D4.3: Media Analysis Tools - Report - Version B

2015-05-01 – 2016-07-31

Project ref. no.	FP7-ICT-2013-7 - 610691
Project acronym	BRIDGET
Start date of project (duration)	1 November, 2013 (36 months)
Document due Date:	2016-07-31
Actual date of delivery	2016-07-31
Leader of this document	Stavros Paschalakis
Reply to	
Document status	Final

Deliverable Identification Sheet

Project ref. no.	FP7-ICT-2013- 610691
Project acronym	BRIDGET
Project full title	BRIDging the Gap for Enhanced broadcasT
Document name	BRIDGET D4.3 v1.2.docx
Security (distribution level)	PU
Contractual date of delivery	2016-07-31
Actual date of delivery	2016-07-31
Document number	
Type	R
Status & version	1.2
Number of pages	35
WP / Task responsible	WP4
Other contributors	Massimo Balestri, Miroslaw Bober, Gianluca Francini, Syed Husain, Skjalg Lepsoy, Alberto Messina, Maurizio Montagnuolo, Karol Wnukowicz
Author(s)	Stavros Paschalakis
Project Officer	Alberto Rabbachin
Abstract	Deliverable D4.3 is the public document outlining Version B of BRIDGET's Media Analysis Tools. This document describes the progress and main results of the research conducted within WP4 from 2015-05-01 to 2016-07-31.
Keywords	Video segmentation, shot detection, scene detection, face detection, face clustering, MPEG-7, scene classification, object classification
Sent to peer reviewer	2016-07-30
Peer review completed	2016-07-30
Circulated to partners	2016-07-31
Read by partners	2016-07-31
Mgt. Board approval	N/A

Version	Date	Reason of change
0.1	2016-07-17	Stavros Paschalakis – Initial version.

0.2	2016-07-25	Stavros Paschalakis, Alberto Messina, Maurizio Montagnuolo, Gianluca Francini –Face clustering, CSM-based programme analysis section, compact video representation section
0.3	2016-07-27	Stavros Paschalakis, Miroslaw Bober – Visual quality assessment, visual scene classification
0.4	2016-07-29	Stavros Paschalakis, Alberto Messina, Maurizio Montagnuolo – Audio quality assessment
1.0	2016-07-30	Stavros Paschalakis, Karol Wnukowicz – Software tools, final clean up, sent for QC
1.1	2016-07-30	QC completed, final comments passed to authors.
1.2	2016-07-31	Stavros Paschalakis – Implemented QC comments.

Table of Contents

Table of Figures

1 Executive summary

Deliverable D4.3 is the public document outlining Version B of BRIDGET's Media Analysis Tools. This document describes the progress and main results of the research conducted within WP4 from 2015-05-01 to 2016-07-31.

The aim of WP4 is to deliver media analysis tools so as to (i) facilitate the manipulation of media content in the authoring tools of WP7, (ii) provide additional contextual information for selecting content in the bridget creation process and (iii) allow a more efficient operation of the media search tools of WP5 and the 3D media tools of WP6, by pre-filtering their input and/or post-processing their results.

WP4 is meeting these objectives by developing (or improving/adapting, where appropriate) tools for (i) the temporal segmentation and structure analysis of audio-visual media, (ii) audio-visual media annotation tools, including content classification, and (iii) audio-visual media quality assessment tools.

Additionally, the work under WP4 is carried out with a view to standardisation, particularly in the context of the MPEG-7 Visual and MPEG-7 CDVA projects.

This report outlines the objectives and progress of WP4 in developing Version B of the Media Analysis Tools, i.e. for months M19 to M33 of BRIDGET.

2 Introduction

The aim of WP4 is to deliver media analysis tools so as to (i) facilitate the manipulation of media content in the authoring tools of WP7, (ii) provide additional contextual information for selecting content in the bridget creation process and (iii) allow a more efficient operation of the media search tools of WP5 and the 3D media tools of WP6 by pre-filtering their input and/or post-processing their results.

WP4 is meeting these objectives by developing (or improving/adapting, where appropriate) tools for (i) the temporal segmentation and structure analysis of audio-visual media, (ii) audio-visual media annotation tools, including content classification, and (iii) audio-visual media quality assessment tools.

More specifically, the media analysis tools of WP4 enable the correct temporal association of the augmentation content within the original content, for example allowing the availability of augmentation content only at specific automatically determined intervals of the original programme, aligned with the structure of the programme.

Furthermore, the media analysis tools of WP4 aim to improve the efficiency and results of the tools of WP5 and WP6. For example, in creating bridgets for a programme exploiting a broadcaster's entire archive, the annotation and understanding of the temporal structure of the archive content results in a faster and more accurate operation of the media search tools of WP5. As another example, in creating bridgets between a programme and user-generated content, which is expected to have lower production values than professional content, identifying unacceptable quality content, e.g., with little visual information, poor illumination or very high motion, will enable better operation of the 3D media tools of WP6 and improve the quality of content produced by the authoring tools of WP7.

Additionally, the work under WP4 is carried out with a view to standardisation, particularly in the context of the MPEG-7 Visual and MPEG-7 CDVA projects.

For Version B of the Media Analysis Tools, our effort was focused on the development of new techniques for compact keyframe-based video representation, the completion of the component technologies for face clustering in video and programme analysis based on computational scene models, both of which provide ways of finding relevant content segments for the creation of bridgets, the completion of our work on visual scene classification, a component technology which provides additional contextual information for the selection of content for the creation of bridgets, and the implementation of a core set of tools to evaluate the audio-visual quality of media based on a variety of static and dynamic visual attributes, as well as audio characteristics.

3 WP4 Tasks

WP4 comprises four tasks, namely

- T4.1: Media Structure Analysis (M3-M28)
- T4.2: Media Annotation (M3-M28)
- T4.3: Media Quality Assessment (M9-M32)
- T4.4: Standardisation (M6-M32)

For Version B of the Media Analysis Tools, the objectives of WP4 were:

- T4.1: Develop tools for media structure analysis, more specifically:
  - Develop new tools for compact keyframe-based video representation for the purposes of visual search.
  - Complete development of component technologies of face clustering and programme analysis based on computational scene models.
- T4.2: Develop media annotation tools, more specifically:
  - Complete development of visual scene classification component technology.
  - Develop a GUI tool for multi-descriptor-based image similarity assessment.
- T4.3: Develop media quality assessment tools, more specifically:
  - Complete development of visual quality assessment tools based on static and dynamic visual features.

- Develop tools for assessing the robustness of audio fingerprint – based synchronisation during bridget authoring.

- T4.4: Contribute work related to T4.1 (Media Structure Analysis) and T4.2 (Media Annotation) to MPEG.

4 T4.1 Media Structure Analysis

4.1 Overview

The goal of the Media Structure Analysis task is to deliver temporal segmentation and structure analysis tools for audio-visual media. The tools developed under this task aim to provide different types of segmentations, e.g., according to shots, scenes, faces, interior/exterior settings, etc. and different kinds of summarisations, e.g., according to shot or speaker clustering or to more complex repetitive structural patterns. The division of videos into such diverse programme structures aims to increase the efficiency of the search tools of WP5 and the quality of the content produced by the authoring tools of WP7.

For Version B, our work under T4.1 focused on the development of the following component technologies:

1. Compact keyframe-based representation of a video for visual search purposes.
2. Face clustering.
3. Programme analysis based on computational scene models.

4.2 Compact keyframe-based representation of a video for visual search purposes

This section describes the techniques developed by BRIDGET as an effort to reach a specific standardization goal: the future MPEG standard Compact Descriptors for Video Analysis (CDVA) – Search and Retrieval. In the Call for Proposals [2][3], MPEG issued an invitation to submit methods for search and retrieval in video. This document states that "*This call addresses descriptor technology for search and retrieval applications, i.e. for visual content matching in video. […] The industry thus needs video descriptors that enable performing this task with smaller descriptor sizes and shorter runtimes as compared to application enabled by single-frame (still image) descriptors (e.g. CVDS) in the video domain.*"

An explicit goal is therefore to capture relevant characteristics of the whole or a part of a video sequence (as opposed to representing each video frame). Since visually similar frames and shots often occur several times across a video sequence, we made the hypothesis that such repetitions could be detected and exploited as a form of redundancy in the video material.

Three different approaches were developed during the WP4 activities:

1. Media segmentation based on Logical Story Units.
2. Keyframe selection based on variation of the RVD descriptor.
3. Keyframe selection based on variation of color histograms.

4.2.1 Media segmentation based on Logical Story Unit

In Version A of the BRIDGET tools [1], we developed a media segmentation technique based on detection of Logical Story Units (LSUs) for the purpose of creating a compact representation of a video for visual search purposes. While the LSUs approach is effective, it can require a great amount of memory and cannot satisfy one of the MPEG CDVA requirements, namely that "descriptor extraction shall be possible with low complexity in terms of memory footprint and computational complexity". Furthermore, the LSU-based approach was developed before MPEG published the CDVA Evaluation Framework and the related dataset containing thousands of videos. Processing the dataset we discovered that the LSU computation had a high impact on the overall visual search descriptor extraction time. Therefore, for Version B of the Media Analysis Tools, we chose to seek methods that do not involve detection of conventional scene structures, described below.

4.2.2 Keyframe selection based on variation of the RVD descriptor

We explored a technique based on the variation in time of the global RVD descriptors developed by BRIDGET in WP5. This method is faster because it is based on the comparison of only global descriptors and does not require the storage in memory of descriptors extracted from the whole video before determining representation units.

In order to represent the video in a compact way, only a subset of the original frames are used to form the CDVA bitstream. The first operation used to reduce the number of frames consist in subsampling the video, keeping 1 frame out of 6. Considering the different frame rate of the videos, this ratio allows to keep samples of all video scenes and it is not affected by temporal information that is not always reliable. The second operation compares the global descriptor of each couple of adjacent frames in order to subdivide the sequence in shots. If the score of the comparison is below a threshold *th1* then a shot transition is detected. All frames of the shot are then analysed with an iterative approach: if the score of the global descriptor comparison between the first and the last frame is above or equal to a given threshold *th2* then it is a static sequence and only the descriptors of the frame in the middle of the sequence is used as a reference for creating the CDVA descriptor. If the comparison is below *th2* the process is iterated splitting the sequence in two pieces, cutting the sequence of frames in the middle. The values used for *th1* and *th2* are respectively 0 and 100. Figure 1 shows the frames extracted from the original video (1 out of 6), highlighting with a red border the frames that are selected by the frame selection process.

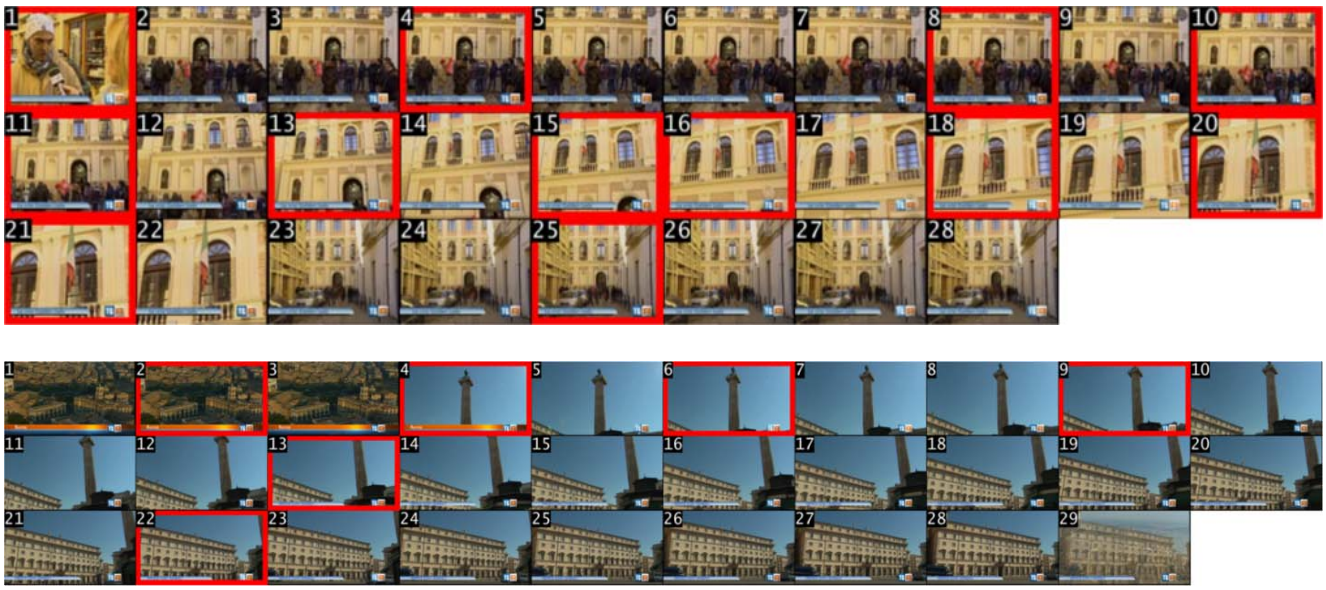

Figure 1. Frames extracted from two MPEG CDVA videos with the selected keyframes highlighted with a red border.

This approach was used to select the keyframe in the proposal M37880 "BRIDGET Response to the MPEG CfP for Compact Descriptors for Video Analysis (CDVA) - Search and Retrieval", submitted to the 114th MPEG meeting held on February 2016 in San Diego – California [4]. The selected keyframes were encoded in a binary format and stored into the CDVA Descriptor, concatenating each one to the other. Therefore, in the proposed implementation, each CDVA Descriptor contained a list of encoded keyframe descriptors (similar to CDVS descriptors, but based on the RVD-W global descriptor).

Table 1 summarizes the complexity measures on extraction of the proposal. The reported times were normalized using the reference platform CPU characteristics (Intel Core I7-5930K), assuming that the CPU is operating at a base frequency of 3.5 GHz, in single thread. The dataset used in MPEG CDVA in order to measure the complexity of a proposal is a subset of the dataset used to evaluate the pairwise matching and retrieval experiments.

Table 1: Complexity measurements evaluated on the MPEG CDVA dataset with the keyframe selection based on variation of the RVD descriptor.

Number of items (files, pairs)	3318
Total video duration (s)	98,916.70
Normalized processing time (s)	182,679.39
Processing time/video duration (s)	1.85
Processing time/item (s)	55.06

The proposals presented at the 114th MPEG meetings were able to compress the information extracted from the video well within the three defined bitrates (16 KB/s, 64 KB/s, 256 KB/s) with promising true positive rates at 1% false positive around of 75%. The main problem was the complexity of the proposed technologies. The BRIDGET proposal required more than a week in order to extract the CDVA descriptor from the whole dataset using a dual Xeon ES-2650 server. The proposal from the University of Peking, due to its extremely heavy associated computational load, had to be run on the *Tianhe* high-performance computing system, using up to 100 nodes, each node being equipped with 2 Intel Xeon E5-2692v2 12 cores processors and 64 GB of RAM.

Due to the high complexity, no proposals were accepted at the meeting and the proponents had to re-submit at the next meeting a new version with a relevant reduction in complexity. Even if keyframe selection based on variation of the RVD descriptor was a technique much faster than the LSU-based approach, it was still too computationally heavy because the RVD descriptors need to be extracted from the SIFT local descriptors of the ALP keypoints. We therefore decided to develop a completely different approach, based on a much faster histogram comparison and described in the next section.

4.2.3 Keyframe selection based on variation of color histograms

This approach is much simpler than the previous two and is based on the comparison of RGB histograms.

The first step consists on a temporal subsampling of the frames, processing one frame out of four. This value is a good compromise between speed and the risk of missing a scene represented by few frames. The temporal subsampling is fixed and not function of the original frame rate of the content because some videos of the CDVA dataset are in formats which don't allow computation of reliable information about frame rate and timing.

The next step is the decoding of the selected frames followed by the computation of the histograms of the R, G, B planes (32 bins each plane). The histogram of the current frame is then L2 normalized and compared with the previous stored histogram using the L2 distance. If the distance is less of equal a given threshold *th* then the frame is dropped, otherwise the frame is selected and used as a keyframe, extracting from it the descriptors that will be used for comparing the video with others. The histogram is stored in memory and used as "previous histogram" in the next iteration.

This process is extremely fast and avoids the computation of the local descriptors from each processed frame. The approach was used as the keyframe selection module in the proposal M38664 "BRIDGET Report on CDVA Core Experiment 1 (CE1)", presented at the 115th MPEG meeting (Geneve, May 2016) [5]. With this keyframe selection technique the complexity of the extraction pipeline was greatly reduced. The results on the CDVA dataset are reported in Table 2.

Table 2: Complexity measurements evaluated on the MPEG CDVA dataset with the keyframe selection based on variation of color histograms.

Number of items (files, pairs)	3318
Total video duration (s)	98,916.70
Normalized processing time (s)	72,161
Processing time/video duration (s)	0.73
Processing time/item (s)	21.75

The keyframe selection approach proposed by BRIDGET provided a relevant gain on speed and was adopted as part of the new CDVA Experimentation Model defined as output of the 115th meeting (W16274 "CDVA Experimentation Model (CXM) 0.2") [6].

4.3 Face Clustering

Face clustering is an important task since it helps in identifying segments of potential interest for bridgets along the programme timeline, such as the appearance of guests in a talk show. Furthermore, it provides valuable information for further data analysis such as content segmentation and summarization. In BRIDGET, face clustering is implemented by a three-step algorithm involving detection, extraction and matching of faces occurring in the analysed broadcasts. Full details of the algorithm are available in [1]. In the second part of the BRIDGET project, we optimized the original algorithm by implementing a fully parallel architecture based on the Java OpenIMAJ library [7]. This allows to take advantage of modern multicore and multithreaded CPUs, while maintaining interoperability and (back) compatibility with existing hardware.

Experiments were conducted on a CentOS release 6.2 (64 bit) virtual machine equipped with 16 GB of RAM and 16 virtual cores at 1.7 GHz. The speed of the algorithm was tested using a reference dataset of 14 programmes including newscasts and talk shows. On average the algorithm performs about 0.67x real time (processing time / video time ratio of 1.5:1). Speed is dependent on the genre of the analysed programme. Best performance was achieved for talk shows with one presenter and few guests getting an average speed of 10x real time (processing time / video time ratio of 0.1:1). On the other hand, worse performance was observed for newscasts, obtaining an average speed of 0.4x real time (processing time / video time ratio of 2.5:1). The parallel implementation of the algorithm outperforms the serial implementation by an average speedup factor of about 3.

4.4 Programme Analysis based on Computational Scene Models

4.4.1 Motivations

This work aims to develop an automatic technique for TV programme structuring. TV programme structuring is the process in which broadcasted content is segmented and organized based on recurring patterns in the programme timeline. Examples of these patterns, called *computational scenes* in BRIDGET, or *structural units* in [8] include the appearance of participants in talk shows, audio-visual jingles in quiz shows, or target events in sports videos. This technology serves as the basis to facilitate tasks such as browsing and annotation of broadcasted material. In the specific case of BRIDGET, this technology supports the user of the Professional Authoring Tool in finding appropriate segments to which associate bridgets.

4.4.2 State of the Art in TV programme structuring

This section overviews the state of the art literature in TV programme structuring. Almost all of the proposed methods rely on supervised approaches. This is the case of sports videos in which information about target events, such as football goals and free kicks, is modelled based on some prior knowledge.

Basing on prior knowledge provides good performance of the algorithms. On the other hand, poor generalization properties are offered, since retrained models are needed when new programme genres are considered. As an alternative, genre-independent approaches aim at identifying a global description of the programme structure while maintaining independency to the programme genre.

4.4.2.1 Event-based programme structuring

Event-based programme structuring aims at providing a summary of the video centred around some target events. Most of these approaches rely on news or sports videos, where events are well defined, such as single news stories in newscasts or goals, free kicks and corners in football matches. Hidden Markov Models have been extensively adopted in these applications [15][16][17][18][19]. Even if these approaches achieve effective results, their applicability is strictly restricted to specific application domains.

4.4.2.2 Genre-agnostic programme structuring

Genre-agnostic programme structuring addresses the issue of enlarging the application domain of automatic video segmentation and summarization. The assumption is that patterns of audio-visual structures can be used to infer structural elements, such as scene separators (e.g. jingles and applauses) or anchor-persons [22], of recurrent programmes. Recurrent programmes are intended as programmes with multiple episodes periodically broadcasted (e.g., daily, weekly). The novelty here is that the target audio-visual structures are not defined a priori, but automatically identified by the analysis of the patterns shared among different episodes of the same programme. Qu et al [20] propose an unsupervised approach to recurrent TV programme structuring casting the problem of structure discovery as a grammatical inference problem. In [21] decision trees are employed to detect short audio-visual sequences that delimit the different parts of a programme. Quiz shows, magazines and newscasts are taken as reference genres. Though these approaches are promising and represent and interesting attempt to perform programme structuring at a general level, they still present some drawbacks, such as limited capability of generalizing to different programmes of the same genre. In fact, even if the same grammar can be considered a good representative for e.g. different newscasts, the same could not be true for e.g. different talk shows.

4.4.3 A Hidden Markov Model approach for TV programme structuring

The approach adopted in BRIDGET assumes that the storyline of a TV programme is modelled by a discrete Hidden Markov Model (HMM). Consecutive frame-based audio-visual shots of the programme are taken as reference timeline. The observations correspond to the set of audio-visual features that can be automatically extracted from the analysed content. The state transition function represents the probability of moving from one state to another.

4.4.3.1 Overview of Hidden Markov Models

A Hidden Markov Model (HMM) is a statistical Markov chain in which the modelled system is supposed to be a Markov process with unobservable (i.e. hidden) states. HMMs have been successfully applied to a variety of fields, including speech recognition [9], video classification [10], and part of speech (POS) tagging [11]. An HMM is characterized by the following elements:

1. The set of possible states in the model, $S = \{S_1, \dots, S_N\}$[1]. For example, in POS tagging applications S represents the set of syntactic categories, e.g. verb, noun, article, etc., to which a word could belong;
2. The set of distinct observation symbols per states, $V = \{v_1, \dots, v_M\}$. The observation symbols correspond to the physical output of the modelled system. Continuing with the POS tagging example, V is the sequence of words in a given text;

[1] We denote with q_t the state at time t.

3. The state transition probability distribution, $A \in [0,1]^{N,N}$, where $a_{i,j} = P(q_{t+1} = S_j | q_t = S_i), 1 \leq i,j \leq N$ denotes the probability that the system is in state S_j at time *t+1* given that it is in state S_i at time *t*. In POS tagging problems, $a_{i,j}$ could be e.g. the probability that the next word is a noun given that the previous word is an article;
4. The observation symbol probability distribution in state j, $B \in [0,1]^{M,N}$, where $b_j(k) = P(v_k \text{ at } t | q_t = S_j),\ 1 \leq j \leq N \text{ and } 1 \leq k \leq M$. In POS tagging applications, $b_j(k)$ could be e.g. the probability that a word is tagged as noun;
5. The initial state distribution $\pi \in [0,1]^N$where $\pi_i = P(q_{t_0} = S_i), 1 \leq i \leq N$, e.g. the probability that a phrase starts with an article.

Given an HMM, there are three types of problems that can be posed:

1. Scoring: given the observation sequence $O = \{O_1 O_2 \dots, O_T\}$ and the HMM model $\lambda = (A, B, \pi)$, what is the probability $P(O|\lambda)$ that the observation sequence is generated by the model?
2. Decoding: given the observation sequence $O = \{O_1 O_2 \dots, O_T\}$ and the HMM model $\lambda = (A, B, \pi)$, what is the most likely state sequence $q = \{q_1 q_2, \dots, q_T\}$ that generated O?
3. Learning: given the observation sequence $O = \{O_1 O_2 \dots, O_T\}$ and the HMM model $\lambda = (A, B, \pi)$, how do we adjust the model parameters so that the probability $P(O|\lambda)$ is maximized?

Solving each of the posed problems allows to tackle many real-world applications, such as pattern recognition, matching and prediction.

4.4.3.2 Developed model

The talk-show genre was taken as reference for this work. This choice was justified by the fact that talk shows cover most of the daily broadcasting time. Furthermore, they represent potentially interesting use cases for bridgets consumers. For example, insights and analytics about the debated topic could be provided in the form of bridgets [13]. In the television domain, a talk show is defined as a *"mainly verbal programme in which more than person participates"* [14]. Even if talk shows can be very different in terms of e.g. duration, studio settings and scopes, they also present some shared aspects that have been studied in semiotic and sociological works [12]. A talk show is generally made of a succession of repetitive structures such as interviews, musical passages, excerpts, jingles, etc. Speaker interventions can be seen as elementary units, i.e. computational scenes. TV reports and guest performances can be seen as transitional elements between speaker interventions. Jingles, cutaway shots and applauses are used as demarcation lines. In light of these considerations a talk show can be considered, from the editorial point of view, as a sequence of the following five main different types of audio-visual segments:

- *Start*, corresponding to the opening credits of the programme. These include e.g. starting jingles or previews;
- *Talk*, corresponding to verbal interactions between people. These include e.g. monologues, interviews and debates;
- *Insert*, corresponding to non-verbal parts of the programme. These include, e.g. guest performances such as playing or singing, advertisements and reports;
- *Separator*, corresponding to demarcation lines in the broadcast timeline;
- *End*, corresponding to the final credits of the programme. These include e.g. ending jingles or summaries.

It has to be noticed that annotators use these elements as reference documentation items and thus they can be used as potential target segments for the creation of bridgets.

4.4.3.3 Processing pipeline

The processing pipeline is illustrated in Figure 2. From top to bottom and left to right, first the audio and video tracks are extracted from the input video. From these tracks, a set of audio-visual features are extracted. The defined features include:

- The sequence of video shots;
- The presence or absence of faces within the detected shots;
- The presence or absence of speech, music, noise, and silence in the audio track. Combinations of these features are possible (e.g., speech with noisy background).

Intuitively, the selected features are representative of the five types of audio-visual segments (i.e. the HMM states) that characterize a talk show. For example, the absence of faces plus music could denote a *start* segment. On the opposite, the presence of faces and pure speech could denote a *talk* segment.

The extracted features are used to generate the observation sequence for the HMM system. More formally, let $s = \{s_{1,\dots,}s_T\}$ be the set of video shots detected by the shot segmentation algorithm. The observation sequence associated to s is $O = \{O_1, \dots, O_T\}$, where $O_i$is a five-dimensional binary vector representing the presence of absence of a given feature at shot s_i. For example, the vector $O_i = [11010]$ would indicate that the i-th shot depicts some faces. Speech and background noise is also detected. The sequence of the observation vectors is the input of the HMM. The output of the HMM is the sequence of states (i.e. types of audio-visual segments) that most likely generated the observation vectors, according to the underlying Markov process.

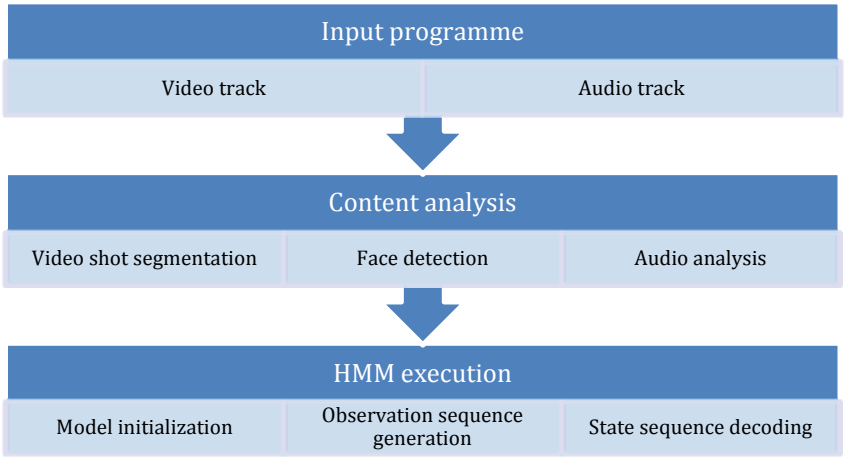

Figure 2. HMM-based processing pipeline for talk show segmentation.

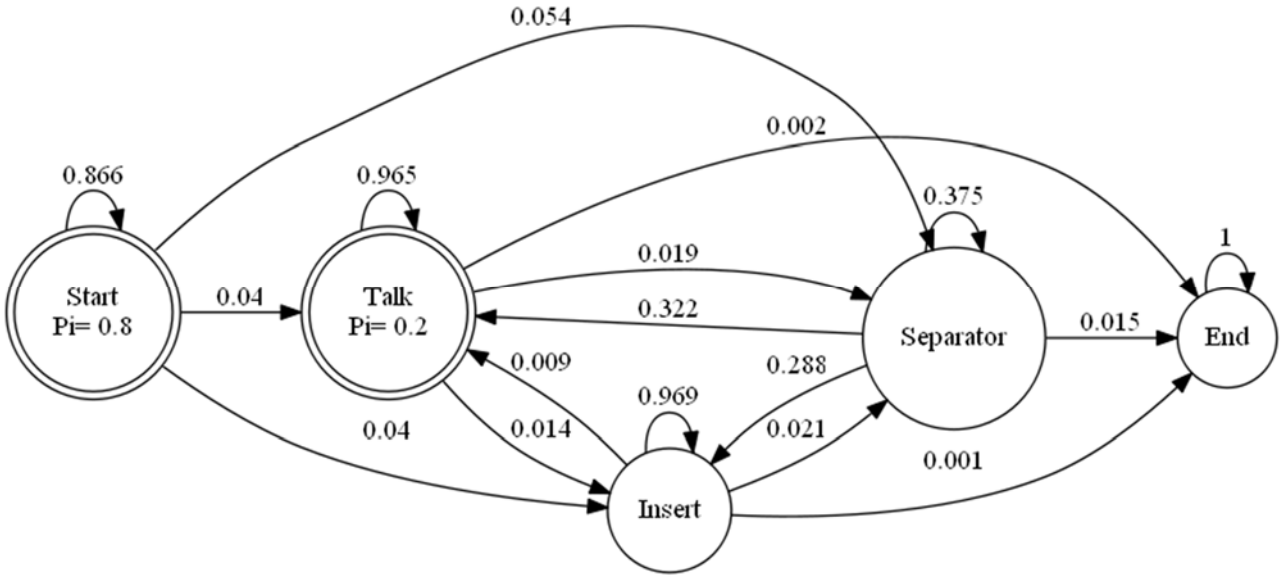

Figure 3. Illustration of the HMM for talk show structuring.

4.4.3.4 Performance tests

This section describes the structure and the performance of the generated HMM. The HMM parameters were estimated empirically using a reference dataset of 24 chapters from two programmes, namely "Ballarò" and "Porta a Porta", two popular Italian talk shows. Chapters were manually defined by annotators. The total duration of the collected material was 11 hours, resulting in 8256 analysed video shots. The length of each chapter was varying from six to sixty minutes (average 28 minutes). The ground truth was generated by labelling each shot according to its state. The created HMM together with interstate connection and initial state probabilities is illustrated in Figure 3. Leave-one-out cross validation was used to evaluate the performance of the method. Overall accuracy was calculated as one minus the normalized Levenshtein distance between ground truth and detected state sequence. Table 3 illustrates the achieved accuracy for each programme, where a value of one denotes a perfect match between the generated segmentation and the ground truth segmentation. Experimental results are encouraging, reaching in many cases an accuracy greater than 90%. The confusion matrix in Figure 4 helps understand which are the successful aspects and the critical points of the method. The rows of the matrix are the actual classes and the columns are the predicted classes. We note that *talks* and *inserts*, i.e. the most important segments from the editorial point of view, are predicted with good accuracy. Some opening segments (i.e. *start* state) are confused with *talk* sequences. This is the case when e.g. a programme starts with the anchor who introduces the episode topics. A potential point of failure of the model is the *end* state. In fact, when the system reaches this state, it loops there at probability one. This means that when an audio-visual shot is labelled as *end*, all the successive shots will be labelled as *end*. However, experimental results demonstrated that misclassification of non-end segments as end segments occurs very rarely (see the green bar in Figure 4). As a final comment, we note that separators obtain the worse results, being often misclassified as either talks or inserts (see the orange bar in Figure 4). However, this is a less critical issue, since it corresponds to a (usually short) shifting of the start (end) of talks or inserts along the generated segmentation.

Table 3: Performance evaluation of the HMM programme segmentation framework.

File	Accuracy	File	Accuracy
Ballaro_2014-09-30_1_Anteprima	0.82	Porta_a_porta_2014-03-21_2_Intervista	0.8
Ballaro_2014-09-30_2_Intervista	0.71	Porta_a_porta_2014-03-21_3_Intervista	0.65
Ballaro_2014-09-30_6_Dibattito	0.86	Porta_a_Porta_2014-04-07_1_Intervista	0.67
Ballaro_2014-09-30_7_Dibattito	0.96	Porta_a_porta_2014-10-01_1_Intervista	0.53
Ballaro_2015-02-24_10_Dibattito	0.68	Porta_a_porta_2014-10-01_2_Intervista	0.81
Ballaro_2015-02-24_3_Dibattito	0.9	Porta_a_Porta_2015-01-15_3_Intervista	0.8
Ballaro_2015-02-24_4_Dibattito	0.95	Porta_a_Porta_2015-02-06_4_Intervista	0.9
Ballaro_2015-02-24_6_Dibattito	0.95	Porta_a_porta_2015-02-19_3_Intervista	0.94
Ballaro_2015-02-24_7_Dibattito	1.0	Porta_a_porta_2015-02-27_1_Dibattito	0.94
Ballaro_2015-02-24_8_Dibattito	1.0	Porta_a_porta_2015-02-27_2_Dibattito	0.97
Ballaro_2015-02-24_9_Dibattito	0.85	Porta_a_Porta_2015-03-10_4_Intervista	0.88
Average	0.88	Porta_a_porta_2015-03-13_1_Intervista	0.89
Average accuracy: 0.86		Porta_a_porta_2015-03-20_1_Intervista	0.97
		Average	0.83

	Start	End	Talk	Insert	Separator
Start	93	0	7	0	0
End	0	66	16	12	6
Talk	0.5	0.5	90	7	2
Insert	2	0	10	84	4
Separator	11	1	20	30	38

Figure 4. Confusion matrix for the HMM states.

4.4.4 Conclusions

We developed an HMM framework for addressing the task of structural segmentation of talk show programmes. Semiotics and editorial aspects were considered in the definition of the elementary units (i.e. HMM states) in which the programme timeline is decomposed. Multimodal features in the video and audio domain have been extracted and used as observables in the model. The experimental results are encouraging. Future work includes planning for the extension of the dataset to further test stability and generalization of the method.

5 T4.2 Media Annotation

5.1 Overview

The goal of the Media Annotation task is to deliver annotation tools for audio-visual media providing complementary descriptions and additional relevance information to be used in the authoring tool of WP7, as well as in conjunction with the tools of WP5 and WP6, by pre-filtering their input or post-processing their results.

For Version B, our work under T4.2 focused on the development of a visual scene classification technology combining deep Convolutional Neural Networks (CNNs) with the RVD-W descriptor of WP5. Additionally, T4.2 also worked on the development of a GUI tool for multi-descriptor-based image similarity assessment, primarily for the purposes of semi-supervised image selection and sorting prior to 3D reconstruction according to WP6.

5.2 Visual Scene Classification

5.2.1 Introduction

Image and video scene classification underpin numerous applications including web search, organization of photo/video libraries, surveillance, biometrics, robotic vision etc. The task of performing accurate and scalable classification is challenging mostly due to large intra-class visual diversity, significant similarities between classes, background clutter and partial occlusions. In the first part of BRIDGET we developed an efficient and effective scene classification system, based on the aggregation of densely-sampled SIFT descriptors into RVD representations.

In the second part of BRIDGET we significantly improved the classification performance by encoding deep Convolutional Neural Network (CNN) features into our novel RVD-W representation, as illustrated in Table 4. To further enhance performance, we propose a method to incorporate second order statistics (diagonal covariance of the residual vectors) into the original RVD-W framework.

Table 4: Comparison of classification performance achieved in phase A and phase B of BRIDGET.

Method	**PASCAL VOC 2007 (mAP %)**	**CALTECH 256 (mCA %)**	**MIT SCENE 67 (mCA %)**
RVD with SIFT (version A)	65.1	49.3	65.4
RVD-W with CNN (version B)	86.4	77.0	77.2

5.2.2 Image classification based on CNNs

CNNs are currently achieving the state-of-the-art in many different areas including object recognition. The current benchmark used in object recognition is the ImageNet Large Scale Recognition Challenge (ILSVRC) dataset which contains over 1.2 million images for training over 1,000 unique object classes [23]. In the latest ILSVRC (2015), an ensemble of deep CNNs with up to 152 layers are trained and averaged to produce 3.57% top-5 error on the test set achieved the state-of-the-art [24].

CNN based visual descriptors typically involve using a pre-trained model on a large database such as ImageNet avoiding the heavy computational resource requirement in training billions of hyper-parameters in the network while also obtaining highly accurate, robust and generalised features. Features are obtained from the convolutional layers outputs in the network, which layer determines the type of features extracted where closer to the input layer are low-level edge-based features and closer to the output layer are high-level features built up from the aggregation of the low-level features forming shapes and textures. Extracted feature descriptors may form an input into a linear classifier for supervised learning, which may be undertaken using a dataset different to the one that the CNN was trained on, allowing for adaptive transfer learning to particular tasks. Azizpour et al. [25] confirms the robustness of features extracted from a CNN trained on ImageNet which outperforms previous methods (using outside training data) on several datasets including PASCAL VOC2007 80% mAP and MIT SCENE67 71% mCA.

5.2.3 RVD-W based on deep CNN features

We propose to encode CNN-based features into RVD-W representation for image classification task. In this approach, all images are resized to $c \times c$ pixels prior to passing through the network. We crop the CNN at the last convolutional layer and regard it as a dense descriptor extractor. The output of the last layer is a $h \times w \times d$ feature map which can be considered as a set $X = \{x_t \in \mathbb{R}^d, t = 1 \dots T\}$ of d-dimensional descriptors extracted at $h \times w$ spatial locations.

Each local descriptor x_t is assigned to its K nearest clusters with corresponding ranks and the residual vectors are L1-normalized and weighted based on the rank assignment weights. The weighted residual vectors are de-correlated and subsequently whitened before aggregation into a cluster-wise representation ζ_j^1.

The second order cluster-wise representation ζ_j^2 is formed by aggregating the average variance of the residual vectors for each cluster. We apply L2-normalisation to the individual ζ_j^1 and ζ_j^2 vectors. The final RVD-W representation R^w of an image is obtained by stacking of the component vectors $\zeta_j^1$and then of the vectors ζ_j^2 for each of the j^{th} cluster. More information on the RVD-W formation can be found in the WP5 report [43].

5.2.4 Experiments and results

The performance of the proposed method is extensively evaluated on three standard image classification benchmarks: Pascal VOC 2007, Caltech 256 and MIT SCENE 67 datasets. The performance evaluation on ImageNet was not completed by the time of writing this report, due to the computational complexity of the extraction of CNN features from this dataset.

First we compare the performance of RVD-W representation with the Fisher Vector. We extract deep descriptors using the state-of-the-art CNN, OxfordNet. Each image is resized to the size 448 × 448 before passing through the network. The output of the last layer is a 28 × 28 × 512 feature map, forming a set of 784 512-dimensional descriptors. The dimensionality of the CNN features is reduced from 512 to 64 via PCA. We learn a codebook of 256 cluster centers, forming a 16k dimensional image signature. Linear Support Vector Machines are used as a classifier.

As can be seen the RVD-W pipeline outperforms the FV pipeline on all datasets. The mean gain over FV is +1.1% on PASCAL VOC 2007 and +2.0% on CALTECH256.

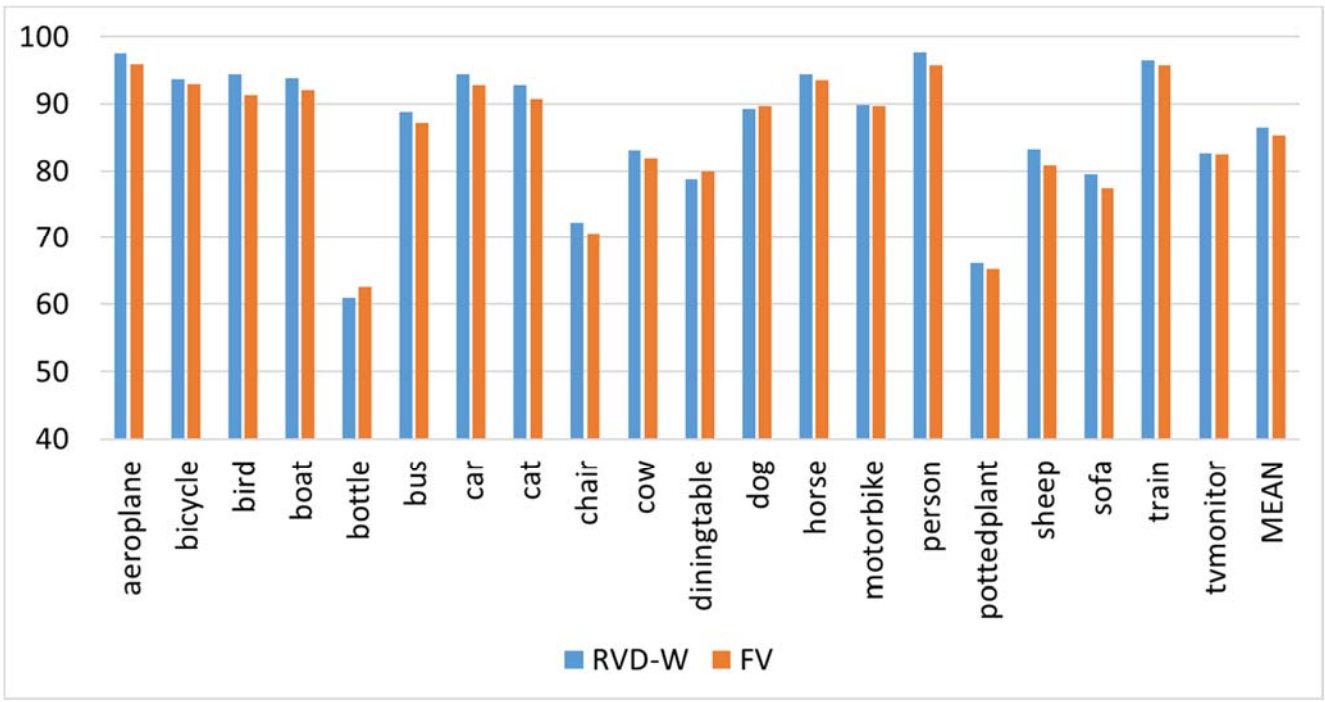

Figure 5. Comparison of the classification performance of FV and RVD-W (PASCAL VOC 2007).

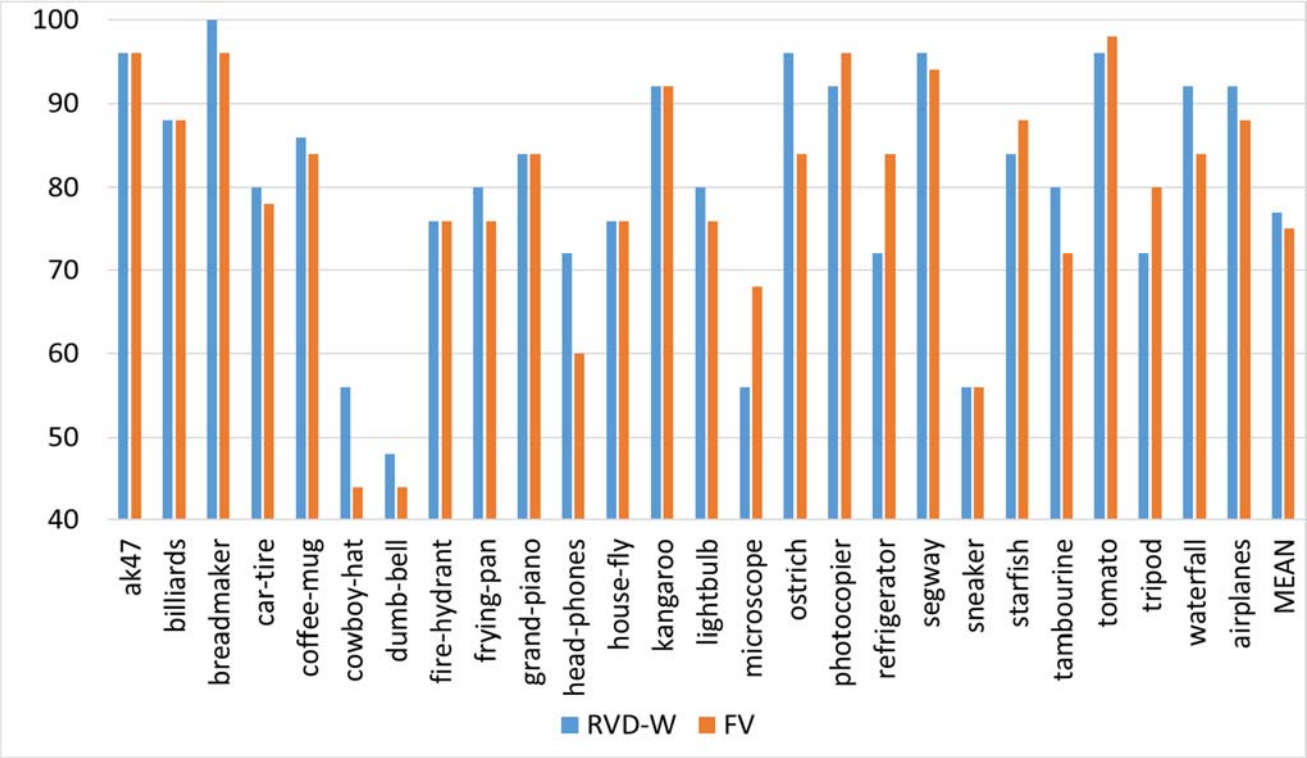

Figure 6. Comparison of the classification performance of the FV and RVD-W (CALTECH 256).

5.2.5 Comparison with the state-of-the-art based CNN-representation

Table 5 compares the performance of proposed method to the latest CNN-based representations, showing a consistent improvement achieved by the RVDW signature over the state of the art.

Table 5: Comparison with the state of the art with CNN-based compact codes.

Method	PASCAL VOC 2007	CALTECH 256	MIT SCENE 67
Max pooling [25]	80.7	-	71.3
MOP-CNN [26]	-	-	68.9
RIFD-CNN [27]	71.0	-	-
CNN+FV	85.4	75.1	76.2
CNN+VLAD	82.2	70.7	69.8
CNN+RVD-W	**86.4**	**77.0**	**77.2**

5.3 GUI tool for image selection and sorting

Under T4.2 we also worked on the development of a GUI tool for multi-descriptor-based image similarity assessment and sorting. The tool aims to facilitate semi-supervised image selection and sorting prior to 3D reconstruction according to WP6.

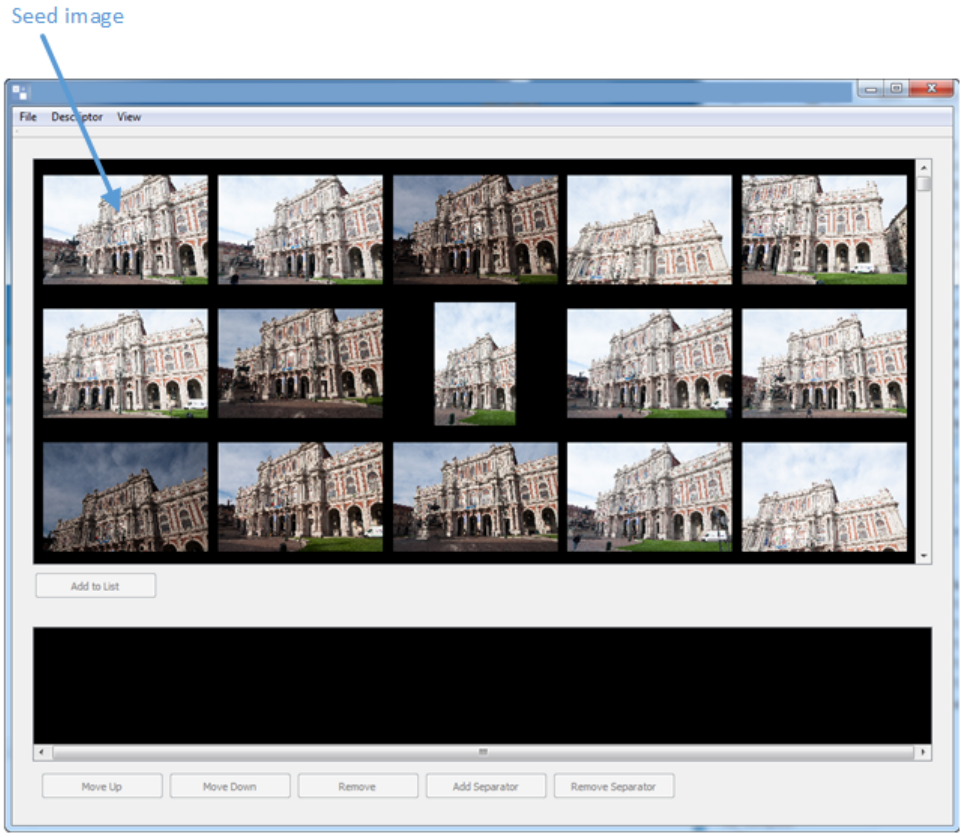

Figure 7. Image sorting using WP5 visual search descriptors on the WP6 "Palazzo Carignano" dataset.

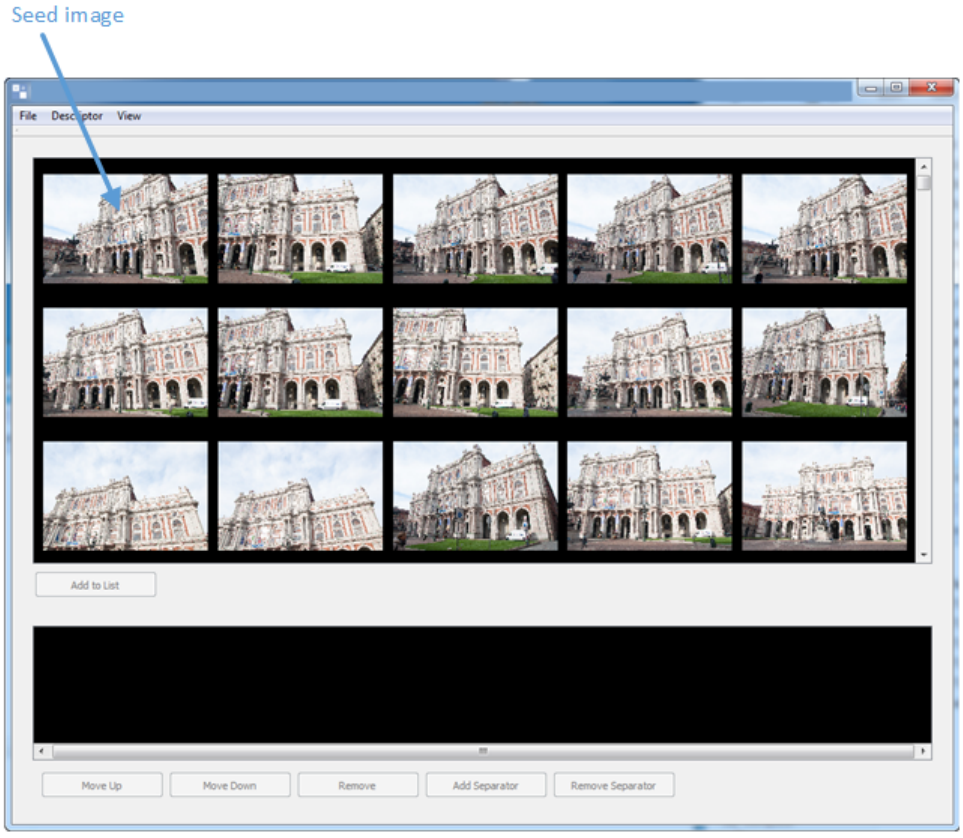

Figure 8. Image sorting using WP5 visual search descriptors, MPEG-7 dominant colour and MPEG-7 colour structure on the WP6 "Palazzo Carignano" dataset.

The tool allows a user to create, update and maintain a database for a set of visual assets, i.e. images or video keyframes. The database comprises visual search descriptors from WP5 and MPEG-7 descriptors, as well as self-similarity matrices for each descriptor type. During normal operation, the user is able to select a "seed" image and view all the other images based on their similarity to the seed image according to one or more descriptors (when multiple descriptors are used, a weighted rank-based fusion scheme is used to merge the multiple similarity lists to a single list). By selecting different descriptors, the user can sort the images according to different criteria, e.g. local information (using the visual search descriptors of WP5), global colour (e.g. using MPEG-7 dominant colour), global colour structure (e.g. using MPEG-7 colour structure), etc., as illustrated in Figure 7 and Figure 8. The user may then select images in a specific order, split them into groups (e.g. representing different sides of a particular building), and create an export list for further processing by the tools of WP6, as shown in Figure 9.

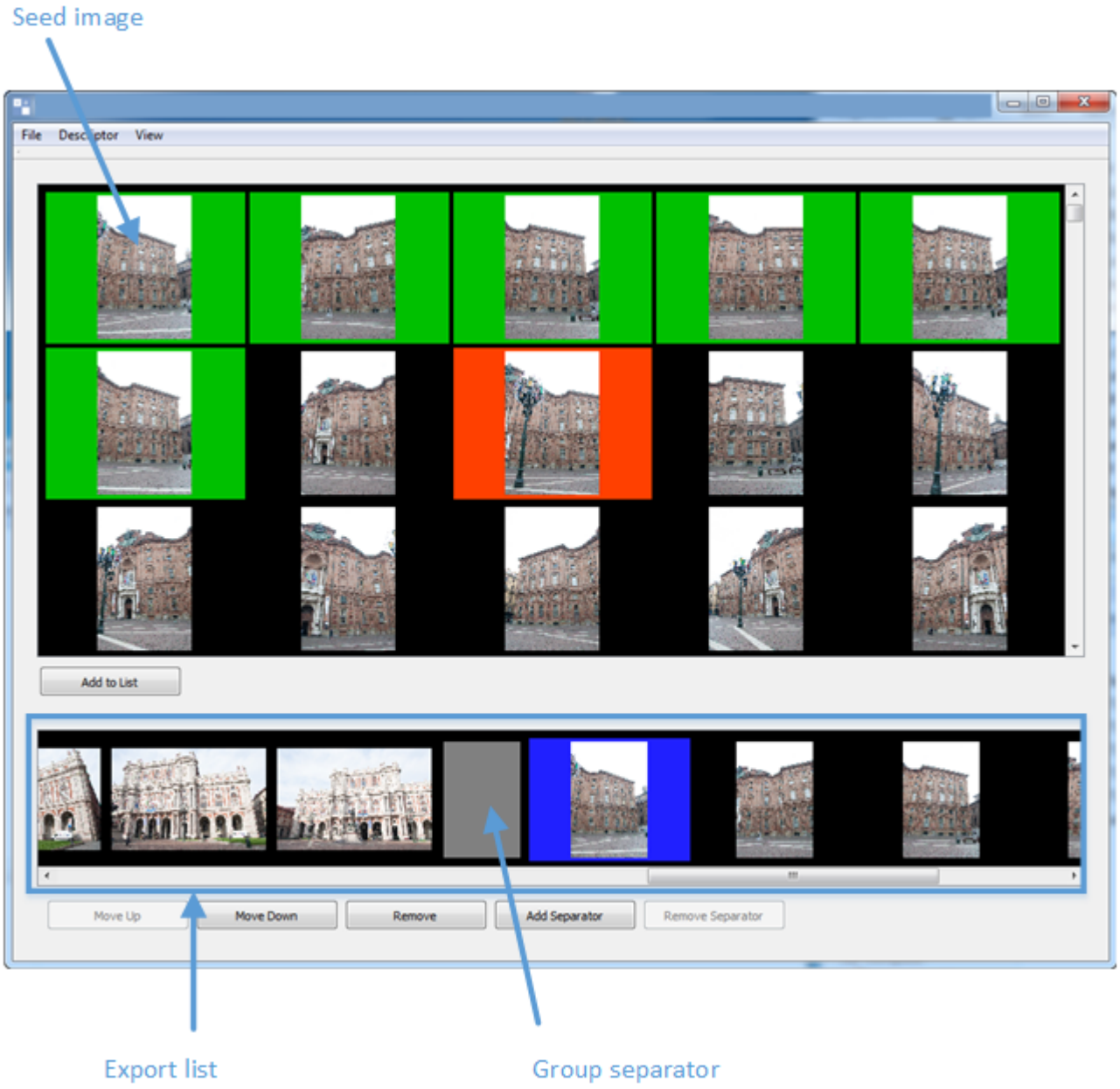

Figure 9. Image selection and export.

6 T4.3 Media Quality Assessment

6.1 Overview

The goal of the Media Quality Assessment task is to deliver tools for audio-visual quality assessment. These tools aim to allow the presentation of the highest quality media to the end users, enhance the operation of the 3D scene reconstruction tools of WP6, by filtering media content of low quality which would adversely affect the operation of those tools, and allow an assessment of the robustness of audio fingerprint – based synchronisation during bridget authoring. The objective is to select a suitable existing algorithm, as there is no requirement to move beyond SOTA.

6.2 Visual Quality Assessment

Our work focused on the implementation of a core set of tools to evaluate the visual quality of media based on: (1) size and resolution, (2) level of JPEG artefacts, (3) sharpness/blurring, (4) brightness and contrast properties, and (5) shakiness.

6.2.1 Image-based tools

6.2.1.1 Image resolution

This metric takes into account the largest dimension D in pixel of a picture w x h, that is $D = \max(w, h)$. A custom conversion table can be built in order to generate a quality index $\mathbf{q}_{\mathbf{RES}}$ between 0 and 5, as shown in Table 6. The boundary values can be adjusted, based on the user preference.

Table 6: Image resolution-quality conversion table.

D	$\mathbf{q}_{\mathbf{RES}}$
>= 4000	5
>= 2560	4
>= 1024	3
>= 256	2
>= 64	1
< 64	0

6.2.1.2 Block artifacts

Block artifacts are usually caused by the application of quantization on lossy compression algorithms on the image. Block transform coding (e.g. the discrete cosine transform) is applied to a block of pixels, and to achieve lossy compression, the transform coefficients of each block are quantized. The lower the bit rate, the more coarsely the coefficients are represented and the more coefficients are quantized to zero.

Statistically, images have more low-frequency than high-frequency content, and the quantisation tables are designed to keep the low-frequency content after quantization, generating blurry, low-resolution blocks. Because the quantization process is applied individually in each block, neighbouring blocks quantize coefficients differently. This leads to discontinuities at the block boundaries, which are most visible in flat areas, where the effect is poorly masked.

In the following we present our solution to estimate the amount of blocking artifacts present in the image, represented by the quality index $\mathbf{q}_{\mathbf{BLK}}$ between 0 and 5, where 5 represents the best quality. After

computing the gradient of the image G_X in the x direction, the gradient magnitude is accumulated per column in $S_X[i]$, where $0 < i <= w - 1$:

$$S_x[i] = \sum_{j=0}^{r-1} G_x(j,i).$$

$S_X[i]$ contains the information about the gradient magnitude difference between column i and column $i + 1$, as shown in Figure 10. If the image contains block artifacts, values $S_X[k]$, $S_X[2k]$, $S_X[3k]$ … , where k is the size of the transformed block, will appear consistently higher than the other sums. Block size k is usually equal to 8.

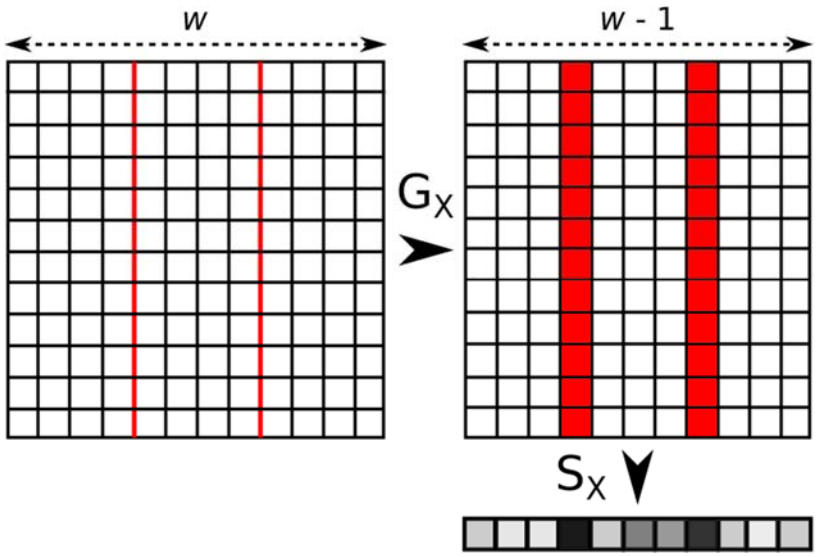

Figure 10. Gradient magnitude sums SX[i].

The image degradation due to block artifacts is computed as ratio r between the average value of S_X for columns at index k, $2k$, $3k$, … and the average value of S_X for the remaining columns. If no artifacts are detected, r should be close to 1, while in case of very sharp block artifacts, r should be much greater than 1. The quality index $\mathbf{q_{BLK}}$ is computed as:

$$q_{BLK} = max(0{,}6 - round(r)).$$

6.2.1.3 Sharpness

Sharpness can be measured as amount of blurriness in the image. Blurriness is caused by the lack of focus in the camera lens or by the change of the image content during a single exposure, either due to rapid movement or long exposure. Given the image gradients G_X and G_y in the x and y directions, the L2 norms $N_X = ||G_X||$ and $N_y = ||G_Y||$ are used to compute an aggregate metric S:

$$S = \frac{N_x^2 + N_y^2}{w \cdot h}.$$

A custom conversion table can be built in order to generate a quality index $\mathbf{q_{SHR}}$ between 0 and 5, as shown in Table 7.

Table 7: Sharpness-quality conversion table.

$1 / S$	$\mathbf{q}_{\mathbf{SHR}}$
<= 0.00005	5
<= 0.00010	4
<= 0.00015	3
<= 0.00020	2
<= 0.00030	1
> 0.00030	0

6.2.1.4 Contrast

Contrast is the difference in luminance or colour that makes the representation of an object in the image distinguishable. Bad light condition or overexposure cause the loss of details in the picture, affecting the overall contrast value. We use the Histogram Spread (*HS*) metric introduced in [29] to measure the image contrast. Given the grayscale (8 bit) cumulative histogram of an image, *HS* is computed as follows:

$$HS = \frac{L_{3rd} - L_{1st}}{255},$$

where L_{3rd} and L_{1st} are the third and the first quartile of the cumulative histogram and 255 is the maximum range of pixel values. *HS* values are in the range [0, 1]. A quality index $\mathbf{q}_{\mathbf{CNT}}$ with values between 0 and 5 is computed as linear map of [0, 1] range into [0, 5].

6.2.1.5 Overall quality index

An overall quality index **q** is computed considering the worst quality index among image resolution $\mathbf{q}_{\mathbf{RES}}$, blockiness $\mathbf{q}_{\mathbf{BLK}}$, sharpness $\mathbf{q}_{\mathbf{SHR}}$ and contrast $\mathbf{q}_{\mathbf{CNT}}$:

$$\mathbf{q} = \min(\mathbf{q}_{\mathbf{RES}}, \mathbf{q}_{\mathbf{BLK}}, \mathbf{q}_{\mathbf{SHR}}, \mathbf{q}_{\mathbf{CNT}}).$$

An image could be perfectly on-focus, with no compression artifacts but as big as a stamp. While the mean operation fails in penalizing the image quality, large differences between the quality indexes are better represented by taking the minimum value.

6.2.2 Video-based tools

6.2.2.1 Keyframe-frame based assessment

Video quality assessment according to (1) size and resolution, (2) level of JPEG artefacts, (3) sharpness/blurring, (4) brightness and contrast properties is performed by combining the techniques described in 6.2.1.1 to 6.2.1.5 with the shot detection and keyframe selection tools developed in Version A of the Media Analysis Tools [1].

6.2.2.2 Shakiness

Shakiness is the distortion of the video recording caused by instability of the camera. This effect is common for videos recorded by operators with hand-held cameras, especially when the operator walks or performs panning and tilting of the recorded scene. The shakiness is usually defined as unwanted motion of high frequency. The global camera motion can be estimated on frame-to-frame basis, and this motion can be decomposed into intentional camera motion such as pan and tilt, and unwanted camera motion using a cut-off frequency of 1Hz [30]. In [31][32] the camera motion is estimated using Luminance Projection Correlation method (LPC). In each frame the luminance pixels are projected in horizontal and vertical directions, the projections are correlated between consecutive frames giving 2 components of motion signal in horizontal and vertical directions. The shakiness quality score is calculated by applying low pass filtering on the obtained camera motion signals.

Our shakiness assessment implementation is based on keypoint detection and tracking of the keypoints across video frames to obtain the motion of camera in consecutive frames. The keypoint features to be tracked are N strong corners (N=50) detected in luminance video frame using the method described in [33]. The keypoints are tracked across frames using Lucas Kanade feature tracker [34] which computes the positions of the corresponding features in the following frame. The keypoint positions in two consecutive frames are the input to the function which computes the optimal affine transform of the 2D points. The translation component of the affine transform represents the basic camera motion used to assess video shakiness. The standard deviations of the extracted geometrical translations in temporal windows of the duration 1 second are computed as the shakiness coefficients. The intentional camera motion usually has low variation within 1 second, thus the variation represents the unwanted shakiness component. To speed up the processing the video frames are resized to the height 256 pixels (preserving aspect ratio), and every second frame is processed.

The shakiness coefficients are calculated for each temporal segment with continuous geometrical translations of keypoints, where the shakiness coefficient for a segment is the average of deviations in a set of 1-second temporal windows of that segment. In most cases the tracking temporal segment corresponds to video shot, but in some cases a shot may be further split into sub-segments – this happens when the tracking function cannot find enough corresponding keypoints in 2 consecutive frames. Finally, the shakiness of a video is computed as a function of the shakiness components of all temporal segments in that video.

The shakiness detection method was tested using 12 videos with different levels of unwanted shakiness. The videos included professional productions with stable camera movements, videos recorded with hand-held cameras and phone cameras with visible unwanted camera movements of different levels. The experiment shows that the shakiness score computed by the software corresponds to the subjective impressions of unwanted camera movements.

6.3 Audio Quality Assessment

6.3.1 Motivation

This work aims at devising an architecture for audio quality analysis targeted at the evaluation of the robustness of audio fingerprint – based synchronisation during bridget authoring. The rationale of the work is to support the bridget author in identifying which segments of the content have the best (or the worst) probability of success in terms of recognition, so to steer her editorial choices accordingly and optimising the final user experience (i.e., minimising false detections and missed matches).

6.3.2 Architecture

Audio fingerprint – based content recognition is an established business area, counting several industrial solutions such as [35][36][37]. Future systems wanting to exploit BRIDGET's results using this approach will therefore rely on a stable marketplace in which software and services components for audio fingerprint based content recognition are readily available. On the other hand, it is not expected to have major breakouts in the academic research concerning this topic in the next few years. Thus, in the context of BRIDGET the attention has moved to consider systems supporting the assessment of the

quality of user experience during the authoring phase, and how this quality depends on the specific performance of the audio fingerprint extraction system.

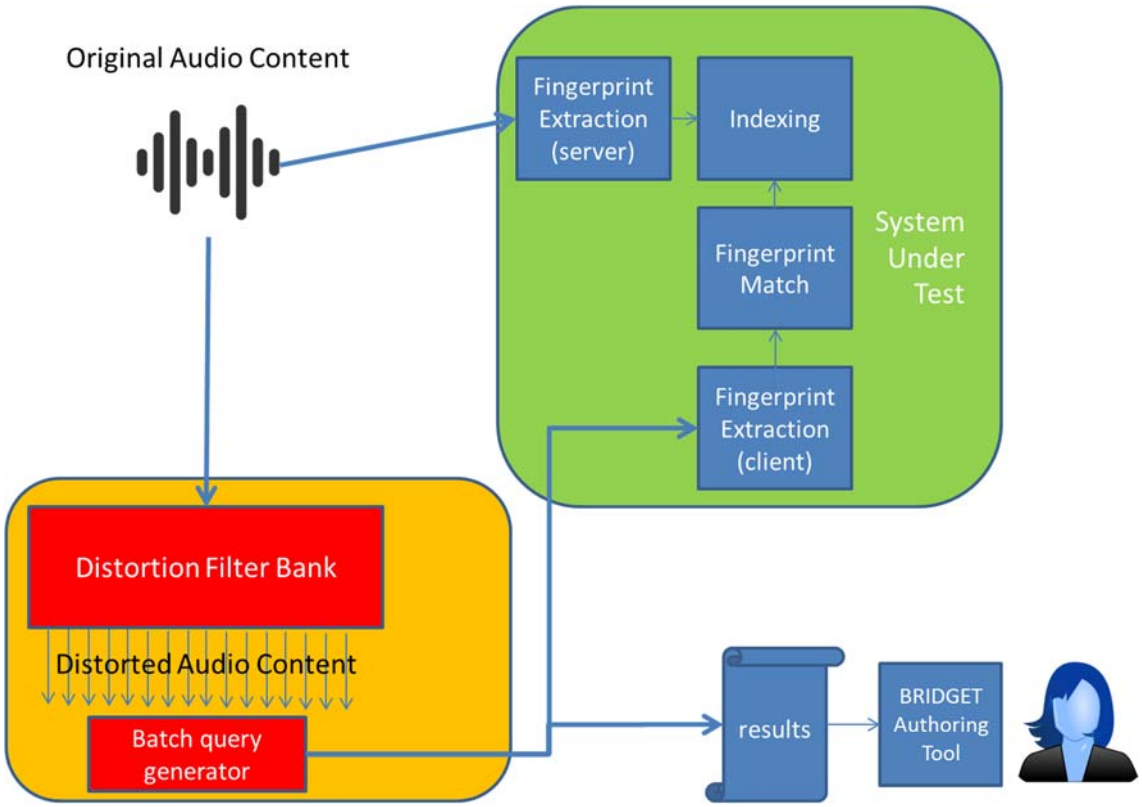

Figure 11. Architecture for Audio Quality Assessment for fingerprint-based synchronisation.

Figure 11 shows the envisaged architecture. In the picture, the green area named "System Under Test" (SuT) represents the audio fingerprint system being tested. It consists of a fingerprint extraction component, an indexing component and a fingerprint match component. The orange block is the quality assessment block, logically part of the BRIDGET Professional Authoring Environment. The original audio (i.e., the soundtrack of the programme being associated to bridgets) is first processed by the fingerprint extraction tools provided by the SuT and the results are stored in its internal database. Then, in the quality assessment block the same content is passed through a distortion filter bank simulating the target listening environment. In typical home listening conditions, this block contains filters like additive noise filters, reverb and echo adding filters, configurable with different levels of severity. The same architectural concept may accommodate more complex toolsets like virtual room environment simulation and multichannel sound propagation models. Regardless from the actual structure of the distortion filter bank and its implementation, the output of this block is a set of distorted audio clips which are used in the subsequent phase of the system as follows.

The distorted audio clips are used as query for the SuT fingerprint match module, which returns the result of the matching. This result typically consists in the ID of the found clip (or NULL if the query does not return any element) and the time of the match, i.e. the approximate point in the original audio clip which corresponds to the query with the maximal confidence. In general, the expected behaviour is that as the distortion parameters are made harsher, the quality of the results in terms of precision and recall are lower. This effect can be used in several ways:

- If there is only one SuT available, the result helps to identify which parts of the clip are more prone to content recognition errors, thus alerting the author not to plan any bridget in those parts. Similar issues can be originated and presented to the author by multiple places in the programme timeline at which exactly the same audio is placed (e.g., jingles, recorded parts);

- If multiple SuT are available, the results help identify the best one for a specific content item, or even – in a more sophisticated setting – which SuT to query at which estimated time in the programme timeline, or how to combine multiple query results over multiple SuT to enforce detection.

6.3.3 Initial Evaluation

To validate the approach we developed a simple testing suite utilising a temporary account on ACR Cloud content recognition services [36]. We implemented the following distortion filters, using available open source audio processing tools like sox and ffmpeg:

- Echo / Reverb
- Pink noise
- Ambient Noise
- Pitch shift

For the tests we used 7 clips taken from the content library collected in Task 8.5 for the project and documented in [38] and [39]. The server side indexing has been done using the ACR Cloud server's default parameters, while on the client side we used a 5 second chunking scheme for deriving the query clips from the distorted versions of the 7 longer clips. Before launching the batch queries, all query audio content has been normalised in terms of loudness following the specifications given in [40], to simulate the audio processing taking place at the head ends of modern broadcasting systems. The cascade application of [40] and of the distortion filter banks thus simulates with good approximation the transmission and decoding / rendering / ambient diffusion chain of a typical broadcast transmission concerning audio.

As an example, Figure 12, reports the average miss rate, i.e. the ratio between the wrongly recognised 5-second segments and the total number of 5-second segments, for each of the 7 tested clips in the case of increasing ambient noise (respectively 0%, 20%, 40% and 80% of the original signal). The data tells that Clip1, Clip6 and Clip3 are more critical to ambient noise than the others.

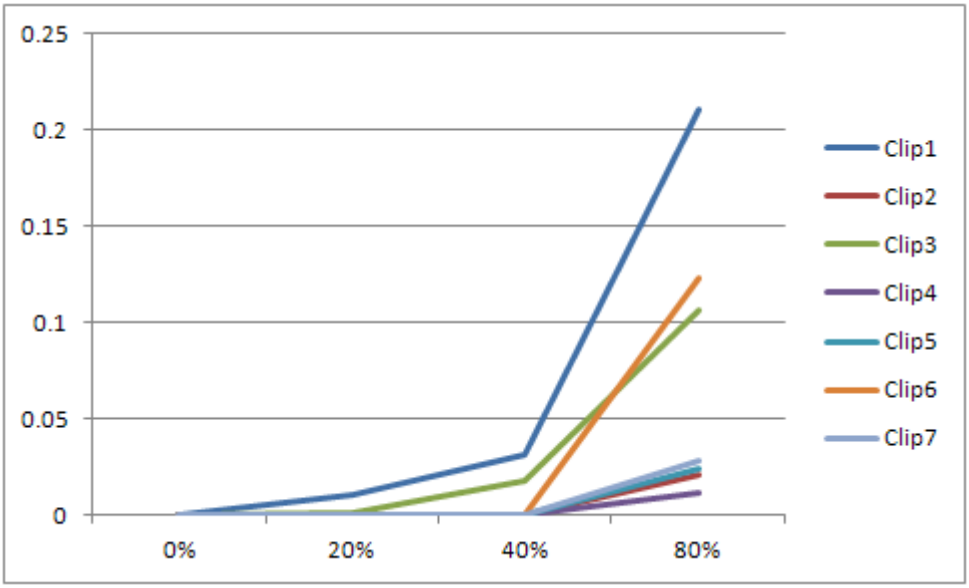

Figure 12. Average miss rate for the 7 tested clips.

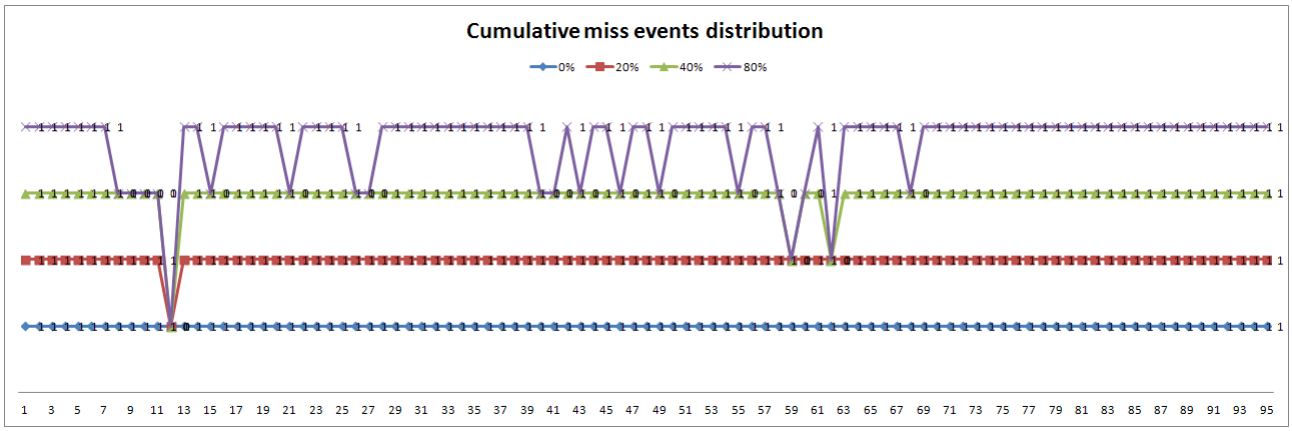

Figure 13. Cumulative miss events distribution over Clip1 timeline.

Figure 13 reports the cumulative miss events distribution over the timeline of Clip1, under the same experimental conditions. For each 5-second segment (X-axis) on the Y-axis are reported the sums of the match events (=1) and of miss events (=0) at each of the 4 ambient noise levels used. Lower sums indicate that misses occur at a higher number of ambient noise levels, and thus indicate segments that are relatively more sensitive to detection errors. In this type of graphs the vertical amplitude of the notches are related to the criticality of the segment at lower levels of artefacts, while horizontal amplitude gives account of how the segment interval becomes greater with increasing levels of artefacts.

Figure 14 reports the cumulative miss events distribution over the timeline of Clip4, which was amongst the less critical overall. This time the picture shows different effects, namely ambient noise, ambient noise plus additive reverb, and pink noise. While the latter generally affects all the timeline (as expected), the effect of additional reverb seems not to be critical in this case, except for one specific segment.

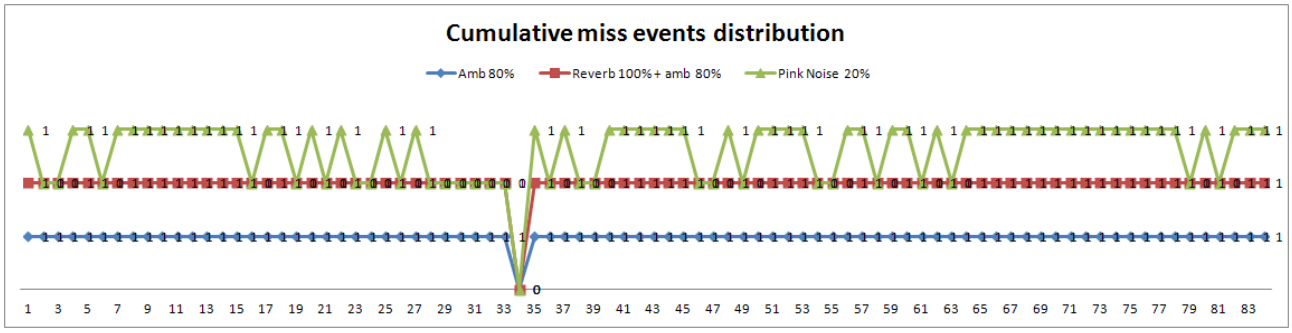

Figure 14. Cumulative miss events distribution over Clip4 timeline.

6.3.4 Conclusions

This study had the objective of devising an automatic method to detect portions of a programme timeline that are more critical than others in terms of quality of audio fingerprint-based detection, in order to support the bridget authors to optimise the quality of the user experience by minimising synchronisation problems. The method has been tested in an experimental setting simulating a real case, using a set of 7 programme soundtracks among the ones collected for the experimentation in the project, and interfacing with a state-of-the-art commercial audio fingerprint solution. Initial results indicate the validity of the approach, although more experiments are needed on bigger data sets and with an increased number of distortion conditions.

Figure 15 shows a possible integration of the results in the BRIDGET Authoring Tool. The coloured bar below the Audio Waveform widget visually indicates to the author what segments may or may not be critical w.r.t. audio fingerprint based content recognition.

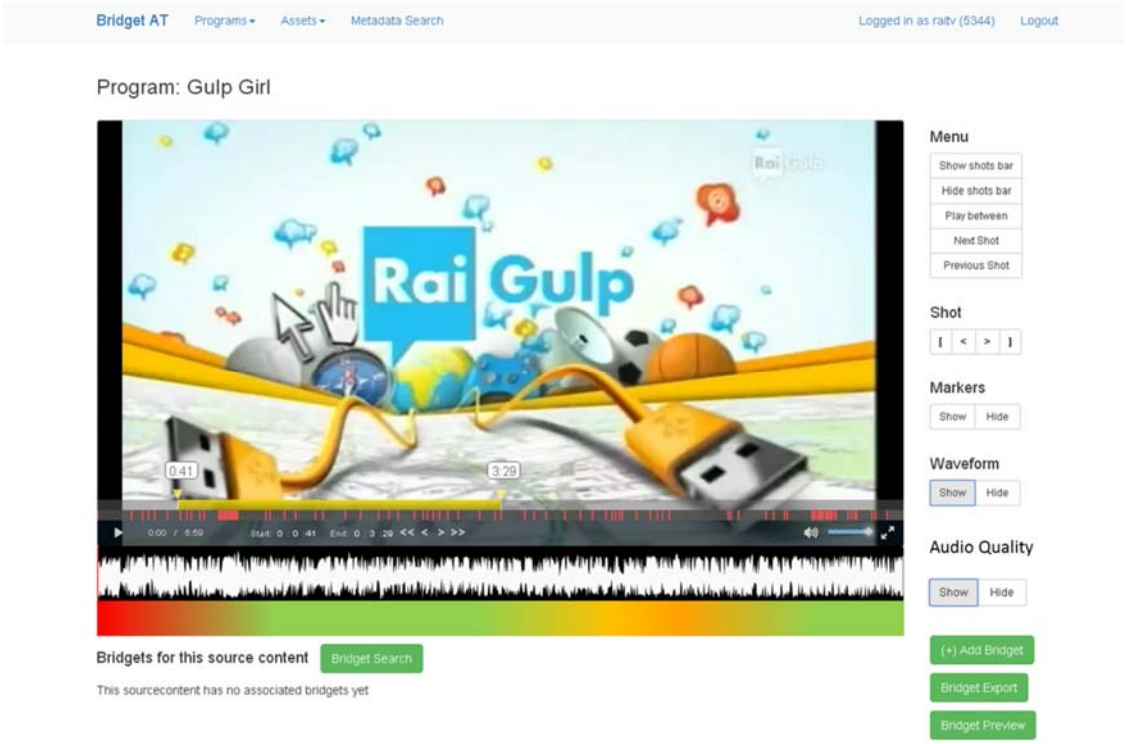

Figure 15. Example of Audio Quality Metrics Integration in the BRIDGET Authoring Tool.

6.4 GUI tool for visual quality assessment

Although the visual quality assessment tools are available as libraries and applications for integration in the BRIDGET workflow, we have also implemented a GUI tool for visual quality assessment, for quick desktop use, as illustrated in Figure 16 and Figure 17.

7 T4.4 Standardisation

As a result of the activities carried out within WP4, sometimes in collaboration with WP5, three proposals co-authored by BRIDGET researchers were submitted to MPEG [4][5][41] ([4] and [5] relating to the compact representation of a video for visual search and [41] relating to visual scene classification) and two output documents co-edited by BRIDGET researchers were approved by MPEG [6][42] ([6] relating to CDVA Experimentation Model and [42] being the final version of the MPEG-7 reference software completed in Version A of the tools [1]).

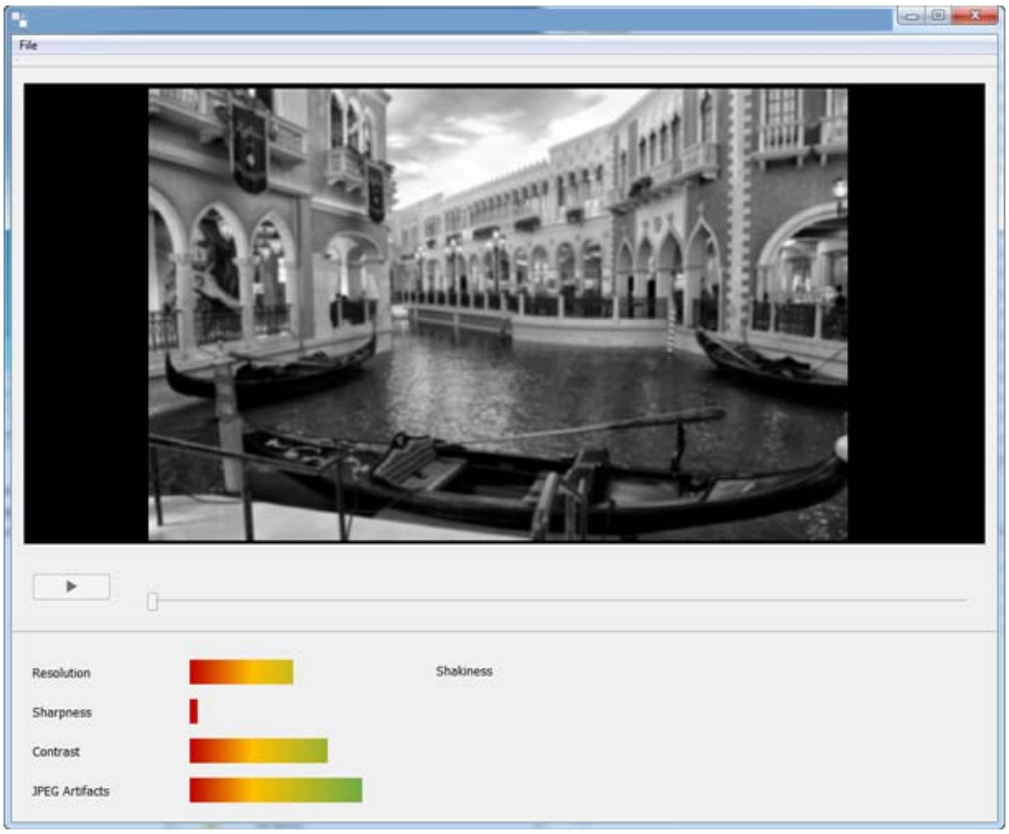

Figure 16. Visual quality assessment tool – image example.

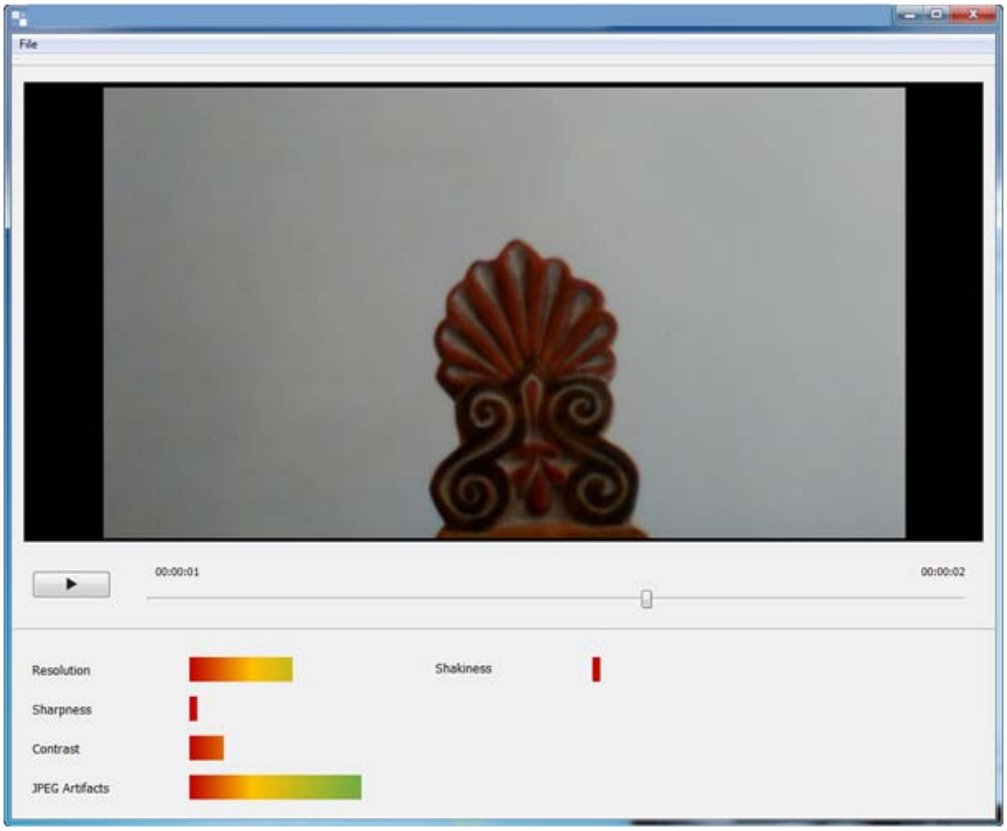

Figure 17. Visual quality assessment tool – video example.

8 Conclusions

This document presented an overview of the work carried out under WP4 for Version B of the Media Analysis Tools, i.e. concerning months M19 to M33 of the BRIDGET project. Task T4.1 (Media Structure Analysis) saw the development of new techniques for compact keyframe-based representation of a video for visual search purposes, the fastest of which was adopted by MPEG as part of the new CDVA Experimentation Model, and the completion of the component technologies for face clustering in video and programme analysis based on computational scene models, both of which provide new ways of finding relevant content segments for the creation of bridgets. In Task T4.2 (Media Annotation), our efforts were focused on visual scene classification, a component technology which provides additional contextual information for the selection of content for the creation of bridgets. Our chosen approach entails the combination of deep Convolutional Neural Network (CNN) methodology with RVD-W and has been shown to achieve higher performance on benchmark datasets (PASCAL VOC 2007, CALTECH 256, MIT SCENE 67) than the latest CNN-based representations. In Task 4.3 (Media Quality Assessment), we focused on the implementation of a core set of tools to evaluate the visual quality of media based on a variety of static and dynamic visual attributes, as well as audio characteristics, aiming to allow the presentation of the highest quality media to the end users and allow an assessment of the robustness of audio fingerprint – based synchronisation during bridget authoring. Finally, as a result of the activities carried out within WP4, three proposals co-authored by BRIDGET researchers were submitted to MPEG and two output documents co-edited by BRIDGET researchers were approved by MPEG.

References

[1] BRIDGET Deliverable D4.1 – "Media Analysis Tools - Version A", March 2016

[2] ISO/IEC JTC1/SC29/WG11 (MPEG), "Call for Proposals for Compact Descriptors for Video Analysis (CDVA) – Search and Retrieval", MPEG output doc. N15339, July 2015, Warsaw, PL

[3] ISO/IEC JTC1/SC29/WG11 (MPEG), "Evaluation Framework for Compact Descriptors for Video Analysis - Search and Retrieval – Version 2.0", output document N15729, October 2015, Geneva, CH

[4] Massimo Balestri (TI), Gianluca Francini (TI), Skjalg Lepsoy (TI), Miroslaw Bober (UNIS), Sameed Husain (UNIS), Stavros Paschalakis (VA), "BRIDGET Response to the MPEG CfP for Compact Descriptors for Video Analysis (CDVA) - Search and Retrieval", MPEG contrib. M37880, 114th MPEG mtg., San Diego CA, USA, February 2016

[5] Massimo Balestri (TI), Gianluca Francini (TI), Skjalg Lepsoy (TI), Miroslaw Bober (UNIS), Sameed Husain (UNIS), "BRIDGET report on CDVA Core Experiment 1 (CE1)", MPEG contrib. M38664, 115th MPEG mtg., Geneva, CH, May 2016

[6] Massimo Balestri (TI), Miroslaw Bober (UNIS), Wernel Bailer (eds.), "CDVA Experimentation Model (CXM) 0.2", MPEG output doc. N16274, 115th MPEG mtg., Geneva, CH, May 2016

[7] Open Intelligent Multimedia Analysis for Java - http://openimaj.org/ (Last accessed: July 25th, 2016)

[8] F. Vallet, S. Essid, J. Carrive, and G. Richard, TV Content Analysis: Techniques and Applications,chapter High-level TV talk show structuring centered on speakers' interventions, CRC Press, Taylor Francis LLC, 2011.

[9] Lawrence R. Rabiner. 1990. A tutorial on hidden Markov models and selected applications in speech recognition. In Readings in speech recognition, Alex Waibel and Kai-Fu Lee (Eds.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA 267-296.

[10] Shyju Wilson, C. Krishna Mohan, K. Srirama Murthy, "Event-Based Sports Videos Classification Using HMM Framework", *Computer Vision in Sports, Advances in Computer Vision and Pattern Recognition*, T.B. Moeslund et al. (eds.), Springer International Publishing Switzerland, pp. 229-244, 20 January 2015

[11] Daniel Jurafsky and James H. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (1st ed.). Prentice Hall PTR, Upper Saddle River, NJ, USA.

[12] F. Vallet, S. Essid, J. Carrive, and G. Richard, TV Content Analysis: Techniques and Applications, chapter High-level TV talk show structuring centered on speakers' interventions, CRC Press, Taylor Francis LLC, 2011

[13] BRIDGET Deliverable D8.5 – "User Trials and Feedback Analysis - Version A", November 2015.
[14] European Broadcasting Union, Editorial Format Classification schema, http://www.ebu.ch/metadata/cs/ebu_EditorialFormatCodeCS.xml (last accessed: July 21th, 2016)
[15] L. Xie, P. Xu, S.-F. Chang, A. Divakaran, and H. Sun, "Structure analysis of soccer video with domain knowledge and hidden markov models," Pattern Recognition Letters, 2004.
[16] E. O. Rajae and O. H. T. Rachid, "Highlights recognition and learning in soccer video by using HMM and the Bayesian theorem", International Conference on Multimedia Computing and Systems, (2009), pp. 304-308
[17] Haitao Yang, Jia Wang and Jingmeng Sun, "Detection of Corner Event Based on Hidden Markov Model in Soccer Video", International Journal of Signal Processing, Image Processing and Pattern Recognition Vol.8, No.12 (2015), pp.409-420.
[18] Shigeru Motoi, Toshie Misu, Yohei Nakada, Tomohiro Yazaki, Go Kobayashi, Takashi Matsumoto, and Nobuyuki Yagi. 2012. "Bayesian event detection for sport games with hidden Markov model", Pattern Anal. Appl. 15, 1 (February 2012), 59-72.
[19] X. Faguo, Z. Xiang and L. Tao, "The tracking method for soccer video in HOG and particle filter based on ball", Electronic Science and technology, vol. 9, (2013), pp. 36-40.
[20] Bingqing Qu, Félicien Vallet, Jean Carrive, Guillaume Gravier, " Content-Based Discovery of Multiple Structures from Episodes of Recurrent TV Programs Based on Grammatical Inference", Multimedia Modeling, Lecture Notes in Computer Science, vol. 8935, pp 140-154, 2015.
[21] A. E. Abduraman, S.-A. Berrani, and B. Merialdo, "Audio/visual recurrences and decision trees for unsupervised tv program structuring," in Proceedings of the 8th International Conference on Computer Vision Theory and Applications (VISAPP), 2013.
[22] Ping Ji, Liujuan Cao, Xiguang Zhang, Longfei Zhang, and Weimin Wu. 2014. News videos anchor person detection by shot clustering. *Neurocomput.* 123 (January 2014), 86-99.
[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge", International Journal of Computer Vision (IJCV), 2015.
[24] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition", in IEEE Conference on Computer Vision and Pattern Recognition, 2016
[25] H. Azizpour, A. Razavian, J. Sullivan, A. Maki, S. Carlsson, "Factors of Transferability for a Generic ConvNet Representation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015
[26] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in European Conference on Computer Vision, 2014.
[27] G. Cheng, P. Zhou, J. Han, "RIFD-CNN: Rotation-Invariant and Fisher Discriminative Convolutional Neural Networks for Object Detection", in IEEE Conference on Computer Vision and Pattern Recognition, 2016
[28] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, "Return of the Devil in the Details: Delving Deep into Convolutional Nets" , British Machine Vision Conference, 2014
[29] A. K. Tripathi, S. Mukhopadhyay and A. K. Dhara, P*erformance metrics for image contrast*, Image Information Processing (ICIIP), 2011 International Conference on, Himachal Pradesh, 2011, pp. 1-4. doi: 10.1109/ICIIP.2011.6108900
[30] Matti Niskanen, Olli Silvén, Marius Tico, "Video Stabilization Performance Assessment", Proc of IEEE International Conference on Multimedia and Expo, July 2006, pp. 405-408
[31] Sophia Bano, Andrea Cavallaro, "ViComp: composition of user-generated videos", Multimedia Tools and Applications, 2016, vol. 75, pp: 7187–7210
[32] Prarthana Shrestha, Hans Weda, Mauro Barbieri, Peter H.N. de With, "Video quality analysis for concert video mashup generation", Advanced Concepts for Intelligent Vision Systems: 12th International Conference, ACIVS 2010, Sydney, Australia, December 2010, pp. 1-12
[33] Jianbo Shi, Carlo Tomasi, "Good Features to Track". Proc. of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, June 1994, pp: 593-600
[34] Jean-Yves Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker", Intel Corporation, Microprocessor Research Labs, 2000
[35] http://www.shazam.com/
[36] https://www.acrcloud.com/

[37] www.civolution.com
[38] D8.3, “Material Library and Ground Truth – Version A”, June 2015
[39] D8.7, “Material Library and Ground Truth – Version B”, June 2016
[40] EBU R128, “Loudness Normalisation And Permitted Maximum Level Of Audio Signals”, European Broadcasting Union, June 2014.
[41] Miroslaw Bober (UNIS), Stavros Paschalakis (VA), Alex Freestone (UNIS) “Evaluation of MPEG-7 descriptors in scene classification tasks”, MPEG contrib. M37466, 113th MPEG mtg., Geneva, CH, October 2015
[42] Stavros Paschalakis (VA), Karol Wnukowicz (VA) (eds.), “Text of ISO/IEC DIS 15938-6:201X Reference software (2nd edition)”, MPEG output doc. N15368, 111th MPEG mtg., Warsaw, PL, July 2015
[43] BRIDGET Deliverable D5.3 – “Media Search Tools - Version B”, July 2016