UNIVERSITÀ DEGLI STUDI DI FIRENZE
Facoltà di Ingegneria - Dipartimento di Sistemi e Informatica

Tesi di Laurea Specialistica in Ingegneria Informatica

Generazione Automatica di Video Musicali
Automatic Music Video Generation

Candidato
Alessio Bazzica

Relatori
Prof. Alberto Del Bimbo

Prof. Marco Bertini

Correlatori
Ing. Giuseppe Serra

Ing. Lamberto Ballan

Prof. Alan Hanjalic *

Cynthia C. S. Liem, MSc MMus *

* TUDelft Delft University of Technology

Anno Accademico 2011-2012

alla mia famiglia
a Iris

“You will learn to lose everything;
we are temporary arrangements.”
Alanis Morissette

Acknowledgments

A substantial part of the thesis work has been carried on in the D-MIR lab, TU Delft (Delft, The Netherlands) under the supervision of Prof. Alan Hanjalic and Cynthia C. S. Liem.

The system resulting from this thesis project was a Grand Challenge Finalist entry at the 20th International Conference of Multimedia.

MuseSync: Standing on the Shoulders of Hollywood
Cynthia C. S. Liem, Alessio Bazzica, Alan Hanjalic
ACM Multimedia 2012 (Nara, Japan)

Contents

List of Figures

List of Algorithms

List of Definitions

Chapter 1

Introduction

The number of uploaded user-generated videos has been increasing since the spread of mobile devices with video recording capabilities and high bandwidth Internet access. Last figures published on `http://www.youtube.com/t/press_statistics/` state that 72 hours of video are uploaded to YouTube every minute. This figure includes 3 hours of video which are fed from mobile devices: this means that every day, more than 4000 hours of mobile videos are uploaded. However, informal videos, usually recorded by mobile devices, are often not particularly *appealing*[1] to a broad audience. Therefore, they are unlikely to become popular. A possible solution to overcome the problem of this lack of appeal could lie in the use of music as a means to enrich visual content. Like in Hollywood movies, the expressive power of the music can amplify visual content and can influence the viewer in many different ways.

In this project music is used for automatic music video generation. It is an audio component which amplifies the message that the user would wish to convey. Music as the audio component can even add a message. This is achieved through a selection of the soundtracks which are related to the theme of the user's envisioned final video object (i.e. the realization which the user has in mind). Music normally has to be aligned with certain sections in the video. This process is normally described as synchronization of music with video. The list of appropriate synchronizations which enhance the impact of the co-occurrences of visual and musical cues is finally used

[1] "Having qualities that people like, being pleasing or attractive" (from Merriam-Webster®).

to provide the user with a small set of synchronized videos.

On one hand, one can believe that the narrative intent of the user is entirely reflected in the audiovisual signal information. In this case, no extra information is needed in order to select suitable music and synchronize the audio and video streams. Thus, a music video generation system can rely on aesthetic features, semantic analysis (e.g. concept detectors) and semiotic analysis.

User-generated videos are usually not the same as professional videos in terms of aesthetic linking between different modalities: this happens due to the limited quality of the recording devices and the fact that users[2] do not act as movie directors. For example, no plot is written in advance and the scene is recorded as it is. The user has little knowledge of the rules regarding audio-visual associations and little knowledge of social conventions. Moreover, every single modality is deficient in strong aesthetic cues for the same reasons. For example, there may be some background noise in the audio. The lighting conditions may be inadequate. Thus, even if visual aesthetic cues are extracted, amplified and employed to find some appropriate music and to synchronize it with the video, the final music video might not show a sufficient degree of connections between the two modalities.

Detecting semantic concepts in the video, such as actions or the presence of objects, is a way of getting a greater insight than the cues obtained through a pure aesthetic analysis. But without any prior information, each detected concept is often as relevant as any other and this may lead to the establishment of spurious links between visual and musical concepts. The following example should illustrate this point even if it involves only the analysis of the video. Consider a video where the camera panned over an area with a river and a house. If the intent of the user is to show his childhood home, extra information is required to grasp that intent. For example, one can sometimes find hints in the video's title or while listening to the original video's speech: both might be reliable sources to infer that the presence of the river is less important than the presence of the house.

Finally, imagine another video recording of a child looking at his birthday cake dotted with candles. The viewer needs no additional information about the user in order to know that the user has recorded a scene of a birthday

[2] The *user* in this chapter is who uploads a user-generated video.

party. The images of a “child” and a “cake” are sufficient for the viewer to understand such context. This happens because the “child” and the “cake” are *semiotic signs*. Semiotics is the science of signs as carriers of sense. In semiotics, “sign” is anything which conveys a sense: words, pictures, sounds, gestures, clothes, etc. Semiotics suggests that signs are related to their meaning by social conventions, i.e. by a specific cultural context. Therefore, it might also happen that a user video contains undisputed semiotic signs, as in the case of the video recording of a birthday party. One can take for granted that a user has always the intent of amplifying a message. Then, in the example of the birthday, a party soundtrack would be a suitable choice. But what if the user intent is to put the focus on the child perusing the cake so that the child looks like a scary subject? In this case, semiotic analysis alone is not sufficient to take into account such user intent. Semiotic analysis could be more suitable when specific audiovisual signs have been thoroughly chosen in advance, as in the case of advertisements and movies. Such case is indeed different from user-generated videos which can be defined as a series of unique events not previously written up.
To summarize, of the three possible analysis strategies, semantic analysis seems to be the most reliable analysis in the user-generated video context, but it can lead to inappropriate choices of music due to the absence of prior information, which causes a visual scene to be ambiguous in terms of interpretation.

Other strategies to make strong connections between music and video should be considered. The common assumption is that the raw video signal is the only source of information for that task; but the person who records the video, what his intent is in telling the story, might reveal curious details about the content of the video. Insights regarding the story can lie in the relationship between the user and the elements within the recorded video (e.g. people, objects, actions). Let us recall the previous examples of videos. Sometimes the context can be obvious, as in the case of the video recording of a child looking at the cake. The viewer immediately understands that the scene that has been recorded is that of a birthday party. Other times, the subject of the video can also be personally related to the user. For instance, the subject of the video can be his childhood home. A third possibility is that a clear message or context emerges from the visual content, e.g. a birthday party, but the user ultimately want to tell a completely different story.

In the two latter examples, asking to upload the video and then to provide extra information can be useful so that the viewers could easily understand the user intent. For instance, linking a video with a music song which recalls the childhood theme could be useful in the first video example: the viewers can be easily understand that the house in the video is related to the user's childhood. Furthermore, receiving some extra information, can help in situations in which the transcript of a speech and/or the video's title are not reliable sources.

What has been said above suggests that an automatic music video generation system should exploit extra information in order to outperform a system based on pure semantic analysis. In respect of the previous examples, two challenging problems can be observed. The first relates to the matching of the video theme to the music theme. This process can be called *thematic matching*. When thematic matching is involved, one focuses on the need to introduce context to the viewers and eventually, by selecting suitable music, set their mood. The second problem relates to aligning cues in the music to visual cues. When the cues from different modalities are well timed, the overall effect of the music video is enhanced. This process can be called *cross-modal synchronization*. It is worth noting that for the former problem, hints given by the user regarding his intent may play a relevant part, thanks to their potential power of disambiguation. As far as the second problem is concerned, a pure semantic analysis alone could extract interesting cues and work at a fine temporal resolution as required by the synchronization.

The most important thing about what has been said above is making good thematic connections: these connections enable the viewer to understand the video better. Without strong connections, only the viewers who have a close relationship with the user might be able to grasp the deeper meaning of the video (relying on prior information regarding him). This is supported by videos watched only by viewers within the user's circles. And vice versa: a large number of popular videos appealed to viewers, just because users added manually an appropriate song. These users could have picked up on the appropriate soundtrack right at the start of making the music video. Alternatively, they could have added the felicitous soundtrack after the music video had been made because they found the video without music boring.

On the basis of the ideas outlined above, an automatic music video generation system has been designed and developed. It has been defined as a framework within which a number of soundtracks are thematically preselected according to a verbal description of the video provided by the user. They are then ranked. For the ranking, scores express the degree of synchronization between the temporal development of motion in the video and novelty in the music track. By providing the top-k ranked items, user can have a degree of the freedom of choice in selecting one of the highly-scored well-synchronized music videos.

This thesis is structured as follows. Chapter 2 discusses related works and presents a series of works on which the proposed approach is grounded; Chapter 3 illustrates the general system design motivating the choices; Chapter 4 explains how the thematic music pre-selection works; Chapter 5 thoroughly presents the audiovisual synchronization framework and the techniques employed to analyze musical and visual contents; Chapter 6 presents designs for evaluation procedures, both regarding the full system and individual components. Finally, a conclusions chapter summarizes the contributes of this thesis project.

Chapter 2

Audiovisual Cross-Modal Interactions

In this chapter a series of resources are reported in order to give a background information about the typical approaches to the audiovisual cross-modal linking. In the first section, some previous works in which music is aligned to videos are presented. Then, a musicological and psychological perspective is given in order to better understand the role of music in music videos.

2.1 Previous Works

The closest work in terms of use case is [6] where home-made videos are analyzed and an appropriate background music (or BGM) is automatically selected from a library. The system uses the original audio track to keep dialogs and narration in the edited video, and employs two probability models trained with a set of professional movie clips. One is used to model speech, background music and video co-occurrence, while the other models transitions between music and speech. The visual content is represented adopting a set of 500 visual words encoding the frequency of colors in the Hue-Saturation-Value (HSV) space, and the optical flow magnitudes. A set of 23 low-level audio features including Mel-Frequency Cepstral Coefficients (MFCCs), spectral centroid and tempo is quantized in 32 audio words.

The evaluation consisted of a user study involving 16 subjects who had to judge the relative quality of the edited video compared to a reference. Two types of pair have been adopted: in one case the reference was the original video without BGM, in the other one the reference was an edited video whose BGM has been found via a baseline method. The results show that the output provided by their system is often the most appreciated in both cases. However, the authors state that they would introduce high-level semantics and scene analysis in future works encouraging to investigate further.

In [17, 8] interesting video segments are automatically selected and aligned via a rule-based approach with incidental music[1] and the speech extracted from the original audio track. Given a music track and a video, the former is analyzed finding the strongest onsets and estimating the tempo which is supposed variable along the time. The video is parsed and then organized into a hierarchical structure consisting of scenes, shots and sub-shots. In order to do that, the authors adopt shot detection techniques and define visual, audio and linguistic attention indexes with sub-shot temporal granularity. The best alignment is finally selected heuristically solving a nonlinear 0-1 integer-programming problem which aims to align shot transitions with music beats, try to match the motion intensity with the music tempo and mix speech with music without breaking the sentences.
The authors adopt both an objective and a subjective evaluation, although the former is not based on a ground truth and seems closer to a validation procedure. They indeed prove that the system's behavior is the expected one looking at the objective function scores.
The authors conclude focusing on two interesting open issues. They first mention the need of better understanding the extent to which motion intensity, matched with musical tempo, affects the perception of the video content. Even more interesting is the second issue which puts the focus on the need of having a more semantic meaning. They indeed call this issue "better semantic storytelling" and express the intention of adopting face detection, annotation and tracking in order to highlight the role of a central character.

In [18] the following approach is presented in order to automatically add

[1] "Incidental music is music in a play, television program, radio program, video game, film or some other form not primarily musical. The term is less frequently applied to film music, with such music being referred to instead as the film score or soundtrack" (Wikipedia).

music to a series of pictures. Given a shot from a movie, the authors mine associations between frames within the shot and the played soundtrack analyzing a set of low-level features defined in the MPEG-7 standard. Providing a series of pictures as input to the system, the authors first cluster them in groups in order to suggest one music track for each group. For each picture a rank of similar frames extracted from movies is defined, and the top ranked frames are mapped to the played songs features through the mined associations. The extracted audio features are then used to retrieve the most suitable song within a library of free music. The authors also present two different approaches to select one music track for each group of pictures.
The evaluation involved 13 subjects grading the suitability of image and music content in the produced video through a scale ranging from 1 (“total miss”) to 5 (“fits very good”). Both the average score and the standard deviation suggest that the subjects prefer the system output over a random music track selection.
In [18] the main contribution is again given by the idea of exploiting the expertise of professional users (in this case movie directors), but the system only relies on low-level features.

Another related use case is the automatic Music Television Video (MTV) generation. In [11], given a raw video and a music song, a new MTV is generated by segmenting the video and the song in clips of fixed length and inferring the most suitable video clip sequence for the given song. A probabilistic model, known as “dual-wing harmonium”, is trained with a large dataset of professional MTV: it combines video and audio features to produce a latent representation for each pair of video and music clips. Clustering these points in the latent vector space defines groups of similar MTV clips. The clusters are then used at the prediction stage in order to select the best video clip for each music song clip.
The visual features span color histograms and structure tensor histograms encoding the motion (intensity and direction), while the audio feature vector consists of some temporal and spectral features such as MFCCs and zero-crossing rate.
In order to assess their system, the authors made a comparison with a commercial software (MUVEE) both adopting an objective and a subjective evaluation. The former consists of a measure of similarity between a

professional MTV and its resynchronized version. Two different similarity measures show that the MTVs generated with their system are more similar to the original ones than those generated with MUVEE. The latter involved five subjects asked to give two scores from 0 to 1 reflecting the extent to what the video change matches with music beats and how excited users feel about the video scenes. The results show that MUVEE is slightly better for the second score and the other way around holds for the first one.

The authors state that the discovered patterns might be not straightforward to understand, especially when a dominant feature is absent, and that they would consider lyrics analysis and highlight detection for future works.

Remarks

In all the works presented above expect one, namely [17, 8], it is implicitly assumed that appealing effects can be obtained through the right choice of association patterns between aesthetic features encoded in the auditory and visual domain, such as spectral power and color histograms. In that direction, in [15] it has been shown that a relationship between auditory and visual domain exists even though neither the audio nor the images possess semantic content.

In order to extract a more semantic description of the visual content, some authors make use of motion analysis techniques.

In both cases, association patterns can be learned exploiting movies or professional videos, where it is assumed that video and audio track match well. This approach avoids the encoding of pre-defined rules borrowed from the field of cinematic production.

Independently of the approach, an important aspect is represented by the system evaluation: it might be hard to formulate a problem of automatic music selection in terms of document retrieval; thus classic scores, such as precision and recall, cannot be applied directly. This is supported by the fact that all the works mentioned in this section employed a subjective evaluation.

2.2 Musicological and Psychological Perspective

Music is employed for storytelling by movie directors and sound designers combining sight and sound in order to enrapture the audience. A number of functions have been listed and discussed in [13] and more recently reordered in [19][2]. They state that music can:

- emphasize movement or a real sound;
- bring to mind a specific location;
- comment the images (even contradicting them like in the case of a mellifluous melody for atomic holocaust);
- be the music present in the scene (e.g. a radio in a car);
- express actor's emotions;
- communicate emotions to the audience;
- represent something known by the audience but not currently part of the narrative;
- anticipate actions or enhance film's structure elements like openings and scene changes.

As it will be shown soon, the way the music is an effective mean to achieve such functions relies on socially established meaning in music and video and cross-modal interactions between them. Thus, music and video are not just time sequences of sound and images, but also semiotic signs for concepts that are not directly related to the audio or video content in these time sequences. This fact can cause a certain expectation regarding congruent narrative structures and corresponding video content: such a knowledge should be helpful to devise a system that automatically generates music videos.

Music Affects the Interpretation of Video

In [5], it is argued that associations between musical and visual elements do not simply add together to produce a composite meaning. This principle does not indeed fully explains the interpretation of incongruent contents across different modalities (e.g. happy music with sad video). Then, the author introduces a *multi-level congruence-associationist framework* (see

[2] A web document, where the original contents from [13] have been adapted and resumed, is available in English at `http://www.tagg.org/teaching/mmi/filmfunx.html`.

Figure 2.1) which shows how music influences the interpretation of film and video.

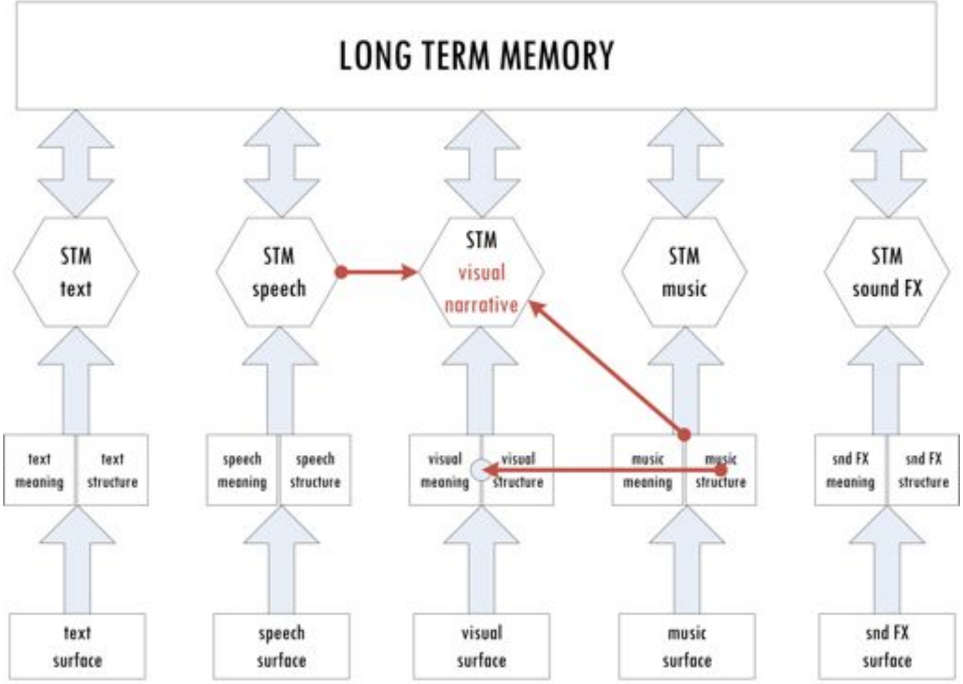


Figure 2.1: Congruence-Associationist Framework

The framework shows a number of elements which are:

- modalities: such as visual content, music and speech;
- surface information: physical information received by the sense organs;
- structure: formal characteristics, style and grammar, holding across any time period or culture (e.g. motion in visual images);
- meaning: associations brought to mind, feelings, interpretations;
- STM: short term memory, or *working* memory;
- LTM: long term memory, repository of knowledge gained through lifelong experiences.

In addition to the vertical connections, other interactions across different modalities may take place. In particular, in the case of structural congruency of audio and visual materials, i.e. congruent accent patterns across different modalities, the attention will be directed towards the part of the visual scene that is structurally congruent with the music or the speech. While the audience is actively engaged in constructing the working narrative, even if the vision generally predominates over audition, music plays a role directing attention to certain features in the visual image domain, feeding information directly in the working narrative and providing associations that establish inferences in LTM.

In this psychology work, the author states that synchronized music and visual streams may enhance the perceived match and stimulate unification of the meaning perceived for the individual modalities. This conclusion is also shared by the researchers in the multimedia field (e.g. [17], [16]).

Having in mind the definition of LTM, that is the source of inferences and contexts that an individual actively generates in order to make sense of the external world known initially through the surface information, its presence in the framework proves that visual and musical contents cannot be treated as the only source of information during the interpretation of a multimedia object. Thus, any approach in which external information is gathered from the user to describe his mental representation of a music video is justified.

The last remarkable aspect of the aforementioned framework is that it does not suggest that happy visual content has to be associated with happy music: this means that any kind of audiovisual link can potentially work.

Chapter 3

Proposed Approach

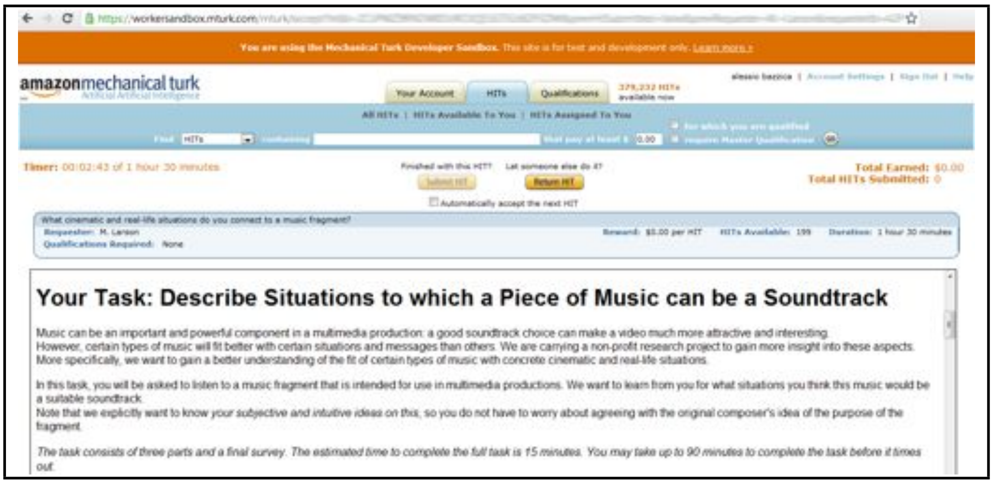
In this chapter the system outline is reported focusing on the motivations behind the design choices. In the first section, a user survey study conducted as part of this project is presented. Then, recalling what emerged from the analysis stage of the previous chapter, both the music pre-selection algorithm and audiovisual synchronization method are presented. In the last section, the novelties of the proposed approach are remarked.

3.1 Cross-Modal Connotative Associations

Consider the scenario of a layman user looking for music to be used as reinforcement or meaning-creating element in a video. It can happen that he has a result in mind but cannot express the corresponding information with the right musical vocabulary. However, if the connotative associations between music and visual narrative are strong enough, music can be characterized in terms of the envisioned multimedia context (i.e. the final realization consisting of a video and a music song). In [12], through a user survey study conducted as a crowdsourcing experiment on the Amazon MTurk platform (see Figure 3.1), it has been shown that strong connotative associations between music and visual narrative indeed exist.
In every survey, a respondent got assigned a random fragment of production music, for which associative descriptions were sought, and had to fill-in the following four parts:

- general questions on characteristics of the music fragments;

- an imaginary cinematic scene description to which the music fragment would be a suitable soundtrack;
- a personal episode from the respondent's life to which the music fragment would be a suitable soundtrack;
- demographic information.


https://workersandbox.mturk.com/mturk/
You are using the Mechanical Turk Developer Sandbox. This site is for test and development only. Learn more »
amazonmechanical turk
Artificial Artificial Intelligence
Your Account
HITs
Qualifications
379,232 HITs available now
alessio bazzica | Account Settings | Sign Out | Help
All HITs | HITs Available To You | HITs Assigned To You
Find HITs containing that pay at least $ 0.00
for which you are qualified
require Master Qualification
Timer: 00:02:43 of 1 hour 30 minutes
Finished with this HIT?
Let someone else do it?
Submit HIT
Return HIT
□ Automatically accept the next HIT
Total Earned: $0.00
Total HITs Submitted: 0

What cinematic and real-life situations do you connect to a music fragment?

Requester: M. Larson

Reward: $5.00 per HIT

HITs Available: 199

Duration: 1 hour 30 minutes

Qualifications Required: None

Your Task: Describe Situations to which a Piece of Music can be a Soundtrack

Music can be an important and powerful component in a multimedia production: a good soundtrack choice can make a video much more attractive and interesting.
However, certain types of music will fit better with certain situations and messages than others. We are carrying a non-profit research project to gain more insight into these aspects.
More specifically, we want to gain a better understanding of the fit of certain types of music with concrete cinematic and real-life situations.

In this task, you will be asked to listen to a music fragment that is intended for use in multimedia productions. We want to learn from you for what situations you think this music would be a suitable soundtrack.
Note that we explicitly want to know *your subjective and intuitive ideas on this*, so you do not have to worry about agreeing with the original composer's idea of the purpose of the fragment.

The task consists of three parts and a final survey. The estimated time to complete the full task is 15 minutes. You may take up to 90 minutes to complete the task before it times out.

Figure 3.1: User Survey Study via Amazon Mechanical Turk

In order to verify whether music fragments can be realistically recognized and retrieved based on the description provided by the respondents, an inverse task was designed based on responses from the original description survey. The respondents were provided with a cinematic scene description and asked to rank and rate three music fragments according to their fit to the given description. The given descriptions have been obtained from the initial survey run making minor clean-up changes. Looking at the outcomes, it is clear that the stimulus music fragments of the first survey are considered as better fits to the description than the random fragments: this proves the existence of strong connotative connections between free-form and spontaneous description of visual narrative and musical information. Hence, a music retrieval system does not need music queries to be confined to musical vocabulary but they can also be constructed in a user-friendly narrative form.

In the user survey study, insights are also given regarding the way respondents described their stories. In most of the cases, there was a majority preference for an event structure class in which events are taken as basic building blocks of a narrative[1]. More specifically, four classes of events have been differentiated: states having no internal structure, activities involving internal change but no endpoint, achievements involving an instantaneous culmination or endpoint and accomplishments involving a build up period and then a culmination. These class of events have been employed to describe episodic changes in correspondence of variations in the musical texture. Such variations have been reported in a mid-level form which is not as advanced as musical themes, but not as basic as low-level features like beat or tempo (e.g. "the parts where there are constant harmonies in the background need a more quiet scene than the parts where there is only the beat"). Thus, reasonable approximations for textural changes in music and salient episodic changes in visual content could be used as a basis for the audiovisual synchronization.

3.2 System Design

One of the main assumption of this work is that a person capturing a user generated video has a clear mental image of what he wants to record, but that the resulting video content on its own only weakly represents this mental image. In order to strengthen the representation and make it truly enjoyable multimedia, suitable soundtrack music should be found and synchronized for the uploaded video.

The user should not have to describe music to be added in the form of specialized musical terminology or dedicated song title vocabulary. Instead, a free-text story-driven approach has been employed, asking users for a textual plot description and several supporting keywords for their video, after which the provided user input is automatically compared to cinematographic plot situations. Folksonomic[2] song descriptors that are commonly associated with existing plot situations are assumed to also be suitable for the given input video, if the intended story of the video resembles that of the cinematic plot. This results in a pre-selection of candidate soundtrack songs

[1] "A spoken or written account of connected events; a story." (from WordReference).

[2] "A folksonomy is a system of classification derived from the practice and method of collaboratively creating and managing tags to annotate and categorize content; this practice is also known as collaborative tagging, social classification, social indexing, and social tagging" (from `http://en.wikipedia.org/wiki/Folksonomy`).

built through cross-modal connections which will be referred as *story-driven soundtrack pre-selection*.

In the multimedia/content-based retrieval community, the video component has traditionally been considered as the strongest modality in a multimodal setting. Audio going with this video component is usually considered to be subordinate, resulting in the expectation that an added audio component shall be modified (typically through time warping) in case of a non-perfect temporal fit to the video. However, in the case of user-generated video, the video component is not a very strong modality. Therefore, both the video and music soundtrack streams will be kept in their original forms, without modifying their internal temporal discourse during the synchronization. Instead, the signal is only shifted by a fixed lag, found through cross-correlation. In this, audio and video features have been chosen in order to capture the type of temporal developments mentioned by the user survey study respondents in [12]. Since videos can have very diverse visual content, one cannot concentrate on a specific class of visual objects or dedicated concept detectors, but only can consider general descriptors. Furthermore, it has been assumed that the visual content consists of a single raw shot. This means that no sudden cuts or cross-fades are present, although fast camera motions like panning might happen. Again, this assumption is considered valid because only mobile recorded videos have been considered.

Each pair consisting of a pre-selected music song and the uploaded video is evaluated assuming that the cross-correlation score gives an accurate assessment of the degree of congruence of video and audio segments. Since multiple pre-selected soundtracks may fit well to the video content, the system returns synchronized results for the three best-scoring soundtracks.

3.2.1 Story-Driven Soundtrack Pre-Selection

When starting the application, a user is asked to enter a short free-text story description of the video for which he searches a soundtrack, as well as several tag-like keywords describing the intended “feel” of the video. Subsequently, a traditional text retrieval approach is employed using length-normalized TF-IDF measures to score documents based on queries. This is done in three steps with different document and query types:

- the user story is compared to movie plots in order to retrieve the song tags associated to the movie soundtracks;

- the previously retrieved song tags together with the tag-like keywords are used in an intermediate clean-up stage in which a song tag co-occurrence index is employed;
- the last retrieved song tag set is finally used to retrieve music in a local repository looking in the indexed music metadata (genre, instrumentation, associated mood, etc.).

In the next chapter, the indexing and retrieval algorithms employed for the story-driven soundtrack pre-selection are thoroughly illustrated.

3.2.2 Video to Music Synchronization

After the pre-selection of a number of music songs, the system computes a synchronization score for each pair given by the uploaded video and a pre-selected music song. In order to do that, the following features, which are intended to approximate the changes described by respondents in [12], are extracted:

- a combination of the signal novelty described in [7] with onset information to find significant sound changes that go together with strong musical accents;
- the *motion activity* score defined in the MPEG-7 standard [9] reflects the intensity of action.

Even if not specifically meant for synchronization purposes, both descriptors are well-founded in literature. They do not rely on training data and they are relatively light-weight in the computational sense. Furthermore, they are intended to go beyond aesthetic linking and a pure physical-to-physical connection as in the case of motion sonification [10]. They are indeed employed to establish audiovisual events links.

Once these descriptors are extracted, the cross-correlation of them is maximized for each music-video pair. This results in having each pair characterized by the synchronization score and the associated best audio lag. By ranking the scores, the three soundtracks having the highest synchronization scores are selected to produce three automatically synchronized videos which are shown to the user.

3.3 Novelties

The idea of considering existing cinematic productions as examples for associating music to video has already been used in most of the related works [18, 11]. However, in all these approaches, audio and video signal features were directly associated with each other, thus implicitly assuming that low-level signal characteristics hold all necessary information for making cross-modal associations. As shown in the previous sections, this is not a realistic assumption. Thus, establishing cross-modal *thematic* connections through movie plots represents a novelty because it is a more expressive way to connect a video with a song.

The synchronization approaches proposed in the past try to make only physical-to-physical connections, such as beat to shot boundary or tempo to motion intensity alignment [17, 6]. In this work, the goal is to reach an event-to-event connection in order to link more semantic concepts without restricting the range of concepts to a small set. The adopted features have been chosen to well encode salient events: for instance in the case of the video analysis they can detect an object which suddenly starts to move while boundaries between different timbres in the music usually detect distinguishable parts of a song. Furthermore, the adoption of a crowd sourcing experiment to forecast the most effective features represents a novelty because most of the mentioned works adopted a large set of common low-level features without strongly motivating the choices.

The proposed approach has also been compared to the YouTube video editing panel through which the uploaders can replace the original audio with a music track from a library of more than 150000 songs. No details are given about either the synchronization procedure or the type of search, but the following example suggests that the search takes into account music metadata only. Using the keywords “relaxed holidays sea sun and fun” the following music songs are pre-selected adopting the system devised in this work: Beach Bum, Feelin’ Good, Mirage, Long Road Ahead, Somewhere Sunny and Blobby Samba. With the YouTube music search engine, no result is given using the whole query. Using then the sub-query “holidays sun”, these songs have been retrieved: Holiday In The Sun, Dreams of the Sun and Day Dreaming. Adding “relaxed” in the query the system cannot retrieve songs anymore. Finally, regarding the synchronization capabilities, in

YouTube there is an option through which only those songs similar in length to the uploaded video are retrieved: this feature alone is by far different from the synchronization strategy adopted in this work.

Chapter 4

Story-Driven Soundtrack Pre-Selection

In order to make a selection of thematically suitable songs from a music repository, the user is asked to enter a short free-text story description of the video for which he searches a soundtrack, as well as several tag-like keywords describing the intended “feel” of the video. In this chapter the full *story-driven soundtrack pre-selection* pipeline is thoroughly explained. Besides, implementation details are given in which the Lucene text search engine, which is briefly presented in Appendix A, has been used.

4.1 Associations between Movie Plots, Music and Social Tags

The link between music to be retrieved and user text entries relies on the entities shown in Figure 4.1:

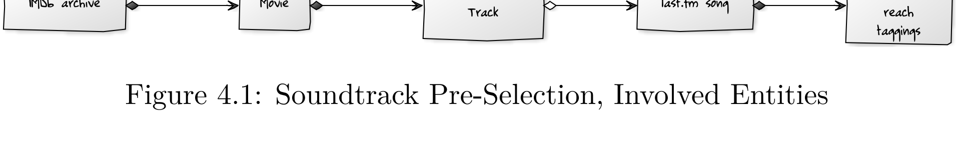


Figure 4.1: Soundtrack Pre-Selection, Involved Entities

The *Local Music Repository*, represented as a set of *Music Song* elements, is the only source of music in the system. This means that no external music

can be employed. Nevertheless, new songs can be added at a second time if opportunely described by the metadata required by the system. Such metadata includes the *title*, a *description* provided by the author, and other information such as the genre, the instruments and the feel.
Besides, information regarding movies, plots, soundtracks and textual folksonomic descriptors are used. In this, the Internet Movie Database (IMDb) archive is the source of information regarding movies. Each movie is described by its plot and the list of employed soundtracks. Mapping the soundtracks in the last.fm social music service, it is possible to obtain a series of tags added by the users for each soundtrack.
The number of mined movie entries is 80909 but only 12573 have both the plot description and one or more soundtracks present in the last.fm archive. In total, 228645 unique song tags were crawled; only retaining those tags used by at least 100 different last.fm users, a vocabulary of 11616 unique social music tags has been kept. For the music, a dataset of 1084 songs with royalty-free production music from three resources[1] has been used.

4.2 Indexing

The music pre-selection is based on a series of indexes which are employed to retrieve music given the user text entries. More in detail, three indexes have been devised[2]:

- the *Music Tag to Plots* index (**T2P**), used for searching documents similar to the user plot entry and returning a list of related music tags;
- the *Music Tag to Music Tags* index (**T2T**), a music tag co-occurrence index used for cleaning up through a step similar to the pseudo relevance feedback;
- the *Song to Music Metadata* index (**S2M**), used for retrieving songs in the local repository related to a series of music tags provided as query which are compared to the music metadata.

[1] `http://www.incompetech.com/` (Kevin MacLeod), `http://www.danosongs.com/` (Dan-O) and `http://derekaudette.ottawaarts.com/` (Derek R. Audette).

[2] The assigned names follow the naming convention *key to indexed content*; refer to the Appendix A for a definition of *key* and *indexed content*.

In order to present the indexing algorithms, some definitions are first given.

Definition 4.1 (Movies, Soundtracks, Music Tags and Music Repository)
M = set of movies
S_M = set of all songs used as sound track in any movie in M
T_{S_M} = set of all tags linked to any song in S_M
PM = set of production music (local repository)

The following propositional functions are also given:

Definition 4.2 (Instances Relations)
hasSong(movie, song): true if the song is used as sound track in the movie
hasTag(song, tag): true if the song has the tag

4.2.1 Music Tag to Plots Index

The *Music Tag to Plots* index contains documents where the key is a music tag and the indexed content is built concatenating the plots of all movies that have a soundtrack song with the particular music tag forming the document key. The index is built through the Algorithm 4.1:

Algorithm 4.1 Music Tag to Plots Index Builder

```
for each $tag \in T_S$ do
    $contents \leftarrow \emptyset$
    for each $song \in S_M \mid \text{hasTag}(song, tag)$ do
        for each $movie \in M \mid \text{hasSong}(movie, song)$ do
            $contents \leftarrow contents \cup movie.plot$
        end for
    end for
    $index.\text{add}(\text{new Document}(tag, contents))$
end for
```

4.2.2 Music Tag to Music Tags Index

In order to clean up a series of music song tags, a step similar to pseudo relevance feedback is achieved through the *Music Tag to Music Tags* index. It is a tag co-occurrence search index having once again song tags as keys.

The indexed content consists of all music song tags that occur together with the key tag within a song. The Algorithm 4.2 shows how the index is built:

Algorithm 4.2 Music Tag to Music Tags Index Builder

```
for each $song \in S_M$ do
    $tags \leftarrow \emptyset$
    for each $tag \in T_{S_M}$ | hasTag($song, tag$) do
        $tags \leftarrow tags \cup tag$
    end for
    for each $tag \in tags$ do
        $contents \leftarrow tags \setminus tag$
        $index$.add(new Document($tag, contents$))
    end for
end for
```

4.2.3 Song to Music Metadata Index

The *Song to Music Metadata* index is built upon metadata descriptions of the songs (genre, instrumentation, associated mood, etc.), as entered by the original song composers. The Algorithm 4.3 details its building procedure:

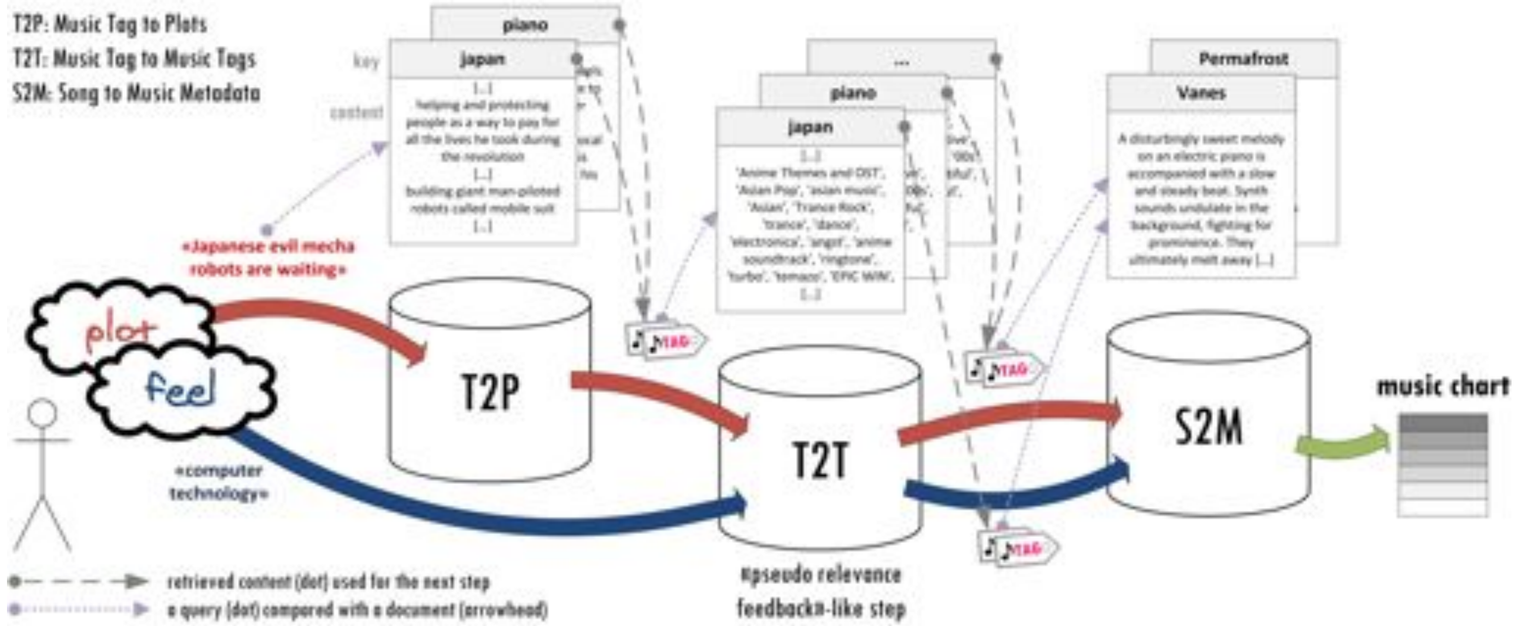
Algorithm 4.3 Song to Music Metadata Index Builder

```
for each $music \in PM$ do
    $contents \leftarrow music$.title $\cup$ $music$.genre $\cup$ $music$.subgenre $\cup$ $music$.instruments $\cup$ $music$.feel $\cup$ $music$.description
    $genre \leftarrow music$.genre $\cup$ $music$.subgenre $\cup$ $music$.instruments
    $index$.add(new Document($music$.title, $music$.file, $contents, genre$))
end for
```

In this case, the fields *title*, *file* and *genre* are stored but not indexed, while the only indexed field is again *contents*. Each retrieved document corresponds to a song in the local music repository.

4.3 Retrieval

Using the aforementioned indexes, the user entries, namely the plot description and the feel-related keywords, are used to retrieve music as shown in Figure 4.2:



T2P: Music Tag to Plots
T2T: Music Tag to Music Tags
S2M: Song to Music Metadata
key
content
piano
japan
[...]
helping and protecting people as a way to pay for all the lives he took during the revolution
[...]
building giant man-piloted robots called mobile suit
[...]
«Japanese evil mecha robots are waiting»
plot
feel
«computer technology»
T2P
T2T
S2M
...
piano
japan
[...]
'Anime Themes and OST', 'Asian Pop', 'asian music', 'Asian', 'Trance Rock', 'trance', 'dance', 'electronica', 'angst', 'anime soundtrack', 'ringtone', 'turbo', 'tomaco', 'EPIC WIN',
[...]
Permafrost
Vanes
A disturbingly sweet melody on an electric piano is accompanied with a slow and steady beat. Synth sounds undulate in the background, fighting for prominence. They ultimately melt away [...]
TAG
music chart
«pseudo relevance feedback»-like step
retrieved content (dot) used for the next step
a query (dot) compared with a document (arrowhead)

Figure 4.2: Music Pre-Selection

Three steps are necessary in order to retrieve music from the local repository:

1. The free-text user description (the “plot” of the video) is compared to IMDb plot data via the Music Tag to Plots index: the text entry is compared to the content of each document in the index and the output consists of the tags stored as keys in the retrieved documents. This is done in order to associate similar plot elements to similar corresponding music keywords.

2. Subsequently, the obtained music song tags are cleaned up through a step similar to *pseudo relevance feedback* employing the Music Tag to Music Tags index so that in the end a set of n tags is retrieved as follows:

   - first half set: the list of song tags retrieved from the T2P index are used to match documents in the T2T index and the top $n/2$ retrieved keys are collected;
   - second half set: the keywords provided by the user (“feel” of the video) are used to match documents in the T2T index; again the top $n/2$ retrieved keys are collected.

3. Finally, the Song to Music Metadata index is used to retrieve songs in the local repository by providing the song tags from the previous step as a query.

4.4 Pseudo Relevance Feedback -like Step

During the preliminary experiments, no tag co-occurrence index had been used and looking at the outcomes, one would say that the thematic connection between the user plot situations and the music is often weak. The reason for that is the following: some song tags are noisy and may cause a *thematic drift*. In order to clean up such tags, the tag co-occurrence index, namely the Music Tag to Music Tags index, and the pseudo relevance feedback technique are combined and used.

On one hand, the tag co-occurrence analysis is used as a basis for finding similar and conceptually related tags. As stated in [21], this is necessary when different tags might have been used for the same concept. In this case, it is difficult to find all items relevant for a certain tag.

An alternative approach to the tag co-occurrence analysis would be to map tags to a thesaurus. But, as it is observed in [4], the vocabulary of folksonomies includes many community-specific terms which do not appear yet into any lexical resource.

On the other hand, the *pseudo relevance feedback* is used as criterion for the automatic selection of relevant tags retrieved from the tag co-occurrence index. In order to understand the rationale behind this choice, the technique is briefly explained.

The relevance feedback is a well-known information retrieval technique which involves the user to refine his query and get rid of irrelevant results. Providing a query to a search engine, the user receives a first list of retrieved documents. Some of theme might not be relevant to the user. Thus, a feedback can be asked in which relevant and non-relevant results are distinguished. This feedback is then used to retrieve a new set of documents aiming to maximize the number of those which are relevant. This is achieved as follows. Queries and documents are represented as points in a vector space model so that the query can be refined as a linear combination of the original query, the relevant documents and eventually those which are non-relevant. The weights determine how far or near the potentially relevant documents have to be with respect to the original query and the manually marked documents. The weights are chosen so that the new query is near to those documents marked as relevant and far from those marked as non-relevant.

In order to automatically obtain a feedback, it can be assumed that the user would mark the top k retrieved documents as relevant. Thus, using the top k documents to refine the query leads to a new query as described above. This technique is known as the *pseudo relevance feedback*.

Chapter 5

Video to Music Synchronization

The Video to Music Synchronization framework is used to synchronize the uploaded video with the pre-selected soundtracks obtained with the method shown in Chapter 4. This is achieved by first analyzing a number of features in the audio and video streams separately. Following this analysis, the streams are synchronized. Based on a score expressing the goodness of the synchronization for every soundtrack to the video, the pre-selected soundtracks are ranked. The top-k ranked synchronizations are returned to the user.

In this chapter, the framework is first presented defining the *Best Audio Lag* problem and its solution. Then, the audiovisual analysis techniques are presented. The last section focuses on the pre-selected soundtracks re-ranking problem.

5.1 Score-based Synchronization Framework

Two separate problems are tackled in the presented framework as shown in Figure 5.1. One is nested into the other one. Due to this structure, the two parts of the framework are referred to as the *inner layer* and the *outer layer.* They are defined as follows:

- *inner layer*: a pair of audio-video streams have to be synchronized; in this, a *synchronization score* is computed; it reflects the degree of match, achieved through the synchronization process, between the streams;

- *outer layer*: iterating over a set of audio streams, pairs consisting of a video stream and the current audio stream are considered; each pair is provided as input to the inner layer; the obtained synchronization scores are then used to rank the set of audio streams so that the highest score is attributed to the best aligned audio stream.

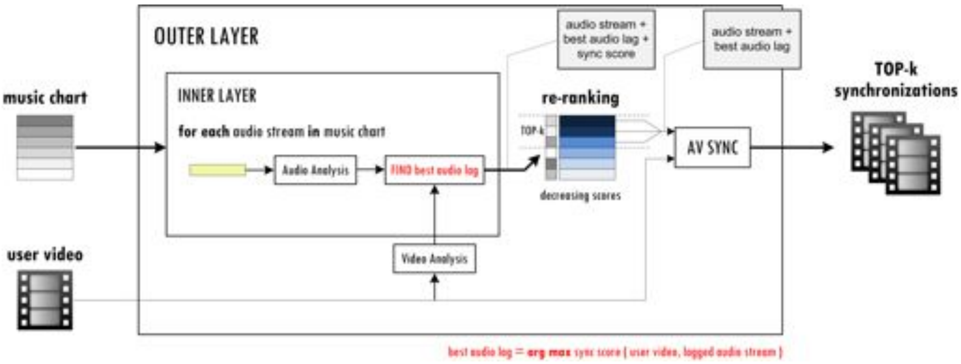

OUTER LAYER
audio stream + best audio lag + sync score
audio stream + best audio lag
music chart
INNER LAYER
for each audio stream in music chart
Audio Analysis
FIND best audio lag
re-ranking
TOP-k
decreasing scores
AV SYNC
TOP-k synchronizations
user video
Video Analysis
best audio lag = arg max sync score (user video, lagged audio stream)

Figure 5.1: Synchronization Framework, Layers

The main element of the presented framework is the *synchronization score* for which the following definition is given:

Definition 5.1 (Synchronization Score)
The synchronization score is a function of an audiovisual stream pair. It assesses the extent to which the streams are temporally *aligned* according to any set of predefined rules or criteria.

In the next part of this section, the two layers are explained in-depth. Before that, some shared definitions are given.

Definition 5.2 (Stream)
$\text{stream}(k) : \{1 \dots L\} \to D$ where

- D is the vector space in which stream points lie (e.g. [-1,1] for a mono-channel audio stream);

- $L \in \mathbb{N}_{>1}$ is the *length* of the stream.

Both the audio and the video streams are defined accordingly to the Definition 5.2. The only difference is given by the unit of measurement: a video stream's length L_V is expressed in number of frames, while an audio stream's length L_A is expressed in number of samples.
Finally, in order to express a stream length in seconds, the following definition is given:

Definition 5.3 (Stream Duration)
Given a stream having length $L \in \mathbb{N}_{>1}$, its duration in seconds is defined as

$$\text{stream duration}(\text{stream}(k)) = \begin{cases} L/\text{sample rate} & \text{if stream}(k) \text{ is an audio stream} \\ L/\text{frame rate} & \text{if stream}(k) \text{ is a video stream} \end{cases}$$

where

- *sample rate* is the number of audio samples per second;
- *frame rate* is the number of frames per second (fps).

5.1.1 Inner Layer - Synchronization

In the synchronization problem, the input video stream is treated as an atomic object. Therefore, no change is made on the video. What changes is the auditory content which is indeed replaced by a delayed and/or zero padded version of the input audio stream. The input audio stream can only be delayed and/or zero padded because no time warping has to be applied. Thus, neither the video nor the audio are time stretched following the ideas reported in Section 3.2.
In order to link strong accents across different modalities, it has been decided to analyze the streams in terms of *intensity features.* For instance, one could measure the amount of motion in a scene or the estimated velocity of a visual object which is tracked along the time. Some examples can also be given for the audio streams: it can be measured the loudness of an audio segment, the strength of a timbral change or the intensity of the onsets. All these intensity features share the following properties:

- they change along the time;
- they are one-dimensional;

- peaks are associated to strong *accents*[1].

Therefore, limiting the analysis to this particular type of features should not represent a bottleneck when the task is the alignment of audiovisual accents. The following definition formalizes what has been said above:

Definition 5.4 (1-Dimensional Intensity Feature)
Row vector $\mathbf{s} \in F^n$ where

- $F = [0\ 1]$ or $F = \mathbb{R}_{\geq 0}$ (respectively with or without normalization);
- n is the length of the stream expressed in number of video frames.

The vector represents a stream along the time.

One may wondering why it has been chosen to express the length of the vector $\mathbf{s}$ in number of video frames even if the Definition 5.4 applies both to audio and video streams. The most obvious motivation is the following: the alignment of audio samples to video frames has to be done at the lowest temporal resolution because there is no method more precise than the lowest resolution available. Such resolution is always the one of the video[2]. A more solid motivation is grounded on psychological knowledge. In [20], the temporal *Just-Noticeable Difference* (JND) of human beings has been measured through a series of experiments in which respondents had to detect asynchrony in an audiovisual stimulus. The best performance shows that, when the asynchrony is under the threshold of 50 ms, the subjects perceive the multi-modal stimulus as synchronous. The video frame frequency usually ranges between 24 and 30 fps: given that the audio has a by far higher temporal resolution, the video temporal resolution can be safely adopted as the common resolution because the greatest synchronization error would then be $1/24$ s ≈ 42 ms that is lower than the previously mentioned threshold of 50 ms. It is worth noting that this choice does not deny an analysis of the two streams at different temporal resolutions; it is just a suitable resolution to align the two streams.

[1] “An emphasized detail or area, a small detail in sharp contrast with its surroundings.” (from Merriam-Webster®).

[2] The audio sample rate is always much bigger than the frame rate. For instance, one can consider a frame rate of 30 fps and an audio sample rate of 44100 Hz. In this case, the audio has to be delayed and/or padded by a number of samples which is a multiple integer of 44100/30.

Each stream can be analyzed from many point of views so that different types of accent have to be detected. Then, it is useful to extend the 1-Dimensional Intensity Feature as follows:

Definition 5.5 (Multi-Dimensional Intensity Feature)
Matrix $\mathbf{S} \in M_{m,n}(F)$ where

- $F = [0\ 1]$ or $F = \mathbb{R}_{\geq 0}$ (respectively with or without normalization);
- n, the number of columns, is the duration of the stream expressed in number of video frames;
- m, the number of rows, is the number of intensity features extracted for the considered stream;
- each row $\mathbf{s}^i$ is a 1-dimensional intensity feature vector so that $\mathbf{S}$ can be written as

$$\mathbf{S} = \begin{pmatrix} \dots \\ \mathbf{s}^i \\ \dots \end{pmatrix}$$

The two sets of audio and visual intensity features are defined as follows:

Definition 5.6 (Auditory Intensity Features Set)
Given a set of auditory intensity features, the set A is the set of labels referring to each feature, e.g. $A = \{\text{Loudness}, \text{Novelty}, \text{Onsets}\}$. Mapping each label to an integer, the set is redefined as follows: $A \leftarrow \{1, \dots, \mid A \mid\}$.

Definition 5.7 (Visual Intensity Features Set)
Given a set of visual intensity features, the set V is the set of labels referring to each feature, e.g. $F = \{\text{Motion Activity}, \text{Tracked Item \#0 Velocity}\}$. Mapping each label to an integer, the set is redefined as follows: $V \leftarrow \{1, \dots, \mid V \mid\}$.

Having to link accents lying in different streams, it is necessary to specify which pairs of 1-Dimensional Intensity Features are compared.

Definition 5.8 (Audiovisual Intensity Feature Pairs Set)
The set of audiovisual Intensity Feature pairs is a subset of all the possible audiovisual Intensity Feature pairs, or $P \subsetneqq \mathcal{P} = A \times V$ (e.g. $P =$

$\{(\text{Novelty}, \text{Motion Activity}), (\text{Loudness}, \text{Tracked Item \#0 Velocity})\}$, using the mapping between labels and integers $P = \{(2, 1), (1, 2)\}$).

This last definition is important because, together with the audiovisual accents comparison method, it formalizes the concept of "temporal alignment made according to any set of predefined rules or criteria" reported in the Definition 5.1.

Following the definitions given above, the intensity features comparison method can be presented. In this work, it has been chosen to rely on a well-established mathematical operator: synchronization scores are based on *cross-correlation*, which is defined as "a measure of similarity of two waveforms as a function of a time-lag applied to one of them" (from Wikipedia). Having to compare $\mid P \mid$ audiovisual intensity feature pairs at different time-lags, the following matrix of cross-correlation scores is defined:

Definition 5.9 (Cross-Correlation Matrix)
Given the Multi-Dimensional Intensity Feature matrices for the audio $\mathbf{S}_A$ and for the video $\mathbf{S}_V$, the Cross-Correlation Matrix $\mathbf{XC} \in M_{p,n_{XC}}(\mathbb{R}_{\geq 0})$ where:

- p is equal to $\mid P \mid$, P is the set defined in the Definition 5.8;
- n_{XC}, the number of columns of $\mathbf{XC}$, is defined as $2 \times \max(n_A, n_V) - 1$
- n_A and n_V are respectively the number of columns of $\mathbf{S}_A$ and $\mathbf{S}_V$.

then $\mathbf{XC}$ can be represented as follows:

$$\mathbf{XC} = \begin{pmatrix} \cdots \\ \mathbf{xc}^k \\ \cdots \end{pmatrix}$$

where $k \in \{1 \ldots p\}$.

Each row $\mathbf{xc}^k$ in the cross-correlation matrix $\mathbf{XC}$ is referred to as *cross-correlation vector* and is defined as follows:

Definition 5.10 (Cross-Correlation Vector)
Given a cross-correlation matrix $\mathbf{XC}$, its rows, called Cross-Correlation Vectors, are defined as:
$\mathbf{xc}^k = (\ldots, \mathrm{xc}_l^k, \ldots)$ where

- $l \in \{1 \ldots n_{XC}\}$ is the column index associated to the lag $l - N$
- $N = \max(n_A, n_V)$;
- $n_{XC} = 2 \times N - 1$ is the number of meaningful lags;
- and xc_l^k is the cross-correlation between the vectors of the k-th pair $(i_k, j_k) \in P$ when the 1D Intensity Feature associated to the audio has lag equal to $l - N$, or

$$\mathrm{xc}_l^k = \sum_{m=-\infty}^{\infty} \mathrm{s}_m^{i_k} * \mathrm{r}_{m+(l-N)}^{j_k}$$

- $\mathbf{s}^{j_k}$ is the j_k-th row of $\mathbf{S}_V$;
- $\mathbf{r}^{i_k}$ is the i_k-th row of $\mathbf{S}_A$;
- $i_k \in A$ and $j_k \in V$.

In summary, the process required to compute the correlation matrix **XC** is the following (see Figure 5.2):

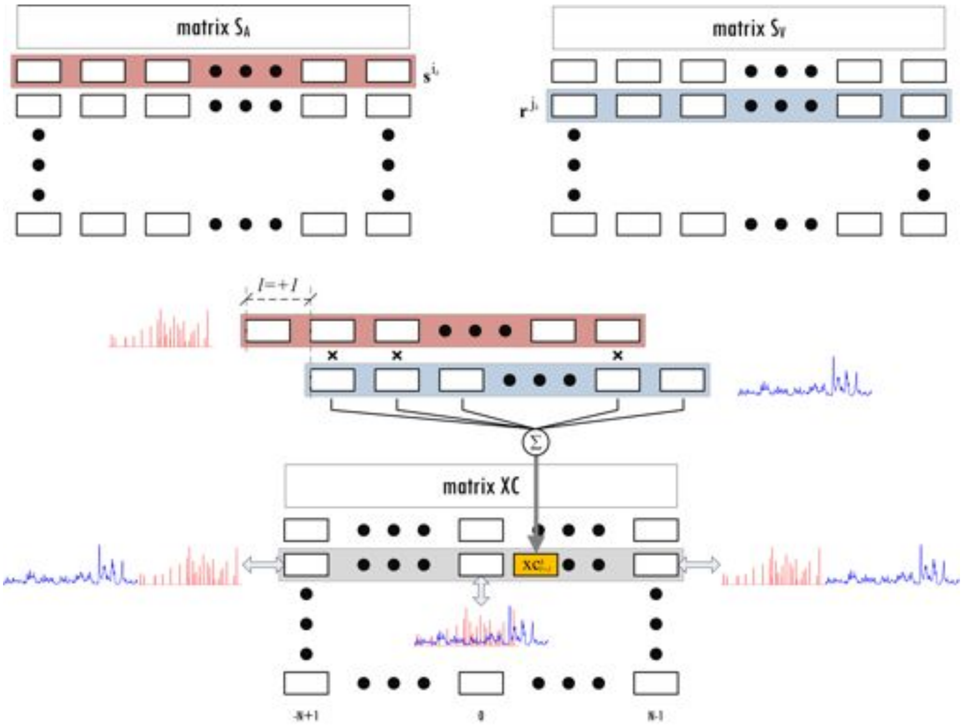


Figure 5.2: Synchronization Framework, Inner Layer

- the audio stream is analyzed extracting the audio intensity features;
  - each feature is encoded as a 1-dimensional intensity feature vector $\mathbf{r}^i$;
  - all these vectors form the multi-dimensional intensity feature matrix S_A;
- the video stream is analyzed extracting the audio intensity features;
  - each feature is encoded as a 1-dimensional intensity feature vector $\mathbf{s}^j$;
  - all these vectors form the multi-dimensional intensity feature matrix S_V;
- some pairs of audiovisual intensity features are combined together to compute the cross-correlation vectors for a set of audio lags ranging from $-N+1$ to $N-1$ where $N = max(n_A, n_V)$;
- all these cross-correlation vectors form the cross-correlation matrix **XC**.

In order to align the audio stream to the video stream, a column from th cross-correlation matrix **XC** has to be selected. The selected column corresponds to an exact audio lag value which is then used to align the two streams. This is achieved reducing the **XC** matrix to a final row vector of synchronization scores in which the highest value determines the column, or the audio lag, to be selected. Such projection of the matrix **XC** onto a row vector is defined as follows:

Definition 5.11 (Synchronization Scores Vector)
Row vector $\mathbf{w} \in \mathbb{R}_{\geq 0}^{n_{XC}}$ where:

- the element w_l is associated to the lag $l - N$;
- $N = max(n_A, n_V)$;

In the particular case in which the matrix **XC** has only one row, the synchronization scores vector corresponds to **XC**. This happens when only one pair of audiovisual intensity features are compared. Otherwise, it is necessary to combine the comparisons in order to reduce **XC** to a row vector. Some examples are given:

- average score, each value in w corresponds to the mean of the values in the correspondent column of **XC**:

$$\mathrm{w}_l = \frac{1}{\mid P \mid} \sum_{k=1}^{|P|} \mathrm{xc}_l^k$$

- max score, each value in w corresponds to the biggest value in the correspondent column of **XC**:

$$\mathrm{w}_l = \max_k \mathrm{xc}_l^k$$

In this work, the idea to integrate more than a single audiovisual intensity feature pair has not been investigated further. It has been left for future developments[3]. Nevertheless, projecting the cross-correlation matrix **XC** onto a synchronization scores vector according to a set of predefined rules or criteria, the best synchronization can always be found. Thus, at this point it is assumed that:

- either **XC** has only one row, i.e. the synchronization scores vector $\mathbf{w} = \mathbf{XC}$;

- or **XC** has been somehow projected onto the synchronization scores vector **w**;

Once the synchronization scores vector is computed, it is possible to find the *best audio lag*:

Problem 5.1 (Best Synchronization)
Given a pair of audiovisual streams, the best audio lag is given by:

$$l^*_{fps} = -N + \operatorname*{arg\,max}_{l \in \{1 \ldots (2 \times N-1)\}} w_l$$

where l^*_{fps} is expressed in number of video frames and can be converted in number of audio samples as follows:

$$\text{best audio lag} = \frac{\text{sample rate} \times l^*_{fps}}{\text{frame rate}}$$

[3] The author is sorry for those who really were so brave to read everything up to this point and, by any chance, also found the multi-dimensional intensity features topic interesting.

This formulation of the best synchronization problem leads to an optimization problem which has a finite feasible set whose size increases linearly with the duration of the longest stream. For instance, given a video of 4 minutes at 30 fps and a music track of 5 minutes, the visual and auditory intensity feature vectors lengths are respectively 7200 and 9000 video frames; then the feasible set for l is $\{-9000 \cdots +9000\}$.
Given the audio and video streams and the best audio lag, a new audio stream having the same stream duration of the video is created through the Algorithm 5.1. This will be merged with the video stream replacing any existing audio content.

Algorithm 5.1 Synchronized Audio Stream

```
audio($k$) $\leftarrow$ audio stream to be synchronized        $\triangleright$ read the input
video($k$) $\leftarrow$ video stream to be synchronized
$lag$ $\leftarrow$ best audio lag (in seconds)

$d_A \leftarrow$ stream duration(audio($k$))                    $\triangleright$ get duration
$d_V \leftarrow$ stream duration(video($k$))

$clip_{from} \leftarrow \max(0, lag - d_A)$                       $\triangleright$ extract the clip from audio($k$)
$clip_{to} \leftarrow \min(d_A, d_V - lag)$
audio$'$($k$) $\leftarrow$ extract(audio($k$), $clip_{from}$, $clip_{to}$)

$pad_{left} \leftarrow -lag$                                      $\triangleright$ left zero padding
if $pad_{left} > 0$ then
    audio$'$($k$) $\leftarrow$ merge(zero padding($pad_{left}$), audio$'$($k$))
end if
$pad_{right} \leftarrow d_V -$ stream duration(audio$'$($k$))     $\triangleright$ right zero padding
if $pad_{right} > 0$ then
    audio$'$($k$) $\leftarrow$ merge(audio$'$($k$), zero padding($pad_{right}$))
end if

create an empty video object
assign video($k$) as video stream
assign audio$'$($k$) as audio stream
```

The Algorithm 5.1 makes use of a number of auxiliary functions; they are defined in the algorithm block 5.2.

Algorithm 5.2 Synchronized Audio Stream - auxiliary functions

```
function EXTRACT(audio($k$), $from$, $to$)
    $from \leftarrow from \times$ sample rate, $to \leftarrow to \times$ sample rate
    create a new empty audio stream audio$'(k)$
    append the audio samples from audio($k$) with $k \in \{from \ldots to\}$
    return audio$'(k)$
end function

function MERGE(audio$_1(k)$, audio$_2(k)$)
    create a new empty audio stream audio$'(k)$
    append the samples in audio$_1(k)$ to audio$'(k)$
    append the samples in audio$_2(k)$ to audio$'(k)$
    return audio$'(k)$
end function

function ZERO PADDING($n$)
    create a new empty audio stream audio($k$)
    append $n$ samples set to zero in audio($k$)
    return audio($k$)
end function
```

5.1.2 Outer Layer - Ranking

Given an input video and a set of music songs, each audiovisual stream pair is provided as input to the inner layer. For each pair, the output consisting of a best audio lag and a synchronization score is collected. Ranking the music songs by the obtained synchronization scores defines a ranked list. From this list, the top-k corresponding music songs are used to make k new music videos through the Algorithm 5.1.

In this framework, the synchronization scores are not normalized at all. For example, one may consider to apply a temporal length normalization. In that case, normalizing by the video stream duration has no effect because the top-k soundtracks will remain the same (the same normalization factor is applied to all the scores). Such normalization would instead be useful if one needs to compare the synchronization scores of audiovisual pairs in which the video component is different.

The Algorithm 5.3 summarizes what has been said above:

Algorithm 5.3 Synchronization-based Ranking

```
video($k$) $\leftarrow$ video stream to be synchronized
$AS \leftarrow$ set of audio streams audio($k$)
$R \leftarrow$ init collection
for each audio($k$) $\in AS$ do
    $(\text{best audio lag}, \text{sync score}, \text{video}) \leftarrow \text{inner layer}(\text{video}(k), \text{audio}(k))$
    $key \leftarrow \text{audio}(k)$
    $entry \leftarrow (\text{best audio lag}, \text{sync score}, \text{video})$
    $R.insert(key, entry)$
end for
sort $R$ by $key$
return $R.top(3)$
```

5.2 Intensity Features

When a user watches a video, his attention level is not always constant. Excluding external factors, such as the presence of a disturbing environmental noise, the attention level can be affected by the audiovisual content and the user himself. On one hand, certain features do sometimes cause watchers to orient automatically (e.g. a sudden loud noise, a rapid movement). However, many features that attract or hold attention of users are informative, signaling content that users are likely to find relevant or entertaining.

When no prior information is available telling what a large audience might find relevant or entertaining, one could seek for a set of general audiovisual features. For instance, in the visual domain rapid movement is quite general in order to span the following events: a car chase, a cat fighting with another one, a camera which pans in order to change subject. While in the music domain, one could detect loudness peaks, onsets or timbral changes (as emerged from the user study in [12]). These musical features are often adopted in film music to attract users' attention.

On the other hand, a source of information could be exploited to focus on specific parts of the audiovisual content. For instance, the person who uploads could select through a bounding box a visual item to be tracked along the time. Its estimated velocity can then be read as an intensity feature. Similarly, the user could also highlight a small part in the music which he finds relevant. Through audio similarity techniques, other occurrences of the highlighted section can be sought so that the computed intensity feature shows high values when it is likely to find similar segments.

As a first attempt to align audiovisual accents, it has been decided to focus on the first case. The reason is that the approach is simpler than the second and previous works in that direction exist [10, 16].

Finally, it is remarked the role of the intensity features. They should well encode *when* and *to what extent* a relevant feature occurs. In this case, they can be exploited to align accents across different modalities so that the perceived cross-modal match is enhanced.

5.2.1 Motion Activity

In order to capture sudden events in the visual content without restricting to a limited set of real world objects, it has been chosen to assess the suitability of the MPEG-7 *motion activity* descriptor presented in [9]. As the authors state, it encodes the overall activity, or pace of motion, which often denotes the level of action. What makes this descriptor definitively interesting is its implementation. Instead of estimating motion with computer vision techniques, it exploits the information regarding motion already available in the video compressed domain, namely the *MPEG motion vector field.* Even if MPEG motion vectors have been devised to reach a good video stream compression, the user study in [9] shows that such vectors revealed to be also suitable in the estimation of the motion activity descriptor. This allows to drastically reduce the required computation resources and even devise online algorithms.

Implementation

The motion activity descriptor is computed for each frame considering the magnitudes of the motion vectors. More precisely, the standard deviation of the magnitudes is computed. Due to the absence of a public available implementation, in this work the descriptor has been implemented as follows:

- the descriptor is computed for each P-frame[4] by the Algorithm 5.4;
- it might be necessary to skip some frames, in this case an illegal value is assigned;
- illegal values for skipped frames are then replaced looking at neighborhood values in order to avoid false falls to zero;
- thresholding and normalization according to the outcomes reported in [9] are applied;
- a median filter is finally used to get rid of spurious spikes.

Algorithm 5.4 Motion Activity Index

```
for each $frame \in \text{video}(k)$ do
    $magnitudes \leftarrow \emptyset$
    for each $\mathbf{v} \in frame.\text{MVF}$ do
        $m = \text{sqrt}(v.\text{dx}^2 + v.\text{dy}^2)$
        $magnitudes \leftarrow magnitudes \cup m$
    end for
end for
$descriptor = \text{stddev}(magnitudes)$
```

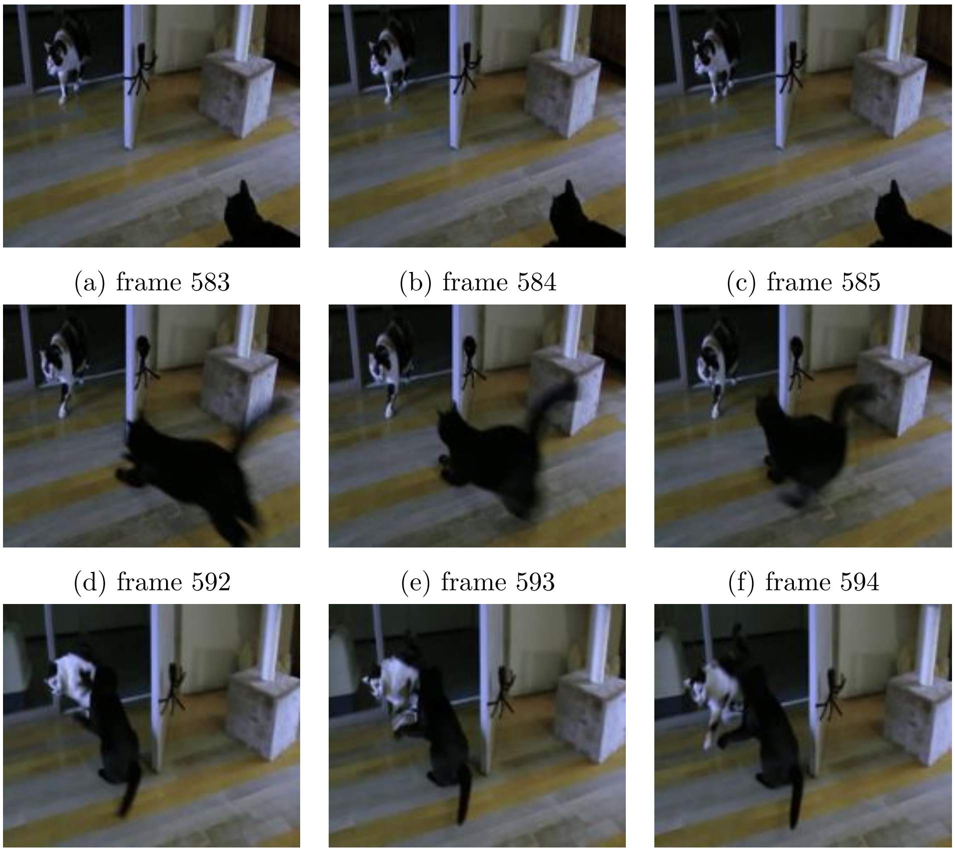
As stated above, the computation of some frames is skipped. This happens when a frame is not a P-frame or when a *duplicated frame* occurs. The latter case may happen for two reasons. Sometimes the video is temporally edited by duplicating frames or audio samples so that music and video are well timed together. A transcoding in which the frame rate has changed may also generate duplicated frames as a temporal adaptation artifact. Both cases lead to an absence of motion in the first duplicated frame dragging down the descriptor to zero. But when there is an isolated duplicated frame, the user will not perceive the absence of motion in a scene. Thus, the descriptor should reflect what the user perceives. However, there is another possibility: the action suddenly stops so that a frame is replicated many times. In this

[4] P-frame is an abbreviation for Predicted-frame. They exist to improve compression by exploiting the temporal (over time) redundancy in a video. P-frames store only the difference in image from the frame (either an I-frame or P-frame) immediately preceding it (from Wikipedia).

case the user perceives absence of motion. Therefore, the descriptor should distinguish true absence of motion from single duplicated frames. This has been done skipping only the first duplicated frames so that a long series of duplicated frames is not skipped and a zero valued descriptor is returned.
In [9] a quantization step has been considered in order to compare the descriptor values with a ground truth obtained through a user study. Even if in this work a real valued feature is sought, it turned to be useful the threshold for the highest activity value. It has been used to limit the standard deviation values up to that threshold. This also enable to normalize the descriptor in the $[0, 1]$ range.
The final consideration regards the requirements for the videos to be analyzed. In order to make use of the quantization threshold reported in [9], it is required that the input compressed video stream adopts the MPEG-1 codec and that the frame size is 352×288. However, if one needs to extract the motion activity index from a video having a different codec or different resolution, the threshold can be linearly scaled. The only problem, for which a solution is not given in [9], occurs when a codec employs macroblocks with different sizes. In this case, it should be investigated first whether the values extracted from a MPEG-1 transcoded copy of the video resemble the values extracted from the original one in which every type of blocks is analyzed. If the values are not too different for a large set of videos, then the results given in the user study in [9] could be considered still valid.

Examples

Consider the sequences of frames in Figure 5.3. One should expect that in the frame interval 583-585 the descriptor presents values just above the zero. Then, when the black cat suddenly attacks the other one, the descriptor should peak. Finally, in the last group of frames, the action is slower than the previous one but not quite as the first; thus the value has to be neither 0 nor 1. This expectation is met as shown in Figure 5.4.



(a) frame 583 (b) frame 584 (c) frame 585

(d) frame 592 (e) frame 593 (f) frame 594

(g) frame 606 (h) frame 607 (i) frame 608

Figure 5.3: Example Frames

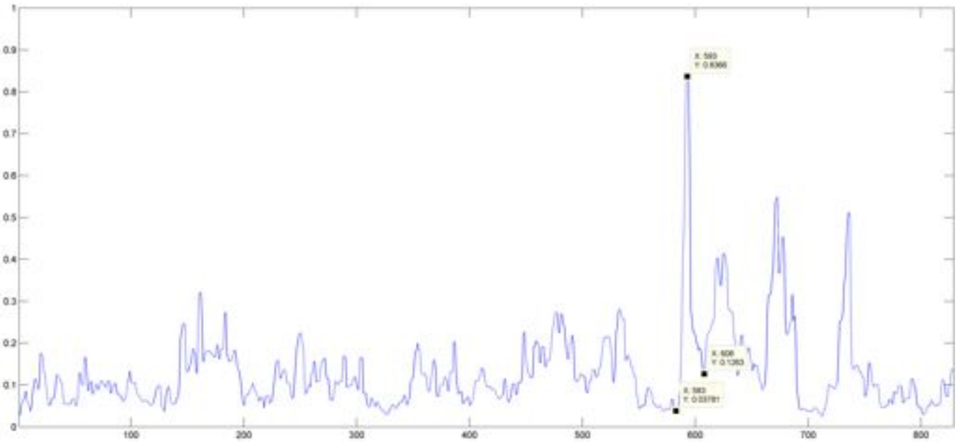


Figure 5.4: Motion Activity

5.2.2 Audio Novelty

As reported in [3], many musicological findings suggest that some time segments of audio may be more *salient* than others when making similarity judgments. The *temporal salience* is the perceptual importance attached to a particular time segment of audio. It can be estimated as a function of the spectro-temporal features. This information can be then incorporated into audio similarity measures and increase their correlation with human similarity when comparing relatively similar audio objects. The authors of [3] also report that three factors are usually associated to the salience: loudness, temporal proximity to onsets[5] and novelty.

Loud sounds are more likely to signal danger. Besides, humans place importance on the *attack*[6] when performing instrument identification and instrument similarity tasks. As for the latter, humans react and involuntarily attend to sounds that are novel more than the sounds that are not, where *novelty* refers to when the sound stimulus is "new or relatively rare in relation to the recent history of stimulation".

Among the aforementioned factors, novelty is the most suitable concept to delineate the way respondents in [12] described variations in the musical texture. For this motivation, it has been decided to devise an audio intensity feature using the audio novelty.

In [3] two different methods of estimating the novelty have been compared: the first is referred to as the Foote's novelty measure [7], while the second has been devised by the authors. It has been decided to adopt the first method because it is well-known and there is a public available implementation in the MIR Toolbox library.

The main idea behind the method described in [7] is to devise a function which peaks when there is a region with high self-similarity transitioning to a dissimilar region with high self-similarity. At such peaks, a listener would not expect a dissimilar region. Such function is computed as follows.

The first step is the computation of a *self-similarity matrix* like the one shown in Figure 5.5. This can be achieved through the following operations:

[5] Onsets are often associated to "transient" regions in the signal, a notion that leads to many definitions: a sudden burst of energy, a change in the short-time spectrum of the signal or in the statistical properties, etc. (from [1]).

[6] The attack of the note is the time interval during which the amplitude envelope increases (from [1]).

- the audio stream is segmented in overlapping frames for a short-time analysis in the frequency domain:
  - sampling rate 44.1 kHz;
  - frame size of 1024 samples (23 ms);
  - step size of 256 samples (25% overlap);
  - a Hanning window is applied;
- a constant-Q magnitude spectrogram [2] is computed for each frame so that a vector $\mathbf{v}_i$ is used as a representation of the i-th frame;
- each pair of frames represented as $\mathbf{v}_i$ and $\mathbf{v}_j$ is compared through a measure of (dis)similarity, such as the *cosine distance*:

$$D_C(i,j) = \frac{\mathbf{v}_i \bullet \mathbf{v}_j}{\| \mathbf{v}_i \|\| \mathbf{v}_j \|}$$

- the similarity matrix $\mathbf{S}$ is built so that the (i,j) element is $D_C(i,j)$.

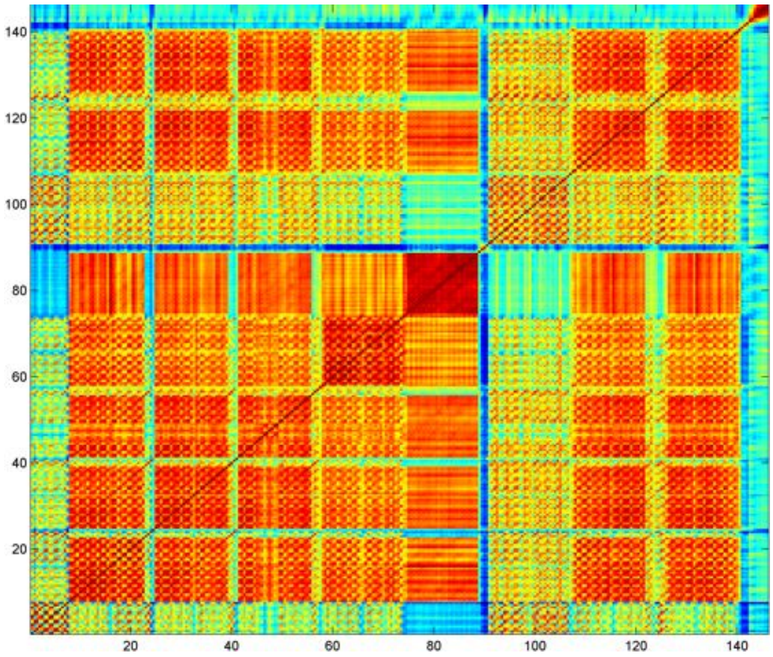


Figure 5.5: Audio Novelty, Similarity Matrix

Given that the cosine distance is symmetric, the similarity matrix $\mathbf{S}$ will also be symmetric. Furthermore, the elements on the diagonal are self-similarities measures; thus the associated value is always 1, which is the cosine between two identical vectors.

Given a self-similarity matrix $\mathbf{S}$, the novelty curve is computed by traversing the diagonal of $\mathbf{S}$ and considering a square sub-matrix $\mathbf{S}'_W$ centered on the current diagonal element. The index W in $\mathbf{S}'_W$ denotes the size of the sub-matrix. It is expressed in number of frames.

When the current frame of $\mathbf{S}$ lies between two dissimilar regions with high self-similarity, then the sub-matrix $\mathbf{S}'_W$ has a particular aspect. Imagine $\mathbf{S}'_W$ divided in four quadrants. Then, the first and the third quadrants will be bright, while the second and the fourth will be dark. In this, brightness is associated to cosine distance values near to 1, darkness is instead associated to values near to -1. This means that:

- the $W/2$ frames just before the current one are similar between themselves;
- the $W/2$ frames just after the current one are similar between themselves;
- the two sets of $W/2$ frames are dissimilar.

Thus, in order to measure to what extent a frame holds the aforementioned properties, the correlation between the sub-matrix $\mathbf{S}'_W$ and a kernel is computed. The kernel is devised so that the maximum value of correlation is reached when the three properties listed above are fully observed. Furthermore, smoothing is applied in order to avoid edge effects. In [7], it has been proposed the *Gaussian Checkerboard Kernel* which is shown in Figure 5.6.

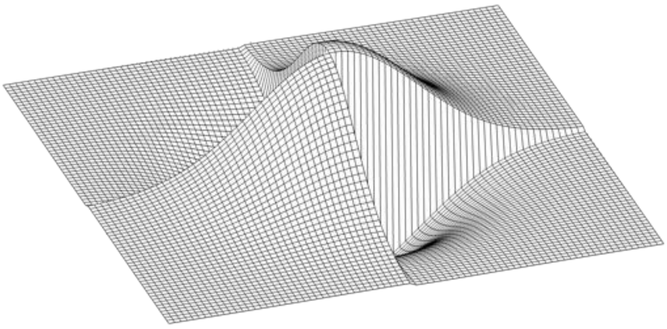


Figure 5.6: Audio Novelty, Gaussian Checkerboard Kernel

The parameter W, namely the width of the kernel, is important in order to detect changes at different time scales. Small kernels detect novelty on

a short time scale. Larger kernels average over short-time novelty, such as notes, and detect longer-term structure. In order to detect significant changes in the audio, it has been chosen to use a large kernel, namely the number of frames within a segment of one second. When peaks in the novelty function have to be associated to salient musical events, a large kernel is useful to reduce the number of false positives.
In summary, the computation process described above is exemplified in Figure 5.7:

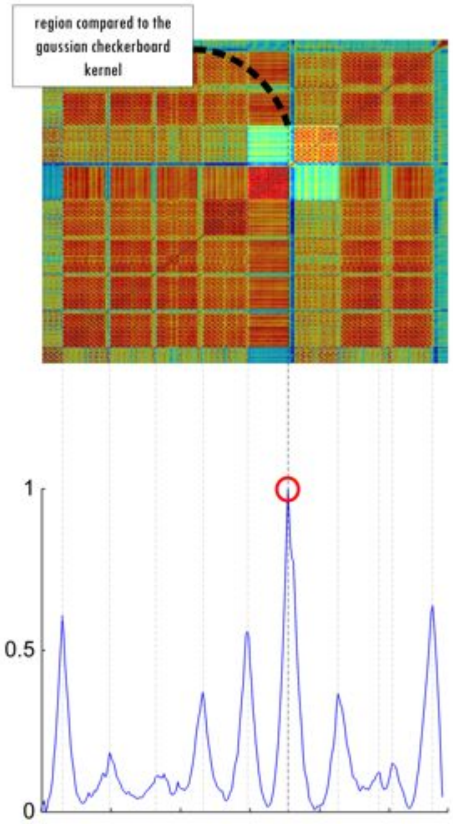


Figure 5.7: Audio Novelty, Change Point

5.2.3 Hybrid Feature: Audio Novelty and Onset Accents

A suitable feature approximating auditory intensity for the purposes of this project was proposed in [7], in which peaks in the audio novelty are used to segment an audio stream and to align video clips to the extracted audio segments. In doing this, the temporal dimension is analyzed to find boundaries in the audio stream, that is determining *when* a salient change occurs. Besides, the novelty amplitude dimension is used to select the highest peaks as the most important boundaries.

Apart from the time, accuracy in the amplitude dimension also is of relevance for the synchronization purposes. In order to ensure that the overall degree of synchronization match perceived by viewers is high, the novelty curve is combined with onset intensity information, which can be seen as a salience measure for occurring musical note events.

Before describing how the audio novelty is combined with onsets, the *onset accents* intensity feature is presented.

Onset Accents

The *onset* denotes the temporal instant in which the transient audio segment associated with the beginning of a note starts. It is commonly used in order to estimate the tempo in a music piece or to track the beat along the time. In [1], a series of techniques to detect onsets are presented. A possible choice is to measure changes in the spectrum along the time. Sudden changes are often associated to onsets. More in detail, one can adopt the *spectral flux* measure which is defined as follows:

Definition 5.12 (Spectral Flux)

Given two consecutive frames at time $n-1$ and n and their spectral representations $X_k(n-1)$ and $X_k(n)$, the spectral flux is defined as:

$$SF(n) = \sum_{-\frac{N}{2}}^{\frac{N}{2}-1} [H(\mid X_k(n) \mid - \mid X_k(n-1) \mid)]^2$$

where $H(x) = (x + \mid x \mid)/2$, i.e. zero for negative argument so that only those frequencies where there is an increase in energy are considered.

The time instants in which $SF(n)$ peaks are denoted as *onsets.* It is worth noting that the function $SF(n)$ does not only encode *when* an onset occurs. It also measures the *extent* to what an onset is relevant as the norm of a spectral power difference. This makes the spectral flux function a suitable intensity feature.

In Figure 5.8, an example of the onset accents intensity feature is shown.

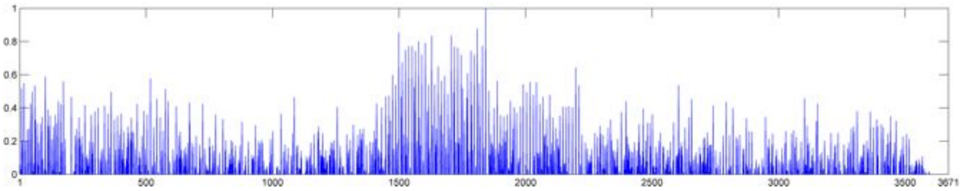


Figure 5.8: Onset Accents

Hybrid Onset-Novelty

The novelty curve is used first to find salient transitions (based on the amplitude, which will make sure that only peaks remain). Then, the novelty peaks are used as a mask for the onset accents function as follows:

- a new vector having the same size of the audio novelty vector is initialized;
- this vector is filled with a series of triangular windows:
  - placed in correspondence to a peak in the novelty vector;
  - the width is 0.3 s;
  - the amplitude is 1.

Such vector is finally multiplied element-by-element with the onset accents vector giving rise to the *hybrid onset-novelty* intensity feature. Combining this, salient sound changes (peaks from the novelty curve) at a salient musical event (strong note starts from the onset accent curve) will be reflected in the hybrid feature vector. In Figure 5.9, both the audio novelty, the onset accents and the hybrid intensity features are compared.

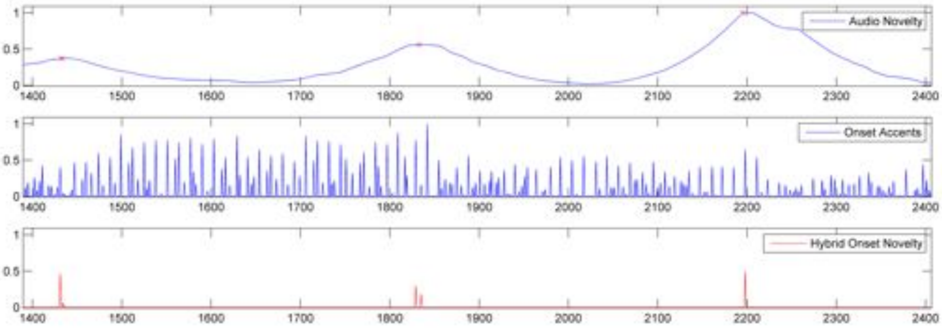


Figure 5.9: Hybrid Onset-Novelty Feature, Comparison

5.2.4 Motion Attention

In this section, a second experimental intensity feature for the visual stream analysis is presented. It has not been used for the audiovisual synchronization, but it should be investigated further for future developments. Thus, a description is given and some results are shown.
The Motion Activity descriptor presented in the Section 5.2.1 is not grounded on semantic analysis. Therefore, when salient events are sought, the descriptor is likely to suffer from false-positives and false-negatives. Unfortunately, no public ground truth had been found to thoroughly assess the Motion Activity as a salient event detector. Thus, it had been decided to look for a more semantic oriented descriptor in which the focus is put on moving objects. This would be subjectively compared to the Motion Activity.

Modeling the Motion Attention

In [14], the MPEG motion vector fields are again employed in order to devise a video skimming solution. Analyzing the motion, highlights are extracted from a video. In this, a motion attention measure and a saliency map are computed.
The main assumption in [14] is that the motion vector field has three types of *inductor*:

- intensity: motion energy, or activity;
- spatial coherence: spatial phase consistency of motion vectors within a spatial window;
- temporal coherence: temporal phase consistency of motion vectors within a temporal sliding window.

These inductors are mapped to values to be computed for each MPEG block (i, j) and each frame as detailed in the Definitions 5.13, 5.14 and 5.15.

Definition 5.13 (Motion Attention, Intensity Index)

$$\text{Intensity}(i,j) = \frac{\sqrt{dx_{i,j}^2 + dy_{i,j}^2}}{\max \text{Intensity}(i,j)}$$

Definition 5.14 (Motion Attention, Spatial Index)

$$\text{Spatial Coherence}(i, j) = -\sum_{t=1}^{n} p_S(t) \log(p_S(t))$$

where

- $p_S(t)$ is a probability distribution function defined as

$$p_S(t) = SH_{i,j}^{w}(t) / \sum_{k=1}^{n} SH_{i,h}^{w}(k)$$

- $SH_{i,j}^{w}(t)$ is the spatial phase histogram whose probability distribution function is $p_S(t)$;

- n is the number of bins in the spatial phase histogram;

- w is the number of blocks determining the size of the $w \times w$ spatial window.

Definition 5.15 (Motion Attention, Temporal Index)

$$\text{Temporal Coherence}(i, j) = -\sum_{t=1}^{n} p_T(t) \log(p_T(t))$$

where

- $p_T(t)$ is a probability distribution function defined as

$$p_T(t) = TH_{i,j}^{L}(t) / \sum_{k=1}^{n} TH_{i,h}^{L}(k)$$

- $TH_{i,j}^{w}(t)$ is the temporal phase histogram whose probability distribution function is $p_T(t)$;

- n is the number of bins in the temporal phase histogram;

- L is the size of the temporal sliding window.

In order to devise a measure of motion attention for each MPEG block, the following relationships between motion vectors and attended motions have been reported in [14]:

- the camera motion is able to give rise to high intensity, which is not the interest of human yet;

- the spatial phase consistency provides two cues:
  - phase of motion vectors in moving object tend to be consistent;
  - disordered phases and large magnitudes implies unreliable information;
- camera motion is always more stable than object motion during a longer time.

Therefore, the overall motion attention of each block in a frame is then computed as follows:

Definition 5.16 (Motion Attention Measure)

$$\begin{aligned}\text{Motion Attention}(i,j) = {} & \text{Intensity}(i,j)\times \\ & \text{Temporal Coherence}(i,j)\times \\ & (1 - \text{Intensity}(i,j) \times \text{Spatial Coherence}(i,j))\end{aligned}$$

The values computed through the Definition 5.16 are then used to detect regions of motion attention through the following image processing methods:

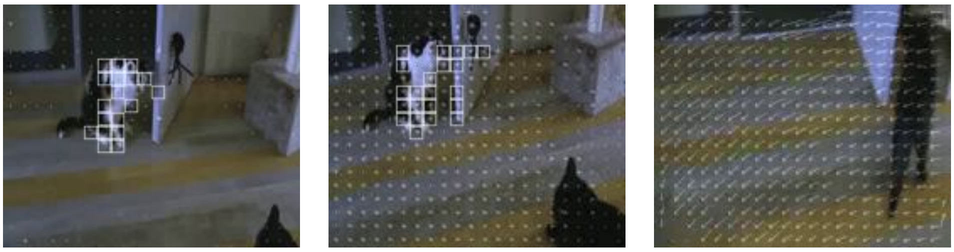
- histogram balance and median filtering;
- binarization, a threshold is used to decide whether a block is salient;
- region growing and selection.

Finally, given a set of salient regions, an index of motion attention is computed as the average intensity of the blocks belonging to the regions. The series of motion attention indexes is used as an intensity feature to describe a video stream along the time dimension.

Examples and Implementation Details

In the following examples, some video frames are shown. The MPEG motion vectors and the salient blocks are also shown. The latter, are highlighted with a white border.

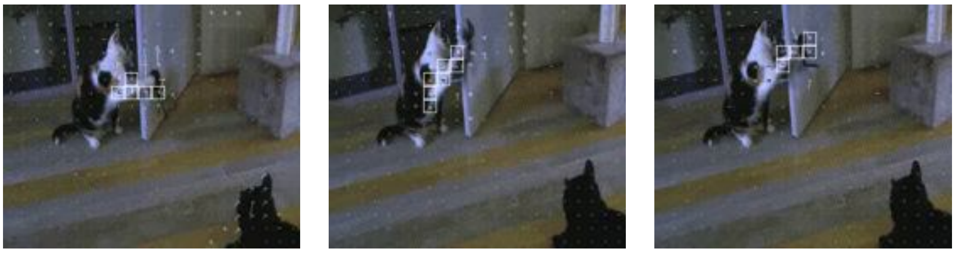
The set of frames in Figure 5.10 shows that the method described in [14] effectively get rid of camera motion:



(a) frame 32 (b) frame 48 (c) frame 738

Figure 5.10: Saliency Detection, Behavior With and Without Camera Motion

The sequence in the Figure 5.11 proves that the method implicitly has tracking capabilities:



(a) frame 60 (b) frame 69 (c) frame 70

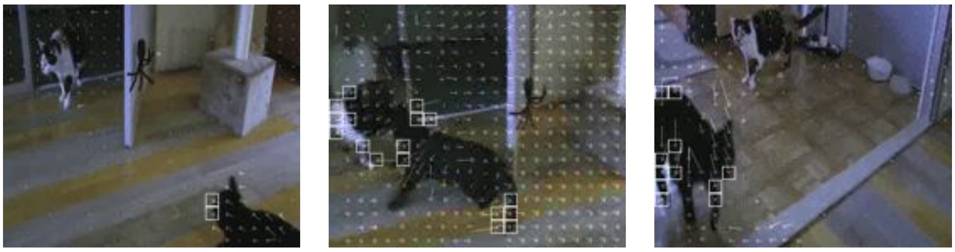
Figure 5.11: Saliency Detection, Tracking

The following example is reported in order to illustrate a problem to be tackled in the future. During the implementation of the Motion Attention descriptor, it has been assessed each intermediate result contributing to the final index computation. This has been useful in order to tune the parameters, such as the number of bins of the phase histograms and the sizes for the spatial and the temporal windows. Besides, the behavior of the descriptor has been tested on videos containing different motion patterns (e.g. pure camera motion, pure object motion, or both).

Looking at the outcomes, it has been decided to add a new factor in the Definition 5.16. This additional factor measures the entropy of the MPEG luma values within a block. Those blocks which contain *interest points*[7] are

[7] An interest point is a point in an image which has a well-defined position and can be robustly detected. It can be a corner, an isolated point of local intensity maximum or minimum, line endings, or a point on a curve where the curvature is locally maximal

likely to score high in terms of luma entropy.
Unfortunately, this method presents a drawback: when a block has a single luma value, the overall score is zero even if the block belongs to a moving object. However, from a subjective evaluation conducted by the author on a small set of videos, the aforementioned factor seemed to be globally beneficial: the precision increased while the recall slightly decreased. In Figure 5.12, a set of frames extracted from the same scene shows a black cat running. As anticipated, only those blocks with high luma entropy are detected as salient.



(a) frame 587 (b) frame 615 (c) frame 669

Figure 5.12: Saliency Detection, Luma Entropy Issue

The plot shown in Figure 5.13 is reported to motivate why the Motion Activity descriptor has been preferred over the Motion Attention. Both descriptors have been extracted as intensity features and plotted together.

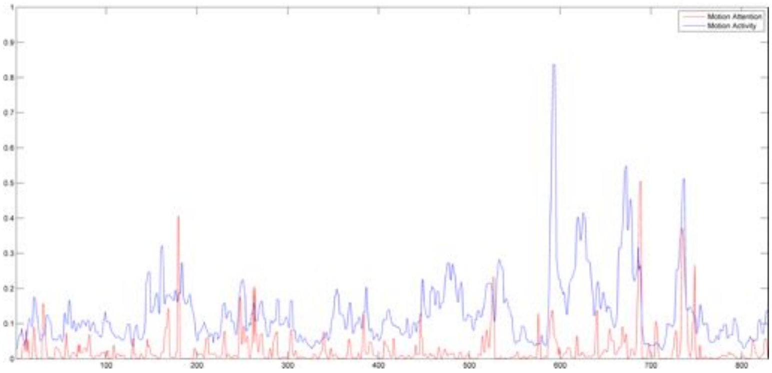


Figure 5.13: Motion Attention vs. Motion Activity

Recalling the example given in the Figure 5.3, one should notice that the most important peak at the frame 593 is not detected by the Motion Atten-

(from Wikipedia).

tion descriptor. The lack of recall may lead to weak audiovisual links, thus it has been decided to conduct the first synchronization experiments using the Motion Activity descriptor.

Finally, to implement the Motion Attention descriptor a first duplicate frame detector has been again employed as discussed in the section 5.2.1 for the Motion Attention descriptor.

5.3 Cross-Modal Accents Alignment

Up to this point, both the audiovisual synchronization framework and audiovisual intensity features have been presented. In this section, it is shown the architecture of a working system which employs the aforementioned components.

The following intensity features have been selected: the *motion activity* feature for the visual analysis and the *hybrid onset-novelty* feature for the audio analysis. When the system is deployed, a local repository of production music has to be built. Each item in such collection consists of a compressed audio file (mp3 format) and a data file containing the values of the hybrid onset-novelty intensity feature vector. Having a local repository of music enables to compute the auditory intensity features in advance so that the time required to answer to a user is reduced.

In Figure 5.14, the visual intensity features aligned to the lagged auditory intensity feature are shown. In this example, it is evident that the highest peaks in both modalities are well aligned.

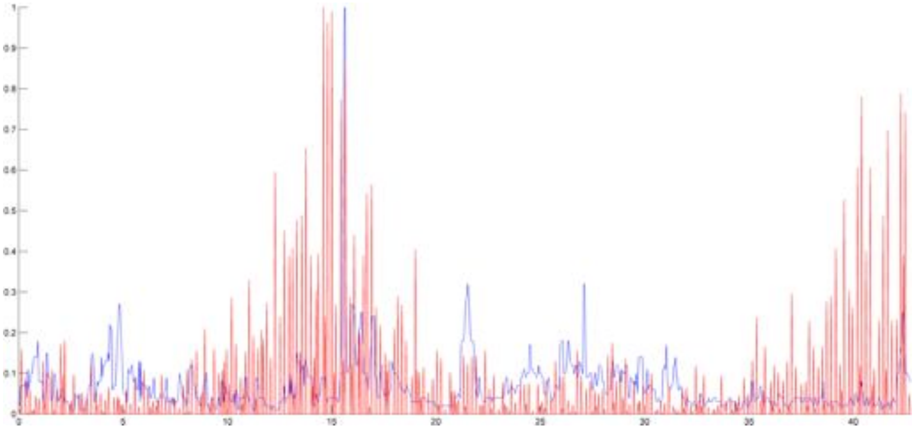


Figure 5.14: Example of Aligned Streams

One may wonder whether the synchronization scores can well discriminate strong and weak alignments. Let us assume that the synchronization score reflects the perceived synchronization match of human subjects. Then, consider an audiovisual pair and compute the synchronization scores vector **w**. Plotting the frequencies of the values in **w**, a histogram like the one in Figure 5.15 is obtained. The best synchronization score always lies in the right most bin. When the other scores are concentrated in a group of consecutive bins and this group is far from the right most bin, then one can infer that the discrimination power is high. Otherwise, if many synchronization scores are concentrated in the right most bins, it can happen that a large group of possible alignments are perceived as equivalent. For instance, this happen when an intensity feature contains a periodic series of peaks whose frequency is high.

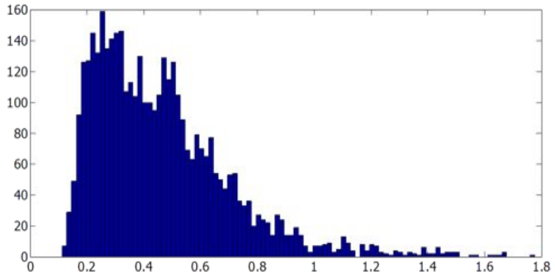


Figure 5.15: Distribution of the Synchronization Scores

The full synchronization pipeline is shown in Figure 5.16.

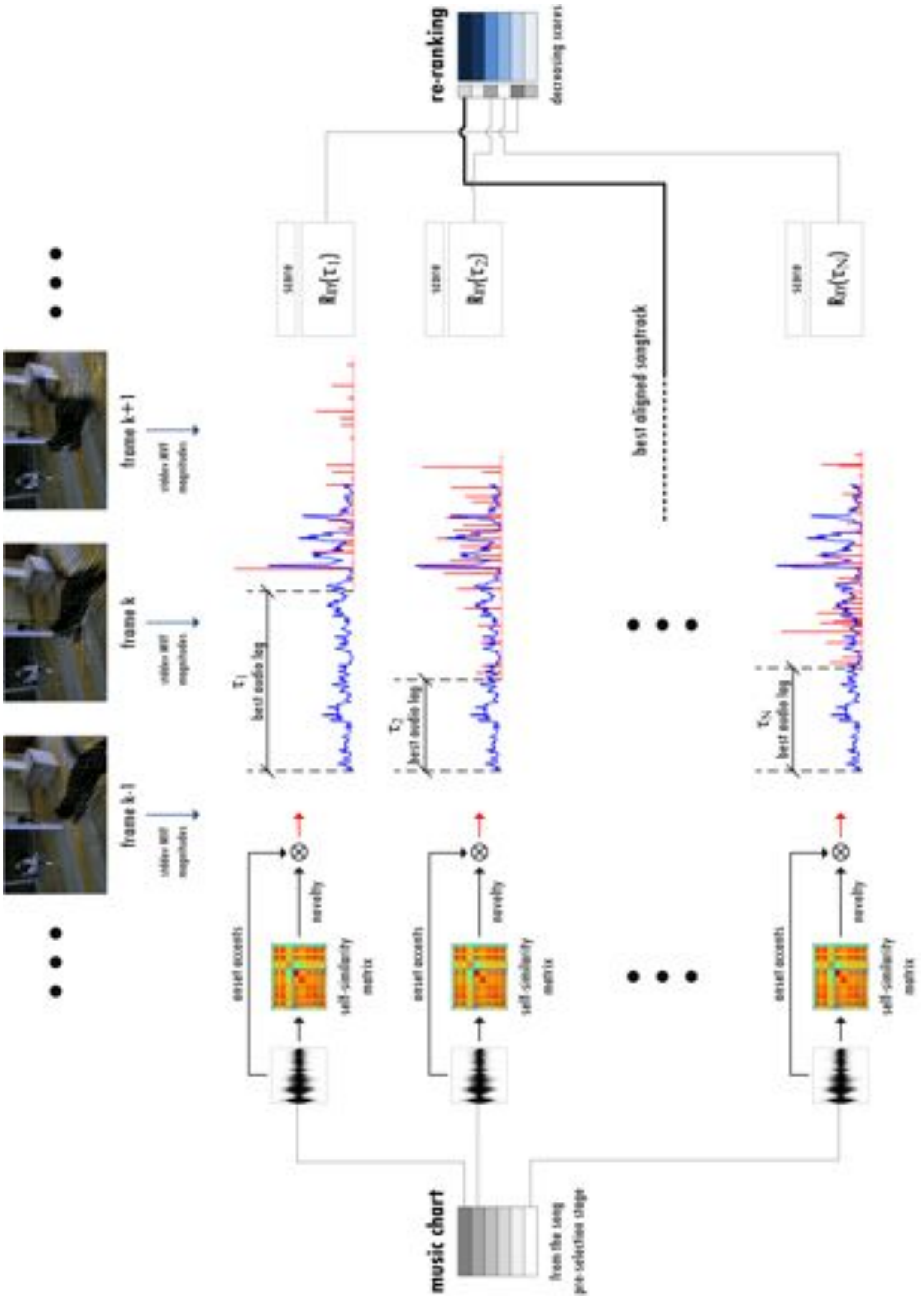


Figure 5.16: Audiovisual Synchronization Pipeline

Chapter 6

Towards a Full System Evaluation

This thesis project includes all the activities described in the previous chapters, namely the analysis stage, the system design and its implementation. In this chapter, a design for a full system evaluation is outlined.

6.1 What Makes the Evaluation Tricky

The problem of automatically generating music videos cannot be formulated as a classical information retrieval (IR) problem in which documents are retrieved given a query. Therefore, well-known IR measures such as the precision, the recall or the F-measure cannot be directly used. Similarly, comparing the system output to a ground truth might not be straightforward.

Apart from adopting objective evaluation methods, subjective assessments can also be considered. In all of the works mentioned in Section 2.1, such kind of assessment has been used. In this, the authors had to accurately setup their surveys. For instance, the system output is compared to one or more reference objects which can be a result of a baseline method. Scales of suitability are devised in order to evaluate the system in terms of average score and standard deviation. In order to reach statistical significance and avoid user-bias issues, the number of subjects involved in the subjective evaluation should be the highest possible. Therefore, instead of "manually" recruiting the respondents, a survey could be carried out through a crowdsourcing framework, e.g. Amazon MTurk. But accessing to a large number of respondents turns to be disadvantageous when the rate of spam is high[1]. All the aforementioned factors suggest that the evaluation of the automatic music video generation system presented in this thesis should be thoroughly designed.

6.2 Assess each Layer First

The system presented in this thesis includes a number of different problems, namely:

- given a user entry in the form of text, music has to be pre-selected from a local repository;
- given a set of soundtracks, each soundtrack has to be synchronized to a given video, in this:
  - each soundtrack has to be analyzed in order to detect salient auditory events;
  - the video has to be analyzed in order to detect salient visual events;
- synchronized videos have to be ranked according to the degree of synchronization match.

Being independent problems, each individual component can be assessed separately. Furthermore, without an intermediate evaluation of the system, it would be useless to run a full evaluation.

6.2.1 Music Pre-Selection

The music pre-selection problem can be seen as a ranked search problem in which a query is given and the top-k documents are retrieved. Returned documents can be judged as relevant or non-relevant. This would suggest to adopt the precision and the recall as scores. But it is desirable to also

[1] When a user survey respondent has not been "manually" recruited, it can happen that he may only be interested in completing as many surveys as possible rather than taking care of the quality of his answers. This scenario usually leads to the submissions of spam answers.

consider the order in which the returned documents are presented. Then, the *Mean Average Precision* (MAP) score can be used to evaluate the music pre-selection component. It is a well-known IR performance measure which indeed takes into account the ranking position of a retrieved document.

In order to compute the MAP score, a ground truth is required. It can be built as follows:

- a set of queries is collected;
- for each query, each document in the collection is labeled as relevant or non-relevant with respect to the query;
- for each query, the ranked set of retrieved documents is stored.

Therefore, in the specific case of the music pre-selection component, it is necessary to build a ground truth as described above. Unfortunately, it is not straightforward because, even for a small collection of music (e.g. 50 items), it takes too much time for a subject to judge whether each music piece is relevant with respect to the query.

Even if other strategies may be considered in order to efficiently build the ground truth, it could be better to first assess the music pre-selection algorithm looking for undesired behaviors. For instance, one could check whether *hubness* occurs. This phenomenon occurs when the algorithm tends to always answer with a small subset of the indexed music. Given a large set of queries, this could be easily verified plotting the number of times that a music item is retrieved.

6.2.2 Audiovisual Content Analysis

Both the audio and the videos streams are analyzed as intensity features (see Definition 5.4). As already explained, an intensity feature encodes two aspects of the audiovisual content: *when* a salient event occurs and the *extent* to which is relevant.

Therefore, the audiovisual content analysis can be seen as a classification problem in which events are salient or non-salient and salient events can be judged as slightly relevant, relevant of very relevant.

A ground truth can be obtained asking to the subjects to add temporal marks in correspondence of salient events; in addition they can specify a label to express the grade of relevancy. Once the data are collected, two assessments can be done. The first one assesses the precision and the recall of the algorithm when seen as a salient event detector. The second one measures the extent to what the amplitude dimension of an intensity feature reflects the relevancy perceived by human subjects. For instance, this can be done considering the three grades of relevancy as labels and defining the thresholds for the intensity features so that labels can be compared and a confusion matrix can be computed.

6.2.3 Audiovisual Synchronization

In order to solve the best synchronization problem of the Definition 5.1, a set of synchronization scores is computed. This set can be used to make a first ranking: given a video and a soundtrack, the set of feasible audio lags is ranked by the synchronization score associated to each lag. The top ranked audio lag is defined as the best audio lag. Then, the Algorithm 5.3 makes a second ranking in the following way: given a video and a set of soundtracks, the set of synchronized audiovisual pairs is ranked by the synchronization score associated to the best audio lag.

Recalling the layout of the synchronization framework presented in the Section 5.1, the first ranking can be associated to the problem solved in the inner layer, while the second ranking can be associated to the problem solved in the outer layer.

Being ranking problems, one could again use the MAP score. In this case, it is necessary to make clear what a query and a document are for each ranking problem. As for the former, the query is the audiovisual pair and a document is a synchronized video in which a specific audio lag has been used to align the music to the video. While in the latter case, the query consists of a video and a set of soundtracks, or a set of audiovisual pairs in which the video stream is always the same. The retrieved documents are the synchronized videos in which the best audio lags have been used to align the streams.

A ground truth for the first ranking problem can be built as follows:

- a set of queries consisting of audiovisual pairs is collected;
- for each query:
  - the top-k audio lags are extracted from the computed synchro-

    nization scores vector[2];
  - the audiovisual pair is synchronized k times considering the top-k audio lags;
  - each synchronized video is judged from a subject as well-synchronized or weakly-synchronized, that is to say relevant or non-relevant.

The choice of considering only the top-k audio lags instead of any feasible audio lag is motivated by the following facts. Considering the whole set of feasible audio lags would lead to a huge number of synchronized videos to be judged (e.g. the number of feasible lags for a video of 4 minutes at 30 fps and a music track of 5 minutes is about 20000). Therefore, a criterion to reduce the number of synchronized videos is required. The typical pattern for the distribution of the synchronization scores has been shown in Figure 5.15: the best audio lag lies in the most right bin and does not belong to the part in which the majority of synchronization scores fall. Considering the top-k audio lags leads to including the best audio lag and other $k - 1$ lags; the latter are quite similar in terms of synchronization scores. Thus, a comparison between the best solution and a random solution is achieved. Similarly, a ground truth for the second ranking problem can be built as follows:

- a set of queries consisting of a video and a set of n soundtracks is collected;
- for each query:
  - each audiovisual pair is synchronized using the best audio lag;
  - a subject select the 3 out of n best synchronized videos, that is to say 3 relevant documents are chosen.

In both cases, the MAP score should be a useful way to assess whether the audiovisual synchronization and the synchronization score based ranking are effective.

6.3 Full System Evaluation

When the individual components are assessed and the effectiveness is proved by experimental results, the whole system could be subjectively evaluated.

[2] The synchronization scores vector is defined in the Definition 5.11.

For instance, a survey can be carried out in which respondents have to judge to what extent the system fulfilled their expectations. A respondent could report whether the output videos are better than the uploaded video and why. In this, a motivation field could help in understanding whether the techniques used in the system to select suitable music and to synchronize it to the video are effective.

Chapter 7

Conclusions

In this thesis project, the problem of automatically adding music to user generated videos has been addressed. Suitable music and suitable synchronizations can make such videos more attractive for sharing on the Web. Two scientific challenges have been identified in this work. Firstly, a number of soundtracks have to be selected from a large collection of music. Then, the selected music pieces have to be synchronized with the user generated video. The best synchronized videos are finally shown to the user.

As for the first challenge, state-of-the-art approaches are mainly based on the idea to select music according to audiovisual contents match criteria. Even if different approaches have been proposed, the researchers always have considered the user generated video as the unique available information to make a choice. Inspired by musicology and psychology insights, and supported by the outcomes of a crowdsourcing experiment, in this work the idea of uncovering the user intent has been used to enrich the set of input data through which suitable music is retrieved.
The second challenge, namely the video to music synchronization, has been the main focus in this thesis project. Two contributions have been given. Audiovisual features have been devised in order to detect salient musical and visual events which are used as anchor points. As it has been suggested by the authors of previous works, new descriptors, which are more semantic oriented than common low level features, have been sought. The idea of looking at the visual motion and the audio novelty is also based on how musical and cinematic plot descriptions have been reported in the

aforementioned crowdsourcing experiment. The second contribute is given by the definition of an audiovisual synchronization framework. Thanks to this framework, many different techniques to analyze and synchronize audiovisual streams can be used and compared.

Details regarding the evaluation of the system are given in the form of design proposals. The problems associated to each individual component have been restated as an isolated retrieval problem so that well-known evaluation measures can be used. As for the whole system, some general ideas to carry out a subjective evaluation has been reported.
Apart from evaluating the system, it has been planned to improve some components. This should be done before a full system assessment is carried out. The motivation id that it might be hard repeated more than once the full evaluation. Some possible improvements are reported below.
When music is retrieved by the user text entries, term boosting can be applied to the plot entry so that each word is weighted. For instance, adjectives and names are first found via Part-of-Speech tagging and then the names in the entry are boosted looking at the number of associated adjectives. This should be useful to put the focus on the most relevant words.
The idea of considering the user intent could be also extended to the audiovisual synchronization component. One could ask to the user to select a specific object in the video (e.g. drawing a bounding box). The object is tracked along the time and its velocity is estimated. The velocity is finally used as a visual intensity feature. As for the music, a similar hint can be asked to the user. Given a music track, a user could highlight a short segment which he considers salient. Through audio self-similarity techniques based on timbral frame representations, those parts which are similar to the highlighted audio segment are detected. The values of an auditory intensity feature vector are then boosted in correspondence of such similar parts.

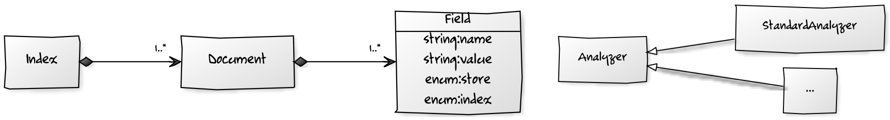
Appendix A

Lucene - Full-Text Search Engine

Lucene[1] is a full-text search engine library adopted for text indexing and searching. It has been used in the implementation of the *Story-Driven Soundtrack Pre-Selection* pipeline presented in Chapter 4. In particular, it has been exploited its feature to rank the search results. In this appendix, details regarding indexing, retrieval and ranked search through Lucene are briefly presented.

A.1 Lucene Framework

The Lucene framework essentially consists of *entities* and *analyzers* which are reported together with their relationships in Figure A.1:



(a) entities

(b) analyzers

Figure A.1: Lucene framework

In the Figure A.1a, the `Index` is the collection of indexed `Documents` which are described by a series of `Fields`. Each Field has a *name*, a *value* and a series of options. For instance, one can choose whether, and eventually how,

[1] `http://lucene.apache.org/core/`

a field has to be stored and/or indexed.
The Figure A.1b refers to the text analyzers which have to implement the interface `Analyzer`. A number of default analyzers are already implemented such as the `StandardAnalyzer`. Analyzers define the way the text is tokenized, which filters are employed (e.g. stopwords removal, stemming) and the recognition of particular types of text (e.g. email address).

A.2 Indexing

At the indexing stage, an index is built instancing the `Index` class and adding to it instances of the `Document` class. The document is the indexed/retrieved unit. In order to index the content, a suitable text analyzer must be specified: for instance the `StandardAnalyzer` class removes stop words and recognize email addresses and acronyms in contrast to the `WhitespaceAnalyzer` class which splits tokens at whitespace. The sequence diagram in Figure A.2 resumes the index building procedure:

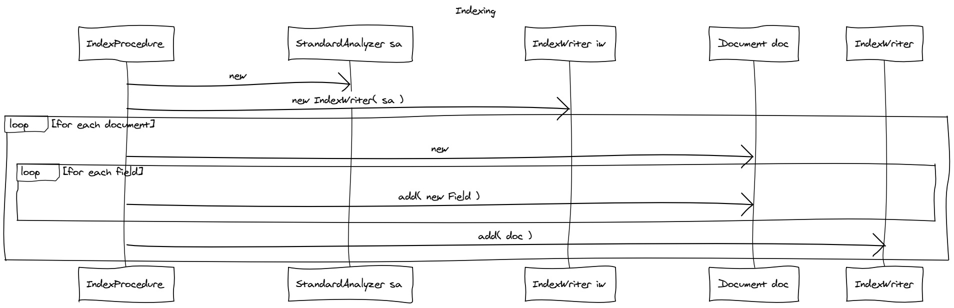


Figure A.2: Indexing with Lucene

In the indexes presented in Chapter 4, two fields have been defined: the *key* and the *indexed content*. The former is stored but not indexed, while the latter is only indexed. This allows to compare the query to the *indexed content* field. The document retrieved through this comparison are used to collect their value in the *key* field. The set of keys is returned as answer set.

A.3 Retrieval

The sequence diagram of the retrieval stage is shown in Figure A.3. The index in which one would search documents is accessed through an instance

of the `IndexSearcher` class. A pointer to the index has to be passed in the constructor as a `Directory` object.
The appropriate analyzer has to be instantiated; it has to be the same used at the indexing time (e.g. `StandardAnalyzer`). This is used to instance the `QueryParser` class which parses the textual query through the `QueryParser::parse()` method. An instance of `Query` is then returned.
Providing the query and an integer specifying the amount of elements to be retrieved to the `IndexSearcher::Search()` method, results are returned as a `TopDocs` instance. This object allows the iteration of the ordered set of documents ranked by computed scores.

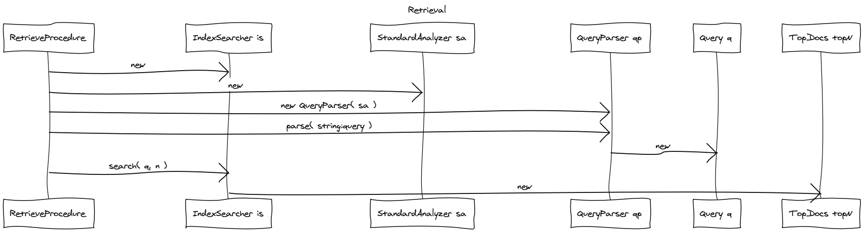


Figure A.3: Retrieval with Lucene

A.4 Ranked Search

The last aspect discussed in this appendix regards the internal documents ranking in Lucene. The predefined score is the well-known measure called *length-normalized TF-IDF* in which the main contribute is given by the following factors[2]:

- **TF**: Term Frequency, how often a term appears in a document
  implication: the more frequent a term occurs in a document, the greater its score
  rationale: documents which contains more of a term are generally more relevant

- **IDF**: Inverse Document Frequency, how often a term appears across the index

[2] This section is strongly inspired by the Kelvin Tan's tutorial available at `http://www.lucenetutorial.com/advanced-topics/scoring.html`; the official Lucene documentation section regarding document similarity is available at `http://lucene.apache.org/core/3_6_0/api/all/org/apache/lucene/search/Similarity.html`.

implication: the greater the occurrence of a term in different documents, the lower its score
rationale: common terms are less important than uncommon ones

- **Coord**: number of terms in the query that were found in the document
implication: a document that contains more terms in the query will have a higher score
rationale: self-explanatory

- **LengthNorm**: measure of the importance of a term according to the total number of terms in the field
implication: a term matched in fields with less terms have a higher score
rationale: a term in a field with less terms is more important than one with more

In the end, the score is defined as follows (to the exclusion of other factors irrelevant in this context):

Definition A.1 (Lucene Scoring Function approximation)
$\text{score}(q, d) = \text{Coord}(q, d) \times \sum\limits_{t \in q}(\text{TF}(t \in d) \times \text{IDF}(t) \times \text{LengthNorm}(t \in d))$
where q is the query, d is the document and t is a term

Further details about the Lucene framework and two comprehensive tutorials are available at `http://www.lucenetutorial.com/` and `http://tinyurl.com/IBM-lucene-tutorial`[3].

[3] `http://www.ibm.com/developerworks/opensource/library/os-apache-lucenesearch/` (full url).

Bibliography

[1] Bello, J., Daudet, L., Abdallah, S., Duxbury, C., Davies, M., and Sandler, M. A Tutorial on Onset Detection in Music Signals, 2005. *Speech and Audio Processing, IEEE Transactions on 13*, 5 (2005), 1035–1047.

[2] Brown, J. Calculation of a Constant Q Spectral Transform, 1991. *The Journal of the Acoustical Society of America 89* (1991), 425.

[3] Cartwright, M., and Pardo, B. Novelty Measures as Cues for Temporal Salience in Audio Similarity, 2012.

[4] Cattuto, C., Benz, D., Hotho, A., and Stumme, G. Semantic Grounding of Tag Relatedness in Social Bookmarking Systems. In *The Semantic Web - ISWC 2008*, A. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. Finin, and K. Thirunarayan, Eds., vol. 5318 of *Lecture Notes in Computer Science*. Springer Berlin, Heidelberg, 2008, pp. 615–631.

[5] Cohen, A. J. How Music Influences the Interpretation of Film and Video: Approaches from Experimental Psychology, 2005. *Selected Reports in Ethnomusicology 12*, Special Issue in Systematic Musicology. R.A. Kendall & R.W. Savage, Eds. (2005), 15–36.

[6] Feng, J., Ni, B., and Yan, S. Auto-generation of Professional Background Music for Home-made Videos. In *Proceedings of the Second International Conference on Internet Multimedia Computing and Service* (2010), ACM, pp. 15–18.

[7] Foote, J., Cooper, M., and Girgensohn, A. Creating Music Videos using Automatic Media Analysis, 2002. *Proceedings of the tenth ACM international conference on Multimedia* (2002), 553.

[8] Hua, X., Lu, L., and Zhang, H. Automatic Music Video Generation based on Temporal Pattern Analysis. In *Proceedings of the 12th annual ACM international conference on Multimedia* (2004), ACM, pp. 472–475.

[9] Jeannin, S., and Divakaran, A. MPEG-7 Visual Motion Descriptors, 2001. *Circuits and Systems for Video Technology, IEEE Transactions on 11*, 6 (2001), 720–724.

[10] Jensenius, A. Motion-sound Interaction Using Sonification based on Motiongrams, 2012. *ACHI 2012, The Fifth International Conference on*, c (2012), 170–175.

[11] Liao, C., Wang, P. P., and Zhang, Y. Mining Association Patterns between Music and Video Clips in Professional MTV, 2009. *Advances in Multimedia Modeling* (2009), 401–412.

[12] Liem, C. C. S., Larson, M., and Hanjalic, A. When Music Makes a Scene - Characterizing Music in Multimedia Contexts via User Scene Descriptions, 2012. *International Journal of Multimedia Information Retrieval* (2012).

[13] Lissa, Z. *Ästhetik der Filmmusik.* Henschel, Berlin, 1965.

[14] Ma, Y. A Model of Motion Attention for Video Skimming, 2002. *Image Processing. 2002. Proceedings.*, 100080 (2002), 129–132.

[15] Mattek, A., and Casey, M. Crossmodal Aesthetics from a Feature Extraction Perspective: a pilot study, 2011. *Traditional Music*, Ismir (2011), 585–590.

[16] Moody, N., Fells, N., and Bailey, N. Motion as the connection between audio and visuals. 2006.

[17] sheng Hua, X., Lu, L., and jiang Zhang, H. Optimization-based Automated Home Video Editing System, 2004. *IEEE Trans. on Circuit and System for Video Technology 14* (2004), 572–583.

[18] Stupar, A., and Michel, S. PICASSO To Sing you must Close Your Eyes and Draw - Categories and Subject Descriptors. In *CIKM.* 2011, pp. 2589–2592.

[19] Tagg, P., and Clarida, B. *Ten Little Title Tunes — Towards a Musicology of the Mass Media.* The Mass Media Scholar's Press, New York, USA and Montreal, Canada, 2003.

[20] Vatakis, A., and Spence, C. Audiovisual Synchrony Perception for Music, Speech, and Object Actions., Sept. 2006. *Brain research 1111*, 1 (Sept. 2006), 134–42.

[21] Wartena, C., Brussee, R., and Wibbels, M. Using Tag Co-occurrence for Recommendation. In *Intelligent Systems Design and Applications, 2009. ISDA09. Ninth International Conference on* (2009), IEEE, pp. 273–278.

Acknowledgments

I would like to thank first Beppone, a.k.a. Giuseppe Serra, and Iris. I used to say that I would have never gone abroad in my life. But thanks to them, I changed my mind even if, in the beginning, I was quite scared. Thus, I can say that they contributed to what happened to me in the last two years, in no small way. They encouraged me to take the initiative.
In particular, Iris had to endure my endless talk about Delft, and my preparations for the trip. Since my departure, she had to listen to my funny Dutch stories without cringing. She also had to drag my luggage for 3 km when Delft was completely frozen.

I also have to apologize to my parents. I have to apologize to them for all the times I stood them up for dinner, and for all the promises I made to go out with them, and failed to keep. I would also like to express my gratitude to them for all the help and assistance they gave me.

I would like to thank Prof. Alberto Del Bimbo, who firmly endorsed my project, and Prof. Alan Hanjalic for accepting me as a visiting student in the D-MIR lab. I am grateful for all the invaluable advice I received from both of them.

My special thanks go to Cynthia Liem. You have been my mentor. You welcomed me in the D-MIR lab. You were very patient with me. You made me feel welcome in the lab by asking me out to join the others for lunch (even if I used to have my Italian lunch in my place). You assisted my process of *dutchification.* You taught me a lot of new amazing things, and you helped me write this thesis. You always found time for my dummy questions.

During my stay in Delft, I had the fortune to share the desk with Cynthia and Stevan. Stevan is my second mentor, he also welcomed me in his lab. I became addicted to his weird songs, e.g. Euro Neuro, and from him I learnt everything there is to learn about the Turkish guy called Baba. But the thing I miss the most are Alessandro and Stevan arguing. They were friends but they hardly ever agreed on anything. Every day, after lunch, Cynthia and I would join their discussions about the *dutchified* Italian recipes or any other topic, in an effort to resolve their disagreements. I am really grateful to Stevan and Alessandro for all the help they gave me and all the fun we had together during my stay in Delft.

Finally, I would like to thank all the others from the D-MIR lab: Martha, Raynor, Christoph, Carsten, Yue Shi, Xu Peng, Xinchao and last but not least, Saskia. It was amazing working with you and taking part to the lab meetings.

I also would like to thank Prof. Richard Heusdens and Richard Hendriks: your lectures were really amazing. I really enjoyed your course.

I am especially grateful to Francesco Maresca. He was the man behind the scene, who made me feel at home in The Netherlands. He welcomed me and Iris at the airport when we arrived in the middle of a snow storm in Schiphol. He sang funny Dutch songs at my birthday party, e.g. “Cowboy Bazzicone”. He used to bring his iPad to Mass so that I could understand at least some of it. He used to visit me in Delft even if it involved him coming to see me on the days before he had meetings at 9 am in Eindhoven.

I would like to thank Gabriëlle, Therese, Pieter, Kenny, Tom, Stan, Remco, Hanneke, Michiel and Giacomo. Thanks to our friendship, to the School of Community in Utrecht and thanks to all the time we spent together, I felt at ease even though I was far from home.
Finally I am especially thankful to Paolo and Irene, who were my Italian friends in Delft.

I would now like to thank a few Italians.

During my studies, I could always rely on some valuable friends. Let me say thank you to: Padre Filippo, Andrea Gamannossi, Alessandro Mancini, Beppone, Beppino, Lamberto, Marco Fontani, Anna Grasso, Andrea Tassi, Giovanni Fabbri, Alessio Vannucci, Lorenzo Usai, Georgios Nikiteas, Irene Pianigiani. And let me also thank my dear mates Luca Del Tongo, Dario D'Amico and Lorenzo Paladini.

I also would like to thank Francesco Bracci. Even if you are not here with us anymore, the memory of your being so curious and so full of wonderment still lingers.

My special thanks go to Prof. Giuseppe Modica. You have always encouraged me to face up to the difficulties and to press ahead. If I had not listened to you, I would have never written this thesis.

I am grateful to the Costanzo's family, who have been my second home. Iris, Angela, Maria Luisa and Gne Gne have been bearing up with me for years.

It is now your turn, Vickie. You have been taking care of my English for the past two years. If it had not been for your tuition, I would not have been eligible to apply for the Erasmus program in the first place. Since we are on the subject, I would like to thank Pio who shared with me the suffering of taking the IELTS exams.

A big thank you to my colleagues at Polimoda. In particular I would like to thank Fabrizio, Stefano, Simone and the new entry Lucia. A special thank you to Fabrizio who helped me in making several important choices.

Finally, I would like to thank the reviewers of this manuscript. They are: Cynthia Liem, Victoria Maximova, Lamberto Ballan, Marco Fontani, Luca Del Tongo, Dario D'Amico. Your contributions have been invaluable!