Multimedia Tools and Applications
https://doi.org/10.1007/s11042-020-09616-9

Show me where the action is!

Automatic capturing and timeline generation for reality TV.

Timothy Callemein[1] · Tom Roussel[1] · Ali Diba[1] · Floris De Feyter[1] · Wim Boes[1] · Luc Van Eycken[1] · Luc Van Gool[1] · Hugo Van hamme[1] · Tinne Tuytelaars[1] · Toon Goedemé[1]

Received: 9 September 2019 / Revised: 26 June 2020 / Accepted: 12 August 2020 /
Published online: 02 September 2020

Abstract
Reality TV shows have gained popularity, motivating many production houses to bring new variants for us to watch. Compared to traditional TV shows, reality TV shows have spontaneous unscripted footage. Computer vision techniques could partially replace the manual labour needed to record and process this spontaneity. However, automated real-world video recording and editing is a challenging topic. In this paper, we propose a system that utilises state-of-the-art video and audio processing algorithms to, on the one hand, automatically steer cameras, replacing camera operators and on the other hand, detect all audiovisual action cues in the recorded video, to ease the job of the film editor. This publication has hence two main contributions. The first, automating the steering of multiple Pan-Tilt-Zoom PTZ cameras to take aesthetically pleasing medium shots of all the people present. These shots need to comply with the cinematographic rules and are based on the poses acquired by a pose detector. Secondly, when a huge amount of audio-visual data has been collected, it becomes labour intensive for a human editor retrieve the relevant fragments. As a second contribution, we combine state-of-the-art audio and video processing techniques for sound activity detection, action recognition, face recognition, and pose detection to decrease the required manual labour during and after recording. These techniques used during post-processing produce meta-data allowing for footage filtering, decreasing the search space. We extended our system further by producing timelines uniting generated meta-data, allowing the editor to have a quick overview. We evaluated our system on three in-the-wild reality TV recording sessions of 24 hours ($\times$ 8 cameras) each taken in real households.

Keywords Autonomous PTZ steering · Event timeline · Sound recognition · Facial recognition · Action recognition · Reality TV

This work was made possible by the Belgian production house Geronimo, the KULeuven GOA project CAMETRON and the Research Foundation Flanders (FWO-Vlaanderen).

✉ Timothy Callemein
timothy.callemein@kuleuven.be

[1] Katholieke Universiteit Leuven, Sint-Katelijne-Waver, Belgium

1 Introduction

These days, collecting audiovisual data is easier and cheaper than ever before. Cameras and microphones have become ubiquitous and the storage for the data they collect has expanded. This allows for devices that continuously record their surroundings. To generate entertaining reality TV shows from these continuous recordings, however, one needs to be able to separate the wheat from the chaff. Recent progress in machine learning offers solutions for this information-extraction exercise.

In this paper, we explore the possibilities of the state-of-the-art in video and audio processing for a real-world use case with such continuous recordings. We demonstrate these in the system proposed, both during recording of the footage with our automatic camera man, and also afterwards with our audio-visual action detection and meta-data timeline extraction.

More specifically, we devised a system to help producers of TV shows capture a reality-TV show. To record spontaneous scenes, families were continuously filmed 24/7 in their own houses with PTZ cameras.

The use case of a reality-TV show, however, brings a challenge: the recorded footage must be of a decent cinematographic quality. One solution would be to hire professional camera men operating the cameras 24/7. In order not to disturb the filmed people showing their natural behaviour, it is best not to have these camera men walking around in the house itself, but remotely operate a set of steerable Pan-Tilt-Zoom (PTZ) cameras. Evidently, this solution is still expensive and very labour-intensive.

In our work, we propose a system that automatically creates cinematographic shots, i.e. the *automatic cameraman*. Based on real-time image interpretation of the situation in the room as observed by a wide-angle static overview camera, the PTZ camera will be steered towards a cinematographically correct shot taking into account canvas composition rules such as the *rule of thirds*, and the *head room* and *nose room* rules. Figure 1 illustrates this idea.

The second challenge tackled in this paper, is making the recordings manageable for the editors of the TV show. In the current work, we propose to create timelines that instantly give an idea of the actions: *what* happened *when* and *where* in the recordings and *who* was involved. Figure 2 illustrates such a meta-data timeline, visualising at what time which people were present and what actions occurred.

The remainder of the paper is structured as follows. Section 2 presents related work, highlighting existing problems and challenges. Section 3 describes the proposed approach to address such challenges. Our automatic camera man is detailed in Section 4, while Section 5 explains how the audiovisual data analysis techniques are used to build rich timelines, helping out the reality TV show editors. We evaluate each component of our system in Section 6

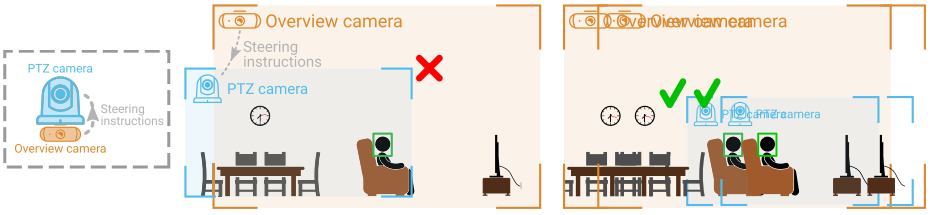


Fig. 1 Example situation where the PTZ is steered to capturing a better shot

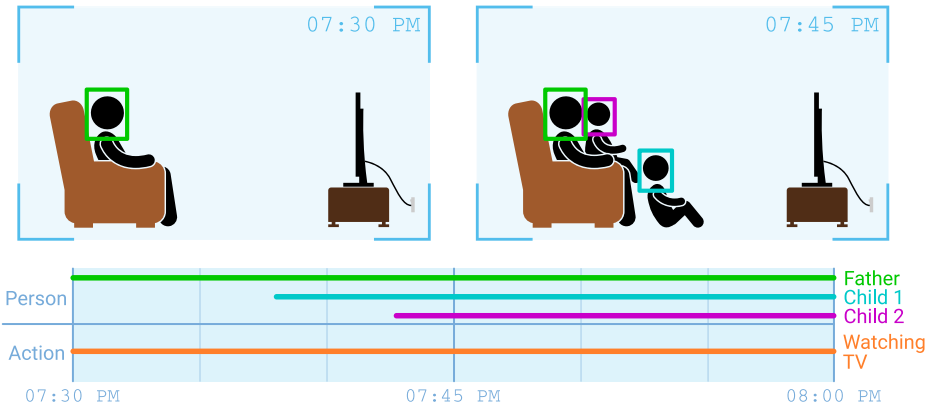


Fig. 2 An example of a meta-data timeline showing the identities and actions

and show results on our large real-life datasets. Finally, Section 7 draws conclusions and discusses future work.

2 Related work

This article comprises many elements and techniques that work together towards a global pipeline. We, therefore, split this related work section into several parts: the first focusing on related work that concerns the autonomous cameraman used to steer the PTZ camera during recording. The next subsections (action, sound, and facial recognition), discuss related work on several techniques capable of generating said footage meta-data during post-processing. This is followed by a subsection discussing techniques that summarise meta-data derived from recorded footage.

2.1 Autonomous camera men

PTZ cameras nowadays are commonly used in surveillance applications. Their degrees of freedom enable to change the viewing direction as well as the zoom factor, making it possible to zoom in to a specific detail of the scene.

In television and cinema, camera men mostly refrain from these cameras unless the format of the production requires it, e.g. for reality TV. Indeed, remotely operated PTZ cameras make the presence of human camera men in the house unnecessary, as they are often considered obtrusive and people tend to adjust their behaviour based on their presence. This is very important here, as opposed to normal productions, authentic reality TV lacks a script and the spontaneity removes any possibility to retake shots. By placing several PTZ cameras at strategical positions, we reduce the awareness of being recorded.

In surveillance applications, it suffices to steer the PTZ to capture a zone of interest [1, 46, 71]. A more related research field is techniques for automatic lecture recording, able to follow a single lecturer while complying with some cinematographic rules [28]. However, as the motion of most lecturers are constrained to a small region in front of the blackboard, this is far simpler than our target application. In our case, we are recording reality TV which often involves small household settings, with much larger motion variations, also along the

depth axis. Moreover, we often see partial occlusions of people, by e.g. dining tables and sofas. Additionally, we aim at taking shots with multiple people, opposed to only a single person.

Work of Callemein et al. [3] uses room information to first limit the number of recordings that are being captured. In addition to triggering recordings based on room activity, they compared both speed and accuracy of a Deformable Parts Model DPM [16] and Aggregate Channel Features ACF [13] model, trained for upper body detection. In addition to upper body detection, they use a frontal and left/right profile face trained cascade classifier models [60] to determine the gaze direction and head location. Both the head position and upper body area were used to propose a medium shot canvas and steer a PTZ. Even though they are capable of steering the PTZ, their models often are incapable of detecting partially occluded people.

Nowadays, Convolutional neural network CNN based techniques greatly outperform these techniques, both in accuracy and speed, showing better robustness when detecting partially occluded object. In this work we start from [3], using the proposed approach allowing for triggered recording, to reduce the collected footage. We also replace the cascade head and upper body detections models with a much more robust and much more informative single detection model, the OpenPose framework [4]. Instead of bounding boxes, this framework outputs frame-by-frame human poses represented by key points. These key points provide more precise spatial information, exempting us from using multiple detectors or estimating certain body positions based on the box coordinates. This allows us to determine more accurate medium shots, based on output from a single detector.

2.2 Action recognition

As we want a compact and uncluttered overview of what happened in the recorded reality TV footage, algorithms are needed to identify the actions visible in these videos. Since the past two decades, many video action recognition methods and pipelines have been proposed by the vision research community. Among the hand-engineered ones that could model effectively the appearance and motion representations across frames in videos are HOG3D [35], SIFT3D [48], HOF [36], and iDTs [63]. Several other techniques were proposed to model the temporal structure in an efficient way, such as the actom sequence model [19], temporal action decomposition [41], dynamic poselets [64] and ranking machines [17].

There are several approaches to end-to-end neural network-based action recognition [15, 32, 52, 57, 65] to exploit the appearance and the temporal information. These methods operate on 2D (individual image-level) [9, 14, 21, 53, 55, 65, 76] or 3D (video-clips or snippets of K frames) [15, 57–59]. The filters and pooling kernels for these architectures are 3D (x, y, time) i.e. 3D convolutions ($s \times s \times d$) [76] where d is the kernel's temporal depth and s is the kernel's spatial size. These 3D CNN are intuitively effective because such 3D convolution can be used to directly extract spatio-temporal features from raw videos. Carreira et al. proposed inception based 3D CNNs [30], which they referred to as I3D [6]. More recently, some works introduced a temporal transition layer that models variable temporal convolution kernel depths over shorter and longer temporal ranges, namely T3D [11]. Further in [10], Diba et al. propose spatio-temporal channel correlation that models correlations between channels of a 3D CNN, with both spatial and temporal dimensions. In contrast to these prior works, our work differs substantially in scope and technical approach. The HATNet [12] proposes an architecture that exploits both 2D CNN and 3D CNN to learn an effective spatio-temporal feature representation. Finally, it is worth noting the self-supervised CNN training works from unlabelled sources for action recognition: Fernando et

al. [18] and Mishra et al. [38] generate training data by shuffling the video frames; Sharma et al. [51] mine labels using a distance matrix based on similarity although for video face clustering; Wei et al. [69] predict the ordering task; Ng et al. [39] estimate optical flow while recognising actions. Self-supervised and unsupervised representation learning is beyond the scope of this paper.

2.3 Sound recognition

Research into sound recognition has recently started to attract a lot of attention as a result of the release of Audio Set [20], a large-scale collection of acoustic signals containing environmental audio events, as well as the inception of challenges involving this classification task such as DCASE 2016 [37], DCASE 2017 [61] and DCASE 2018 [44].

A critical issue in this research domain concerns labelling as the majority of the data used is weakly annotated: the onsets and durations of the sounds present in the auditory signals are usually unknown. In other words, typically only clip-level event information is provided and temporal details are omitted.

Current state-of-the-art approaches to sound recognition typically involve neural networks. For instance, convolutional recurrent architectures utilising gated linear units [26, 72, 73] have been very popular in prior work. Densely connected convolutional networks [7] have also been shown to be successful for this task, as well as capsule-based neural structures [31].

2.4 Facial recognition

In recent years, facial recognition systems have made significant progress in accuracy [42, 47, 54, 62]. This is mainly due to the rise of deep learning techniques that came along with an increase in computing power, along with the public availability of large face datasets like LFW [27], CASIA-Webface [74], VGGface2 [5] and MS-Celeb-1M [22]. Moreover, current face datasets contain faces in almost unconstrained settings. These conditions create opportunities for applying facial recognition in a true real-world context like ours.

One of the recent trends in facial recognition models, is the use of large margin, angular-based loss functions [8, 62, 66, 67] with typically a ResNet-like [25] backbone network. One particularly interesting loss function is the recently-published ArcFace loss [8], which has shown to surpass previous state-of-the-art methods [62].

2.5 Audiovisual summarisation

With the current trends of easy data storage and relatively cheap audiovisual recordings, it is easy to get overwhelmed by a large amount of data. This increased the research attention towards techniques that reduce the workload required to get through all this information. Video summarisation is a field that is related to our proposed method. The goal of video summarisation is to process a large amount of video data and produce a more condensed representation of the input, while retaining as much information as possible. Most of these condensed representations are comprised of video skims (short clips) [23, 40, 45, 49], but they can also be a series of representative static images [34]. Some proposed methods also allow for user input, in the form of queries [49, 50].

To find the important segments a variety of techniques are used, but usually the visual data is used. In [40], the authors propose extracting key frames from several videos using a weakly learned video saliency model. This saliency model, based on Histogram-of-Gradient

features, is trained to detect popular concepts. These keyframes and their neighbouring frames are than combined to create short clips using a probabilistic model that optimizes for desirable attributes such as camera shakiness.

It is important to realise that summarization is inherently rather subjective. Not everyone is looking for the same information, so methods that can adapt to user preferences are useful. In [50], the authors propose to extract high level and contextual video features from 5 second shots using a set of pretrained detectors [2]. These features are then fed into a neural network that has been conditioned using the user query to estimate the importance of the shots. The segments are processed sequentially to select the most important shots based on the query, and then combined into the final summary.

Of course, summarizations skims are not only summarisation technique. In our work, we automatically construct visual timelines to show the events that occur throughout the input data. Closest to this aspect of our method is the work of Xiong et al. [70]. They construct a story-line representation of a set of videos. This representation contains a timeline of actors, events, locations and objects, which can be used to respond to queries. In this work, we construct a timeline from meta-data produced from a set of chronological videos, there is no temporal overlap between these videos. In this way, it is different from our use case, in which there are several audiovisual streams being recorded simultaneously.

Tapaswi et al. [56] propose a timeline visualisation of which characters interact with each other. These interactions were determined by first segmenting the video in ”scenes” containing several shots, and then using person identification to determine which characters were sharing a scene together. This was tested on clearly structured data, i.e. TV series. They specifically focus on creating a timeline representation of the input video that is visually appealing.

The aforementioned timelines only show person interaction and our proposed method additionally shows event information. To our knowledge, this is the best fitting related work available.

3 Approach overview

Figure 3 shows an overview of the overall approach of the proposed system. The goals of our system are (i) to automate the recording of reality TV by replacing a team of human camera operators with PTZ cameras controlled by automatic camera men, and (ii) to drastically reduce the effort of a human film editor who has to select video material from the huge amount of recorded footage and compose it into an entertaining TV show. Our system combines and evaluates several state-of-the-art techniques that are able to cope with the demanding real-world circumstances in in-the-wild household scenes, with the additional challenge that we can not use any labelled (re)training data that is situation-specific.

The first part (described in Section 4) focuses on video captured by the overview camera used for online PTZ steering and recording triggering. Then, we worked out various offline post-processing techniques for the recorded audiovisual data, producing meta-data used to generate timelines that reduce the search space of the production house, as detailed in Section 5.

The block diagram in Fig. 3 illustrates the global system and each separate component, starting from both the assisting wide-angle overview camera and the PTZ camera that will be steered and captured.

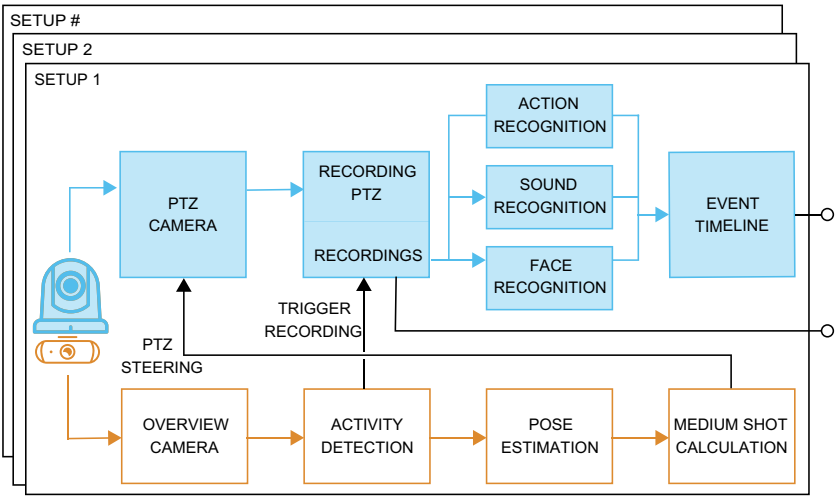


Fig. 3 Overview of our proposed system

We used the activity based triggering proposed by Callemein et al. [3] on the images from the overview camera, to switch on and off the PTZ cameras already reducing the amount of recorded footage (see Section 4.1). On the same images, we estimate the poses of the people present with OpenPose [4]. Based on these poses we calculate a medium shot, discussed in Section 4.2, to steer the PTZ camera accordingly.

From these PTZ recordings we generate meta-data output by performing action recognition (Section 5.2), sound recognition (Section 5.3) and person re-identification (Section 5.1). Based on the output of all these techniques we output several event-timelines, as discussed in Section 5.4. These timelines can be used by the production house to see at a glance what was happening during the recordings.

4 Automatic camera man

To maintain authenticity while capturing real-life reality TV, without having a script, the production house has chosen to place multiple PTZ HD cameras around the house. Our first goal is to operate these automatically, which comprises of two tasks: switching cameras on and off based on activity in the room, and steering the camera to a aesthetically pleasant shot by taking into account cinematographical rules.

We combined every PTZ camera with a wide-angle overview camera, which is mounted as close by as possible the PTZ. This simplifies the geometrical calibration between the two cameras down to an approximated linear relationship. The overview camera images are processed in real time, generating the controls for the high resolution PTZ camera.

4.1 Camera selection

The best guarantee to capture all events in a house, is allowing the installed cameras to record continuously 24/7. However, this would produce an enormous amount of footage, which would need an immense amount of disk storage capacity, taking into account the

broadcasting quality resolution of the video. Therefore, we used techniques from Callemein et al. [3] to decrease the amount of data.

In particular, their approach triggers the cameras on and off based on room-specific background subtraction activation. This approach requires some manual effort by first configuring which cameras are triggered together and the annotation of trigger regions on each overview camera.

The concept of *buckets* is used, holding a value that decreases over time. Detected foreground action on the overview cameras increase the value of the according *buckets*. When a bucket reaches a set threshold, a signal is sent to trigger the camera recording function.

This system has the unique property that *camera pre-roll* can be achieved: when e.g. a person approaches a door, the camera behind the door will be triggered to start recording, such that the cinematographically interesting event of the opening of the door and the appearance of the person in the room behind the door will be captured.

Another advantage is that this technique reduces the search space greatly for the editor afterwards and increases the speed by which the analysis methods, discussed in the next sections, are executed.

4.2 PTZ camera steering

As in live reality TV filming only spontaneous acting occur and re-acting a certain scene is not done, it is important that our system never misses any event outside the Field-of-View FOV of the cameras. By steering the PTZ cameras to always keep the people present inside the FOV we can prevent this issue. This can be achieved by autonomously steering the camera based on a person detector, as seen in [1, 3, 28].

Our goal is illustrated in Fig. 1. It is clear that the shot on the left is not very interesting: the person is located very near the edge of the frame and it is not clear what he is looking at. The goal of our system is to automatically steer the PTZ camera to the more aesthetically pleasing shot on the right.

The work of Callemein et al. [3] compared several upper body detectors in a similar household setup. In this paper, we want to build further on their work and improve the accuracy by implementing a real-time pose estimator instead of a DPM or ACF based upper body model. The accuracy and speed of state-of-the-art pose estimators [29, 68] on monocular images greatly outperforms the upper body models used in [3]. We, therefore, replaced the previous upper body detector with a state-of-the-art pose estimator available in the OpenPose framework [68].

The main goal is to take the closest aesthetically pleasing medium shot of the largest number of people visible in the overview image. A medium shot commonly consists of a close up of one or more people with the upper frame border just above the head (i.e. the *head room* rule) and the bottom of the torso near the lower border of the frame. To position the people in this manner, the cameraman usually uses the location of the eyes, the torso and the gaze direction. The latter is used to frame individual people slightly off-center, such that they have more canvas space in their viewing direction (i.e. the *nose room* rule). Both the location of the eyes and the torso can be calculated based on the body key points provided by the OpenPose framework. Only the gaze direction is not directly available but can be derived from the facial landmarks.

OpenPose provides five face key points: one for the nose, and two for the ears and eyes. By combining the distances between these points the gaze can be estimated. The face seen from a frontal perspective will show an equal distance difference from the eyes and ears to the nose. If the face is turned towards the left, the distance between the left ear and nose will

be smaller than the distance between the right-ear and nose. The same distance difference occurs between the nose and ears. With a simple vector subtraction, we can derive the gaze direction (left or right), which is sufficient to provide all the necessary information to take a medium shot.

As previously discussed, multiple PTZ cameras were installed on various strategic places in multiple rooms. Each of these cameras has been equipped with at least one supporting overview camera that will retain an overview of the room at all times, independent of the current orientation of the PTZ. Based on the poses derived from these overview images, we propose a single medium shot to be taken by the PTZ. This medium shot is calculated firstly based on the torso bounding box that needs to fall within this medium shot, with the eyes placed on the 1/3 y-axis to comply to the cinematographic *rule of thirds*. This rule, however, is not strict, and a certain deviation margin is allowed. For example, when two persons are nearby looking at the same direction, then the 1/3 y-axis will commonly be placed in between both eyes. The same allowed margin applies when placing people on the 1/3 and 2/3 x-axis (complying the rule of thirds), based on their gaze direction. When 2 people are facing the same direction and are relatively nearby each other, we can place this rule-of-thirds lines in between them. We, therefore, go over each possible canvas, minimising the size of the canvas containing as many people as possible, while minimising the margins from the rule of thirds directions. This proposed shot, comprising of a centre x, y position and scale, is afterwards translated to pan-tilt-zoom values using a calibrated lookup table and send to the PTZ to trigger repositioning.

Figure 4 illustrates four overview frames with a proposed medium shot canvas indicated by the red rectangle.

We further extend this algorithm by adding a time delay and allowing for inter-camera communication. This means that a certain shot has to be kept still for a minimum time duration before steering the PTZ, filtering out sudden movements and possible false positives. While inter-camera communication disables or enables autonomous steering, i.e. some cameras hold a static shot, while the only the remainder is allowed to move.

Fig. 4 Example of four overview frames with their proposed medium shot canvasses (red)

5 Generating rich meta-data timelines

After limiting made recordings based on a human activity triggers, as described in Section 4.1. Still a huge amount of footage is gathered each day due to our system that is being used in a real-life household. However, as the aim is to create entertaining reality TV shows of it, a movie editor needs to isolate some interesting story lines from these daily life videos.

As it is clearly very time consuming to go over all of the recorded footage, we propose in this paper a way to speed-up this process. We aim to automatically extract as much as key data as we can from the videos, and represent these in rich timelines as a handy tool for the editor. The following information is detected in the videos. First, we detect people and identify their identity by means of face recognition. Then we classify their actions from the video, and use the audio to recognise what is happening. As we know where the cameras are installed in the house, we can also link each of these detections with a room label and a time stamp. All extracted data is then represented in rich meta-data timelines, offering an overview of what happened in this household. The two formats used to generate the timelines are described in detail in Section 5.4.

5.1 Person identification

We use facial recognition to recover the identity of each person whose face is visible in a given frame. In most cases, a facial recognition system consists of three steps: face detection, face alignment and face classification. For face detection, we employ a pre-trained MTCNN [77] model. MTCNN also detects facial landmarks that are used to align the face. After the face alignment, we use a ResNet-101 [25] network trained with ArcFace [8] loss on the MS-Celeb-1M [22] dataset to extract a deep face embedding.

For each person of interest, there are 5 manually selected reference images with which query faces can be compared. With the embeddings of these reference images, we trained an SVM that classifies the queries. We use an RBF SVM implementation from the Scikit-learn Python library [43].

5.2 Action recognition

To localise different people activities in the videos, we use a recently proposed human action recognition method based on the 3D convolutional networks. The method uses a sequence of video frames as input, incorporating temporal information as well as appearance features. For the training, we need some video clips from the desired action categories to recognise. For action recognition in our scenario, we do not crop the person region but instead use the whole scene of frames to do the task more accurately with respect to the pre-trained model and training pipeline since the whole scene can help to recognise actions.

The action recognition module is trained by a deep 3D Convolutional Neural Network (3D-ResNet50)[24, 58] which is one of the state-of-the-art networks for action recognition. The model is pre-trained on the Kinetics-600 dataset [33] which has 600 different human action and activity categories. For our application, we have selected almost 70 categories of indoor actions from Kinetics which are likely to happen in our videos like cooking, going to bed, watching TV, etc. After the selection of those categories, the 3D CNN model was fine-tuned on the selected set and ready to do inference on the camera footage. The input of the model to the inference task is a short clip of 32 sequential frames ($\sim$ 2 seconds) from

the video and the output is a label for the related action happening in the corresponding clip or no label in case of no specific action.

An issue with the action recognition pipeline is the different angle of captured targets compared to the Kinetics videos. This makes the result less robust than results obtained on videos more similar in style to the Kinetics dataset. For instance in the cooking class the accuracy performance in our data is 4% less compared to the same category in Kinetics dataset.

5.3 Sound recognition

We chose to employ a convolutional recurrent neural network with gated linear units [72] to perform sound recognition. Notably, this type of architecture was deemed the winner of task 4 of the DCASE 2017 challenge [61].

We trained the aforementioned network utilising Audio Set [20], an extremely large weakly annotated multi-label collection consisting of over two million audio fragments (mostly 10 seconds long) covering a multitude of environmental events. In doing so, we did not make use of the full data set corresponding to the complete Audio Set ontology [20], which comprises over 500 classes. Instead, we first created a custom set of sound labels that could provide valuable information in the context of this project by manually picking and aggregating a limited amount of classes from the original hierarchy. Table 1 contains an overview of the combinations made to ultimately obtain 13 categories of relevant auditory events. Secondly, because of practical restrictions, we selected a limited amount of data samples (21039 in total) from Audio Set [20] containing at least one of these resulting sounds to optimise the considered model.

We use the trained neural architecture to detect relevant sounds in the audio captured by the PTZ cameras used in this project. We cut the collected auditory signals into non-overlapping pieces of 10 seconds and perform recognition on each of these frames separately. Doing so allows us to build timelines of auditory events.

Table 1 Combinations of Audio Set classes [20] used to obtain sound labels

Sound labels	Corresponding classes in Audio Set ontology [20]	
Dog	Dog	
Cat	Cat	
Door	Door	
Kitchen sounds	Dishes, pots, and pans	Cutlery, silverware
	Chopping (food)	Frying (food)
	Microwave oven	Blender
Water	Water tap, faucet	Sink (filling or washing)
Speech	Speech	
Shout	Shout	
Screaming	Screaming	
Laughter	Laughter	
Crying	Crying, sobbing	
Singing	Singing	
Kids playing	Children playing	Children shouting
Radio/TV	Radio	Television

Some of the employed cameras contain more than one microphone and therefore capture multiple audio signals, which can be only weakly correlated. For example, this is the case when the microphones are directed towards different parts of the recorded scene. To deal with this issue, recognition is executed on the different audio channels independently and the resulting predictions are aggregated in the following simple way: if a certain sound is detected in at least one of the channels, that type of auditory event is considered to be active in the recorded scene.

5.4 Generating timelines

To provide the user with information at a quick glance, we build visual timelines.

To construct these timelines, we do a pass over all of the "detections" found by our method. These include action, sound and face detections. To each action and sound detection, we assign a face detection, as an identity. This association is done by checking if an action/sound co-occurs with face detection, within the same room.

We generate timelines in two different formats. The first is a timeline containing all the detections associated with a single person along with their location throughout the day. Activities are displayed using colour coded bars, their location is visible on the y-axis. For examples, see Section 6.5. With this format, it's easy to quickly get an idea of what one person has been doing throughout the day.

The second format is a full overview of everyone seen throughout the sequence. This format includes everyone that was detected in the sequence and shows who did which activity at what time. Examples are available in Section 6.5.

We apply temporal filtering on these plots to remove spurious detections and provide a clearer overview. This requires consistent detection over the relevant event during a predefined time range. This range represents a trade-off between getting a clearer overview of the detections throughout the sequence and finding rare, and short events. This time range can be set by the user of the system, depending on what they are looking for.

6 Evaluation

This article combines several components as illustrated in Fig. 3. To evaluate this system we first evaluated all elements separately and then demonstrate the generated timelines for our real-life dataset.

We start by discussing the framework we designed to evaluate the automatic medium shots that were proposed in Section 6.1. We then continue in Section 6.2 that describes the dataset containing random shots procured by the automatic camera man system, followed by the evaluation of the three detectors in Sections 6.3 and 6.4 on this dataset to eventually output event timelines as described in Section 6.5.

6.1 Automatic camera man evaluation

A strict evaluation of our automatic camera man is not straightforward, because determining the aesthetical quality of the generated shots is challenging due to its subjective nature. In Section 4 we proposed an approach that uses OpenPose to propose an aesthetically pleasing medium shot around the people present. Yet there is no direct objective metric to determine how good the overall system works. To overcome this challenge, we created an online platform that enabled the production house to take part in a user study, where they evaluate

the proposed canvasses. They were informed to only limit their evaluation on the people present, not the context of the scene.

The online platform follows three stages. First (i), we gather information about the user. Then, we show an overview camera image with the proposed canvas and overview cut-out based on that canvas, followed by (ii) asking the user to score the aesthetical quality of the given images and (iii) asking the user to adjust the medium shot to a better one. Stage (ii) and (iii) are repeated 20 times, on randomly selected proposals. Stage (i) allows us to learn more about the experience and knowledge of the user concerning cinematographic shots.

The user is asked to self-evaluate his/her subject knowledge based on two criteria, from 0 (none at all) to 4 (highest):

Cinematographic knowledge How good do you think your overall cinematographic knowledge is?

Time spending How much time do you spend concerning photography/cinematography?

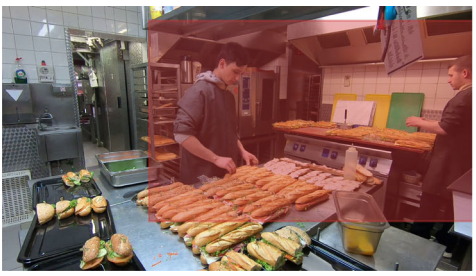
After gathering the user info, we continue our user study by repeating stage (ii) and stage (iii), showing 20 random medium shot proposals. Figure 5 illustrates an example of a medium shot proposal. Figure 5a illustrates the complete overview image of scene, while Fig. 5b illustrates the cut-out proposal as the PTZ would provide. We chose only to use the overview images to evaluate our process, without the actual PTZ steering, to make sure the proposals are consistent for each user, allowing for better comparison. The user is then asked to evaluate this initial proposal, with a score, 1 (very bad) to 5 (very good), for each of these three criteria:

Head room Is there enough or too much head room?

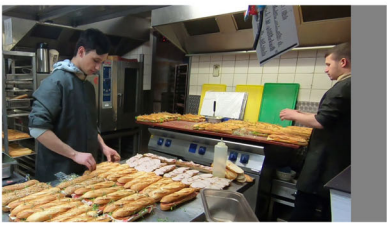
Rule of Thirds Does the shot comply to the rule of thirds?

Overall How does the overall medium shot score?

During this study, 14 members of the production house participated, evaluating 20 frames each. Table 2 summarises these user study results. When reviewing the evaluation output, we noticed our system sometimes produced an erroneous canvas. It turned out that some of the scenes contained wall posters, showing real-size photographs of people, which are indeed detected as real people by our automatic system. In contrast to our automatic system, the testers of the user study never chose canvases that also encompass these wall posters. To overcome this issue, these poster pictures could be removed from the room before using this system. An automatic alternative could be a person tracker to calculate the movement of certain people detections. Then, the system could ignoring detections with little movement over a long period of time, as they are probably not people. However, since only a limited



(a) overview image with the proposed canvas (red)



(b) cut-out of the proposed canvas

Fig. 5 Images that are shown during the second stage in the user study

Table 2 Scores derived from the automatic cameraman validation user study

Evaluation	Score	Cinematographic Knowledge					
		Full dataset			Filtered dataset		
		1	2	3	1	2	3
Head room	1	0.10	0.05	0.36	0.00	0.01	0.28
	2	0.25	0.19	0.24	0.25	0.19	0.26
	3	0.25	0.33	0.19	0.25	0.34	0.21
	4	0.40	0.34	0.17	0.50	0.34	0.22
	5	0.00	0.10	0.04	0.00	0.11	0.03
Rule of thirds	1	0.10	0.05	0.32	0.00	0.01	0.23
	2	0.10	0.13	0.19	0.06	0.09	0.20
	3	0.60	0.35	0.25	0.69	0.40	0.29
	4	0.20	0.41	0.21	0.25	0.43	0.25
	5	0.00	0.06	0.04	0.00	0.07	0.03
Overall	1	0.15	0.04	0.38	0.00	0.01	0.29
	2	0.30	0.09	0.23	0.38	0.06	0.24
	3	0.15	0.38	0.17	0.13	0.39	0.20
	4	0.40	0.40	0.20	0.50	0.43	0.25
	5	0.00	0.10	0.04	0.00	0.11	0.03
Average IoU		0.565	0.676	0.521	0.642	0.713	0.580

number of frames contained these posters, we chose to filtered them out of the dataset. The results of which are shown in Table 2 as *filtered dataset*.

Table 2 shows for both the full and filtered dataset for each criterion (Head room; Rule of Thirds and Overall) the score distribution normalised and grouped by the *Cinematographic Knowledge* user score, gathered in stage (i). There were no users with a score of 0 or 4 and therefore we only show the results for users with a score of 1 to 3.

In Table 2 the aesthetical evaluation scores show a tendency to be lower for users with a higher cinematographic knowledge score. Indeed, these results show that our system still does not receive high scores from users with high cinematographic knowledge scores, who are more strict on this than average maybe because of their education. Yet more than 80% of our canvasses are scored above average by the remaining users, showing that our system already proposes sufficient enough canvasses to meet their expectations.

In stage (iii) we asked the users to correct the proposed canvasses. By allowing the users to adjust the proposed canvas we can see towards what direction our system can improve. The bottom row in Table 2 shows the average Intersection of Union (IoU) between our proposed canvas and the adjusted canvasses.

These average IoU show similar behaviour as mentioned before, that the canvasses perform best according to the users with a cinematographic knowledge of 1 or 2, while the users with knowledge 3 score worst.

When reviewing the proposed adjustments by the users, it was clear that in some cases our system was a bit conservative, a human camera man should rather zoom in closer. We see that a human camera operator would in some cases also chose to only take a subset of the people into view, because e.g. they were too far apart. However, our approach aimed towards capturing all the people present, over capturing the closest canvas, decreasing the possibility of missing a key-scene greatly.

6.2 Event evaluation dataset

Section 4 described the approach used to first limit the number of recordings, based on the activity detection, while also steering the PTZ cameras to take medium shots. This approach has been evaluated in the previous Section 6.1 on real-world image data and showed to already achieve shots of sufficient quality. We, therefore, decided to work further on the real output of this algorithm and collect our own large scale reality TV dataset to further evaluate our event detection approach.

This dataset contains footage that was captured during 24 hours, containing three different household settings and families, in which each of them were 6 to 8 camera systems installed.

Each family has a different social focus. One of them is a reconstituted family with the main focus on the kids and the parents. In the second family, a couple has been trying to get pregnant and had in vitro fertilisation as a last resort and gave birth to twins. They are at home now, where their grandparents currently take residence while their new house is being renovated. The parents and the baby are the main focus, with the grandparents mostly in the background. The last family focuses on a senior that is still living on his own, despite being over 90 years old. His daughters still visit him frequently and are currently preparing for the wedding of one of them. The daily routine and habits of the senior are the main focus, he is also the only family member who spends a lot of time with his dog and cat.

All three families lead different lives and each family holds a different social focus on which afterwards the production team hopes to build a story around. We, therefore, let multiple people annotate 100 random segments for each family of 10 seconds. In order to limit the number of possible labels that could be used, a subset of the labels discussed in Sections 5.2 and 5.3 was selected, based on the domain knowledge.

In Table 3 an overview is given of all the labels and their instances in the 300 segments. Apart from these labels for action and sound recognition, the people present in the segment was also annotated. Even though a subset of labels was selected based on the domain knowledge, the nature of this case leads to a further decrease in annotated classes. As a result, we will mainly focus on the labels that occurred multiple times in the next sections.

6.3 Action and sound recognition evaluation

Our suggested approach to perform action recognition, described in Section 5.2, detects a label each 2 seconds. A global threshold of 0.01% on all classes ensures minimal confidence must be reached before detections are evaluated. Because our annotation segments' start frame, stop frame and duration differ from the segments' used during recognition, we aggregate all detections together, retaining the highest detection confidence. We then evaluate each annotation individually, to see if the label occurs in the aggregated set of detections. We follow a similar approach for evaluating the sound recognition labels, detected for segments of 10 seconds, aggregating each detection that falls within the segment time frame.

By determining the optimal threshold for each label, i.e. maximising Youden's index [75] to maximise both sensitivity and specificity, we can measure how good these detectors perform. Tables 4 and 5 shows the thresholds, Youden's index, specificity and sensitivity with criteria Youden's index > 0. The remainder of the labels had a Youden index < 0, with an optimal threshold set at 1.00. This can partially be explained by the lack of sufficient instances in the annotation set.

Table 3 Annotation labels based on domain knowledge

Sound labels	Instances	Action Labels	Instances
Dog	1	Baby waking up	0
Cat	0	Brushing Teeth	0
Door	6	Carrying baby	2
Kitchen Sounds	33	Celebrating	0
Water	1	Clapping	4
Speech	98	Cooking or preparing food	15
Shout	0	Crawling Baby	0
Screaming	10	Crying	0
Laughter	5	Cutting Watermelon	0
Crying	0	Doing Laundry	7
Singing	1	Dying Hair	0
Kids Playing	8	Eating or Drinking	28
Radio/TV	41	Grooming Dog	1
		High Jump	2
		Hugging	0
		Ironing	1
		Kissing	1
		Kitchen Activities	10
		Laughing	1
		Making Bed	1
		Petting Animal (not cat)	2
		Petting Cat	3
		Reading Book	3
		Reading Newspaper	0
		Setting Table	2
		Shaking hands	0
		Sitting at table	62
		Smoking	0
		Taking a Shower	0
		Training Dog	0
		Using remote controller (not gaming)	6
		Walking the dog	0
		Washing the dog	0
		Writing	2

In our annotation set, we see a large number of radio/TV instances, while in Table 5 we see that this class has a very low sensitivity. We examined the detection output and noticed that in most of these cases the TV had people talking, leading to speech as detector output.

Each segment can hold multiple labels for each separate recognition part. Our action recognition model, however, was trained on a public dataset with single-class classification output. Therefore, in cases where two labels are labelled, our output will only have a single action recognition label. On segments that hold similar contextual labels that often occurring

Table 4 Action recognition labels maximised by Youden's Index showing the optimal threshold, specificity and sensitivity

Label	Threshold	Youden's Index	Specificity	Sensitivity
cooking or preparing food	0.130	0.112	0.846	0.267
eating or drinking	0.160	0.505	0.934	0.571
kitchen activities	0.540	0.159	0.959	0.200
sitting at table	0.780	0.032	0.983	0.048

simultaneously, the model will only output the strongest classification label, leading to more false negatives for each remaining annotation label. In Table 4 we can see that *sitting at table* has a low sensitivity due to only being classified when no stronger action was present. When people were sitting at the table, while talking or writing the action will not be overshadowed by another.

In our case, the production house has less interest in the exact set of labels, as opposed to a single label that already allows filtering and contributes to an event timeline.

6.4 Person recognition evaluation

As explained in Section 5.1, we use a combination of face detection and face re-identification using a small gallery of each family member to recognise the people in the camera views. Due to the nature of the use case, the identification evaluation should be done on a segment-based level, instead of a frame-based level. Indeed, it is of no interest for the editors of the TV-show to know who was present in each frame. It suffices if editors know who was visible during certain ranges of time.

As such, our evaluation does not take the position of the detection into account, nor whether the person was truly visible in a particular frame. Rather, face classifications will be aggregated per 10-second video *segment*, as defined by the annotations in the dataset described in Section 6.2. We apply facial recognition on every tenth frame or 25 frames per segment. For each face classification, we multiply the detection confidence with the classification score. A person can only appear once in a frame, so when a frame has multiple classifications of the same identity, we keep the classification where this product of detection confidence and classification score is the highest. This gives a single score for each identity per frame. The scores are summed up for each frame in a segment and divided by the number of frames per segment, i.e. 25. This yields a final score between 0 and 1 for every identity in a segment.

Table 5 Sound recognition labels maximised by Youden's Index showing the optimal threshold, specificity and sensitivity

Label	Threshold	Youden's Index	Specificity	Sensitivity
Kitchen Kounds	0.67	0.220	0.947	0.273
Speech	0.78	0.530	0.683	0.847
Laughter	0.95	0.400	1.000	0.400
Radio/TV	0.76	0.118	0.996	0.122

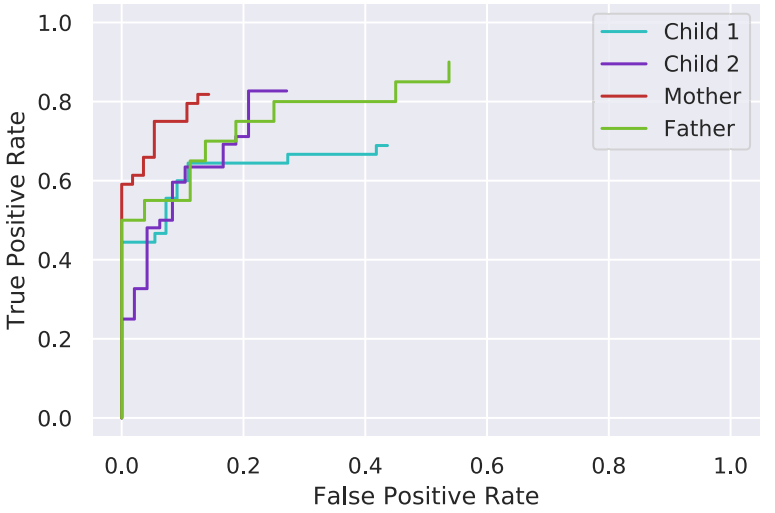


Fig. 6 ROC-curves for the family with the most annotations, using a segment-based evaluation

By comparing these scores with the segment-level annotations, we can create a ROC-curve for each person. Figure 6 shows these ROC-curves for the family with the most annotations, which we call "Family A".

Table 6 shows the optimal threshold for the labels in each family. The thresholds are optimised by maximising Youden's index. To frame the results, we added the column "N_{annot}", which shows how often a certain label was annotated over all segments.

Table 6 Face recognition results for the three families; the thresholds are determined by maximising Youden's index

	Threshold	Youden's index	Specificity	Sensitivity	N_{annot}
Child 1	0.024	0.535	0.891	0.644	45
Child 2	0.003	0.619	0.792	0.827	52
Mother	0.016	0.696	0.946	0.750	44
Father	0.035	0.562	0.862	0.700	20
Baby	0.041	0.695	0.895	0.800	5
Grandfather	0.023	0.467	0.667	0.800	25
Daughter 1	0.013	0.143	1.000	0.143	7
Daughter 2	0.009	0.376	0.964	0.412	17
Mother	0.105	0.547	0.947	0.600	5
Grandmother	0.035	0.531	0.937	0.595	37
Grandfather	0.017	0.482	0.947	0.535	43
Father	0.138	0.909	0.909	1.000	1

The column N_{annot} shows the number of annotations that were present for the respective label

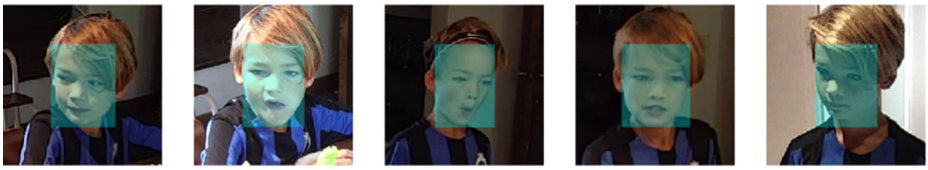


Fig. 7 References that were used for classifying *Child 1* from *Family 1*

It is clear that the state-of-the-art face recognition techniques are indeed suitable for the current use case. To find out where our approach encountered difficulties and where it worked well, we inspect the classification results for *Child 1* from *Family 1* (the first block in Table 6) when using the threshold of 0.024 from Table 6. Figure 7 shows the reference images that we manually selected for *Child 1*.

The best true positive segment—i.e. the segment with the highest aggregated score (= 0.52) for *Child 1* where *Child 1* was indeed present—is shown in Fig. 8 and is from a scene where *Child 1* sits still with his face directed towards the camera. The high score per frame and the face's visibility throughout the segment causes a high aggregated score.

Figures 9 and 10, on the other hand, show a frame from, respectively, the worst false positive segment (aggregated score of 0.08) and the worst false-negative segment for *Child 1* (aggregated score of 0.00). The false-positive detection shown in Fig. 9 is the classification with the highest confidence in that segment and can be attributed to physical similarities between mother and son, along with low resolution and poor lighting. The false-negative is shown in Fig. 9, which is understandable because this scene combines a lot of motion blur with bad lighting and a very challenging camera angle. Additionally, in that segment, *Child 1* is only visible for 10 of the 25 frames that we use for facial recognition.

6.5 Generated timelines

In this section, we show the timelines in the 2 formats we discussed in Section 5.4. We set the time range for the temporal filter to 6 seconds, which approximately corresponds to the system requiring 2 consecutively and consistent action detections.

Fig. 8 Frame showing a true positive with the highest aggregated score of 0.52 for *Child 1* (blue box)

Fig. 9 Frame showing a false positive with the highest aggregated score of 0.08 for *Child 1* (blue box)

We show the individual timelines generated for two of the families in Figs 11 and 12. These contain the locations and the activities of the family member throughout the sequences. We also show an example of the second timeline format in Fig. 13. This format contains the different activities happening throughout the day, along with the person associated with this activity. This representation also contains the activities to which no identity has been assigned.

7 Conclusion

In this article, we proposed a system that reduces the manual labour required to record a reality TV show. Our proposed system consists of two major components.

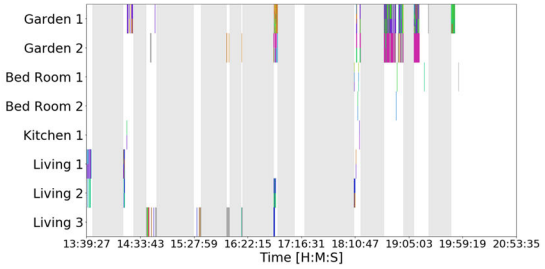
The first component steers PTZ cameras, guided by people detection on a supporting overview camera, aiming to compose aesthetically pleasing medium shots. Our user study

Fig. 10 Frame showing a false negative with the lowest aggregated score of 0.00 for *Child 1*

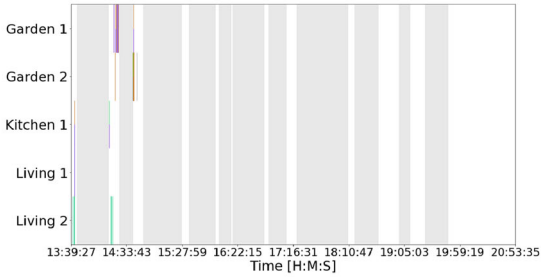
Fig. 11 Timelines showing the activities of the people in the "Family A" sequences

showed that we are capable of proposing canvasses that meet the level of a moderate camera man. Although the system does not reach a professional level, its quality of capturing is acceptable for the commissioning production house to use during moments with low levels

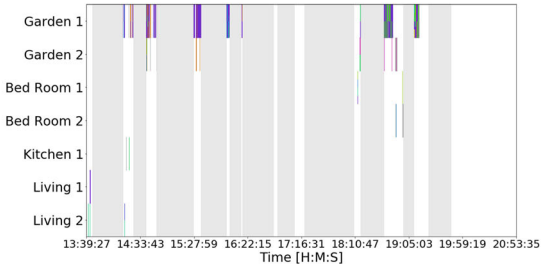
(a) Baby



(b) Grandfather



(c) Daughter 1



(d) Daughter 2

Fig. 12 Timelines showing the activities of the people in the "Family B" sequences

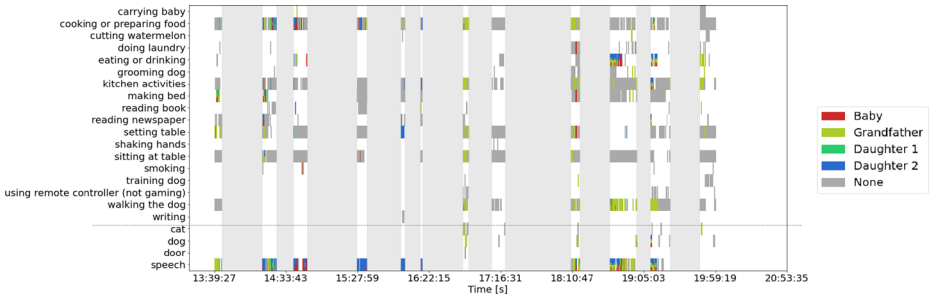


Fig. 13 Full overview of the activities of "Family B". The detections above the gray line are detected actions, underneath are recognised sounds

of activity and only a limited number of people (it is understandable that, at key moments of the storyline, they opted to remotely operate the PTZ cameras manually, in order not to miss any important events of the reality TV story). Currently, we only focused on capturing medium shots of the highest number of people. In future work, experiments on different shots with different criteria could increase the usability of our proposed system further.

Our second component combines the output of several audiovisual content analysis techniques, using footage that was recorded based on our proposed medium shots. While there is room for improvement on the individual event detection components, the resulting timelines reduce the search space for the production house when editing the footage. Nevertheless the challenging nature of our real-life large scale dataset, we have proven in a series of evaluation experiments that each component yields acceptable results and that the generated timelines are informative for a film editor.

References

1. Al-Hadrusi MS, Sarhan NJ, Davani SG (2016) A clustering approach for controlling ptz cameras in automated video surveillance. In: 2016 IEEE International symposium on multimedia (ISM), IEEE, pp 333–336
2. Borth D, Chen T, Ji R, Chang SF (2013) SentiBank: large-scale ontology and classifiers for detecting sentiment and emotions in visual content. In: Proceedings of the 21st ACM international conference on Multimedia - MM '13. ACM Press, Barcelona, pp 459–460. https://doi.org/10.1145/2502081.2502268. http://dl.acm.org/citation.cfm?doid=2502081.2502268
3. Callemein T, Van Ranst W, Goedemé T (2017) The autonomous hidden camera crew. In: Machine vision applications (MVA), 2017 fifteenth IAPR international conference on, IEEE, pp 47–50
4. Cao Z, Simon T, Wei SE, Sheikh Y (2017) Realtime multi-person 2d pose estimation using part affinity fields. In: CVPR
5. Cao Q, Shen L, Xie W, Parkhi OM, Zisserman A (2018) Vggface2: a dataset for recognising faces across pose and age. In: 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018), IEEE, pp 67–74

6. Carreira J, Zisserman A (2017) Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR
7. Choi I, Bae SH, Kim NS (2019) Deep convolutional neural network with structured prediction for weakly supervised audio event detection. Appl Sci 9(11):2302
8. Deng J, Guo J, Xue N, Zafeiriou S (2019) Arcface: additive angular margin loss for deep face recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4690–4699
9. Diba A, Sharma V, Van Gool L (2017) Deep temporal linear encoding networks. In: CVPR
10. Diba A, Fayyaz M, Sharma V, Arzani MM, Yousefzadeh R, Gall J, Van Gool L (2018) Spatio-temporal channel correlation networks for action classification. In: ECCV
11. Diba A, Fayyaz M, Sharma V, Karami AH, Arzani MM, Yousefzadeh R, Van Gool L (2018) Temporal 3d convnets using temporal transition layer. In: CVPR Workshops
12. Diba A, Fayyaz M, Sharma V, Paluri M, Gall J, Stiefelhagen R, Van Gool L (2019) Holistic large scale video understanding arXiv
13. Dollár P, Appel R, Belongie S, Perona P (2014) Fast feature pyramids for object detection. IEEE transactions on pattern analysis and machine intelligence 36(8):1532–1545
14. Donahue J, Anne Hendricks L, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, Darrell T (2015) Long-term recurrent convolutional networks for visual recognition and description. In: CVPR
15. Feichtenhofer C, Pinz A, Zisserman A (2016) Convolutional two-stream network fusion for video action recognition. In: CVPR
16. Felzenszwalb PF, Girshick RB, McAllester D (2010) Cascade object detection with deformable part models. In: 2010 IEEE computer society conference on computer vision and pattern recognition, IEEE, pp 2241–2248
17. Fernando B, Gavves E, Oramas JM, Ghodrati A, Tuytelaars T (2015) Modeling video evolution for action recognition. In: CVPR
18. Fernando B, Bilen H, Gavves E, Gould S (2017) Self-supervised video representation learning with odd-one-out networks. In: CVPR
19. Gaidon A, Harchaoui Z, Schmid C (2013) Temporal localization of actions with actoms PAMI
20. Gemmeke JF, Ellis DPW, Freedman D, Jansen A, Lawrence W, Moore RC, Plakal M, Ritter M (2017) Audio set: An ontology and human-labeled dataset for audio events. In: Proceedings IEEE ICASSP, New Orleans, LA, p 2017
21. Girdhar R, Ramanan D, Gupta A, Sivic J, Russell B (2017) Actionvlad: learning spatio-temporal aggregation for action classification. In: CVPR
22. Guo Y, Zhang L, Hu Y, He X, Gao J (2016) Ms-celeb-1m: challenge of recognizing one million celebrities in the real world. Electron Imaging 2016(11):1–6
23. Gygli M, Grabner H, Van Gool L (2015) Video summarization by learning submodular mixtures of objectives. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, pp 3090–3098. https://doi.org/10.1109/CVPR.2015.7298928. http://ieeexplore.ieee.org/document/7298928/
24. Hara K, Kataoka H, Satoh Y (2017) Learning spatio-temporal features with 3d residual networks for action recognition. In: ICCV
25. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: CVPR
26. Hou Y, Kong Q, Wang J, Li S (2018) Polyphonic audio tagging with sequentially labelled data using CRNN with learnable gated linear units. In: Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018), pp 78–81
27. Huang GB, Mattar M, Berg T, Learned-Miller E (2008) Labeled faces in the wild: a database forstudying face recognition in unconstrained environments. In: Workshop on faces in'real-life'images: detection, alignment, and recognition
28. Hulens D, Goedemé T., Rumes T (2014) Autonomous lecture recording with a ptz camera while complying with cinematographic rules. In: 2014 Canadian conference on computer and robot vision, IEEE, pp 371–377
29. Insafutdinov E, Pishchulin L, Andres B, Andriluka M, Schiele B (2016) Deepercut: a deeper, stronger, and faster multi-person pose estimation model. In: European conference on computer vision. Springer, New York, pp 34–50
30. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML
31. Iqbal T, Xu Y, Kong Q, Wang W (2018) Capsule routing for sound event detection. In: 2018 26Th european signal processing conference (EUSIPCO), IEEE, pp 2255–2259
32. Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-Fei L (2014) Large-scale video classification with convolutional neural networks. In: CVPR

33. Kay W, Carreira J, Simonyan K, Zhang B, Hillier C, Vijayanarasimhan S, Viola F, Green T, Back T, Natsev P, Suleyman M, Zisserman A (2017) The kinetics human action video dataset. arXiv:1705.06950
34. Khosla A, Hamid R, Lin CJ, Sundaresan N (2013) Large-scale video summarization using web-image priors. In: 2013 IEEE conference on computer vision and pattern recognition. IEEE, USA, pp 2698–2705. https://doi.org/10.1109/CVPR.2013.348. http://ieeexplore.ieee.org/document/6619192/
35. Klaser A, Marszałek M., Schmid C (2008) A spatio-temporal descriptor based on 3d-gradients. In: BMVC
36. Laptev I, Marszalek M, Schmid C, Rozenfeld B (2008) Learning realistic human actions from movies. In: CVPR
37. Mesaros A, Heittola T, Benetos E, Foster P, Lagrange M, Virtanen T, Plumbley MD (2018) detection and classification of acoustic scenes and events: outcome of the DCASE 2016 challenge. IEEE/ACM Trans Audio Speech Language Process (TASLP) 26(2):379–393
38. Misra I, Zitnick CL, Hebert M (2016) Shuffle and learn: unsupervised learning using temporal order verification. In: ECCV
39. Ng JYH, Choi J, Neumann J, Davis LS (2018) Actionflownet: learning motion representation for action recognition. In: WACV
40. Nie L, Hong R, Zhang L, Xia Y, Tao D, Sebe N (2015) Perceptual attributes optimization for multivideo summarization. IEEE Trans Cybern 46(12):2991–3003
41. Niebles JC, Chen CW, Fei-Fei L (2010) Modeling temporal structure of decomposable motion segments for activity classification. In: ECCV
42. Parkhi OM, Vedaldi A, Zisserman A, et al. (2015) Deep face recognition. In: Bmvc, vol 1, p 6
43. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. J Mach Learn Res 12:2825–2830
44. Plumbley MD, Kroos C, Bello JP, Richard G, Ellis DP, Mesaros A (2018) Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE 2018)
45. Potapov D, Douze M, Harchaoui Z, Schmid C (2014) Category-specific video summarization. In: European conference on computer vision. Springer, New York, pp 540–555
46. Rameau F, Demonceaux C, Sidibé D, Fofi D (2014) Control of a ptz camera in a hybrid vision system. In: 2014 International conference on computer vision theory and applications (VISAPP), IEEE, vol 3, pp 397–405
47. Schroff F, Kalenichenko D, Philbin J (2015) Facenet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 815–823
48. Scovanner P, Ali S, Shah M (2007) A 3-dimensional sift descriptor and its application to action recognition. In: ACM MM
49. Sharghi A, Gong B, Shah M (2016) Query-focused extractive video summarization. In: European conference on computer vision. Springer, New York, pp 3–19
50. Sharghi A, Laurel JS, Gong B (2017) Query-focused video summarization: Dataset, evaluation, and a memory network based approach. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4788–4797
51. Sharma V, Tapaswi M, Sarfraz MS, Stiefelhagen R (2019) Self-supervised learning of face representations for video face clustering. In: International conference on automatic face and gesture recognition
52. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. In: NIPS
53. Sun L, Jia K, Yeung DY, Shi BE (2015) Human action recognition using factorized spatio-temporal convolutional networks. In: ICCV
54. Taigman Y, Yang M, Ranzato M, Wolf L (2014) Deepface: closing the gap to human-level performance in face verification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1701–1708
55. Tang P, Wang X, Shi B, Bai X, Liu W, Tu Z (2016) Deep fishernet for object classification. arXiv:1608.00182
56. Tapaswi M, Bauml M, Stiefelhagen R (2014) Storygraphs: visualizing character interactions as a timeline. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 827–834
57. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3d convolutional networks. In: ICCV
58. Tran D, Ray J, Shou Z, Chang SF, Paluri M (2017) Convnet architecture search for spatiotemporal feature learning. arXiv:1708.05038

59. Tran D, Wang H, Torresani L, Ray J, LeCun Y, Paluri M (2018) A closer look at spatiotemporal convolutions for action recognition. In: CVPR
60. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001, IEEE, vol 1, pp I–I
61. Virtanen T, Mesaros A, Heittola T, Diment A, Vincent E, Benetos E, Elizalde BM (2017) Proceedings of the detection and classification of acoustic scenes and events 2017 workshop (DCASE 2017)
62. Wang M, Deng W (2018) Deep face recognition:, a survey. arXiv:1804.06655
63. Wang H, Schmid C (2013) Action recognition with improved trajectories. In: ICCV
64. Wang L, Qiao Y, Tang X (2014) Video action detection with relational dynamic-poselets. In: ECCV
65. Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Van Gool L (2016) Temporal segment networks: towards good practices for deep action recognition. In: ECCV
66. Wang F, Cheng J, Liu W, Liu H (2018) Additive margin softmax for face verification. IEEE Signal Process Lett 25(7):926–930
67. Wang H, Wang Y, Zhou Z, Ji X, Gong D, Zhou J, Li Z, Liu W (2018) Cosface: large margin cosine loss for deep face recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5265–5274
68. Wei SE, Ramakrishna V, Kanade T, Sheikh Y (2016) Convolutional pose machines. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4724–4732
69. Wei D, Lim J, Zisserman A, Freeman WT (2018) Learning and using the arrow of time. In: CVPR
70. Xiong B, Kim G, Sigal L (2015) Storyline representation of egocentric videos with an applications to story-based search. In: 2015 IEEE International Conference on Computer Vision (ICCV),IEEE, pp 4525–4533. https://doi.org/10.1109/ICCV.2015.514. http://ieeexplore.ieee.org/document/7410871/
71. Xu Y, Song D (2010) Systems and algorithms for autonomous and scalable crowd surveillance using robotic ptz cameras assisted by a wide-angle camera. Auton Robot 29(1):53–66
72. Xu Y, Kong Q, Wang W, Plumbley MD (2018) Large-scale weakly supervised audio classification using gated convolutional neural network. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE, pp 121–125
73. Yan J, Song Y, Guo W, Dai LR, McLoughlin I, Chen L (2019) A region based attention method for weakly supervised sound event detection and classification. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE, pp 755–759
74. Yi D, Lei Z, Liao S, Li SZ (2014) Learning face representation from scratch. arXiv:1411.7923
75. Youden WJ (1950) Index for rating diagnostic tests. Cancer 3(1):32–35
76. Yue-Hei Ng J, Hausknecht M, Vijayanarasimhan S, Vinyals O, Monga R, Toderici G (2015) Beyond short snippets: Deep networks for video classification. In: CVPR
77. Zhang K, Zhang Z, Li Z, Qiao Y (2016) Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process Lett 23(10):1499–1503. https://doi.org/10.1109/LSP.2016.2603342