A Web-based Tool for the Visual Analysis of Media Annotations

Pierrick Bruneau*, Mickaël Stefas*, Hervé Bredin†, Anh-Phuong Ta†, Thomas Tamisier* and Claude Barras†
*CRP - Gabriel Lippmann
41, rue du Brill, L-4422 Belvaux
Email: name@lippmann.lu
†LIMSI-CNRS
BP 133, F-91403 Orsay
Email: name@limsi.fr

***Abstract*—Multimedia annotation algorithms infer localized metadata in multimedia content, *e.g.* speakers' voices or subjects' faces. There is a growing need of experts from this domain to perform advanced analyses, that go beyond medium-scale quality metrics. This paper describes a novel visual tool, that addresses the concerns of multimedia experts using interactive visualization principles. Multiple coordinated views, augmented by interactive inspection facilities, ease both the navigation in media annotations and the visual detection of relevant information. The usefulness of our approach is supported by experimental scenarios using a real multimedia corpus.**

***Keywords*-Visual Analysis; Timeline; Media Annotations;**

I. Introduction

Well established social networks and video hosting platforms now make it easy to capture and share video data. For efficient search and retrieval, these video collections are often annotated, *e.g.* with the covered topic or the people mentioned in the video. Though some of these tasks are partially addressed ad-hoc (*e.g.* Youtube users' tags), the growing pace of these corpora remains to high for sole manual processing, and automated means are needed to deal with them.

An annotation can often be formalized as a time interval in a specific medium, to which a metadata element (*e.g.* name, location) is attached. In this paper, we restrict our attention to speaker (*i.e. who is speaking?*) and face annotations (*i.e. who is seen?*) in video media. The automated inference of people that are seen or speak in audio or video is an active area in computer vision and speech processing research [1].

In general, in the machine learning literature, algorithmic results are evaluated by matching them against a ground truth, in our example manually annotated media. Aggregate metrics can then be computed and compared to the state of the art. However, doing so gives no hint about strengths and weaknesses of algorithms. Advanced inspection tools are thus needed to get this insight.

In the context of multimedia analysis challenges, several manual annotators and algorithm designers may collaborate to study common corpora. Current practices often rely on sheer file exchanges, causing risks of data inconsistencies. A unique platform, with collaborative work support, is there needed.

Within the CHIST-ERA CAMOMILE project (http://www.chistera.eu/projects/camomile), we developed a novel tool for the visual and interactive analysis of media annotations. Collaborative interactions are outside the scope of this paper, but the framework the visual tool is based on potentially maintains a consistent view of the data for multiple simultaneous clients. This paper rather highlights how this tool can help multimedia experts gain insight from their algorithmic results.

After a focused review of the related work in Section II, the annotation data framework we rely on is presented in Section III. The expert analysis tasks, whose practical importance was discussed above, are detailed in Section IV. The central contribution of the paper, our web-based visual analysis tool, is then described in Section V. In Section VI, its practical usefulness is illustrated by detailed scenarios, run using real data. Perspectives on this ongoing work are finally given in Section VII.

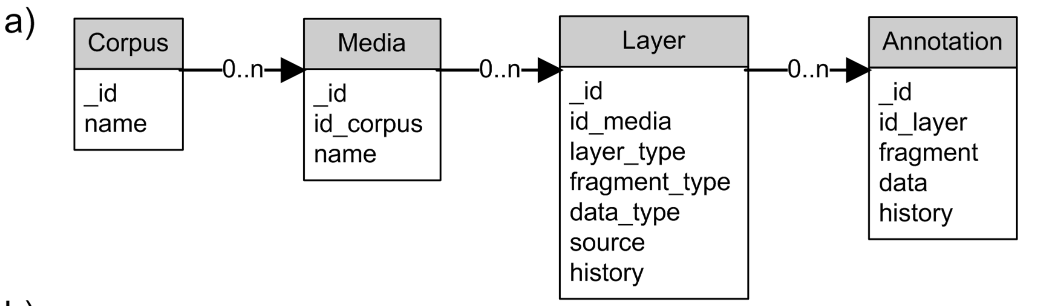
II. Related Work

In this work, we are primarily concerned about visualizing media annotations. Incidentally, most of the work quoted here also supports annotation input, but this aspect lies out of the scope of our proposition.

Advene [2] is dedicated to the annotation of video media. The tool supports the input of a range of annotation types, from free-form to strongly structured, via user-defined specifications. The resulting annotation layers can be visualized in a timeline view, where each layer is mapped to a lane. Annotations within a lane are represented as rectangles, reflecting the underlying time interval. This timeline is synchronized with a video playback, and time scale indicators help contextualize the annotations. Yet, its navigation features are too limited for convenient inspection, and it does not support the color mapping of annotation metadata.

Goldszmidt [3] implemented a Javascript library for audio and video representation. The library synchronizes a timeline view with media playback, and offers the possibility to edit annotation lanes, similarly to Advene. However, it suffers from the same limitations as Advene, with inadequate navigation and color mapping features.

1550-6037/14 $31.00 © 2014 IEEE
DOI 10.1109/IV.2014.28



b) **URL template**: http://[server]/corpus/[corpus_id]/media/[media_id]/layer/[layer_id]/annotation/[annotation_id]

Query all corpora: http://[server]/corpus (returns a JSON array)

Query all media associated to corpus XYZ: http://[server]/corpus/XYZ/media (returns a JSON array)

Query the medium ABC from corpus XYZ : http://[server]/corpus/XYZ/media/ABC (returns a JSON object)

Fig. 1. *a)* Entity-relation diagram of the annotation data. *b)* Template for REST API calls, with some illustrative examples.

Other timeline-based tools, such as *Lignes de Temps* [4] and MediaScope [5] are also relevant to our work. However, these tools are dedicated to a standalone use, and are not adaptable to tasks involving several distant users.

Alternatively, the CAMOMILE project uses a centralized management of the annotation data, with a distant access through a Representational State Transfer API (REST API [6]), introduced in Section III. This approach facilitates the implementation for a variety of platforms and ensures multiple clients have a consistent view of the data. This framework was established by multimedia experts, who also contribute to this paper.

III. Data Framework

The annotation data is structured according to the entity-relation diagram in Figure 1a. In brief, an annotation is characterized by a time interval (*i.e.* fragment) in the associated medium, and a metadata element (*e.g.* speaker name). It belongs to a layer (*e.g.* ground truth for subjects faces, or generated by a given algorithm).

For a specific medium (*i.e.* video sequence), several annotation layers may thus exist. These annotated media are gathered in corpora, that serve as units of reference for challenges in the domain of video analysis [1]. For example, REPERE challenge administrators [7] formed a corpus with manual annotation layers [8]. Participants to the challenge are provided with an annotated media subset from the corpus to serve as training data, and raw media from the same corpus for testing purpose. Algorithmic results form additional annotation layers, that are sent back to the REPERE administrators, who compute the relevant quality metrics and establish the rank among participants. All these exchanges happen manually, with no central data repository.

This observation motivated the development of a server that implements the diagram in Figure 1a. These resources are accessible via a REST API [6]. In short, with this stateless architecture, resources are managed as JSON objects, in response to specific HTTP requests. For the data structure at hand in this paper, the template URL to formulate requests is shown in Figure 1b. By filling in the id's in the URL specification, or leaving them blank, one can locate any data resource (see Figure 1b for examples). All required queries can then be performed using the HTTP implementation of the CRUD (Create, Read, Update, Delete) operations.

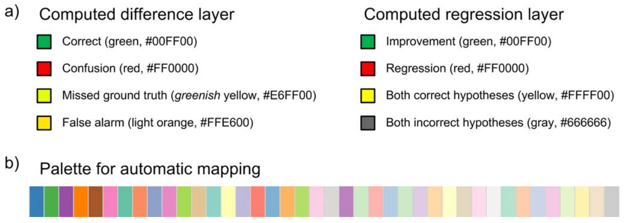

a)
Computed difference layer
Correct (green, #00FF00)
Confusion (red, #FF0000)
Missed ground truth (greenish yellow, #E6FF00)
False alarm (light orange, #FFE600)
Computed regression layer
Improvement (green, #00FF00)
Regression (red, #FF0000)
Both correct hypotheses (yellow, #FFFF00)
Both incorrect hypotheses (gray, #666666)
b)
Palette for automatic mapping

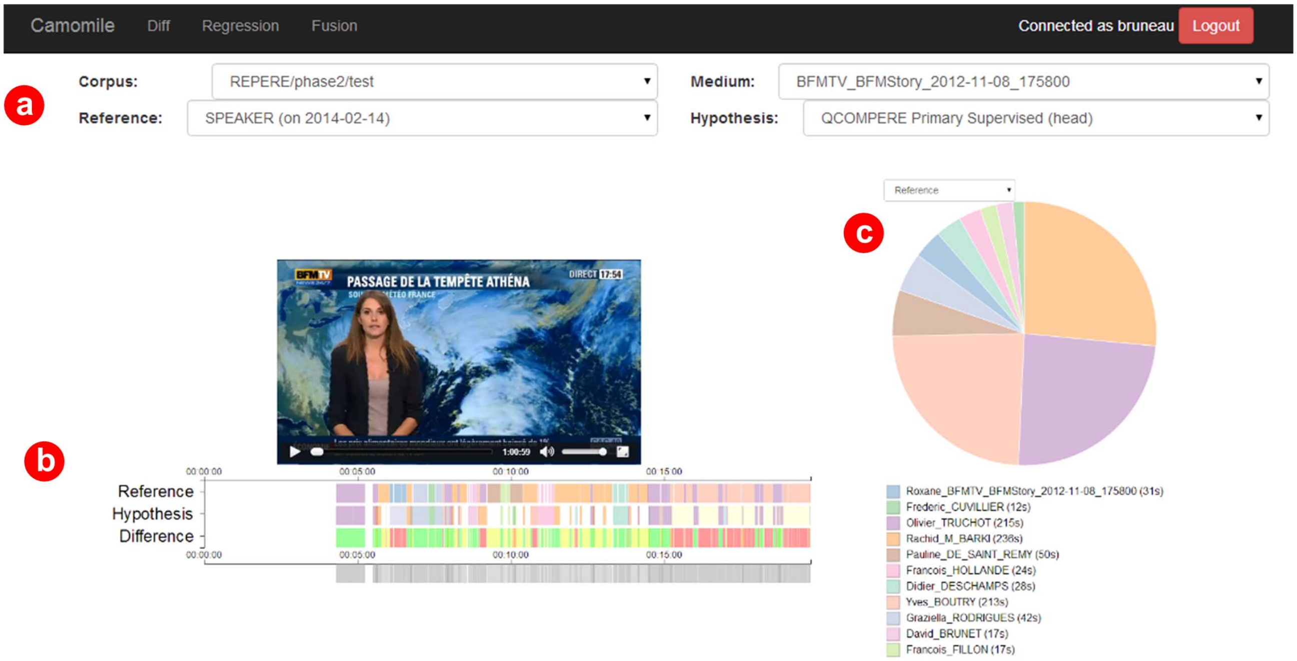
Fig. 2. *a)* Static color mapping used for the computed layers. *b)* Palette used for the automatic mapping of arbitrary metadata.

The REPERE corpus contains several hours of manually annotated media, recorded from 2 French TV channels (BFMTV and LCP) [8]. This data, stored in the above-described framework along with layers resulting from several competing algorithms, is used for illustration in the remainder of the paper.

IV. Task Description

As discussed in Section I, aggregate quality metrics are generally the only relevant indicators in common comparative analyses. However, experts might want to perform a finer-grained analysis, *e.g.* algorithmic errors might be correlated to a given speaker, or to a certain type of background clutter in the video signal. Therefore the two following tasks are considered in this paper:

- *Difference*: The expert wants to inspect how a *hypothesis* layer (*i.e.* computed by an algorithm) differs from a



Corpus: REPERE/phase2/test

Medium: BFMTV_BFMStory_2012-11-08_175800

Reference: SPEAKER (on 2014-02-14)

Hypothesis: QCOMPERE Primary Supervised (head)

Fig. 3. Overview of the annotation analysis tool. *a)* The user selects the layers to be analyzed via combo boxes. *b)* The layers are loaded dynamically in the interactive timeline view, with associated video playback. *c)* The user can select a layer to summarize with a pie chart view. A legend also reports the metadata mapping.

reference layer (*i.e.* ground truth as input manually by annotators).

- *Regression*: The expert wants to compare two algorithms, and understand which patterns are associated to improvement or regression in the respective hypothesis layers.

To effectively support these tasks, we use *differential* layers, in which the annotation metadata values are set with error or regression markers referring to current reference and hypothesis layers. Their visual mapping is facilitated by the use of the data format exposed in Section III. The value set possibly taken by this metadata is known beforehand and reported in Figure 2a. We use a green-red scale with yellow and gray as neutral hues to reflect the polarity of the results. The differential layers are computed via a service call, also accessible through a REST API (see Section III), PyAnnote-REST [9].

V. Visual Tools for Algorithmic Results Analysis

An overview of our visual tool is shown in Figure 3. Its views, described in Sections V-A and V-B, are implemented with SVG drawing primitives, that are linked to the data using the D3.js Javascript library [10]. The angular.js framework [11] is also used to easily integrate multiple coordinated SVG views, user controls, and asynchronous data access to the above-described framework via the REST API. In support of the tasks described in Section IV, we also derive an adapted color mapping.

A. Timeline View

As an effect of user selections from a range of available layers or resulting from a differential layer computation (see Section IV), a given collection of layers has to be displayed to the user. Taking inspiration from Advene [2], we use a timeline to represent these annotations. As in Advene, each layer is mapped to a lane in the timeline, and annotations are represented by rectangles in their respective lane (see Figure 3b). A dynamic time scale indicates the extent of the annotations. The associated metadata value set is mapped to a set of categorical colors. Overlapping annotations within a lane may occur, so the displayed colors are alpha composited.

The metadata value set for the differential layers is static, and its color mapping has been determined in conjunction with the multimedia experts (see Figure 2a).

All other metadata (in this case people's names) are not known beforehand, so their color mapping has to be determined automatically. Standard categorical palettes are available from D3.js [10], or Color Brewer [12]. However, we have two specific constraints:

- The number of distinct metadata values to be simultaneously mapped can potentially exceed 20, far beyond the cardinality of most categorical color scales.

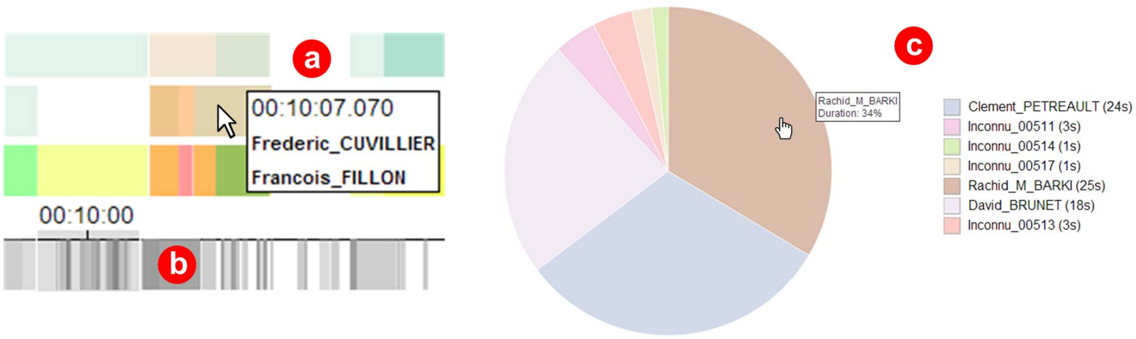


Fig. 4. Zoom on the timeline and pie chart views. *a)* Metadata are revealed in a tooltip when hovering annotations. When multiple annotations overlap, the tooltip summarizes the related metadata and the glyph colors are alpha-blended. *b)* The context lane supports an editable brush that interactively updates the focus lanes. *c)* The aggregate duration of a metadata value is reported as a percentage in the pie chart tooltip, and on an absolute scale in the legend.

- The colors at the vicinity of the static mappings shown in Figure 2a have to be excluded to avoid collisions.

Accounting for these constraints, we propose a combination of Color Brewer categorical palettes. The palette names we use hereafter refer to the R implementation of Color Brewer [13]. To overcome the limited cardinality of Color Brewer categorical palettes (at most 12), we aggregate *Set1*, *Set2*, *Set3*, *Pastel1* and *Pastel2*, as they cover varying luminance and chrominance levels. In this aggregated palette, we then remove the colors that lie too close from the static mapping (see Figure 2a). This vicinity was evaluated by converting colors to the perceptually uniform Lab color space, and using this representation to compute L2 distances between colors in the aggregated palette and the static mapping. We removed the 6 colors closest to any of the statically mapped ones. We thus obtain a 40-color palette, available to map arbitrary metadata automatically, shown in Figure 2b.

To ease the navigation in the timeline view, we took inspiration from the D3.js example implementation of the focus+context principle [14, Chapter 10]. The bottom lane of the timeline supports brushing, and maintains the context by displaying the sum of all the loaded layers in alpha-blended gray shades (see Figure 4b). The focus of all layers and the associated time scale adapt dynamically to brush interactions.

In a given focus, a user can hover annotations, revealing the underlying metadata in a tooltip. Some additional context is there given by the precise timestamp associated to the mouse pointer, reported on top of the tooltip. The tool supports overlapping annotations, as may occur in the data (see Figure 4a). By clicking an annotation glyph, the user can then view the associated sequence in the video playback.

B. Summary View

As a complement to the timeline view described above, a pie chart view summarizes one of the currently loaded layers (see Figure 3c). The color palette of the timeline (see Section V-A) is used consistently in this view. The users may switch the layer they want to inspect using a standard combo box. A section is defined in the pie chart view for each value taken in the metadata of the respective layer. The size of the sections in the pie chart view are computed according to the cumulated durations of the annotations matching the respective value.

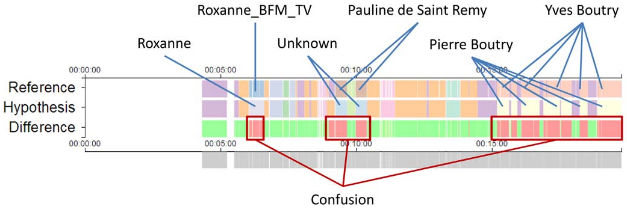


Fig. 5. Differential layer for the *Difference* task, displayed with associated reference and hypothesis. Confusions are highlighted, and guide the visual inspection.

By hovering over a section, the duration can be read as a percentage in a tooltip. The same duration is reported in an absolute scale in the legend accompanying the pie chart (see Figure 4). Using this view, a user can have a glance at the distribution of speech time in a layer, or get the aggregate quality metrics of the current difference or regression task. This summary view is also synchronized to the focus+context mechanism described in Section V-A: modifying the brush interactively updates both the focus in the timeline and the scope of data summarized in the pie chart view. With these coordinated views, the users are then able to refine their initial glance by inspecting patterns specific to an arbitrary subpart of the medium.

VI. Experimental Scenarios

Sections V-A and V-B presented two component views rather independently from their context of use, with a focus on their functionalities. This section shows how they are integrated to support the expert tasks.

As already evoked in Section IV, the first common task for experts is to inspect the *difference* between a layer inferred by an algorithm and its respective ground truth.

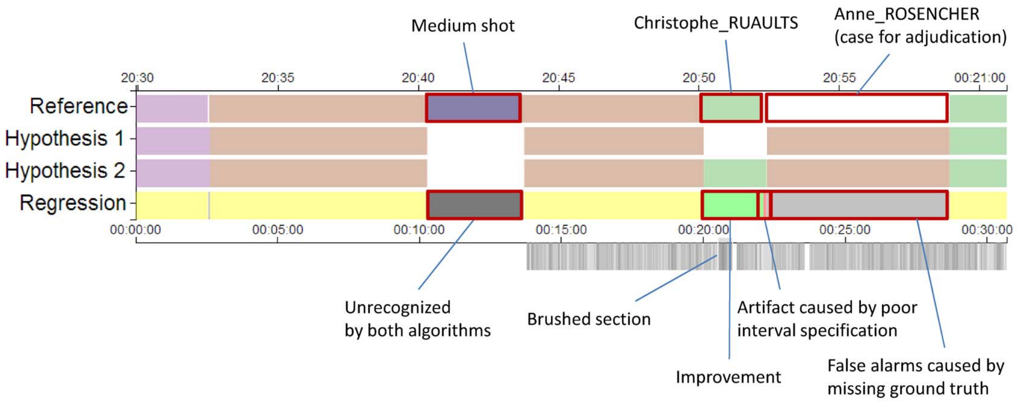


Fig. 6. Differential layer for the *Regression* task, displayed with associated reference and hypotheses. Expert observations are highlighted.

For example, let us assume that the experts want to inspect the results of their novel supervised QCOMPERE algorithm on the *BFM Story* show aired on November the 8th, 2012, at 17:58. Using the combo boxes displayed by the interface (see Figure 3a), they pick the appropriate corpus and medium. They then define the REPERE speaker ground truth layer as the reference, and the results of the supervised QCOMPERE for speaker recognition as the hypothesis. Algorithmic details about the QCOMPERE approaches are out of the scope of this paper and may be found in [1].

The visual mapping used for the differential layer displayed by the timeline view (see Figure 5) instantly reveals areas of interest for the experts. Confusion errors, highlighted in red, are visual cues for local inspection using the brush interaction (see Section V-A). Actually, they want to see which people are not correctly recognized by their algorithm. This is easily done by hovering over the reference and hypothesis layers above the error patterns.

The experts notably find that the ground truth *Roxanne_BFM_TV* is identified as *Roxanne* by the algorithm. Though this recognition looks correct, it counts as an error in quality metrics computation. The experts record this case for the next *adjudication* step, where mistakes in manual annotations can be notified to REPERE administrators.

Pauline de Saint Remy fails to be recognized by the algorithm. To determine possible causes to this particular error, they click on the associated annotation glyph, triggering the video playback. The speaker is revealed to be a background speaker. She appears only once in the timeline, though her intervention is rather long (50s, as seen using the summary view). From these observations, the experts hypothesize a possible cause to this error: as supervised approaches use a training set to learn speaker models, maybe she does not speak there or her interventions are too short for a robust speaker model to be estimated.

Another notable confusion is *Yves Boutry* being detected as *Pierre Boutry*. Actual occurrences of *Pierre Boutry* should be looked for in the rest of the corpus: however, this observation questions the experts on the algorithmic sensitivity to speakers sharing the same surname. Finally, by displaying the summary pie chart of the differential layer, they also find that misses and false alarms only account for 2% of the total speech time in the medium, which indicates their algorithm for speech activity detection is performing fairly good.

Now say the multimedia experts want to compare several algorithms, *e.g.* identify the *regression* patterns that may occur with their newest algorithm w.r.t. a gold standard. The experts have two versions of their QCOMPERE face detection algorithm: a supervised one, that uses the training set for learning face models; and an unsupervised one, that instead jointly extracts and uses other modalities in the medium (such as overlaid text recognition and named entity detection) to recognize faces.

They want to inspect potential regression patterns of their unsupervised algorithm, w.r.t. the supervised version, on the *LCP Entre les lignes* show, aired on March the 16th, 2013, at 21:24. They first select the relevant reference layer for their task, the ground truth faces for the show. The layer originating from the supervised (resp. unsupervised) algorithm is selected as the first (resp. second) hypothesis. The differential *regression* layer, that shows the potential improvements or regression of the second hypothesis w.r.t. the first one, is computed quickly once all hypotheses are loaded.

The experts want to start their analysis by an overview and thus opens the summary view (see Figure 4c). They map the differential layer to the view and notice that the results obtained by the unsupervised algorithm are slightly better: the improvements (22s) outweigh the regression patterns (14s). Using the same view, they also notice that 40% of the ground truth annotated time in the medium is incorrectly detected by both algorithms. They then look for an explanation to this phenomenon by inspecting some examples. As face annotation patterns are difficult to figure out from the zoomed-out timeline, they brush approximately from 00:20:30 to 00:21:00 for an easier inspection (see Figure 6). They identify a ground truth annotation undetected by both algorithms and triggers the associated video playback. It turns out that this is a medium shot of the show guests and host, where most guest are seen from the side. This configuration is typically problematic for face detection algorithms.

In the same brush, the experts remark that the previously undetected *Christophe Ruaults* is recorded by their unsupervised algorithm (see Figure 6). They also notice a case for further adjudication: the ending timestamp for the latter annotation looks incorrect in the ground truth (causing an artifact in the differential layer), and *Anne Rosencher*, annotated just afterwards by both algorithms, and indeed showing up at this point in the video, is missing from the ground truth.

VII. Conclusion

This paper showed the application of interactive visualization principles as a support for automatic media annotation algorithms. Expert concerns were carefully considered in the design of the views and interactions. Detailed usage scenarios using actual data illustrated the interest of the tool.

The tighter integration of the video playback with the timeline, via the synchronization with a current time marker in the view and an adapted brush interaction, would facilitate the analysis even further.

This ongoing work focuses on a low-granularity use case. However, as shortly envisioned in Section VI, the ability to *zoom out* at the corpus level would help in certain cases, *e.g.* to check how frequently a given speaker appears in the whole corpus. Seamless navigation between these two points of view is currently under investigation.

The support of annotation edition in a collaborative context (*e.g.* multiple users operating simultaneously on a layer) is also an important direction of research. Such support would improve the quality of manual annotations and facilitate the adjudication phases.

VIII. Acknowledgments

This work was done in the context of the CHIST-ERA CAMOMILE project funded by the ANR (Agence Nationale de la Recherche, France) and the FNR (Fonds National de la Recherche, Luxembourg).

References

[1] H. Bredin, J. Poignant, M. Tapaswi, G. Fortier, V. B. Le, T. Napoléon, G. Hua, C. Barras, S. Rosset, L. Besacier, J. Verbeek, G. Quenot, F. Jurie, and E. H. Kemal, "Fusion of speech, faces and text for person identification in tv broadcast," *LNCS 7585 (ECCV 2012)*, pp. 385–394, 2012.

[2] O. Aubert and Y. Prié, "Advene: active reading through hypervideo," *ACM conference on Hypertext and hypermedia*, pp. 235–244, 2005.

[3] S. Goldszmidt, "Javascript library for audio/video timeline representation," *WWW Developer Track*, 2012.

[4] V. Puig, "Lignes de temps, un logiciel pour l'annotation de films," *Culture et Recherche*, vol. 112, pp. 25–26, 2012.

[5] "Mediascope," http://www.inatheque.fr/consultation/mediascope.html, 2014.

[6] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, University of California, Irvine, 2000.

[7] J. Kahn, O. Galibert, L. Quintard, M. Carre, A. Giraudel, and P. Joly, "A Presentation of the REPERE Challenge," in *International Workshop on Content-Based Multimedia Indexing*, 2012, pp. 1–6.

[8] A. Giraudel, M. Carré, V. Mapelli, J. Kahn, O. Galibert, and L. Quintard, "The REPERE Corpus: a Multimodal Corpus for Person Recognition," in *LREC*, 2012.

[9] H. Bredin, https://pypi.python.org/pypi/PyAnnote-REST, 2014.

[10] M. Bostock, "Data-Driven Documents," http://d3js.org/, 2014.

[11] "Angular.js," http://angularjs.org/, 2014.

[12] M. Harrower and C. A. Brewer, "ColorBrewer.org: An online tool for selecting colour schemes for maps," *The Cartographic Journal*, vol. 40, no. 1, pp. 27–37, 2003.

[13] E. Neuwirth, "RColorBrewer: ColorBrewer palettes," http://cran.r-project.org/web/packages/RColorBrewer/index.html, 2011.

[14] C. Ware, *Information Visualization: Perception for Design*. Elsevier, 2004.