

The crowd as a cameraman: on-stage display of crowdsourced mobile video at large-scale events

Steven Bohez¹ · Glenn Daneels² · Lander Van Herzele¹ · Niels Van Kets³ ·
Sam Decrock⁴ · Matthias De Geyter⁴ · Glenn Van Wallendael³ ·
Peter Lambert³ · Bart Dhoedt¹ · Pieter Simoens¹ · Steven Latré² ·
Jeroen Famaey²

Received: 9 July 2016 / Revised: 28 November 2016 / Accepted: 8 December 2016
© Springer Science+Business Media New York 2016

Abstract Recording videos with smartphones at large-scale events such as concerts and festivals is very common nowadays. The se videos register the atmosphere of the event as it is experienced by the crowd and offer a perspective that is hard to capture by the professional cameras installed throughout the venue. In this article, we present a framework to collect videos from smartphones in the public and blend these into a mosaic that can be readily mixed with professional camera footage and shown on displays during the event. The video upload is prioritized by matching requests of the event director with video metadata, while taking into account the available wireless network capacity. The proposed framework's main novelty is its scalability, supporting the real-time transmission, processing and display of videos recorded by hundreds of simultaneous users in ultra-dense Wi-Fi environments, as well as its proven integration in commercial production environments. The framework has been extensively validated in a controlled lab setting with up to 1 000 clients as well as in a field trial where 1 183 videos were collected from 135 participants recruited from an audience of 8 050 people. 90 % of those videos were uploaded within 6.8 minutes.

Keywords Crowdsourcing · Mobile video · Wireless upload scheduling

✉ Steven Bohez
steven.bohez@ugent.be

¹ Department of Information Technology - IBCN, Ghent University - iMinds,
Technologiepark-Zwijnaarde 15, B-9052, Ghent, Belgium

² Modeling of Systems And Internet Communication, University of Antwerp - iMinds,
Middelheimlaan 1, B-2020, Antwerp, Belgium

³ Electronics and Information Systems Department, Ghent University - iMinds,
Sint-Pietersnieuwstraat 41, B-9000, Ghent, Belgium

⁴ iMinds Krook.LAB, Technologiepark-Zwijnaarde 19, B-9052 Ghent, Belgium

1 Introduction

Attendees of live events, such as concerts and festivals, are intensively using their smart-phones to record videos of themselves, friends or the stage. Some of these videos are then immediately shared on social media. Even though the resolution and sensor quality of smartphone cameras has significantly increased over the last years, currently the videos and images shown on the large on-stage screens originate exclusively from the limited set of professional cameras installed throughout the event venue. Integrating this User-Generated Content (UGC) as an additional source of footage into the live event media production would not only offer more camera perspectives to the director, outside the scope of professional camera coverage, but also increase the engagement of the public in the event [4].

This paper presents the design and validation of a framework for collecting and showing user-generated videos at large-scale events. The concept is visualized in Fig. 1. Short video clips are captured via a dedicated smartphone app. Based on video metadata (e.g., quality, resolution, location) and the available Wi-Fi capacity, the framework prioritizes the uploads to ensure a good mix of videos, e.g. from different areas in the event venue. The videos are semi-automatically combined into a mosaic that is presented as an additional video input channel to the director of the event, next to the video streams from professional cameras. The collected videos are also used in post-production: the framework can automatically generate a personalized video souvenir, which combines time-synchronized fragments of professional video with user videos, ready for sharing on social media. With this live integration of personal video as well as the video souvenir, organizers can add value to events through user-generated content which has a personal meaning to the audience. Within this context, the users ‘become part of the event’.

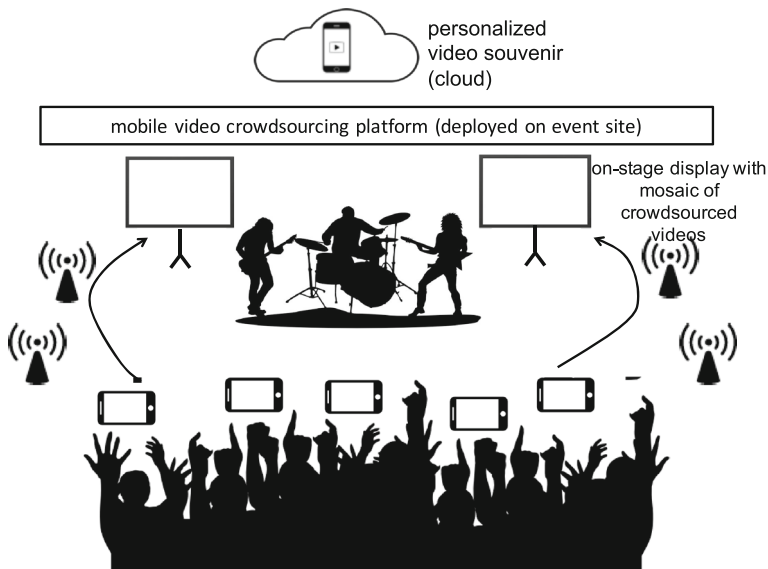


Fig. 1 Videos captured by the audience are collected by a back-end platform. Uploads are prioritized based on the preferences of the director and the available wireless resources. After the event, participants may download a personal video souvenir

The framework has been thoroughly validated in a lab setting as well as in a realistic environment during a large-scale field trial. During a 90 minute commercial show, 1 183 unique videos were uploaded by 135 participants recruited from approximately 8 050 attendees present in the venue.

This article provides the following research contributions:

- A scalable architecture for a mobile video crowdsourcing framework that prioritizes the upload process based on metadata and available wireless resources.
- Performance measurements obtained in a controlled lab setting, as well as from a large-scale realistic field trial on attainable wireless throughput in highly dense Wi-Fi networks, the video upload prioritization and the delay introduced by the proposed framework in making a recorded video available to the director for play-out.
- Insight in the technical characteristics and content of user-generated videos, as well as the recording behavior of users. All our results from the field trial are made available as open datasets to the research community.¹

The remainder of this paper is structured as follows. In Section 2, we discuss related work on collecting and uploading mobile video. Section 3 enumerates technical requirements for the video collection framework. We present the architecture and implementation of our system in Section 4. Validation results on the system scalability in a controlled lab setting are discussed in Section 5. In Section 6, we review results and insights obtained during a field trial at a commercial event. Finally, Section 7 summarizes our main findings and outlines future work.

2 Related work

2.1 Collecting and annotating mobile video

Movisode [25] is a collection framework for on-demand retrieval of crowdsourced mobile video at well-attended events. Smartphones upload metadata on the start and end time, view angle and content of captured videos. Upload decisions are taken by a central algorithm, based on a cost model that includes an estimation of the energy required to upload the video. Similar to our approach, Movisode aims to fulfill coverage requests based on video metadata. The evaluation of Movisode is however limited to a trace-driven lab simulation on the Jiku data set (with only 473 videos). Our approach has been tested in realistic environments where it collected one order of magnitude more videos. Moreover, our framework involves a network-aware upload strategy and provides additional features such as integration with professional production equipment and personal video souvenir generation. Lastly, video queries are not limited to points of interest, but can span any available metadata.

Other work focuses on generating metadata in the cloud for mobile videos, with videos being pushed by the user and not making any prioritization or synchronization of video uploads. In the work of Jain et al. [15], live user video streams are clustered automatically and in real-time by line-of-sight, using multi-view stereo reconstruction and inertial sensing to reason about the relative position and orientation of videos. Also metadata from social networks is used to annotate video streams. EpicPlay [24] selects highlights in live broadcast

¹See <https://www.uantwerpen.be/en/rg/mosaic/projects/selvie/>

sports games, specific for fans of each team and reflecting the emotions of a fan during the game.

While video websites like YouTube provide an abundant supply of video material, these videos are not systematically labeled, nor do they contain additional metadata like smartphone sensor readings. As an outcome of the work presented in this paper, we present a new dataset to the research community, comprising 1383 videos taken at a single event, complemented with mobile sensor data. This dataset complements existing mobile video datasets. Jiku [21] is a dataset containing 473 videos taken at 5 different events. Each video is accompanied with readings from smartphone sensors. The iCosole [3] dataset comprises multi-view and spatial audio, taken in a controlled environment using a range of professional- and consumer-grade equipment. Ground truth annotations have been created for temporal synchronization and spatial alignment between videos.

2.2 Traffic scheduling in dense wireless networks

Wi-Fi networks deployed at large events are heavily stressed by traffic generated from social media, news websites and messaging apps. Aiming to collect as many videos as possible, our framework must carefully schedule the uploads without compromising the crowd's access to other Internet-enabled apps. Scheduling data transmissions in networks has been a popular research subject the past decade. A well known approach in fixed networks is to use advance bandwidth scheduling [16, 20], where a certain amount of bandwidth is reserved during a certain time frame for each requested data transmission. For wireless networks, however, much research focuses on packet-level scheduling using e.g. network coding [13], whereas the system described in this paper schedules on a content- or file-level. Borst [5] describes a set of flow-level channel-aware scheduling algorithms for dynamic wireless environments that aim to maximize user-level throughput. A two-layer scheduler is proposed by Ganguly et al. [11] that first provides guarantees on content download times and uses opportunistic scheduling to maximize the throughput of the entire system.

However, our system poses different challenges: instead of content being downloaded to the mobile device, it is uploaded from it. Moreover, the specific content being transmitted is not requested by the mobile users but by the supervising directors. Finally, these existing algorithms are built for cellular networks and do not account for the issues and opportunities specific to the Wi-Fi network that is deployed on-site. More similar to our work is NEWSMAN [22], which aims at uploading user-generated newsreports in places with very weak network infrastructure using so-called adaptive middleboxes. These middleboxes provide a short-range wireless network and perform upload scheduling taking into account the varying network conditions as well as video transcoding parameters. Chen et al. [7] consider a surveillance use case, using mobile cameras for human detection. They propose a coding rate allocation scheme that improves human detection success rate given a total data rate constraint. Our work however focuses on very dense network deployments which suffer from a high level of interference. Moreover, we want to prioritize content upload based on (dynamic) requirements posed by a human director instead of the mobile user or a predetermined model.

In our previous work, we evaluated scalability of Wi-Fi for upload content scheduling in dense environments [9]. This showed the need for more intelligent scheduling approaches to avoid interference and the resulting performance degradation. However, its evaluation was limited to a controlled lab setup.

2.3 Automated processing/classification of video and director support for mobile video

While UGC recorded at large-scale events may provide great value for the event production, a human director may quickly become overwhelmed by the influx of new content if no additional support is provided for efficient querying and composing the available videos.

The system described in [4] provides a video browsing tool allowing users to efficiently find relevant UGC in large collections. The tool employs iterative clustering on both readily available metadata as well as computed visual features. A more automated system aimed at live video composition called CrowdDirector is presented in [26]. CrowdDirector combines a mobile app on which the quality of the recording is shown in realtime with a backend that generates compositions in a semi-autonomous way. Content, after pre-filtering, is fed into an algorithm to generate possible compositions. Some of these are then offered to a pool of workers that rate the compositions. The ratings are then used to improving the composition algorithm in realtime. The Jiku Director [18] employs a similar autonomous method of creating video mashups by adopting machine learning techniques on pre-filtered videos. However, the learning procedure occurs offline by learning from professional video mashups. The algorithms first classifies each video based on the viewing angle with respect to the subject, then selects the ones which have a suitable viewing angle with regards to the current shot in the composition and finally selects a video from the remaining set based on quality and diversity. Finally, in [1], computer vision algorithms are used in order to determine the position and orientation of each camera. Based on the overlap between the viewpoints, the location of the important content in the scene is determined. A video composition is then created that maximizes the coverage of the important content, as well as follows a number of cinematographic guidelines.

Different from these works, the system described in this paper extends human director support into the scheduling algorithm that decides when content is uploaded, avoiding an unnecessarily large influx of new content. The human director is able to request desired properties for the content, and the scheduler will automatically prioritize uploading the videos matching these criteria.

3 System requirements

Collecting videos from the crowd and integrating them as User-Generated Content (UGC) in the live event production and in personal video souvenirs raises important technical challenges. Professional-grade system scalability and resilience are mandatory, and the system design should take into account requirements from the end-user as well as from the event director.

From an *end-user* perspective, our framework should at least lower any barrier to record videos, and preferably provide additional incentives for users to record as many videos as possible. This observation can be broken down into the following requirements:

- Maximize the potential user base, by imposing as few compatibility restrictions as possible on the myriad of mobile device features, such as mobile OS (Android, iOS), OS versions, supported camera resolution, and wireless connection type.
- Preserve the native look-and-feel of the mobile OS when recording videos. Users should be able to use basic camera features such as switching between front and back cameras, pinch-to-zoom, and turning on the flashlight for additional illumination.

- Ensure adequate Wi-Fi coverage at the event venue to avoid users being reluctant to share videos because of data caps on their mobile subscriptions. A portion of the Wi-Fi throughput must be reserved for video upload, but we must preserve the possibility for attendees to keep using other Internet-enabled applications like Facebook seamlessly.
- Provide an additional incentive for the user by creating a personal video souvenir. This souvenir is a short video fragment, where a user video is merged with fragments from the professional camera feeds. The blending between the user video and the professional video feed must be time-synchronized. The souvenir must be available for all users as soon as possible after the end of the event, otherwise users will lose their interest in sharing this video on social media.

From the perspective of the *event director*, the framework must allow the director to quickly select mobile videos for immediate play-out. This observation can be broken down into the following requirements:

- Prioritize video uploads based on preferences formulated by the director in content and technical characteristics of the UGC, such as videos from a specific area in the event venue, landscape or portrait orientation, and minimum resolution.
- Complete the upload of videos from the smartphone no longer than 1.5 minutes after the time they were recorded. This target is set in accordance to the typical 3–4 minute length of (pop) songs during a show. Directors will prefer to display videos that are in sync with the event storyboard. For example, during a ballad, no videos should be shown that were captured during the previous up-tempo song.
- Transcode and compose the UGC into a mosaic ready for the director to immediately play-out on the on-stage displays without any further need for manual intervention.
- Fix the duration of recorded videos to 7 seconds. Although the system can be flexibly configured to support other fixed, as well as variable, video durations, the decision for 7 seconds was made based on feedback from both end-users and professional directors. This short duration ensures that users will create more catchy and visually attractive videos. An equal length of all videos allows for more efficient upload scheduling and avoids any additional manual processing by the event production crew before the videos can be blended into a mosaic.

4 Platform architecture and implementation

The architecture of our framework for collecting videos from the crowd and integrating them as user-generated content in the professional event production is visualized in Fig. 2.

The Wireless Broker is responsible for Wi-Fi resource management and communication with the Mobile App installed on the users' devices. The UGC Collector prioritizes the video uploads based on preferences set by the director via the Director Console (DC). This component also combines collected UGC into a mosaic that can be played out on screens along the stage. The Post-Production Support (PPS) module generates personal souvenir videos containing a mix of user and professional content. Figure 3 depicts the interactions between the framework's components.

The UGC Collector receives a content request message from the Director Console, specifying the video preferences from the event director. Although the overall goal is to collect as many videos as possible, in practice the wireless throughput is limited and traffic management by coordinating the uploads is needed. For this reason, the client device initially

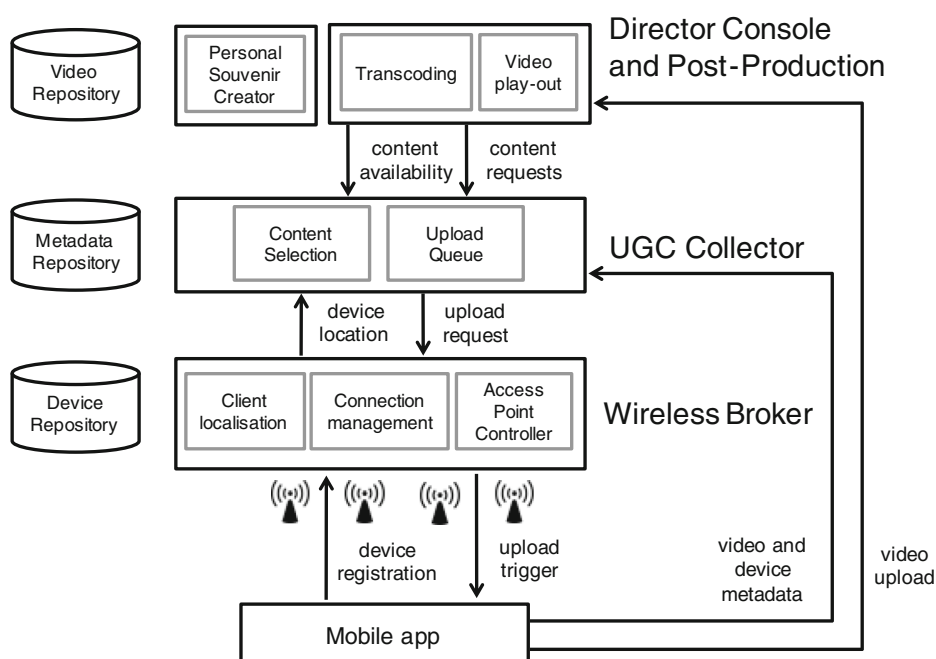


Fig. 2 Architectural diagram of our framework for collecting user-generated content

only uploads metadata of a recorded video to the UGC Collector, instead of immediately initiating an upload of the video file itself. The UGC Collector ranks all advertised videos in an upload queue by mapping the metadata to the content preferences of the director. This queue is continuously updated as new videos are announced or the director's preferences are changed. The Wireless Broker is responsible for the upload of the videos in the order set by the UGC Collector. In particular, this Wireless Broker handles all complexity related to the wireless connection of participating devices and monitors the different Wi-Fi access points that are installed at the event venue. It informs the UGC Collector about the location of the connected clients in order to optimize the upload schedule. Videos are uploaded directly to the Video Repository, part of the Director Console module. This module then prepares the videos for play-out as a mosaic on the on-stage displays and creates the personalized video souvenirs.

4.1 Mobile app for collecting videos

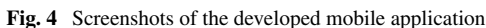
Several options are available for collecting videos from smartphones during events: (i) allow users to record videos with the camera app of choice, and provide incentives to manually share videos, (ii) a mobile web app and (iii) a native app. These three options offer increasing development effort, but also increasing ease-of-use and options to control the upload process.

The choice was made for a native app for the following reasons:

- **Performance:** Native app performance is still unrivaled by web apps, which is important for a camera app where you do not want to miss “the moment”.

- **Ease-of-use:** Native apps more easily provide the OS look-and-feel users are accustomed to, which stimulates the acceptance of our app by the crowd.
- **Controllability:** Native apps have access to more advanced APIs, which allows us to fully manage important configuration parameters. For example, we can automatically enable Wi-Fi and connect to the SSID of the dedicated Wi-Fi network installed at the venue. Moreover, the calculation of various metadata parameters related to the content and device sensors is more straightforward to implement. We can also control recording settings such as the video length and resolution, so that videos are recorded in the format most appropriate for the Director Console module.

We have developed a version of the app for both Android and iOS. The screenshots in Fig. 4 show its user interface. The app offers the same basic functionality as other tried-and-true camera apps, such as zooming, activating the flashlight, and switching between front- and back-facing cameras. Flashlight functionality proved indispensable due to the general lack of illumination in the audience area of concert venues. While the user has freedom to record in either landscape or portrait mode, the duration of all video recordings is limited to 7 seconds (cf. Section 3 for the motivation). The 7 second countdown is clearly indicated to the user as a timer in the corner of the screen as well as a larger circular progress bar.



We also considered other features to be included, such as the possibility for the user to switch to a mode in which videos of unlimited length can be recorded. However, user feedback on an early prototype indicated that this was too complicated, with users forgetting to switch back to the fixed duration if they wanted to record a video for displaying on the large screen.

Next to the main functionality of recording and uploading a video, the app includes a video gallery, as shown in Fig. 4b. Although all recorded videos are normally also accessible from within the phone's default Gallery app, we felt this was a much-needed feature as it allowed us to develop custom actions on videos in this gallery. On top of the default sharing and deleting behavior, we implemented functionality for the user to select one video to include in his personal video souvenir. We limit the choice to videos that were recorded by the app during the show. This provides a strong guarantee to the organizer that personal souvenirs are not abused by users selecting non-relevant videos with possibly inappropriate content.

The smartphone app reports different metadata parameters about the device as well as the recorded videos to the other parts of the framework. The captured parameters include: resolution, target frame rate, start and end time, file size and format, video brightness and shakiness, and device orientation.

4.2 Wireless broker

By deploying a dedicated Wi-Fi network at the event venue, we are able to do more advanced wireless traffic management. The Wireless Broker (WB) monitors and manages the wireless spectrum in order to maximize the throughput of the requested video uploads. It maintains a connection to each smartphone to do rigorous client management. The WB is also connected to each of the Wi-Fi access points at the event venue to perform detailed monitoring and management. Moreover, it determines a location for each client connection which is passed to the UGC Collector to allow for more intelligent upload scheduling.

4.2.1 Access point monitoring

The dedicated wireless network is set up using several APs. As this framework is deployed in a Wi-Fi DenseNet environment, the network should deal as well as possible with high-interference levels and the high probability of packet collisions that are typical in such dense networks. To mitigate these negative effects, the APs were strategically placed in the event venue and the channels of neighboring APs were chosen intelligently so their overlap would be minimal. Each AP is managed and monitored by a centralized AP controller. The Wireless Broker communicates with this controller to fetch monitoring details about the connected devices such as device type, throughput, RSSI values, frequency band usage and which AP a device is associated with. The latter two are used to determine the location of a device, as described in Section 4.2.3.

4.2.2 Client connection management

An end-user can only use the system when successfully registered with the Wireless Broker. Therefore, immediately after launching, the mobile app sets up a connection to the WB and sends a registration message, as shown in Fig. 5. After registration, the connection is maintained at the WB and is marked by a unique identifier that is used throughout the whole

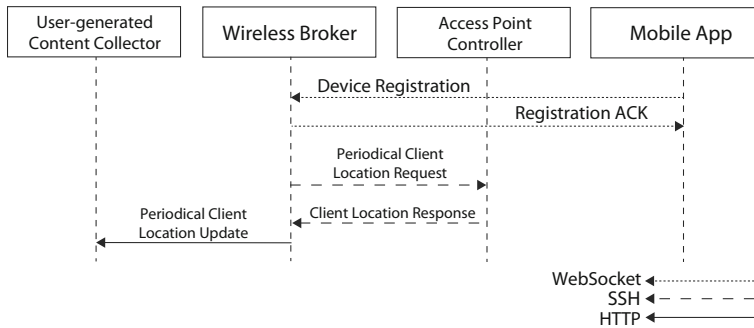


Fig. 5 Sequence diagram of a client registration at the Wireless Broker and the periodical client location update directed towards the UGC Collector

system. The client connections are used to request a video upload: the UGC Collector sends a request to the WB containing a video upload message accompanied by a client identifier, as shown in Fig. 3. The Wireless Broker forwards the request to that particular client and the latter uses the inverse route to acknowledge the video request. In order to avoid unnecessary scheduled upload slots or video request messages for already disconnected clients, a list of connected clients (together with their locations, cf. Section 4.2.3) is periodically sent to the UGC Collector. To keep track of all the connected clients, the WB periodically monitors the access points to check if a client is still associated. In case the client is not observed anymore or the mobile app actively closed the connection, it is removed from the list of registered clients.

4.2.3 Client localization

To allow the UGC Collector to schedule video uploads without saturating the wireless spectrum, the Wireless Broker provides a location of each connected and registered client, as shown in Fig. 5. This location consists of the access point identifier to which the client is connected and the frequency band on which it is connected. This information is fetched periodically using the access point controller and is added to the periodical update of active connections to the UGC Collector. The access point identifier determines the users' (estimated) position in the event hall and is used by the director to select videos from a specific area. The frequency band is used to avoid interference among users when scheduling uploads (cf. Section 4.3).

4.3 UGC collector

The UGC Collector selects and requests the subset of videos that are most likely to be used in the event production, by matching the metadata of the videos with the content requests from the director. Using the resulting matching score as a priority, the UGC Collector maintains an upload queue from which it will select videos suitable for upload, taking into account the available wireless spectrum resources and location of the clients. A fairness mechanism is also included to balance different director requests. In the following sections, we will first describe the scheduling algorithm in the case only a single request is made by the director and afterwards extend it to multiple requests.

4.3.1 Single-request operation

Director requests are formulated in terms of the same video and device metrics used when announcing metadata of recorded videos (cf. Section 4.1).

Each announced video e is modeled as a collection of l attributes a_i with accompanying value intervals:

$$e = \{a_1 : [v_1, v'_1], \dots, a_l : [v_l, v'_l]\} \quad (1)$$

Working with intervals allows for more expressive matching than a single scalar value, as certain metrics such as brightness will fluctuate during the video. Scalar attributes can however still be described by using identical values for the start and end of the value interval. Categorical values can be modeled as discrete (integer) scalar attributes. Sets or sequences of (numerical) values can be seen as a set of different attributes.

The requests made by the director are modeled as n conjunctions of “constraints” on the attributes. Each constraint has an associated weight, expressing its importance relative to other constraints.

$$\phi = \wedge_{i=1}^n \delta_i : w_i, \text{ with } \delta_i = a_i \in [v''_i, v'''_i] \quad (2)$$

Finally, a prorated top-k matching score [8] is calculated for each video e that takes into account interval overlap between attributes and constraints.

$$\text{score}(\phi, e) = \sum_{\forall i: [v_i, v'_i] \in [v''_i, v'''_i]} w_i \times \left(\frac{C_i + \min(v'''_i, v'_i) - \max(v_i, v''_i)}{C_i + v'_i - v_i} \right) \quad (3)$$

Here, $C_i = 0$ for continuous attributes and $C_i = 1$ for discrete attributes in order to account for the overlap at the endpoints. The \in operator evaluates to TRUE when both intervals overlap at least partially. Essentially, the score is calculated as the sum of the (partial) weights of (partially) satisfied constraints.

This top-k score is offset by the time of recording to provide an ordering in the content even in the absence of requests, as well as to bias towards more recent videos. In this way, constraint weights can be interpreted as a time period as well, trading the “age” of the video for more desirable properties.

The \in operator evaluates to FALSE when an attribute is not present in either the metadata or the director request. This implies that the Collector is agnostic to the metrics present in either the metadata or the request: not only do not all metrics have to be added to each metadata post (e.g. when a sensor is not available), the same holds true for video requests.

Using the calculated scores for all available videos, a priority queue is constructed. The queue order reflects the degree of matching between the video metadata and the director preferences. However, in practice multiple videos are uploaded simultaneously and we employ backfilling to make sure that a single device only uploads a single video at a time. The UGC Collector allows for a fixed number of concurrent uploads per access point and frequency band. The UGC Collector assigns this number of upload slots to each channel and effectively tries to occupy them as much as possible by queue backfilling. Our previous experimental results, reported in [9], have shown that 3 simultaneous uploads maximizes overall throughput on a single channel. Content for which the client’s location is currently unknown (e.g. a device is disconnected), is never scheduled for upload.

Each time the Mobile App reports metadata on a new video, a corresponding item is created in a Metadata Repository. Each video in the Metadata Repository is associated with

one of five possible states: advertised, requested, uploading, available or failed. The state transition diagram is shown in Fig. 6.

When a video is selected to be uploaded, an upload request is first passed down to the WB, which checks if the client is online and passes the request. The Mobile App will upload the video into the Video Repository of the DPPS module. The video is only removed from the content queues when the upload is completed. If either the Wireless Broker reports a failure when requesting the content or a timeout occurs during the upload, the video is marked as failed and removed from the queue as well. We allow up to 3 timeouts to occur, re-sending the request after the first 2 timeouts. Timeouts need to be chosen carefully in order to avoid both under-use of the wireless network as well as duplicate uploads. The UGC Collector maximizes throughput by selecting and requesting the next content as soon as a previous upload has been either completed or failed and new content is still available.

4.3.2 Multiple-request operation

Event directors may want to receive (collections of) videos with different characteristics concurrently, i.e. to have multiple requests simultaneously active. One example is the desire to collect videos from different locations at the event venue.

The UGC Collector maintains one priority queue for every active request. Each queue is filled using the top-k score for the corresponding request and emptied using the backfilling mechanism introduced in the previous section. Each queue will thus hold all announced videos, but in a different order. Hence all the queues have the same length and are updated at the same time. As we assume the number of concurrent requests to be fairly low, this does not introduce significant performance overhead.

A fairness mechanism governs the selection from which request queue the next upload should be selected. Each request is associated with a priority (a numerical value between 1 and 9). Using these priorities as weights, a request queue is selected for the next upload. Moreover, an exponential decay is used, halving the priority every one hundred seconds. This introduces a bias towards recently-made requests, as these are likely to be of more importance to the event director. However, given sufficient videos being announced by the crowd, every active request is still selected from time to time.

Two exceptions regarding the fairness system exist however. If a request has priority 10, it has absolute priority and only videos at the top of the corresponding queue will be selected. This allows the event director to quickly gather videos with specific characteristics if needed. If a request has priority 0, it will never be selected unless no other active requests are present. Such a request is automatically added when the UGC Collector is started and contains no constraints, which implies that by default the UGC Collector will upload as much content as possible in a most-recent-first fashion.

Moreover, a request can optionally be registered with an expiration time or a limit on the number of selected videos. When either the expiration time is exceeded or enough videos are

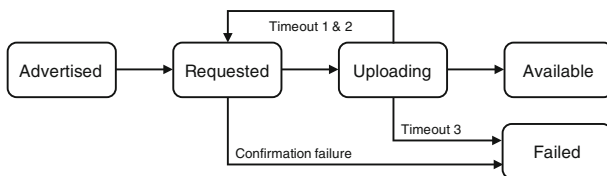


Fig. 6 State transitions of content in the Metadata Repository

collected for that request, it is disabled automatically. Finally, when a video is successfully uploaded, the UGC Collector will check if that video also would have been suitable for any other active requests, and if so, notify the Director as well. Note that each video is only uploaded once, even if it is suitable for multiple requests.

4.4 Director console

This module is a combination of hardware and software components that enable content management, visualization and playout features for large amounts of UGC. This component is the bridge in the video chain that enables creative use of user-generated content, originating from different source devices, in a professional event production as can be seen in Fig. 7. The DC acts as a browsable repository for UGC and a transcoder, composer and playout server towards a professional production chain. For scalability reasons, the DC is split over two physical systems: the visualization and the playout system.

4.4.1 Visualizing and selecting UGC

The user interface of the Director Console, shown in Fig. 8, allows the (human) event director to navigate through incoming content and to manually select individual video clips to be used in the creative composition process. To assist the selection process, the event director can filter the content by setting ranges on video metadata parameters.

4.4.2 Content composition and physical playout

The event director can use a number of creative layouts to combine the uploaded videos. The DS module contains a number of built-in layouts, see e.g. Fig. 9, but the director can also create his own lay-outs, containing up to 16 videos that might partially overlap.

The DC is fully integrated in a professional event production chain. Specifically, the DC creates a number of physical playout channels outputting video according to the HD/SDI standard [23], a common connection type on professional equipment such as cameras, video mixers, production screens, etc. The director can select one layout (e.g., 1x3, 2x2, 3x3, or 4x4 mosaic) for each playout channel, and the DC module will automatically fill the channel layout with the selected mobile videos. Although the mobile app provides some control on

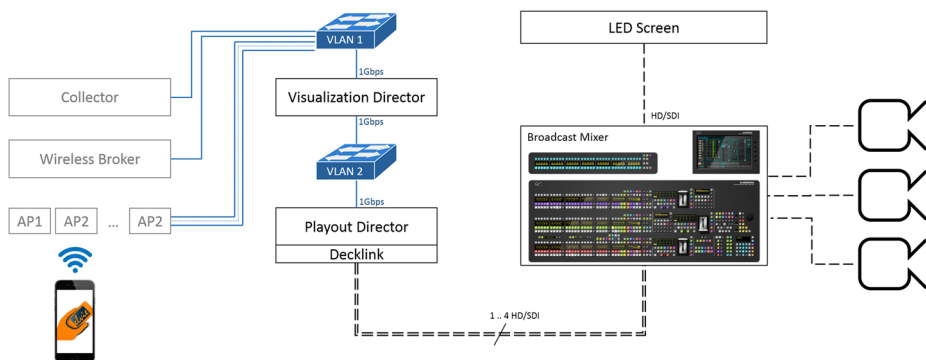


Fig. 7 Director hardware architecture

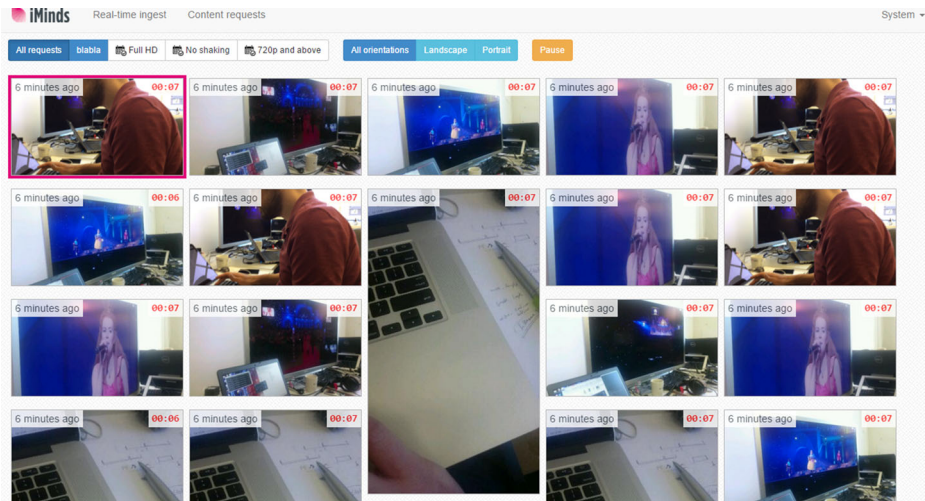


Fig. 8 User interface that allows navigation through incoming UGC content

video resolution and framerate, transcoding the uploaded UGC is a necessity. If the payout reaches the end, the DC will start over to make sure that the output never freezes.

4.5 Post-production support: personalized video souvenir

To prolong the experience of seeing personal video being integrated with professional content, as well as to provide an additional incentive for the users to use the app instead of their regular camera app, the Post-Production Support module generates a personalized

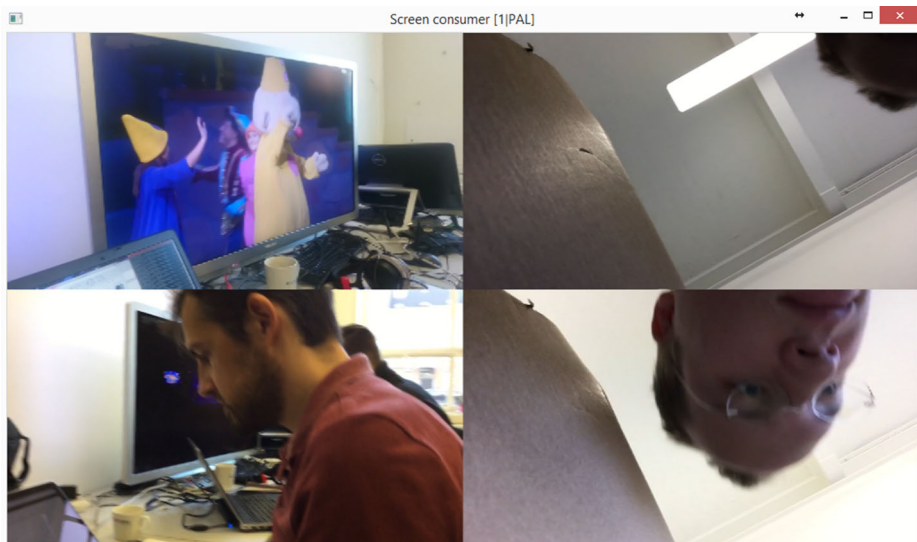


Fig. 9 Output on the physical HD/SDI channel

video souvenir. The video souvenir consist of 5 seconds of professional video content that are pre- and post-appended to a 7 second video manually selected by the user. The event audio stream (the singing band) is time-synchronized over the entire video. The app is remotely triggered to show a notification to the user as a call for action. The user receives a notification once the merged video is available.

The Post-Production Support module is deployed in the cloud, as a peak in the video processing load is to be expected immediately after the event. The professional event recording is streamed to the cloud as 10 second HTTP Live Streaming video fragments, next to additional creation timestamps. Since the professional video encoder (Blackmagic H.264 Pro Recorder) introduces some delay, a sample video recorded before the event is used to estimate this delay so it can be compensated for and the user video stays time-synchronized with the professional audio.

4.6 System implementation

To facilitate on-site deployment, our system is located in a self-contained rack holding the major components (Wireless Broker, UGC Collector, Director Console) on separate servers, as shown on the right hand side of Fig. 10. Table 1 lists the hardware specifications. The Director Console has been split into 2 separate machines to better cope with the load. The different servers are connected via a managed 1 Gbps switch. To allow for ease of management without hindering video throughput, separate VLANS were created for management and video data. The software frameworks and libraries used for the implementation are listed in Table 2. All communication is handled through RESTful web services.

5 System scalability

Before testing our system at the large-scale field trial, we validated its performance in a more controlled setting. A setup was built on the iLab.t Virtual Wall testbed [14], which supports

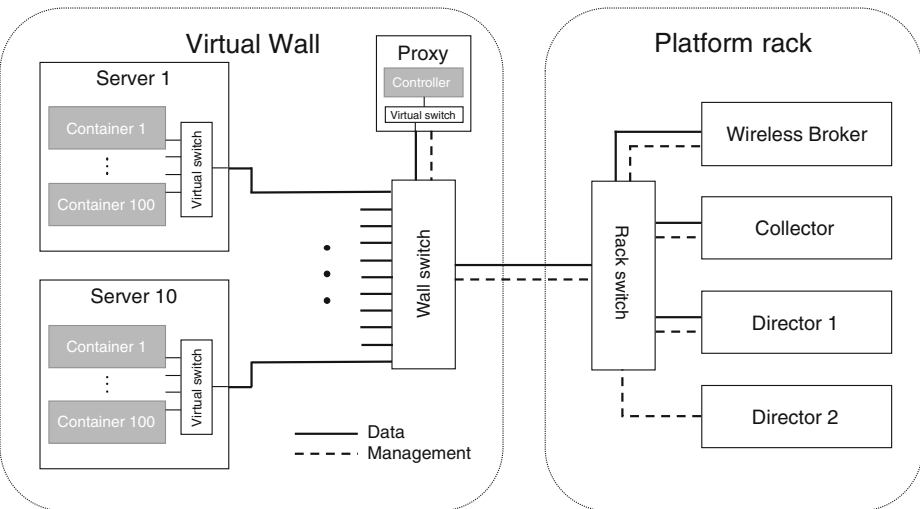


Fig. 10 Setup for scalability tests on the iLab.t Virtual Wall testbed

Table 1 Hardware specifications of the entire system

Component	Hardware specifications
UGC Collector	Intel XEON E3-1220 v3, 16GB RAM
Wireless Broker	Intel XEON Haswell, 8 GB RAM
Director Console	2x Intel XEON E5-2650 v2, 32 GB RAM
Video Repository	Intel NVMe 1.2 TB
Post-Production Support	Amazon EC2 m4.10xlarge, 40 vCPU, 160 GB RAM Blackmagic H.264 Pro Recorder
Access Points	Meru 3x3:3 AP832e, 802.11a/b/g/n/ac
Access Point Controller	Meru MC1550 WLAN controller
Virtual Wall (client simulation)	Intel XEON E3-1220 v3, 16 GB RAM

fast automated network emulation. The test set-up is shown in Fig. 10. A number of servers on the Virtual Wall are connected to the data VLAN via a managed switch. An additional server, the Proxy, allows for remote access to the entire setup and acts as controller for the servers simulating the clients. All connections are 1 Gbps links. In the absence of a sizable number of smartphones, we use the Docker container platform [17] to simulate clients. Containers are a more lightweight alternative to virtual machines, allowing us to rapidly start and stop new clients and to co-locate up to 100 clients per server. In total, we utilized ten physical servers resulting in 1 000 simulated clients simultaneously advertising and uploading content. The containers are connected to the data VLAN using a virtual bridge and their lifecycle is managed by the Proxy server. This implies that no wireless connection is present or simulated. However, this limitation does not affect the conclusions on system scalability.

The simulated clients themselves are written in Python and provide and use the same interfaces as the smartphone app. However, unlike the smartphone application, the simulated clients do not have a camera to record videos, nor a user to press record. Instead, we provide each client with a configuration file that lists the times during the experiment at which it should announce metadata to our framework. We also include in each container the same 10 mobile videos. The start time of recording a 7s video is generated as a Poisson process with mean interarrival time of 30 seconds. During the simulation, the client container randomly maps these recordings to one out of the 10 mobile videos. While the video size is

Table 2 Libraries and software frameworks used for the implementation

Component	Libraries and frameworks
Mobile App	RenderScript (Android, video metadata calculation)
Wireless Broker	Python, Flask [10] (REST interfaces), Paramiko [19] (SSH to AP controller), Autobahn [2] (WebSocket connections to mobile app)
UGC Collector	Node.js, Java 8 (top-k matching, content queue)
Director Console	Node.js, CasparCG [6] (playlist middleware), AngularJS
Post-Production Support	Node.js, FFmpeg

taken from the length of the video selected, the other metadata parameters are sampled uniformly at random (and independent per metadata parameter) from the valid ranges for each metric. For 1 000 simulated clients, this results in over 30 videos per second being advertised on average. If the UGC Collector triggers an upload, the corresponding mobile video is transmitted.

A final distinction between this setup and the real system is the lack of an access point controller, which is required by the Wireless Broker in order to determine the locations of each client. Instead, an additional container is started on the Proxy server which simulates the access point controller. Given a list of client IDs the simulated controller assigns each client randomly to one of 10 access points, and simulates client movement by changing the access point with a small probability.

Performance results for the UGC Collector component are shown in Fig. 11. The system one-minute load average (as reported by `top`) of the UGC Collector never exceeds 50 %, even when a thousand clients are actively registering and uploading videos. As full system load is 400 % for a quad-core processor, there is substantial capacity left. A similar trend is visible for the CPU usage, never exceeding 12 %. An important thing to note, however, is that from about 300 clients onward, the gigabit link used for the video uploads from the client to the Video Repository becomes fully saturated. This implies that from that point on, the UGC Collector queues will continue to grow in size until the full memory capacity has been reached. If needed, the size of the UGC Collector queue can be limited by discarding the oldest videos for upload.

Although the system load and CPU usage of the Wireless Broker are substantially higher than for the UGC Collector, the Wireless Broker is still well equipped to deal with all the client connections, as proven by Fig. 12. We remind the reader that the presented results are for a quad-core processor, hence the maximum system load and CPU usage are 400 %, which indicates that the server still has sufficient capacity left even when 1 000 users are simultaneously connected. Starting and managing a different thread for each upload WebSocket request to a user device (to prevent blocking other Wireless Broker functionality) and periodically communicating with the access point controller to construct the locations list

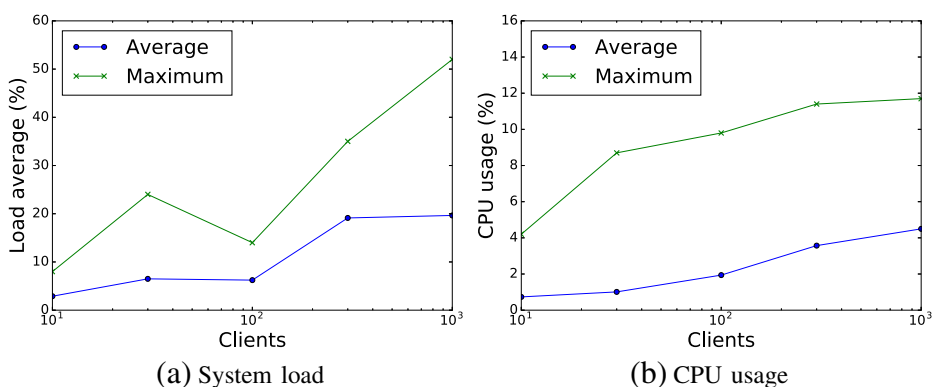


Fig. 11 Performance of UGC Collector component for different numbers of clients. Numbers are collected during a five-minute period after the final client container was started. CPU load and usage are as reported by `top`

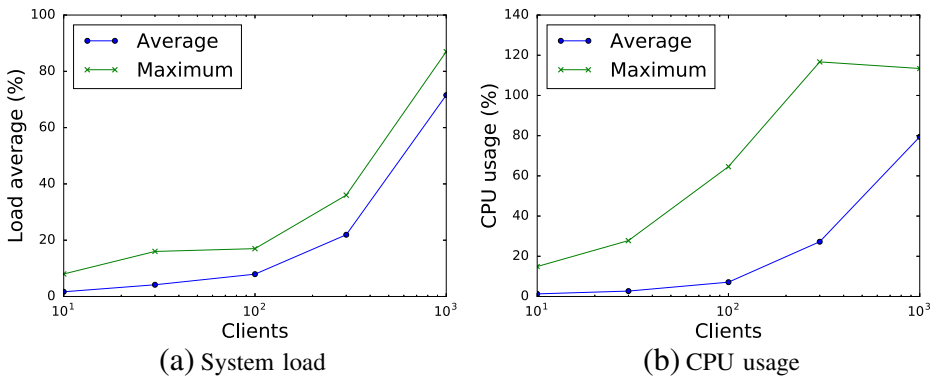


Fig. 12 Performance of Wireless Broker component for different numbers of clients. Numbers are collected during a five-minute period after the final client container has started. CPU load and usage are as reported by `top`

could explain the higher resource consumption in comparison to the more powerful UGC Collector server.

6 Large-scale field trial

Our platform has been validated during a commercial event in February 2016, at the Lotto Arena in Antwerp, Belgium. The event was a show of K3, a Belgian girl band aimed mainly at children and their parents. The show is part of a series where the three former members pass the torch to three new girls, selected via a TV show. The show was sold out, with the Lotto Arena having a capacity of 8 050 persons. All attendees were seated on the central square or on the higher ring. By analyzing the wireless traffic during the show, we detected 6 414 unique devices. Note that the show is explicitly targeting children who might not all possess a smartphone. The format of the event makes it very likely that parents will upload videos of their children dancing and enjoying the show.

We reserved 500 tickets in 4 different seat blocks in the event venue. Participants were recruited by announcements in the official sales channels and in the national press [12] and given a 20 % reduction on the ticket fee. A few days prior to the show, ticket owners were provided with layman instructions to install and use the app. At the sole discretion of the event director, uploaded videos were shown 41 times during the show of 90 minutes, as a 4x4 or 1x3 video mosaic blended with the usual event production (cf. Fig. 13).

6.1 Set-up

At both sides of the stage there was a large LED screen on which the uploaded videos were shown (cf. Fig. 13). Participants were sitting in one of the 4 seat blocks marked in Fig. 14. A dedicated Wi-Fi network was deployed at the event venue: six access points were deployed at strategic locations and supported both 2.4 and 5 GHz associations. The two seat blocks closest to the stage were covered by two APs each and the two seat blocks at the end of the stage were also both covered by an AP. The Meru AP832e APs were connected to the Meru MC1550 WLAN access point controller over a 1 Gbps wired network that also connected

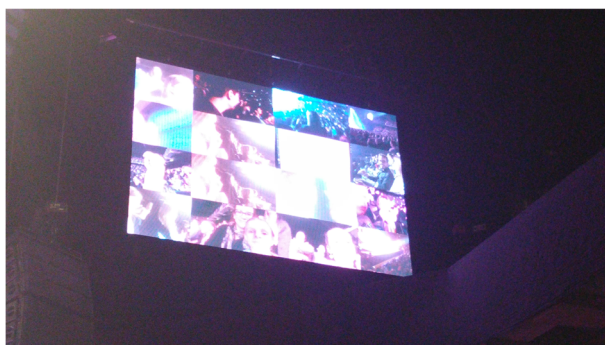


Fig. 13 A 4x4 video mosaic shown on the large displays next to the podium stage, blended with the professional camera video streams

all the platform components which were stacked backstage. Our self-contained platform rack was connected to the production environment of the event.

6.2 Video and device statistics

In total, 366 devices connected to the dedicated wireless network at least once. The open-access network not only allowed users to register with the framework and upload videos but also provided free Internet access. Out of those 366 devices, 135 devices (including 14 devices of the researchers on-site) interacted at least once with the platform.

The distribution of mobile operating systems is shown in Fig. 15. While we expected that this distribution would follow the market, almost half of all devices (47.3 %) were Apple devices running a version of the iOS operating system. The second most used operating system was Android (32.5 %), followed by Windows phones (19.1 %) and a small percentage (1.1 %) could not be identified. Of the devices that interacted with the platform, there were more than twice as many Apple devices than Android devices. The right-hand side

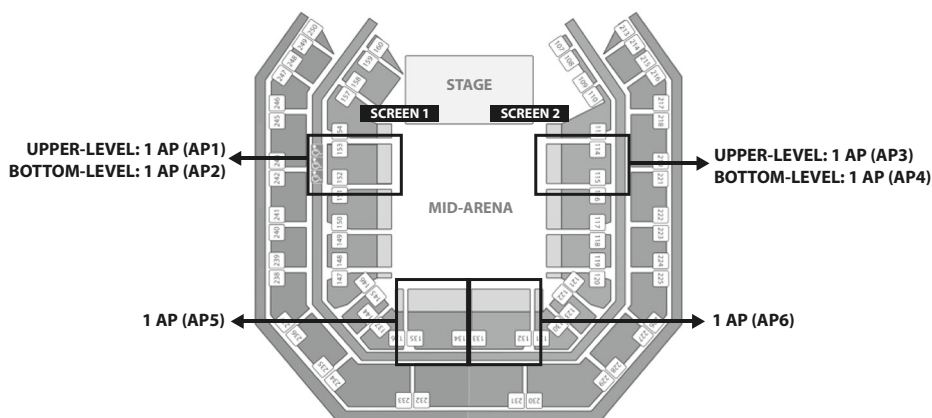


Fig. 14 The set-up of the event venue where the platform was tested. Six Meru AP832e APs deployed a dedicated network for the platform participants sitting in the outlined seat blocks, allowing them to upload their videos and see these on the two large screens next to the stage

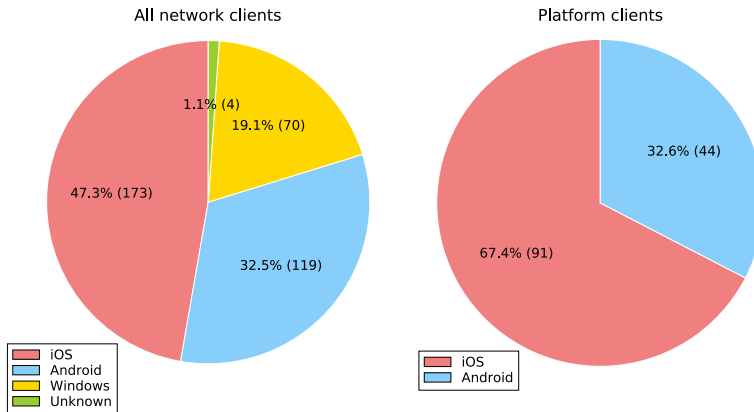


Fig. 15 Distribution of different operating systems as used by all the clients (including non-participants) associated with the dedicated wireless network and the clients that were registered with the platform

of Fig. 15 indicates the distribution of the 135 devices that interacted with the platform via our app. As the application only supported these two operating systems, no Windows users participated.

In total, 1 383 videos were recorded, of which 1 183 were successfully uploaded to the platform. Table 3 gives the distribution of the video orientation per resolution and per content type. The content types were manually labeled as one of following categories. “Stage” is used for videos that mainly capture the performance of the band, “close-up” is used for videos of nearby people (e.g. parents filming their children) and “ambience” is used for videos that mainly capture the atmosphere (e.g. by panning over the event venue). “Other” covers the remaining videos, which include for example accidental recordings of the ceiling. Finally, videos of which the metadata was registered but the video itself was not uploaded are labeled as “unknown”, as these could not be labeled.

A first observation is that the majority (87.0 %) of the videos were recorded in landscape orientation. Three quarters of the uploaded videos (74.9 %) were close-ups of members of the audience, mostly children. Almost two thirds (62.0 %) of videos were recorded in full HD, with the remaining videos being mostly (35.1 %) recorded in 720p, illustrating the presence of high-resolution cameras in today’s smartphones. Our app automatically selects the highest resolution available in the device.

Table 3 Number of uploaded videos per resolution and per type

	480p	720p	1080p	2160p	Total
Stage	0 (0/0)	11 (2/9)	128 (17/111)	2 (0/2)	141 (19/122)
Close-up	4 (0/4)	388 (32/356)	469 (72/397)	25 (11/14)	886 (115/771)
Ambience	0 (0/0)	10 (2/8)	120 (19/101)	4 (2/2)	134 (23/111)
Other	0 (0/0)	2 (0/2)	19 (1/18)	1 (0/1)	22 (1/21)
Unknown	0 (0/0)	75 (14/61)	122 (7/115)	3 (1/2)	200 (22/178)
Total	4 (0/4)	486 (50/436)	858 (116/742)	35 (14/21)	1383 (180/1203)

The numbers between brackets indicate portrait/landscape split

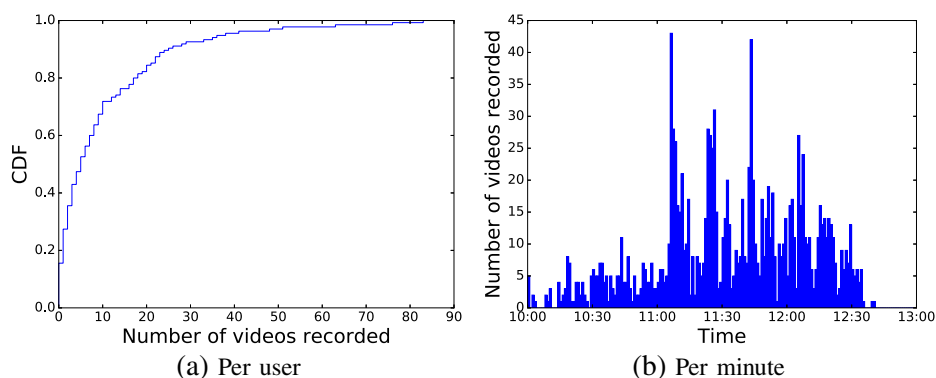


Fig. 16 Characteristics of the number of videos recorded per user and over time during the event

Figure 16 provides more details on how these videos are distributed over users and time. 21 out of 135 users did not record any videos at all even though they did activate the app, while the most active user recorded 83 videos. On average, a single user uploaded 10 videos. In Fig. 16b, it can be seen that users started experimenting with our app as soon as they entered the venue (doors of the venue opened at 10 AM). During the entire show, there was a continuous ingress of new videos, but some clear peaks can be observed. The first, and highest, peak is at the start of the show at 11.10 AM,² during which 43 videos were recorded in one minute. We analyzed the show storyboard and observed that the other peaks can be attributed to highlights in the show. For example, a significant peak is observed around 11.25 AM, which corresponds to the new members of K3 entering the stage for the first time. Two other significant spikes occur at 11.45 AM and 12.10 PM, both corresponding with the reappearance of the new members as well.

6.3 Network statistics

This section discusses usage of the wireless network in more depth. Figure 17 shows the number of associated network clients and the number of registered clients with the platform. During the first hour people started entering the event venue. The number of clients increased linearly until it reached the maximum number of 285 concurrent associated users at the moment the artists entered the stage at 11.10 AM. The number of platform clients peaked at the same moment with 70 concurrent registered devices. Between 11.00 AM and 12.45 PM there were on average 270 clients associated with the Wi-Fi network and 37 clients registered with the platform.

The frequency band usage is plotted in Fig. 18 and shows that the number of devices that used the 2.4 GHz frequency band was less than half the number of 5 GHz users. On average from 11.00 AM until 12.45 PM there were 86 2.4 GHz clients and 176 5 GHz clients. For the platform registered devices, the difference was even larger as there were on average 29 5 GHz and only 7 2.4 GHz clients. These numbers are explained by the fact that there was a majority of Apple devices (see Fig. 15) of which the modern versions all support the 5 GHz frequency band while for Android and Windows phones this is not yet the case.

²The management delayed the start of the show for 10 minutes (until 11.10 AM) because of traffic jams.

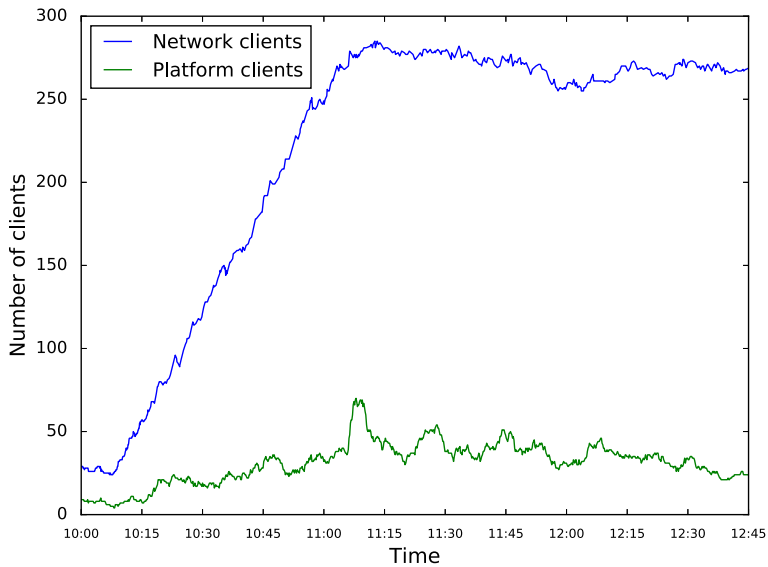


Fig. 17 The number of clients connected to the wireless dedicated network and the number of clients registered with the platform over time

6.4 UGC collection statistics

Over the entire trial, 135 clients registered 1 383 videos in total of which 1 308 were requested by the UGC Collector. The remaining 75 videos were never requested because

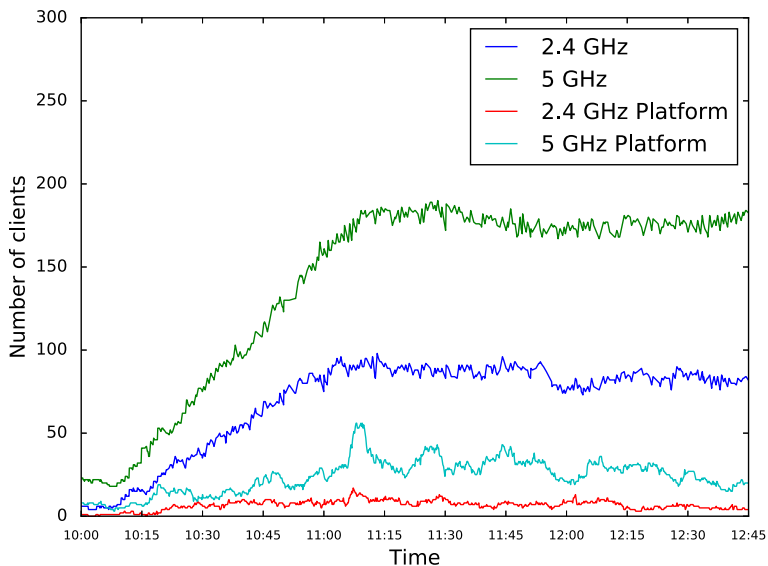


Fig. 18 The frequency band usage of all clients connected to the wireless network and the registered clients over time

they were never at the top of the queue while the client was online. 1183 videos were eventually successfully uploaded. The other 125 uploads were either reported as failed by the Wireless Broker or timed out. Figure 19 shows the number of videos for which a certain amount of requests were made, and how many of those succeeded. 519 retransmission attempts were made, resulting in a total of 1 827 upload requests. Most videos (exactly 1000) were successfully uploaded on the first try. Of the 221 videos for which three attempts were made, 108 were eventually never successfully uploaded. Of the videos that did upload successfully, 57 were uploaded twice and 51 three times due to the retry mechanism.

Using location information fetched from the access point controller, Table 4 shows different statistics per AP. The exact location of each AP can be seen in Fig. 14. The total number of unique registrations is higher than the 135 unique clients that registered with the platform, meaning that multiple devices connected to different APs at different times. Most clients were in the range of more than one AP and depending on the current signal quality the client device could have changed to another AP in order to have the best possible connection. Both the number of unique and average registrations clearly show that the audience sitting in the range of AP6 were most active. Clients sitting near AP6 also recorded most videos (37.5 %) and the UGC Collector requested most videos (34.0 %) from this part of the event venue. This could be explained by the fact that these users had a front view of the stage and were able to record the best-looking videos.

The reader should note that the number of upload requests is more evenly distributed than the number of recorded videos. This is the result of the UGC Collector optimizing bandwidth utilization over the different APs. While the total number of requested videos is less than the total number of recorded videos, for all APs besides AP6 there are more requests than recorded videos. This means that some clients first recorded a video while associated to a particular AP, re-associated with the network and received the request via another AP for the video.

The fill rate of the UGC Collector queue is drawn in Fig. 20. As described in Section 4.3, the UGC Collector maintains one queue per request of the director. These queues are all of the same size. The number of items in the queue reflects the number of videos that are

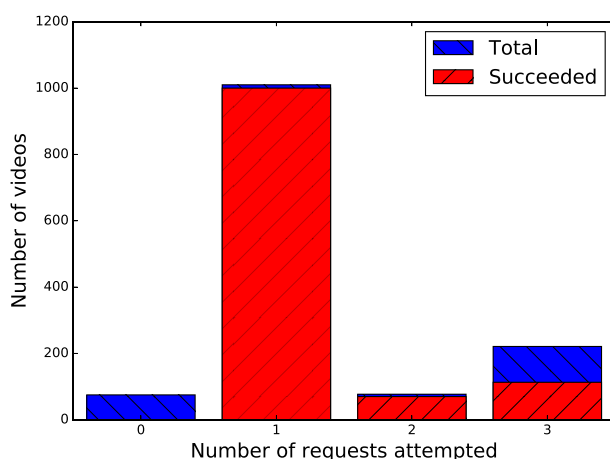


Fig. 19 The number of videos for which a certain number of requests were attempted, as well as how many of those succeeded

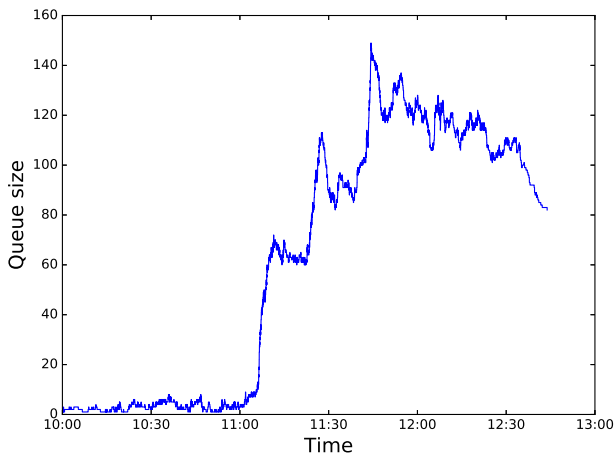
Table 4 Detailed information on the number of associations, recorded and requested videos per AP

AP	Unique registrations	Average registrations	Recorded videos	Video requests
AP1	12	1	63 (4.6 %)	67 (5.1 %)
AP2	48	7	330 (23.9 %)	351 (26.8 %)
AP3	21	2	66 (4.8 %)	68 (5.2 %)
AP4	59	7	345 (25.0 %)	356 (27.2 %)
AP5	12	0	10 (0.7 %)	21 (1.6 %)
AP6	86	19	518 (37.5 %)	445 (34.0 %)

Note that an additional 51 videos were recorded at an unknown AP at the time of submission.

currently either being uploaded or that were advertised but not yet selected for upload. As can be seen, the system failed to upload 82 videos, 75 of which were never selected because they did not arrive at the top of the upload queue. One reason is that when users put their smartphone in their pocket, the Wi-Fi connection is lost and the user's videos are never selected for upload until they reconnect. When they do, however, the UGC Collector is unlikely to select their videos as it prioritizes the upload of newer videos. The remaining 7 videos in the queue were pending uploads when the UGC Collector was shut down at the end of the trial after all clients had disconnected, so these were about to timeout as well.

Another important aspect of the UGC Collector's performance is its ability to prioritize videos. We recall the reader that the Collector prioritizes more recent videos, although its overall purpose is to collect as much content as possible. Figure 21 gives more insight in the age of uploaded videos. Figure 21a is a CDF of the time between the video advertisement (metadata received by the UGC Collector) and the (first) upload request. While 63.4 % of videos is requested within 1 second of advertisement, some videos are only requested after tens of minutes. This illustrates that while the UGC Collector prioritizes more recent videos, it will collect as much content as possible in its current configuration. Figure 21 shows the CDF of the time between the metadata advertisement and the video being available at the

**Fig. 20** No. of video items in the upload queue of the UGC Collector

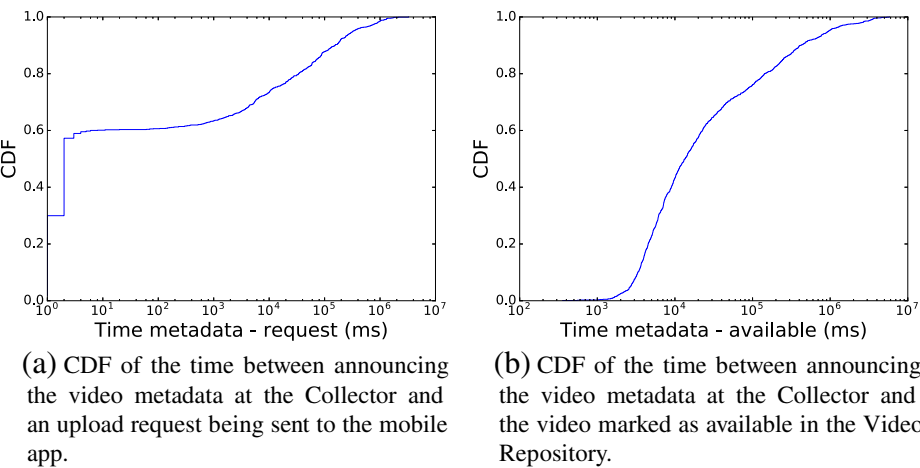


Fig. 21 Information of the age of videos before they are available to the director.

Director Console module. Here we observe that 50 % of videos is uploaded within 13.5s after the end of recording and 90 % within 6.8 minutes. Inversely, 75.2 % of videos were uploaded within 1.5 minutes.

6.5 Playout and post-production statistics

During the event, an assistant director was constantly selecting videos to display in a 4x4 or 3x1 mosaic. The main director decides at his own discretion to show these mosaics on the large screens. Table 5 gives a distribution overview for all items that entered and were selected for playout based on the video labels described in Section 6.2. From these distributions, a 2-sample chi-square test for equality of proportions reveals that the assistant director had a slight preference for selecting close-up videos (269/336 compared to 617/847 p-value = 0.0122). The other type of videos have been selected and displayed at the same ratio as they were recorded and provided by the users.

As discussed in Section 4.5, attendees could request a personalized video souvenir after the event. This was handled via a separate cloud service. Figure 22 gives an overview of the number of personalized video requests based on certain timespans. This table shows that a personalized video service is mainly used on the event day itself, or maximum 2 days after the event. Almost half of the requests are made between 0 and 5 hours after the end of the event.

Table 5 Number of uploaded videos per type and whether or not these are selected for display by the human director

	Stage	Close-up	Ambience	Other	Total
Selected	37	269	28	2	336
Not selected	104	617	106	20	847
Total	141	886	134	22	1183

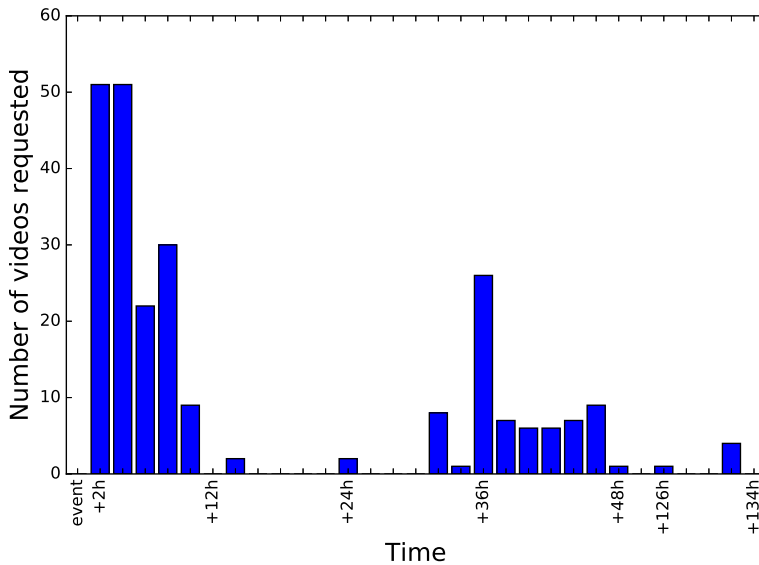


Fig. 22 Number of personalized videos requested over time

7 Conclusion and outlook

In this article, we presented a scalable platform to increase the overall audience involvement at large-scale events by collecting user-generated content and combining it into mosaics that are readily available for integration in the event production. Our platform was tested in a large field trial at a commercial show. The collected content was shown on the event screens, as well as used in the cloud to generate a personalized video souvenir.

Using commodity hardware servers, we are able to serve up to 1000 concurrent users. The framework has been validated during a concert of the K3 band, aiming at parents filming their children. During this event, the audience offered 1383 videos to our platform. We also gathered valuable insight in the recording behavior of the audience. The high proliferation of cameras capable of recording in 1080p thus was observed during the event is clearly proves that there is sufficient potential to capture videos with high resolution. Our framework is able to upload 90 % of the recorded videos within 6.8 minutes.

As future work, we are considering to validate our platform during an event covering a larger geographical venue, e.g. a music or street festival. In this case, the videos recorded by users can be used to capture the long tail of content on the event. One example are circuit-based sports like cyclo-cross, where professional cameras cannot capture every angle. If an important event happens at a location not covered professionally, user videos could be quickly collected and uploaded via a targeted director request. Another interesting research direction would be to (partially) automate the video selection process, for example based on the statistics of previously, manually selected videos or using a feedback loop.

Acknowledgments The results presented in this paper were partially funded through the MIX-ICON program of iMinds. Steven Bohez is funded by a research grant of VLAIO. The authors would like to explicitly thank the colleagues from Studio 100, Videohouse and Multicap.

References

1. Arev I, Park HS, Sheikh Y, Hodgins J, Shamir A (2014) Automatic editing of footage from multiple social cameras. *ACM Transactions on Graphics (TOG)* 33(4):81
2. Autobahn—python a websocket and wamp library for python, <http://autobahn.ws/python/>
3. Bailer W, Pike C, Bauwens R, Grandl R, Matton M, Thaler M (2015) Multi-sensor concert recording dataset including professional and user-generated content Proceedings of the 6th ACM multimedia systems conference, pp 201–206. ACM
4. Bailer W, Weiss W, Wechtitsch S (2016) Selecting user generated content for use in media productions Multimedia modeling, pp 388–393. Springer
5. Borst S (2005) User-level performance of channel-aware scheduling algorithms in wireless data networks. *IEEE/ACM Transactions on Networking* 13(3):636–647. doi:10.1109/TNET.2005.850215
6. Casparcg Pro graphics and video playout system, <http://www.casparcg.com/>
7. Chen X, Hwang JN, Lee KH, de Queiroz RL (2015) Quality-of-content (qoc)-driven rate allocation for video analysis in mobile surveillance networks. In: *Multimedia Signal Processing (MMSP)*, 2015 IEEE 17th International Workshop on, pp 1–6. doi:10.1109/MMSP.2015.7340838
8. Culhane W, Jayaram KR, Eugster P (2014) Fast, expressive top- k matching. In: *Proceedings of the 15th International Middleware Conference on - Middleware '14*, pp. 73–84. ACM Press, New York, New York, USA. doi:10.1145/2663165.2663326
9. Daneels G, Famaey J, Bohez S, Simoens P, Latre S (2015) Upstream Content Scheduling in Wi-Fi DenseNets during Large-Scale Events. In: *2015 IEEE Globecom Workshops (GC Wkshps)*, pp 1–7. IEEE. doi:10.1109/GLOCOMW.2015.7414130
10. Flask a microframework for python, <http://flask.pocoo.org>
11. Ganguly S, Chatterjee M, Izmailov R (2006) Non-real-time content scheduling algorithms for wireless data networks. *IEEE Transactions on Computers* 55(7):893–905. doi:10.1109/TC.2006.106
12. <http://www.hln.be/hln/nl/946/Muziek/article/detail/2599686/2016/01/28/K3-show-pioniert-met-videoselfie-de-Selvie.dhtml> [in dutch]
13. Huang S, Izquierdo E, Hao P (2016) Bandwidth-efficient packet scheduling for live streaming with network coding. *IEEE Transactions on Multimedia* 18(4):752–763. doi:10.1109/TMM.2016.2530411
14. iMinds iLab.t Virtual Wall, <http://ilabt.iminds.be/iminds-virtualwall-overview>
15. Jain P, Manweiler J, Acharya A, Beaty K (2013) Focus: clustering crowdsourced videos by line-of-sight Proceedings of the 11th ACM conference on embedded networked sensor systems, p 8. ACM
16. Lin Y, Wu Q (2013) Complexity analysis and algorithm design for advance bandwidth scheduling in dedicated networks. *IEEE/ACM Transactions on Networking* 21(1):14–27. doi:10.1109/TNET.2012.2189127
17. Merkel D (2014) Docker: lightweight linux containers for consistent development and deployment. *Linux Journal* 2014(239):2
18. Nguyen DTD, Saini M, Nguyen VT, Ooi WT (2013) Jiku director: a mobile video mashup system Proceedings of the 21st ACM international conference on multimedia, pp 477–478. ACM
19. Paramiko, a python implementation of sshv2, <http://www.paramiko.org/>
20. Rao N, Wu Q, Ding S, Carter S, Wing W, Banerjee A, Ghosal D, Mukherjee B (2006) Control plane for advance bandwidth scheduling in ultra high-speed networks. In: *25th IEEE International Conference on Computer Communications (INFOCOM)*, DOI doi:10.1109/INFOCOM.2006.35, (to appear in print)
21. Saini M, Venkatagiri SP, Ooi WT, Chan MC (2013) The jiku mobile video dataset Proceedings of the 4th ACM multimedia systems conference, pp 108–113. ACM
22. Shah RR, Hefeeda M, Zimmermann R, Harras K, Hsu CH, Yu Y (2016) NEWSMAN: Uploading Videos over Adaptive Middleboxes to News Servers in Weak Network Infrastructures, pp. 100–113. Springer International Publishing, Cham. doi:10.1007/978-3-319-27671-7_9
23. Smpte 292m, bit-serial digital interface for high-definition television systems, 1996
24. Tang A, Boring S (2012) # epicplay: crowd-sourcing sports video highlights Proceedings of the SIGCHI conference on human factors in computing systems, pp 1569–1572. ACM
25. Venkatagiri SP, Chan MC, Ooi WT, Chiam JH (2015) On demand retrieval of crowdsourced mobile video. *IEEE Sensors Journal* 15(5):2632–2642
26. Wilk S, Kopf S, Effelsberg W (2015) Video composition by the crowd: a system to compose user-generated videos in near real-time Proceedings of the 6th ACM multimedia systems conference, pp 13–24. ACM



Steven Bohez obtained a M.S. degree in Computer Science at Ghent University, Belgium, in 2013 and is currently working towards a Ph.D. at the IDLab research group at Ghent University - iMinds, funded by the Flanders Innovation & Entrepreneurship (VLAIO) agency. His research focuses on advanced and media-rich edge cloud applications that are distributed between mobile devices and the cloud.



Glenn Daneels is candidate Ph.D. student at the IDLab (imec) research group, University of Antwerp, Belgium. He received his B.S. and M.S. degrees in Computer Science from the University of Antwerp, in 2011 and 2013 respectively. In 2014, he graduated with a second M.S. degree in Artificial Intelligence from the University of Leuven, Belgium. His current research focuses on the improvement of IoT communication protocols and technologies, including the highly reliable networking standard IEEE802.15.4e.



Lander Van Herzele got his Master's degree in Applied Computer Science Engineering from Ghent University College, Belgium, in 2013 and has worked almost three years for iMinds as a research engineer. He has worked on various research projects involving scalable cloud backends.



Niels Van Kets received the M.Sc. degree in information and communication technologies from the Katholieke Hogeschool Kempen, Geel, Belgium, in 2011. Until 2013, he worked as system engineer for VRT, the Flemish public broadcasting station. These were the roots for his passion towards media. Currently he is a research engineer at IDLab, Ghent University. His research interests include new media formats, virtual reality, user interaction and quality of experience.



Sam Decrock is has a Master's degree in Electronics from the Katholieke Hogeschool Brugge-Oostende, Belgium, as well as a M.Sc. degree in Computer Science from the University of Leuven. Sam has worked as research engineer at iMinds for 5 years, working on both hardware and software prototype implementations for various research projects.



Matthias De Geyter Matthias De Geyter obtained his M.Sc. in Electronics Engineering at Ghent University, after which he joined iMinds-IPI-UGent. Two years later, he started at VRT's R&D department (the public broadcaster in Flanders), where he performed research on media adaptation and media exchange formats and their use in system integration. He joined VRT-medialab upon its foundation and continued this work, while also being involved in projects about 3DTV, file-based quality control and production systems in the cloud. In 2011, Matthias joined iMinds-MiX, where he contributed to various research projects related to media, such as personalized news, visual radio and smart toys.



Glenn Van Wallendael obtained the M.Sc. degree in Applied Engineering from the University College of Antwerp, Belgium, in 2006 and the M.Sc. degree in Engineering from Ghent University, Belgium in 2008. Afterwards, he worked towards a Ph.D. at IDLab, Ghent University, with the financial support of the Research Foundation Flanders (FWO). Currently, he continues working in the same group as a post-doctoral researcher. His main topics of interest are video compression including scalable video compression and transcoding.



Peter Lambert is a full-time Associate Professor at the Multimedia Lab of Ghent University – iMinds (Belgium) since 2013. He received his Master's degree in science (mathematics) and in applied informatics from Ghent University in 2001 and 2002, respectively, and he obtained the Ph.D. degree in computer science in 2007 at the same university. In 2009 he became a Technology Developer at Ghent University, which he combined with a part-time Assistant Professorship at the Multimedia Lab since 2010. His research interests include (mobile) multimedia applications, multimedia coding and adaptation technologies, and 3D graphics.



Bart Dhoedt received a Masters degree in Electro-technical Engineering (1990) from Ghent University. His research, addressing the use of microoptics to realize parallel free space optical interconnects, resulted in a Ph.D. degree in 1995. After a 2-year post-doc in opto-electronics, he became Professor at the Department of Information Technology. Bart Dhoedt is responsible for various courses on algorithms, advanced programming, software development and distributed systems. His research interests include software engineering, distributed systems, mobile and ubiquitous computing, smart clients, middleware, cloud computing and autonomic systems. He is author or co-author of more than 300 publications in international journals or conference proceedings.



Pieter Simoens received his M.Sc. degree in Electronic Engineering (2005) and Ph.D. degree (2011) from the Ghent University, Belgium. During his Ph.D. research, he was funded by the Fund for Scientific Research Flanders (FWO-V). In 2012, he was a visiting researcher at the School of Computer Science of Carnegie Mellon University, USA. Currently, he is assistant professor affiliated with the Department of Information Technology of the Ghent University and with iMinds. His main research interests include mobile cloud offloading, serviceoriented networking, edge/fog computing paradigms, and service engineering for advanced mobile applications. In these fields, he is author and co-author of more than 70 papers published in international journals or in the proceedings of international conferences. He has also been involved in several national and European research projects (FP6 MUSE, FP7 MobiThin, H2020 FUSION).



Prof. Steven Latré is an assistant professor at the University of Antwerp and the research centre imec in Belgium. He received a Master of Science degree in computer science from Ghent University, Belgium and a Ph.D. in Computer Science Engineering from the same university. He is currently co-leading the IDLab research group, which focuses on Internet of Things and data analytics. His own research expertise focuses on management and control of Internet of Things environments. W He is author or co-author of more than 75 papers published in international journals or in the proceedings of international conferences. He is the recipient of the IEEE COMSOC award for best PhD in network and service management 2012, the IEEE NOMS Young Professional award 2014 and is a member of the Young Academy Belgium.



Jeroen Famaey is an assistant professor associated with imec and the University of Antwerp, Belgium. He received his M.Sc. degree in Computer Science from Ghent University, Belgium in 2007 and a Ph.D. in Computer Science Engineering from the same university in 2012. He is co-author of over 60 articles published in international peer-reviewed journals and conference proceedings, as well as 8 submitted patent applications. His research resulted in 4 best paper awards and he was the recipient of the 2015 IEEE Communications Society Outstanding Reviewer award. His research interests include MAC protocol and algorithm design for wireless communications, coexistence of wireless networks, and deterministic lowlatency wireless communications, with a focus on 802.11 and 802.15.4 technologies.