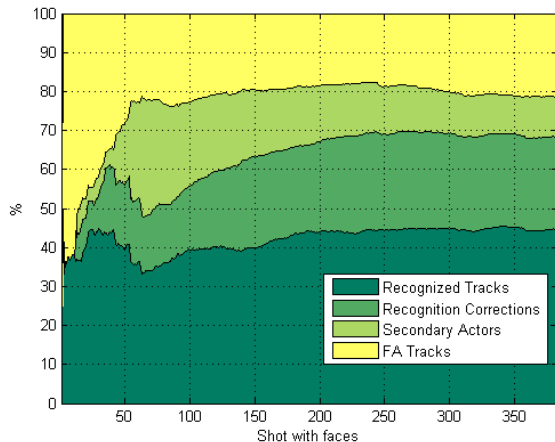


Figure 3. Recognition performance history of key-faces for a VD session

4.4. Action and synopsis

In addition to the above plug-ins, two others allow user to enter manual information that are either too difficult to detect automatically or not associated to audio-visual content: action (e.g. human action or particular situation in the scene) and synopsis.

5. Usability performance

The sequence of tasks to finalize a VD production with the VManager is the followings:

- Create a project and load a film
- Select the speech/non-speech segmentation plug-in and any other visual content extraction plug-ins appropriate to the film
- Launch the batch detection
- Validate the speech/non-speech segmentation
- Validate the key-places, key-faces, text and camera motion detections
- Synchronize the cue times for VD rendering
- Edit the automatically generated VD
- Generate the various VD types (standard or extended)
- Launch the internal VDPlayer for quality check
- Save the production

In order to measure the production performance, we have implemented a log time function that requires the user to indicate when a group of specific tasks was being done for validating actors, validating scenes, validating speech/non-speech segments, adding actors, adding actions descriptions, synchronizing VD, etc. The log time does not include the batch detection, audio files creation or synchronization of video and audio. Batch processing takes typically 3-5 times the film duration on a 4-core processor.

We have done usability tests on various films types: short films, TV series and documentaries. Below are some

performance results for the speech/non-speech and key-place plug-ins, as well as for the overall VD production time performance.

5.1. Speech/non-speech segmentation

Table I gives user interaction measures for the speech/non-speech segmentation for 4 episodes of a TV series (about 20 min each). The column “Detected” gives all the non-speech segments detected by the plug-in, “Validated” gives those validated by the user (true positives) and “Added” gives those added by the user. We see that the results are quite variable. Two main factors contribute to that: variable quality of the audio band and non-optimal adaptation of the algorithm on each series. An audio adaptation step will be added soon in the VManager GUI to improve those results. On average, the precision is 72 % and the recall is 85 %.

TABLE I

Nb. of user interactions with the speech/non-speech plug-in			
Production code	Detected	Validated	Added
DGPCV_23	86	38	14
DGPCV_24	249	192	7
LCR_06	159	135	11
Minuit6	39	20	37
	533	385	69

5.2. Key-places identification

Table II shows how many times the user must interact with the film in order to validate (label) the various places with or without the VManager. Results are presented for 3 different productions: a full-length adventure film of about 150 min. (“Babine”) and two episodes of a TV series of about 20 min. each. The results show a significant reduction in the number of interactions.

Nb. of user interactions with for key-place identification on film shots with or without the VManager

Production code	Nb. of interactions without VManager	Nb. of interactions with VManager
Babine	2193	1235
DGPCV_24	195	106
LCR_06	370	116

5.3. Key-faces identification

Table III gives results for the key-faces plug-in. The results show that, on average, 54% of the actor faces are correctly assigned to an actor class and no principal actor face is missed.

TABLE III
Nb. of user interactions with the key-faces plug-in

	1	2	4	8
Production code	All detected faces	Detected faces assigned to an actor class by the plug-in	Detected faces validated as good class assignment by the user	Added by the user
DGPCV_23	286	221	95	0
DGPCV_24	366	298	160	0
LCR_06	374	303	211	0
Minuit6	141	106	36	0
	1167	928	502	0

5.4. Global production time performance

Table IV gives the average production time ratio (VD production time / film length) we obtained so far on about 700 minutes of various film types.

We observe that the production ratio is significantly lower than the average 25:1 done in the industry, especially when the user becomes familiar with the film type. So even though the automatic audio-visual content extraction plug-ins is not perfect (and they will never do) a powerful GUI interaction mechanism can compensate for that. Those results are very encouraging taken into account that the tool is yet in its beta version. The large ratio value of 39 obtained on one episode of the films #1 is because it was the first of all done by the user.

TABLE IV
Average VD production time on various film types using the VDManger

	Series title	Film type	Nb. of episodes	Length per episodes (min.)	Max/Min production time ratio
1	<i>Kino Québec</i>	Short films	3	5	39/10
2	<i>La vie en vert</i>	Environmental documentary	2	20	7/5
3	<i>Rebut globale</i>	Environmental documentary	8	20	12/7
4	<i>Découverte</i>	Scientific documentary	12	10	15/11
5	<i>Le coeur a ses raisons</i>	Comic TV series	4	20	16/10
6	<i>Dans une galaxie près de chez-vous</i>	Comic TV series	9	20	19/10
7	<i>Minuit le soir</i>	Suspense TV series	4	20	18/16
		TOTAL:	42	695 min.	

We also observe that the production ratio for films #2 is

smaller than for the similar films #3. This is related to the characteristics of the contents which are different. Indeed, looking at the data presented in the Table V, we see that on average the number of non-speech segments for films #2 is fewer, thus much less actions could be described. In fact, we observed in all films that the time to manually insert description of the action is crucial bottleneck since it is not yet assisted by any automatic action recognition.

TABLE V
Average duration value (min.) of semantic data to process in 3 film types

Data	<i>Kino Québec</i>	<i>Rebut global</i>	<i>La vie en vert</i>
Length	0:12:26	00:23:53	00:24:49
Shots	152	254.5	348.5
Non-speech	45.0	51.6	38.5
Scenes	6.5	8.5	7.5
Faces	5	10.1	6.5
Action	72.5	41.6	16
Text	11.5	9.6	18.5

Another interesting point we found from our performance analysis is that the first episodes of a TV series required more production time than the followings. This makes sense since at the beginning of a new series more time is needed to understand the important visual element to describe. However, time is saved later for the remaining productions since places are often the same as well as the same principal actors.

6. VDManger usage example

In this section, we present some snapshots of the VDManger GUI when used for a VD production session.

Figure 3 shows a typical timeline usage. The white arrow on the left-hand side of each tab (place, actor, action, etc.) allows the user to place the VD rendering “cue” wherever it is best suited. Here, the user has placed the face of the actor named “Brade” in the non-speech segment at left. We can also see that it is possible to insert a “cue” in a non-speech segment such that the VD starts at a specific time. Here, the user has chosen to start the action description “He takes out a tape recorder” slightly after the beginning of the non-speech segment.

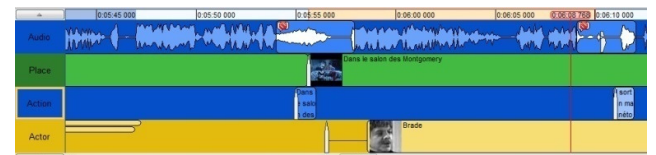


Figure 3. Example of using the interactive timeline to change the VD description cue.

Figure 4 shows an example of a key-place validation process. The key-place plug-in detects a location change in

the film and associated the correct place to the image. The detected place is inserted automatically on the timeline after user validation. In the window “Place identified”, we see the places validated so far by the user.

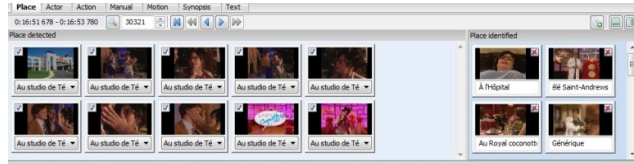


Figure 4: Example of using the plug-in panel for key-place validation.

Figure 5 shows an example of a key-face validation process. The key-face plug-in detects faces and associates the correct names for the characters. In this case, the plug-in detects faces of the two actors named “Criquette” and “Ridge” (both previously validated by the user) and place them on the timeline when they appear. In the “Panel name” window, we see the characters already validated by the user.

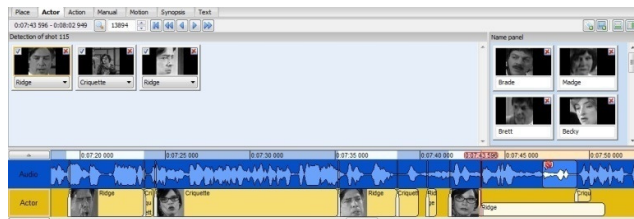


Figure 5: Example of using the plug-in panel for key-face validation.

Finally, Figure 6 shows an example a VD editing process done within the VD editor panel. The “Validate” button tests whether the VD rendering will fit in the available non-speech time segment (depending on the selected synthesis voice). Highlight (red marked) is indicative of a too much longer VD.

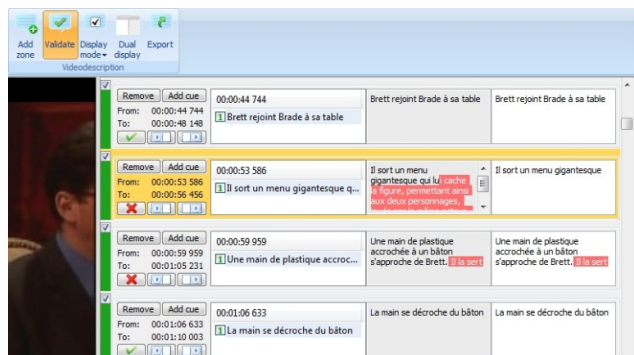


Figure 6: Example of using the VD validation panel to enter and edit the VD script.

7. Conclusion

The computer-assisted VD production system presented in this paper has been developed over the last 5 years by a group of 10 people. We think it is the first of its kind and an original application of video indexing/summarization. The system is in a beta version and currently used for VD production by a dedicated VD scripiter.

Such tool aims at reducing workload and helping VD scripiter to produce more consistent content, more quickly, in order to increase the production quantity and ultimately the accessibility of media documents for people with vision loss. This is timely because the VD industry is growing due to the imposition of regulations requiring broadcasters to add more descriptive narration in their programming.

In our system, all interactions (mouse clicks, windows openings, detection validation, false alarm correction, VD script modifications, elapsed time between each task, etc.) are logged during a production session in order to accumulate user interaction performance. We think this is the best way to determine the overall performance of a user-oriented tool since automatic audio-visual content extraction is never perfect. A simple and powerful GUI interaction mechanism can compensate for that.

We are continuously improving the plug-in detection and computational performance. In the near future, we will test the system at the facilities of our VD producer partners and add a specific module to output video indexing data. The same system architecture can serve for both usages in the film as well as the television production sectors.

Acknowledgements

The authors would like to thank the following organizations: Kino Québec, BlueStorm Inc. and National Film Board of Canada and SETTE Inc. for providing video data professional VD production feedback.

References

- [1] J. Lakritz, A. Salway, “The Semi-Automatic Generation of Audio Description from Screenplays”, Dept. of Computing Technical Report CS-06-05, University of Surrey, 2006
- [2] A. Salway, A. Vassiliou, K. Ahmad, “What Happens in Films?”, IEEE Conference on Multimedia and Expo, ICME 2005.
- [3] C. Branje, S. Marshall, A. Tyndall, D. I. Fels, “LiveDescribe”, Proceedings of the Twelfth Americas Conference on Information Systems, Acapulco, Mexico, August 2006
- [4] E. Schmeidler, C. Kirchner, “Adding Audio Description: Does it Make a Difference?”, Journal of Visual Impairment & Blindness, 95, 197-203, 2001
- [5] E-Inclusion Research Network: <http://e-inclusion.crim.ca>

- [6] Canadian Network for Inclusive Cultural Exchange, "Online video description guidelines", <http://cnice.utoronto.ca/guidelines/video.php>
- [7] J. Clark, "Standard Techniques in Audio Description", <http://www.joeclark.org/access/description/ad-principles.html>
- [8] Office of Communication, "ITC Guidance On Standards for Audio Description", http://www.ofcom.org.uk/static/archive/itc/itc_publications/codes_guidance/audio_description/index.asp.html
- [9] Canadian Network for Inclusive Cultural Exchange: <http://cnice.utoronto.ca>
- [10] P. J. Piety, "The Language System of Audio Description: An Investigation as a Discursive Process", *JVIB*, Vol. 8, pp. 1-36, 2004
- [11] J. M. Turner, "Some Characteristics of Audio Description and the Corresponding Moving Image", Proc. of the 61st ASIS annual meeting, Pittsburgh, PA, October 24-29, 1998, ed. Cecilia M Preston. Medford, NJ: Information Today, 108-117, 1998
- [12] J. M. Turner, E. Colinet, "Using Audio Description for Indexing Moving Images", *Knowledge organization* 31, no 4 : 222-230, 2004
- [13] A. Salway, "A Corpus-Based Analysis of Audio Description", pp 151-174, in "Media for All, Subtitling for the Deaf, Audio Description, and Sign Language", DÍAZ CINTAS, Jorge, Pilar ORERO and Aline REMAEL (Eds.), Amsterdam/New York, NY, 2007
- [14] L. Gagnon, S. Foucher, F. Laliberté, M. Lalonde, M. Beaulieu, "Towards an Application of Content-Based Video Indexing to Computer-Assisted Descriptive Video", *Proceedings of Computer & Robot Vision*, 2006
- [15] L. Gagnon, S. Foucher, M. Héritier, M. Lalonde, D. Byrns, C. Chapdelaine, J. Turner, S. Mathieu, D. Laurendeau, N. T. Nguyen, D. Ouellet, "Towards Computer-Vision Software Tools to Increase Production and Accessibility of Video Description to Visually-Impaired People", *Universal Access in the Information Society*, Springer-Verlag, Vol. 8, pp. 199-218, 2009
- [16] D. I. Fels, J. P. Udo, J. E. Diamond, J. I. Diamond, "A First Person Narrative Approach to Video Description for Animated Comedy", *Journal of Visual Impairment and Blindness*, 100(5), 295-305, 2006
- [17] S. Mathieu, J. M. Turner, *Audiovision interactive et adaptable*, Technical Report, 2007, <http://hdl.handle.net/1866/1307> (in French)
- [18] P. Orera, "Sampling Audio Description in Europe", pp. 111-126, in "Media for All, Subtitling for the Deaf, Audio Description, and Sign Language", DÍAZ CINTAS, Jorge, Pilar ORERO and Aline REMAEL (Eds.), Amsterdam/New York, NY, 2007
- [19] C. Chapdelaine, L. Gagnon, "Accessible Videodescription On-Demand", *Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility*, Pittsburg, 2009
- [20] L. Gagnon, "Automatic Detection of Visual Elements in Films and Description with a Synthetic Voice - Application to Video Description", *Proceedings of the 9th International Conference on Low Vision*, Montreal, 2008, 4 pages (on CD-ROM)
- [21] DTV4ALL <http://www.psp-dtv4all.org/>
- [22] F. Schaffalitzky, A. Zisserman, "Multi-view Matching for Unordered Image Sets, or "How do I Organize my Holiday Snaps?" Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark, 2002.
- [23] P. Viola, M.J. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features", *CVPR*, 2001
- [24] E. Lienhart, J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection", *ICME*, 2002
- [25] P. Wang, Q. Ji, "Multi-View Face and Eye Detection Using Discriminant Features", *CVIU*, Vol. 105, 2007
- [26] S.C. Brubaker, M.D. Mullin, J.M. Rehg, "Towards Optimal Training of Cascaded Detectors", *ECCV*, 2006
- [27] R.C. Verma, C. Schmid, K. Mikolajczyk, "Face Detection and Tracking in a Video by Propagating Detection Probabilities", *IEEE Trans. On PAMI*, Vol. 25, No. 10, 2003
- [28] J. Yang, D. Zhang, A.F. Frangi, J. Yanf, "Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition", *PAMI*, 26(1): 131-137, 2004
- [29] M. Siegler, B. Jain, R. Stern, "Automatic Segmentation and Clustering of Broadcast News Audio", *Proc. DARPA Speech Recognition Workshop*, Feb. 1997, pp. 97-99.