Context not Content: A Novel Approach to Real-Time User-Generated Video Composition

Denny Stohr*, Stefan Wilk*, Iva Toteva*, Wolfgang Effelsberg*, Ralf Steinmetz†

Technische Universität Darmstadt

**Distributed Multimedia Systems,{dstohr, swilk, wolfgang.effelsberg}@cs.tu-darmstadt.de, iva.toteva@stud.tu-darmstadt.de*

†Multimedia Communications Lab, ralf.steinmetz@kom.tu-darmstadt.de

***Abstract*—Instant sharing of user-generated video recordings has become a widely used service on platforms such as YouNow. Yet, it still poses technical challenges, as mobile upload speed and capacities are limited. One proposed solution to address these issues is video composition. It allows switching between multiple video streams–selecting the best source for a given time–for composing a live video of a better overall quality for viewers. Previous approaches require visual analysis of the video streams, usually limiting the scalability of the system. In contrast, our work allows the stream selection to be realized solely on context information, based on video- and service-quality aspects from sensor and network measurements. The implemented monitoring service for context-aware upload of video streams is evaluated in varying network conditions, with diverse user behavior, including camera shaking and user mobility. We show that a higher efficiency for video upload as well as QoE for viewers can be achieved.**

I. INTRODUCTION

Live streaming of User-Generated Video (UGV) has experienced a widespread adoption driven by increasingly powerful mobile devices and networks. A common scenario when a significant number of users record and want to share their experience with friends or the public are entertainment happenings, such as sports events and concerts. As the spectators have a similar view on the event of interest, this has naturally given rise to collaborative video sharing.

Essential challenges for live streaming UGV are presented by the rather low quality of the videos and the limited upload capacities of the wireless network. Viewers often experience poor video quality due to degradations introduced during recording such as camera shake, or because of compression-induced artifacts. Despite the compression of the video streams, it is challenging to upload recordings as bandwidth requirements are high (e.g. compared to text messaging), and users compete for scarce upload capacity in wireless networks. Furthermore, the available capacity for a future video streaming session is nearly unpredictable and varies with the current network utilization. This either leads to an underutilized potential to stream higher video bitrates or interrupted video streams due to over-utilized capacities.

We propose a monitoring service to observe network conditions and user behavior for supporting video composition applications in the selection of the best video stream at a given time, thus achieving a better video quality in composition.

The major contributions of this work are the design and implementation of this novel monitoring service for mobile devices, including algorithms for the reliable detection of network conditions, user activity and video quality in real-time. The proposed monitoring service allows the prediction of video quality independent of the Mobile Video Broadcasting Service (MBS) and could thus be used with any existing or future streaming services. For the evaluation purposes in this work, the proposed system is evaluated using Live Video Upload (LiViU) [16] which features a live video upload and composition framework for mobile devices.

The remainder of this paper is structured as follows: In Section II we give a brief overview on some of the existing MBSs and show their characteristics as well as technical foundation. Section III introduces the model and architecture of the monitoring service, while Section IV gives an overview of LiViU. The evaluation is described and discussed in Section V, and conclusions are presented in Section VI.

II. BACKGROUND AND RELATED WORK

A. Recently Proposed MBS

The work of El Essaili *et al.* [2] investigates the uploading of video when the scheduling in an LTE network can be controlled. They propose a centralized quality-oriented decision for the uplink transmission of client-side video, which is controlled by the eNodeBs. In contrast to their work, we focus on providing an efficient application layer video uploading scheme, without direct control on lower layer network components, by using active network measurements.

Seo *et al.* [8] discuss how Moving Pictures Expert Group (MPEG) dynamic adaptive streaming over HTTP (DASH) can be used for the upload of media by leveraging hypertext transfer protocol (HTTP) POST requests to continuously upload video segments. The proposed system achieves transcoding and transmission of a 480p video with a start-up delay of approximately the duration of one segment under good Wireless Fidelity (WiFi) conditions.

Johansen *et al.* [4] propose a system designed to generate video segments and upload them immediately in order to generate a low-delay video streaming experience. The authors report on dynamically adapting the bitrate of a video during the streaming session, which is a sophisticated concept to cope with under varying network conditions.

A recently proposed system by Siekkinen *et al.* [10] focuses on a better uplink utilization using scalable video codec (SVC) and thus adapting to changing network conditions. Unfortunately, at this point there is no efficient implementation of this video encoding standard for mobile devices.

B. Mobile Video Upload Monitoring

In previous work, a concept for MBS-monitoring was proposed [12], introducing the *director* concept for selecting uploading video sources based on monitored information such as bandwidth, activity, location and device shaking. By selecting only devices that promise a good video quality given their current context, video not offering a good-quality stream does not have to be uploaded.

Richerzhagen *et al.* [6] employ a similar concept for source selection in collaborative video upload. The proposed system selects uploading sources based on derived data such as the client's network bandwidth, where unused nodes do not upload video. Further, they suggest local network resource sharing between nearby clients to address situations where the upload capacity is insufficient for transmission of the desired video source. The system's performance has been evaluated in a simulation, showing that the hybrid concepts allow continuous playback for low bandwidth situations in cases where direct upload strategies do not deliver such good results.

A deployed monitoring system for live video upload was analyzed by Stohr *et al.* [11]. The observed platform *YouNow* collects various device and network related parameters (such as connection type and provider) during live video upload. Depending on the used network type, a significant variation in the video bitrate could be seen, suggesting a variation of the video quality based on the current connection context. As the system does not allow video composition, a selection between multiple source-streams is not addressed.

C. Mobile Video Composition

The proposed monitoring system for UGV is evaluated in combination with *LiViU* [16] which is responsible for live recording and upload of video streams from mobile devices. Further, it allows for live video *composition* and broadcasting [15].

Other composition systems have been proposed, as e.g., Engström *et al.* [3], considering video quality to compose a video mix. It is a semi-automatic system and therefore not considered for our work.

MoviMash [7] introduces an automatic composition, combining a video quality metric with an analysis of video recording degradations. The latter describe phenomena common in user-generated video, caused by lack of skills of the recording user or technical limitations of the recording device. Their video composition algorithm combines video streams from different sources, analyzes them and neglects those views in which recording degradations occur.

Cricri *et al.* [1] show a first step towards replacing video content analysis with mechanisms that leverage different sensors in the recording devices to compose the video. The advantage of such an approach lies especially in the significantly reduced processing time. Cricri *et al.* solely inspect the camera movement for composition decisions. Thus, network related aspects, addressed in our work, are not considered.

A sophisticated, formalized automatic composition is introduced by Shrestha *et al.* [9] (see details in Section IV-C), offering an algorithm using an objective quality function. It maximizes the quality for music video clips, integrating the completeness of the composed video, suitable cutting points, length of the individual shots and diversity of views.

We leverage the approach by Shrestha *et al.* in our work and extend it to be used in combination with ideas of Cricri *et al.* to make decisions in real-time by not inspecting the video itself, but by employing auxiliary metrics provided by the monitoring service.

III. MONITORING FRAMEWORK

The key goal for the proposed Mobile Video Composition (MVC) monitoring framework is the efficient measurement and collection of client and network metrics to support the selection and upload of live streams to a server, thus allowing quality optimization of the live video. Ultimately, the aim is to provide a better overall quality of experience (QoE) for viewers of generated collaborative live streams and to eliminate the upload of unused streams. Metrics in four categories provide indicators for the expected QoE of recorded streams and can be classified into: *i)* Static parameters such as camera properties–they are only collected and transmitted once to the server, at the time when a session is initiated; *ii)* Activity-related dynamic metrics including shake estimation and activity recognition–derived based on device sensor information; *iii)* Network properties–measured and compared with an estimation of the uploading bandwidth; *iv)* Content relevance for a given Point of Interest (POI)–derived based on the recording location. All non-static metrics are periodically collected.

We will now introduce the system's components and architectural considerations beginning with the employed messaging protocol.

A. Messaging Protocol

For messaging, an asynchronous client-server pattern[1] was implemented using JeroMQ. The process of establishing a monitoring session is explained next.

First, the client sends a `JoinInfo` message to the server, which includes static information about the device, such

[1] http://zguide.zeromq.org/page:all/#toc76

as the model and camera characteristics (megapixels, image stabilization capabilities etc.), along with dynamic parameters, including the current location. The server replies with a `JoinInfoAck` message, which not only acknowledges the receipt of the `JoinInfo` message, but also specifies an update interval at which the device is to periodically send information about the quality of service (QoS) parameters. After that, the client prepares and sends such messages, which include positioning information, shake-related data, presumed activity that the holder of the device is currently performing, and network information. The server may also request an immediate update if more up-to-date data is required. At the end, the client can gracefully quit by sending a `LeaveInfo` message.

In addition to this messaging protocol, the server checks the time of arrival of `UpdateInfo` messages from each active device. If the maximum time between update intervals is exceeded, the server assumes that the node has gone offline and removes it as a potential stream producer.

B. Metrics

Information is collected on the device and sent to the monitoring server with the specified periodicity. This includes location data for the device, shaking, detected activities, the network connection type. In addition, the server stores the results of the recent upload bandwidth estimations. In the following, we will describe the collection process for each of these metrics in detail.

The shake detection algorithm is implemented using data from the linear accelerometer of the device. The values from the sensor are recorded along with a timestamp, and are periodically sent to a shake detector. The shake detection estimates whether there is shaking based on a predefined threshold in the allowed deviations of the values. If shaking is detected, its amplitude, duration, and velocity are calculated.

Information about the current user behavior, which potentially impacts the quality of the recording, is based on the activity detection implemented using Google Play Services.[2] Possible detected attributes can be found in Table I.

With regard to the network connection, the first parameter that the MBS Monitoring Framework records is the type of network that the device uses. Devices connected to WiFi or LTE networks are preferred over those using a slower connection, given that such networks provide superior streaming quality (see Table I).

Last, the available bandwidth is estimated using active probing. The architecture allows implementing various algorithms and reusing most common functionality used for bandwidth estimation. This provided infrastructure includes a control communication channel, which uses standard Java TCP sockets, and a packet channel for UDP datagrams, which is explained in detail in Section III-C.

[2]https://developers.google.com/android/guides/overview

C. Bandwidth Estimation

For bandwidth estimation, the WBest algorithm proposed by Li *et al.* [5] was implemented, as it provides fast and low-overhead estimates in a mobile context. In the process of bandwidth detection, WBest first sends pairs of packets to determine the effective capacity C_e, followed by several trains of packets sent at the rate of the estimated effective capacity. Afterwards, the achievable throughput R is calculated based on the number of bytes divided by the packet dispersion, while the achievable bandwidth A is derived by: $A = C_e(2{-}C_e/R)$. For a detailed description of the algorithm, we refer to the authors' original publication [5].

D. Recording Score

Indicator	Weight	Attributes	Value
Activity	0.5	Still, Walking, Tilting, In vehicle, *other*	5, 4, 3, 2, 1
Shaking	0.5	Yes/No	1, 5
Distance	1	x = Meters	$f(x) = -0.05x{+}3.7$
Network	1	x = Bandwidth y = Video Bitrate	$f(x, y) = x/y * 5$
Network	1	WiFi, LTE, HSPAP, 3G, UMTS	5, 5, 3, 2, 1

Table I: Recording score parameters

In addition to storing the data acquired from the devices, upon receiving an update message, the server triggers the processing of the recording score based on the received data. It takes into account parameters included in the most recent update, as well as the initial join message for the particular device to calculate a recording score for each active device. The parameters can be factored with different coefficients. The estimator accepts a location (POI) and a required bitrate as initialization parameters, used to assign scores for all devices currently connected to the orchestration system. For the calculation of *device location score*, the distance between the device (based on the GPS-location contained in the join/update message) and the defined POI is derived (great circle distance). In case this distance is between 5 and 50m, the linear formula for a concert venue is used, as specified in Table I [14]. For larger and smaller distances, 1 and 5 are assigned as scores, respectively.

For the *device activity score*, we consider the results from the shake estimation and activity detection mechanisms. If shake is detected, the minimal score of 1 is factored in; otherwise, the maximal grade of 5 is used. The activity detection algorithm assigns a maximal score to *Still* devices, and a minimal score when the detected activity is *Running*, *On bicycle* or *In Vehicle*. For *Walking* and *On foot* a value of 3 or 4 is assigned. The average of both indicators is the overall *device activity score*.

Next, the *device network score* measures the current network capabilities by assessing two possible scenarios. It uses the bitrate of the video recording and assigns a maximal score to devices that have an available bandwidth equal or higher to the required one. For lower measurements, the score is linearly scaled between 1 and 5. In case a recent bandwidth estimate cannot be derived (e.g., due to high channel fluctuation), is not yet available or is too old (more than 60 seconds), the score is based on the network context. If the devices are connected by Wi-Fi or LTE, a score of 5 is assigned, while lower scores are assigned to slower network technologies.

The sum of the weighted parameters is divided by the sum of weights, resulting in a score in the range 1-5. An overview of the indicators and weighting factored is given in Table I.

E. Integration of the Monitoring Framework

The client-side functionality is packaged as an Android service that can be started via an Intent. The information about the connected devices is exposed by the server via methods as well as through a REST API. In this way, the framework can be easily used by other composition systems.

IV. The Video Recording Application: LiViU

LiViU [16] is an *adaptive mobile video upload protocol* that can be used in conjunction with TCP or UDP. It is quality-adaptive as it supports transcoding multiple video representation in parallel on Android mobile devices. Also, it enables adaptations in the scheduling mechanisms in order to allow a video composition system to request video chunks. Both concepts are depicted in Figure 1.

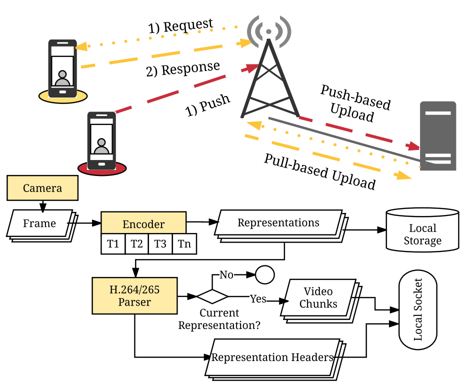

Figure 1: Overview on the concepts and mechanisms of LiViU.

A. Adaptive Video Upload

LiViU allows the creation of multiple video representations in real-time. By generating different bitrate versions of a video, it can ensure that the available bandwidth is utilized.

To do that, LiViU extends the media recording Application Programming Interface (API) on Android phones to set up different encoding threads. A realization is achieved, as the hardware encoding of current smartphone generations can be leveraged for transcoding videos. Each video frame recorded is handed over to the video encoding thread. The graphics rendering API of Android is used to run the transcoding on a GPU of a mobile device. Therefore, each raw video frame retrieved by the camera is converted into two-dimensional texture, which is represented as a three-dimensional texture if a set of frames is available. To access the GPU, the OpenGLES library for mobile devices is used[3], which allows for a quick manipulation of the resolution and frame rate. Each encoding thread operates on a copy of the texture and manipulates it according to the desired frame rate and resolution properties. The final step hands the texture buffer to the respective video encoding object which leverages the built-in hardware to encode the representation at the desired bitrate. The resulting H.264/AVC raw video representations are consecutively written to the device's main memory.

A real-time capable video parsing service analyzes the consecutively written video files and offers them to the transmission functionality of LiViU. The complexity of understanding when a switch can be conducted without any artifacts on the receiver side is hidden within the parsing service. Video chunks of the selected representation are handed over to the LiViU transmission using a local socket on a mobile device.

B. Adaptive Scheduling

LiViU can adapt between the different scheduling scheme: push-based delivery of media messages and pull-based retrieval of the same (see Figure 1). In this work, LiViU uses solely a push-based delivery to allow a low-delay streaming at minimal overhead. Yet, a switch between the two modes can be triggered by the server, which stops the default push-based delivery. The pull-based delivery of chunks is controlled by the application on the receiver side, which can determine when to request which media chunk.

C. Composition

Composition can be performed as an independent component of the application. Generated streams are saved by LiViU with a given `deviceID` and `sessionID` as files on the server and can thus be used for further processing. Currently, a Python-based system (using the *moviePy* library[4]) is employed for an offline composition of video streams, as detailed in the Section V.

V. Evaluation

The prototype system is evaluated with the goal to compare the performance of video composition with or without the proposed monitoring service.

We recorded a dataset at five locations with multiple devices. For each of the recording locations two composed videos–one using the monitored data as a basis for composition and one with arbitrary switching between available streams. In a second step, we employed user studies to derive the user satisfaction for the composed video sequences.

[3]https://source.android.com/devices/graphics/arch-egl-opengl.html
[4]http://zulko.github.io/moviepy/

Location	Device (Network)	Movement	Shaking	Panning
		(o: none; +: some; ++: intensive)		
	1 (LTE 7.2 Mbps)	o	o	o
	2 (3G 7.2 Mbps)	+	++	o
Location 1	3 (WiFi 50 Mbps)	+	+	o
	4 (WiFi 50 Mbps)	o	o	+
	5 (LTE 50 Mbps)	+	o	o
	1 (LTE 7.2 Mbps)	+	+	o
	2 (3G 7.2 Mbps)	o	o	+
Location 2	3 (WiFi 50 Mbps)	+	o	o
	4 (WiFi 50 Mbps)	o	o	o
	5 (LTE 50 Mbps)	+	++	o
	1 (LTE 7.2 Mbps)	+	o	o
	2 (3G 7.2 Mbps)	o	o	o
Location 3	3 (WiFi 50 Mbps)	+	++	o
	4 (WiFi 50 Mbps)	+	+	o
	5 (LTE 50 Mbps)	o	o	+
	1 (LTE 7.2 Mbps)	+	++	o
	2 (3G 7.2 Mbps)	+	+	o
Location 4	3 (WiFi 50 Mbps)	o	o	+
	4 (WiFi 50 Mbps)	+	o	o
	5 (LTE 50 Mbps)	o	o	o
	1 (LTE 7.2 Mbps)	o	o	+
	2 (3G 7.2 Mbps)	+	o	o
Location 5	3 (WiFi 50 Mbps)	o	o	o
	4 (WiFi 50 Mbps)	+	++	o
	5 (LTE 50 Mbps)	+	+	o

Table II: Device context based on location

A. Evaluation Setup

As a basis for the evaluation, video sequences are simultaneously recorded with five Nexus 5 smartphones in five different locations, each carried by one individual user. The POI is configured in the beginning of each session.

Each device is operated by one recording user for a duration of at least three minutes. During recording, monitoring data is collected by the proposed framework and transmitted to the server to be stored in a PostgreSQL database, along with the streamed video sequences. Both services are hosted on an Ubuntu 16.04 server connected to the university network with a bandwidth of 1 Gbps. The network connections of the recording devices are configured so that a wide range of connection attributes is achieved. This includes two devices using a WiFi network hosted by a mobile access point connected via LTE to the Internet with a maximum upload bandwidth of 50 Mbps. Two additional devices use LTE directly with up to 7.2 Mbps and 50 Mbps bandwidth respectively, as well as one device using a 3G connection with up to 7.2 Mbps. Each device maintains the network settings while the user instructions are rotated for each round corresponding to one location. An overview of the setup is given in Table II.

We define the streaming quality in LiViU to a bitrate of 3 Mbps with 30 Frames per second (FPS), a video resolution of 1280×720 pixels, encoded with H.264. During recording, LiViU adapts the bitrate of the recording dynamically between 3 Mbps and 1.5 Mbps based on bandwidth measurements provided by the monitoring framework.

After recording, the transferred video sequences are used to generate compositions of different streams according to the collected scores. Here, each generated sequence has a duration of 30 seconds, where switches between views are allowed at most every 5 seconds. For each 5-second video segment, the device with the highest achieved score is selected as the video source. For comparison, we compose one corresponding video sequence of the same total duration which randomly selects one of the available views in 5-second intervals. Given the slightly different start time of the recording sessions, we normalize the time based on the arrival time of a streaming session on the server. The 10 generated sequences (5 score based, 5 random) are then evaluated in a crowd-sourcing study using the Crowdee service. Here, both classes of generated sequences are displayed to users on mobile devices, gathering a subjective opinion score using a continuous slider to select ratings between 1 and 5, thus evaluating the overall quality of each sequence pair independently. The workers did not know which sequence is based on scores. After each pair of video sequences is shown to the workers, a forced choice experiment is applied, asking which sequence they preferred. In a last question, the workers are asked to identify objects in the videos to verify a satisfactory level of attention during the test.

B. Results

The user study is conducted with 20 individuals, with an average age of 27 years (12 male, 8 female). In Figure 2, Mean

Video ID	% Score	% Random	JND unit
Location 1	88.9	11.1	1.7
Location 2	90	10	1.77
Location 3	66.7	33.3	0.65
Location 4	66.7	33.3	0.65
Location 5	78.9	21.1	1.18

Table III: Just Noticeable Difference experiment results.

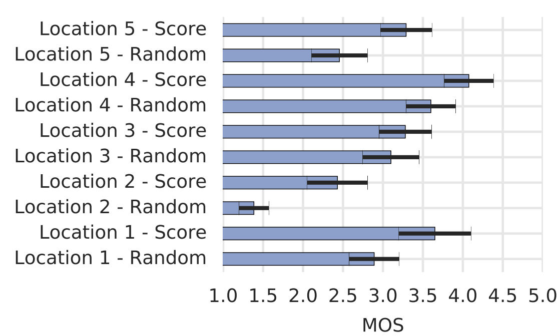

Figure 2: MOS by location

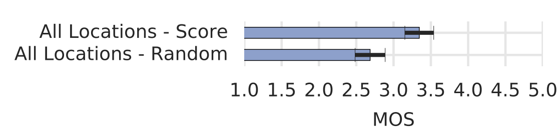

Figure 3: Overall MOS

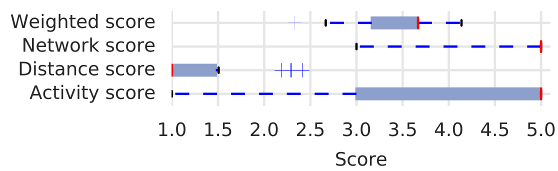

Figure 4: Comparison of generated scores for video selection

Opinion Scores (MOSs) are shown for each of the 5 recording locations. The entire spectrum of MOSs can be observed, indicating heterogeneous quality and user preferences in both random and score-based compositions. For each location, the MOS value for score-based compositions outperforms the random compositions. However, the overall difference in mean values strongly depends on the recording location. For location 3 and 4 the confidence intervals indicate a wide spectrum of results, and no statistically significant preference for score-based vs. randomly selected compositions can be seen. Yet, there is a significantly better score for location 1, 4 and 5. A similar trend can be observed for the summarized results including the results for all recording locations, as shown in Figure 3. Here, the score-based compositions show a higher mean, with both indicators showing a high standard deviation of roughly 0.5 on the MOS scale, indicating the high subjective variance in preferences of users, with a significant preference towards score-based compositions. The results for the Just Noticeable Difference (JND)-based forced choice experiments, as proposed by Watson *et al.* [13], in Table III show that 3 out of 5 score-based compositions are significantly better ($> 75\%$; JND > 1). This result verifies the trend seen on the MOS based results.

Next, in Figure 4 we show the distribution of the metrics used as the basis for the score-based video composition. It can be observed that the overall weighted score is mostly distributed around a mean of 3.5, whereas larger differences in the single indicators exist. The network score is never lower than 3 showing that results from the performed active measurements exceed the required bitrate for video upload. The distance score shows very low values overall, indicating in some cases imprecise location estimates or a high distance of the devices from the POI. Last, the activity score shows a wider spectrum of values, which is to be expected given the range of disruptive actions performed by the recording users, as shown in Table I.

VI. Conclusions and Future Work

In this work, we propose a system for live upload and composition of UGV based on measuring device context information. We introduced a practical system implementation that was evaluated and tested in a crowd-sourced user study. In particular, activity recognition and active bandwidth estimation were included as a novel concept to support mobile live video upload from multiple sources.

Using the created datasets, we show that a higher mean QoE can be achieved when selecting video streams based on network-, activity-, and location-based indicators, *without analyzing the video content directly*. This allows the selection of only those devices that are used for the final view in a composed video stream, minimizing the overall data that needs to be uploaded by all participating devices.

Acknowledgment

This work has been co-funded by the German Research Foundation (DFG) as part of the subproject C3 within the CRC 1053–MAKI as well as by the German Federal Ministry of Education and Research (BMBF): 01IS12054.

References

[1] F. Cricri, I. Curcio, *et al.* "Sensor-based analysis of user generated video for multi-camera video remixing." in *Advances in Multimedia Modeling*. ser. Lecture Notes in Computer Science. vol. 7131. 2012.

[2] A. El Essaili, Z. Wang, *et al.* "Qoe-based cross-layer optimization for uplink video transmission." *ACM Transactions on Multimedia Computing, Communications, and Applications*, pp. 1–22. 2015.

[3] A. Engström, M. Esbjörnsson, *et al.* "Mobile collaborative live video mixing." in *ACM MobileHCI*. 2008.

[4] D. Johansen, H. Johansen, *et al.* "Davvi: a prototype for the next generation multimedia entertainment platform." in *ACM MM*. 2009.

[5] M. Li, M. Claypool, *et al.* "Wbest: A bandwidth estimation tool for ieee 802.11 wireless networks." in *IEEE LCN*. 2008.

[6] B. Richerzhagen, J. Wulfheide, *et al.* "Enabling crowdsourced live event coverage with adaptive collaborative upload strategies." in *IEEE WoWMoM*. 2016.

[7] M. K. Saini, R. Gadde, *et al.* "Movimash: Online mobile video mashup." in *ACM MM*. 2012.

[8] B. Seo, W. Cui, *et al.* "An experimental study of video uploading from mobile devices with http streaming." in *Proceedings of ACM MMSys*. 2012, pp. 215–225.

[9] P. Shrestha, P. H. N. de With, *et al.* "Automatic mashup generation from multiple-camera concert recordings." in *ACM MM*. 2010.

[10] M. Siekkinen, E. Masala, *et al.* "Optimized upload strategies for live scalable video transmission from mobile devices." *IEEE Transactions on Mobile Computing*. 2016.

[11] D. Stohr, T. Li, *et al.* "An analysis of the younow live streaming platform." in *IEEE WNM*. 2015.

[12] D. Stohr, S. Wilk, *et al.* "Monitoring of user generated video broadcasting services." in *ACM International Workshop on Internet-Scale Multimedia Management*. 2014.

[13] A. B. Watson and L. Kreslake. "Measurement of visual impairment scales for digital video." in *SPIE–Human Vision, Visual Processing, and Digital Display*. vol. 4299. International Society for Optics and Photonics, 2001.

[14] S. Wilk and W. Effelsberg. "Systematic assessment of the video recording position for user-generated event videos." in *ACM MM*. 2014, pp. 1009–1012.

[15] S. Wilk and W. Effelsberg. "Mobile video broadcasting services - combining video composition and network efficient transmission." in *ACM MM*. 2014.

[16] S. Wilk, R. Zimmermann, *et al.* "Leveraging transitions for the upload of user-generated mobile video." in *ACM MoVid*. 2016.