Automatic Selection of Live User Generated Content

Stefanie Wechtitsch, Marcus Thaler, Albert Hofmann, Andras Horti, Werner Bailer
JOANNEUM RESEARCH, DIGITAL
Steyrergasse 17, 8010 Graz, Austria
{firstname.lastname}@joanneum.at

Wolfram Hofmeister, Jameson Steiner, Reinhard Grandl
Bitmovin
Lakeside B01, 9020 Klagenfurt, Austria
{firstname.lastname}@bitmovin.com

ABSTRACT

User generated content (UGC) is a valuable source for improving the coverage of events such as concerts, festivals or sports events. Integrating UGC in existing professional production workflows is particularly challenging in live productions. UGC needs to be checked for quality in this case, and metadata captured by the mobile device and extracted from the content are relevant for filtering the UGC streams that go into a live production system. We propose a system for capturing live audio and video streams on a mobile device, performing automatic metadata extraction in real-time and indexing the metadata for access by a production system. The system receives an audio, video and metadata stream from the mobile device, and creates additional metadata from the ingested audiovisual content. The metadata (e.g., location, quality) are then used to automatically select and rank streams, either selecting a stream to show to a viewer or a list of streams from which a human operator can select.

ACM Classification Keywords

H.5.1 Information Interfaces and Presentation: Multimedia Information Systems; I.4.1 Image Processing and Computer Vision: Digitization and Image Capture

Author Keywords

user generated content, content selection, sensor, mobile, content analysis, live

INTRODUCTION

User generated content (UGC) is a valuable source for improving the coverage of events such as concerts, festivals or sports events. In order to integrate user generated content into existing production workflows, both the quality of UGC needs to be checked and metadata needs to be extracted. Such metadata, together with sensor information from the mobile device, will help the production team to assess the context, quality and relevance of the user contribution.

A particularly challenging scenario are live productions, where such metadata needs to be available with small latency. Live streaming of UGC from mobile devices has recently gained popularity, among others through the use of apps like Meerkat[1] or Periscope[2]. However, these apps provide a stream "as is" for viewing on the web, without integration in production workflows. The end users manually need to select a particular stream and have to discover themselves whether there are alternative streams of the event available, in case the one they are watching becomes boring or turns out to be of insufficient quality (both are unfortunately not so uncommon on today's live streaming platforms). Thus, a system that integrates professional and user generated content of an event needs to provide support for content selection. Content selection can be supported by metadata either captured on the mobile device (e.g., capture location) or extracted from the content (e.g., content quality).

We propose a system for capturing live audio and video streams on a mobile device, performing automatic metadata extraction in real-time and indexing the metadata for access by a production system. The system receives an audio, video and metadata stream from the mobile device, and creates additional metadata from the audiovisual content. All metadata are available as a stream (with low latency from the extraction), and are indexed in a metadata store. Metadata needed in the real-time process can be read directly from the stream, and earlier metadata can be queried from the store. The metadata are used to automatically filter content that matches defined quality levels, to select the best stream among alternative ones and to provide a set of content options.

The rest of this paper is organised as follows. The Section Capture and Analysis System describes the capture tools and the analysis framework and modules. The approach to content selection and the results are discussed in Section Content Selection, followed by a Conclusion.

CAPTURE AND ANALYSIS SYSTEM

System Overview

Figure 1 shows an overview of the proposed system. The system consists of a dedicated capture app, which sends video, audio and metadata as separate streams. This saves the muxing/demuxing effort and also facilitates distributed processing of different modalities on different machines in the cloud. All data are provided as RTP streams. The processing system (dashed box in the diagram) performs the necessary decoding and transformation for the content, and also includes a set of interconnected analysis modules. These modules may not only use the content as input, but may also use metadata from the device or from other modules. All extracted metadata are provided as streams again, and a logging module listens to these streams and indexes data in the metadata store. The audiovisual streams can be connected to viewers or to an editing system. A web application performs content selection

[1] https://meerkatapp.co

[2] https://www.periscope.tv

4th International Workshop on Interactive Content Consumption at TVX'16, June 22, 2016, Chicago, IL, USA

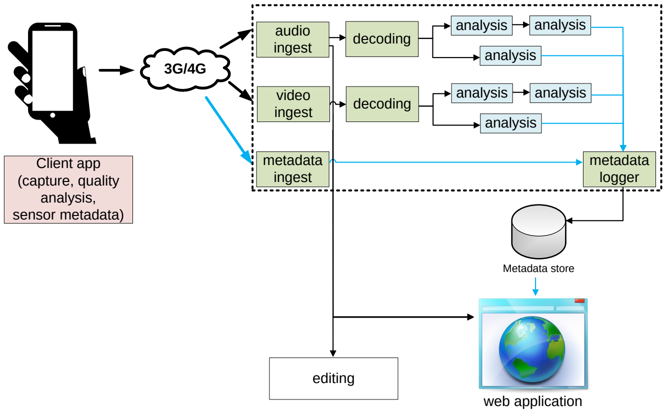

Figure 1. Overview of the proposed system

and displays the audiovisual data together with the extracted metadata.

We decided to build on an existing framework with many standard components which is able to handle the decoding of commonly used media formats. Thus, the GStreamer[3] open source multimedia framework is used for this purpose.

Content Capture

The integrated capture application for Android enables users to perform quality analysis while capturing sensor data and streaming captured video. The main features are: (a) audio and video recording, via the built-in microphone and camera respectively, (b) metadata capturing from different sensors available on the device, (c) on-device analysis of captured essence to meet quality constrains, (d) en-/transcoding and packaging of recorded content and (e) the up-streaming functionality to servers for processing.

Raw video and audio data is captured through the camera and microphone of the device and encoded using Android's *MediaCodec API*, while at the same time the quality of video frames is analysed. As encoding of video frames typically is more time consuming than for audio frames, a buffer synchronizes both streams. Once the encoding for a frame has finished, it is committed into the buffer and/or sent to an RTP packager. In parallel, a live on-device preview, containing visual quality-related notifications, as discussed in the upcoming sections, is presented. Synchronization is done by keeping track of the latest PTS for each stream.

During initialisation of the capture application, various types of static metadata (such as properties and technical parameters of the mobile device) are sent to the processing system. Moreover, together with the content, sensor metadata of the on-device sensors are captured to support real time quality analysis, by recording the following sensors: location, accelerometer, gyroscope, magnetic field, orientation, rotation, ambient light, proximity and pressure. For example, the accelerometer can be used to detect fast and shaky movements of the mobile device. For the analysis of video frames, several lightweight algorithms, which identify defects in the captured content, were implemented. Thus, contributing users who have this app installed are capable of performing visual quality analysis on the mobile device while capturing video and obtaining direct feedback about the quality of the captured content. The application continuously measures sharpness, noise, luminance, exposure and detects the use of brightness compensation before streaming captured video [9]. This way, users are notified during capture if one of the quality measurements falls outside the target range. For each quality measure, an overlay including a descriptive icon and message is displayed to immediately notify the user to avoid the quality impairment.

Algorithms for sharpness, noise and over-/underexposure detection have been implemented in the app. Details on these quality algorithms can be found in [9]. For sharpness estimation we use the Laplace operator for edge detection. By subsampling the response image into equally sized blocks, the sharpness value for each block is represented by its maximum slope response of the corresponding edges. The blocks with the highest values (strongest slopes) are selected to obtain the global sharpness value. For noise estimation the luminance component of each of the analysed image is calculated and the similar block scores from the sharpness estimation are used to find the most homogeneous blocks (with few edges). For the remaining blocks the average absolute differences between the original and the median filtered image are computed representing the block's noise score. The global noise level is then estimated by taking the median of the appropriate block values. To detect the use of brightness compensation the average brightness progression of images within a certain time frame is approximated. If the summed up positive or negative brightness variation values exceeds a predefined threshold, the algorithm reports overexposure or underexposure, respectively. Using a Samsung Galaxy S5 the runtime for all proposed quality analysis algorithms for one frame of the captured HD image sequence is about 200ms. Due to gradual temporal changes of image quality problems (e.g., noise) it is sufficient to process every sixth frame, enabling real-time operation.

Content and Metadata Streaming

In order to perform the RTP streaming, the encoded audio and video frames are pushed into a buffer and wrapped into the ISOBMFF file format. During the entire recording session of a user, each video segment is uploaded to the processing system. After the capture is finished, the full video is accessible via built-in Android functions. In order to stream packets over RTP, a packetizer which generates the RTP headers and splits the data into several packets (if necessary) is used. Every encoded audio/video frame is pushed into the respective packetizer. To ensure synchronization, a similar buffer-based approach as described in Section Content Capture is applied.

The captured video metadata such as device and sensor metadata are accumulated locally with records indexed by time and type of metadata. Incremental metadata are made available as segments of the metadata stream. The segments available as strings in JSON format (following the format defined in [2]) are sent in chunks periodically as a UDP stream from the mobile device to the processing system. For the analysis chain a dedicated module was developed to receive the UDP packets,

[3] http://gstreamer.freedesktop.org

put the string chunks together to restore the JSON message and forward them to the metadata store handler. The analysis chain receives the media stream over the RTP protocol as a UDP stream. The GStreamer standard components are taking care of the reception, management and decoding of the audio and video streams and finally converting the video frames to RGB8 images.

Analysis Framework

The analysis components are implemented as GStreamer plugins and we create a flexible and powerful analysis chain by combining the standard GStreamer modules and our components. The GStreamer framework's messaging concept assures the optimal configuration for each plugin. For our purposes we applied a simplified configuration manner to create simple plugins with arbitrary number of input and output pads having different formats though in some cases manual hints are necessary. Since the standard RTP stream contains only relative timestamps, the synchronization of audio and video content from different devices is realized on the basis of the timestamps of the RTCP stream. A custom plugin handles the extraction of the timestamps and the difference calculation between the internal clock and the absolute timestamp (e.g., synchronized with a PTP [7] clock).

Analysis Modules

The visual quality of a user generated video is a good indicator for an early decision whether the video might be useful to be considered, e.g. in a production, or whether it can conversely be sorted out due to an insufficient quality. In particular, the quality is an important decision criterion when having a huge amount of data available which should be reduced automatically. In order to obtain an overall quality measure of a user generated video, all available individual quality indicators are considered. The metadata received from the mobile device directly as well as the more complex quality measures obtained after transmitting to the server are fused as described in the following.

As mentioned earlier, the mobile device provides quality estimates of how blurry the content is, how much noise it contains, if there are parts suffering from over or under exposure and if the video was recorded under shaky conditions or not. On the server side, we may get additional measures by using a more complex algorithm for the blurriness and the contained noise. Furthermore, an estimate for macro-blocking artefacts is determined. At first we compute one representative value for each measure and combine them with all others by fusion.

All these measures may be available or not, they are optional. The sampling steps for each measure are individual but constant over the whole duration of the video. Noise and blurriness can be measured on both the mobile device or on the server. Since on the server a more complex algorithm can be used, the results may differ a bit. Depending on the use case and the computational complexity that can be afforded, a subset of measures is computed and used for the overall quality measure. Under real-time requirements, we rather use the blur and noise estimation from the mobile device. Independent of the source of the blur and noise measure the computation for the overall quality measure stays the same.

All sampled values are collected individually for each measure and are sorted in increasing order. An appropriate subset of the sorted list is chosen to compute the average value for each measure. It was empirically established that a well correlated measure emerges if the subset is chosen from the higher values (high values indicate lower quality in this case), causing bad quality frames to have a higher influence on the result than good quality frames (so bad quality frames are over weighted compared to good quality frames). Thus, a video where only parts appear as very blurry or have a high level of noise will be rated as being of poor quality. The quality is represented by a floating point number in the range of 0 to 1, where 0 indicates excellent quality and 1 corresponds with very poor quality. This representation is used for each individual measure as well as for the overall quality value.

We have chosen to use the upper 25% of quality scores (i.e., representing the 25% segments with worst quality) to compute an average value for all measures of the involved quality metric. Finally, those values have to be combined. Simply averaging the individual measures is not a good strategy, since having one or two bad quality measures out of our set of five metrics would result in a non appropriate quality measure, distorted by the good quality measures.

The measure which causes the highest impact on the content quality should have the highest impact on the final quality measure. Thus, we apply a weighted sum where the highest values are disproportionately weighted higher.

Metadata Store

The data exchange between the analysis platform and the production system is realised via a metadata store. This metadata store is a persistent hybrid repository accessible over a REST interface. Short term data are kept in a Redis[4] in-memory data structure store whereas long term data are archived into a MySQL[5] database. The repository type is transparent for the client, the difference is only noticeable in the query response time.

The extracted metadata are used for automatic content filtering of the UGC streams, e.g., discarding streams based on overall quality metadata or their location. By querying the metadata store with the appropriate criteria, the relevant streams can be selected for live editing.

CONTENT SELECTION

When multiple concurrent live streams are available for an event, automatic and real-time selection of the best quality content is advocated. The selection strategies implemented so far are rule-based. They use the metadata available in the metadata store as input, i.e., the metadata captured on the device and the results from quality analysis on the mobile device and the server. The metadata does not only contain raw sensor and analysis data, but also the annotations of segments where pre-defined minimum quality limits have been violated.

[4] http://redis.io

[5] http://www.mysql.com

Prior work on automatic video production in [1] and [4] aims for automating the selection of captured content, but these approaches have been developed for professional content and therefore do not exploit video quality at all as a cue for selection. For supporting or automating home video editing some specific approaches for quality based video production and selection have been studied, e.g., in [10] and [6]. Although they address some quality detection requirements specific for user generated content, these approaches are intended to be applied in an off-line fashion on pre-recorded video. An approach for creating mashups of multiple camera concert recordings using video quality cues has been proposed [8], which comes closest to our requirements. Signal quality measures extracted from the individual recordings are used for selecting best quality segments. The approach is applied in a file-based off-line scenario, an on-line real-time scenario has not been investigated.

Approach

In our approach, content is discarded when quality metrics violate thresholds for minimum quality, and the same thresholds are applied for all streams. In addition, the average quality measure determined as described in Section Analysis Modules is compared against a threshold. Temporal filtering of selection decisions is applied, in order to avoid switching streams on and off when quality values fluctuate around the thresholds. The choice of the size of the temporal filter is a trade-off between more frequent switching between streams and more robust decisions that come at the cost of higher latency of the analysis result. If the system is used in a semi-automatic mode, an operator may override automatic filtering decisions based on quality if the clip is the only showing content that should be included.

After filtering, ranking of the remaining streams is applied. For content-based ranking, we use a strategy that is similar to approaches that boost diversity in search results: (i) we prefer streams showing a different area of the event over more of the same, and (ii) from a group of similar ones we select the one with the best quality. We use location information, where available from the metadata of the stream and/or additionally from determining the visual overlap between streams as described in [3]. The spatial distance and the visual similarity are used to determine a pairwise measure for diversity between two streams, in analogy to the affinity graph described in [11]. However, as we do not start from a specific query, we always rank the entire set of streams available at a current time segment. In the current implementation, we only update the location metadata when streams end or are added. The ranked list of streams can be provided as input to a user interface, or an automatic method can be used to select from the top entries in the list, such as the virtual director approach proposed in [5].

Results

The content selection application is implemented as a web application, which implements the selection rules and also includes an HTML5 metadata viewer (see Figure 2). The metadata store is polled in defined intervals for recent data. The selection rules are executed and the UI is updated accordingly. As described in Section Analysis Modules, both quality annotations done at the mobile client and server are used. For each annotation type, a chart with the continuous quality measure is shown, and an additional event view displays segments that do not meet predefined quality standards. This level of detail is only shown for the currently selected video stream. For other concurrent streams, the overall quality metrics are additionally retrieved from the metadata store and visualised in a compact form. When switching to another stream, the views are switched accordingly. To provide audiovisual content to the HTML5 viewer, the incoming media stream is re-streamed by the analysis platform. This can be done as RTP stream with very low latency (requiring a browser plugin) or providing a stream for consumption by an HTML video player, with possibly higher latency.

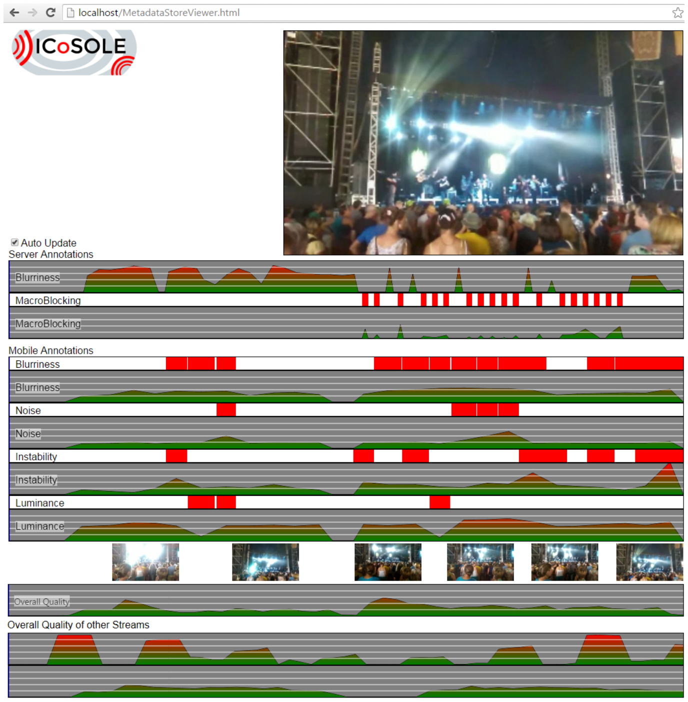

Figure 2. Web-based content and metadata visualisation. For each quality metric, a line chart with the continuous evolution of the measurement is shown. An additional event view on the top of each quality metric highlights segments that do not meet predefined quality standards (indicated by a red bar).

CONCLUSION

In this paper, we have presented a framework for automating content selection in order to complement professional coverage of live events such as concerts, festivals or sports events with user generated content. We have described a system for capturing live audio and video streams on a mobile device, performing automatic metadata extraction in real-time and indexing the metadata for access by a production system. The system creates additional metadata from the audiovisual content, and all available metadata are then used for automatic filtering and ranking of streams, using a rule-based approach.

ACKNOWLEDGMENTS

The authors would like to thank Jürgen Schmidt and Mario Sieck from Technicolor. The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 610370, ICoSOLE ("Immersive Coverage of Spatially Outspread Live Events", http://www.icosole.eu).

REFERENCES

1. Gulrukh Ahanger and Thomas D. C. Little. 1998. Automatic Composition Techniques for Video Production. *IEEE Trans. Knowl. Data Eng.* 10, 6 (1998), 967–987.
2. Werner Bailer, Gert Kienast, Georg Thallinger, Philippe Bekaert, Juergen Schmidt, David Marston, Richard Day, and Chris Pike. 2015a. *Format Agnostic Scene Representation v2*. Technical Report D3.1.2. ICoSOLE project.
3. Werner Bailer, Marcus Thaler, and Georg Thallinger. 2015b. Spatiotemporal Video Synchronisation by Visual Matching. In *Proceedings of the 3rd International Workshop on Interactive Content Consumption co-located with ACM International Conference on Interactive Experiences for Television and Online Video (ACM TVX 2015)*.
4. Xian-Sheng Hua, Lie Lu, and Hong-Jiang Zhang. 2004. Automatic Music Video Generation Based on Temporal Pattern Analysis. In *Proceedings of the 12th Annual ACM International Conference on Multimedia (MULTIMEDIA '04)*. 472–475.
5. Rene Kaiser and Wolfgang Weiss. 2013. *Virtual Director*. John Wiley & Sons, Ltd, 209–259.
6. Tao Mei, Xian-Sheng Hua, Cai-Zhi Zhu, He-Qin Zhou, and Shipeng Li. 2007. Home Video Visual Quality Assessment With Spatiotemporal Factors. *IEEE Trans. Cir. and Sys. for Video Technol.* 17, 6 (June 2007), 699–706.
7. The Institute of Electrical and Electronics Engineers. 2008. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, version 2. (2008).
8. Prarthana Shrestha, Peter H.N. de With, Hans Weda, Mauro Barbieri, and Emile H.L. Aarts. 2010. Automatic Mashup Generation from Multiple-camera Concert Recordings. In *Proceedings of the 18th ACM International Conference on Multimedia (MM '10)*. 541–550.
9. Stefanie Wechtitsch, Hannes Fassold, Marcus Thaler, Krzysztof Kozłowski, and Werner Bailer. 2016. Quality Analysis on Mobile Devices for Real-Time Feedback. In *MultiMedia Modeling*. Springer, 359–369.
10. Si Wu, Yu-Fei Ma, and Hong-Jiang Zhang. 2005. Video quality classification based home video segmentation. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*.
11. Benyu Zhang, Hua Li, Yi Liu, Lei Ji, Wensi Xi, Weiguo Fan, Zheng Chen, and Wei-Ying Ma. 2005. Improving Web Search Results Using Affinity Graph. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '05)*. 504–511.