Watching Together but Apart

Shared Video Experiences during the COVID19 Outbreak

Jannik Bryld
IT University of Copenhagen, Copenhagen, Denmark, janb@itu.dk

Lucas Schønrock
IT University of Copenhagen, Copenhagen, Denmark, lucs@itu.dk

Anders Løvlie
IT University of Copenhagen, Copenhagen, Denmark, asun@itu.dk

Louise Barkhuus
IT University of Copenhagen, Copenhagen, Denmark, barkhuus@itu.dk

ABSTRACT

Watching television together in the living room is a joyful and social activity that most people can relate to. As a result of the ongoing COVID19 outbreak, many friends and family members are separated due to national lockdowns and quarantine. In this demo we show how a web application can support real-time socially based video-content by facilitating a virtual living room for friends and family where they can watch synchronized video content while interacting. Our prototype uses readily available web-technologies, allowing users to get started in just one click by logging in with their existing YouTube account. Data from the YouTube API are then fed into novel recommender algorithms that then suggest videos based on the user's collective interests.

Info available at https://beta.letswatch.net

CCS CONCEPTS

• Human-centered computing • Collaborative and social computing • Collaborative and social computing systems and tools

KEYWORDS

Remote video; interactive tv; YouTube; video content.

1 Introduction

Throughout the 20th century television was often discussed as an isolated, anti-social activity by research and popular media alike; however other scholars have also pointed to the profound sociability that video media contributes to:

"TV and other mass media, rarely mentioned as vital forces in the construction or maintenance of interpersonal relations, can now be seen to play central roles in the methods which families and other social units employ to interact normatively." [2, p. 1]

Television can actually cultivate multiple forms of sociability: Direct (for example, when chatting with friends or family during a movie night at home) or indirect (for example when discussing the latest Game of Thrones episode at work or school), both being important to social life in the modern world. Both these types of sociability are being challenged by the recent and still developing outbreak of COVID19. At the time of writing, most of Europe and The United States are in national lockdown. While some countries including Spain and Italy have banned all outside activity except grocery shopping, most others are strongly recommending that their citizens stay at home and minimize social

contact. This means that friends and family members are being separated with very limited ways of socializing, and even fewer ways that resemble everyday activities. In this demo, we present the possibility for allowing people to experience a living-room-like social experience by enabling personal interaction through audio and video chat while watching synchronized video content from YouTube, effectively creating a virtual living room.

2 Related Work

Synchronizing video-playback between users to create a shared video experience is not a novel concept. Services developed for research such as STV3 [4], CollaboraTV [3] and AmigoTV [1] were already prototyped and tested more than a decade ago. Most of these services, however, required installation of third-party software and required users to go through a somewhat lengthy onboarding process. A current commercial service, Watch2gether, comes closest to the prototype demo described here, and works by synchronizing video-content from YouTube between users. Watch2Gether (www.watch2gether.com) requires no prior setup, but requires that users share a unique link with each other to ensure that a connection can be established between them. Another current service is the airtime app for smart phones (www.airtime.com), however, this has the limitations of smartphone size and bandwith. None of the services investigated allows users to personalize their experience, and none of them use existing viewing history or other kind of data about users to recommend video content. In this demo we use readily available web-technologies to create an instantaneous onboarding process, but one that simultaneously allows users' existing viewing habits to be adopted using open API's to enable the development of recommender systems.

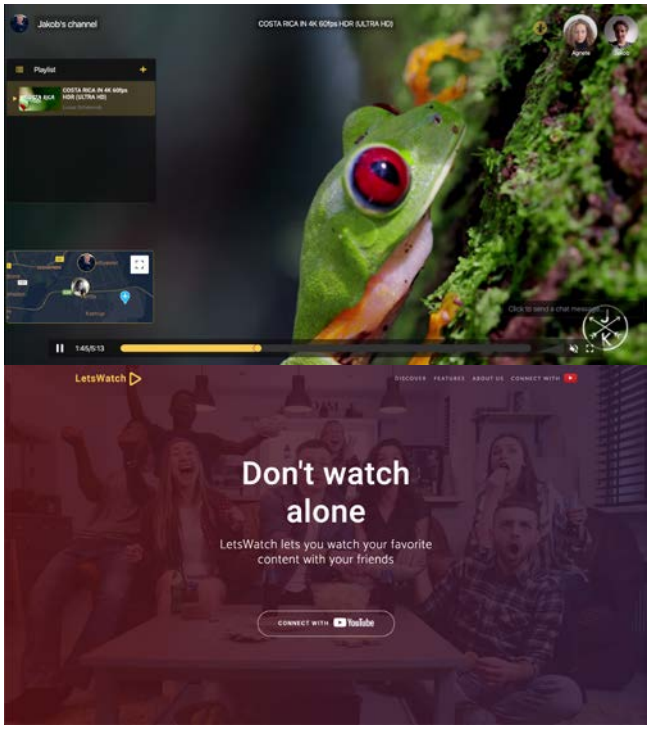

Figure 1: The frontpage and channel page of the prototype.

3 Prototype

We present the prototype LetsWatch, which is a web-based prototype that enables real-time social through distributed video viewing. It consists of two primary pages (see fig 1-2). The prototype allows users to watch and socialize about videos from Youtube. Upon logging to the prototype using a Youtube account, the user's persona and viewing habits are adopted. This enables recommender algorithms to automatically suggest content that suits the group.

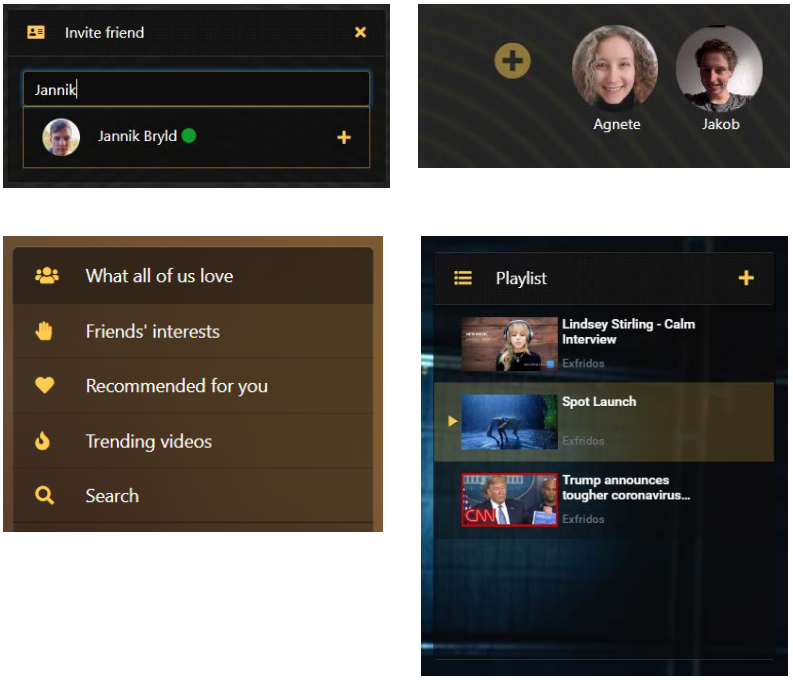

Figure 2: Invitation system (top left): Other active users can be located and invited to the channel by searching for their name. Webcam and audio (top right): Two users using the webcam and voice functionality to communicate together naturally while watching video. Video selection (lower left): Videos can be selected by browsing the different categories. Co-curation playlist (lover right): Videos are added to the queue of the playlist and can then be selected or removed from the playlist.

The software ecosystem of LetsWatch is composed of an online webpage, accessible on any common browser, and a synchronization server, which handles and appropriately relays video interactions and messages between users who are watching together.

3.1 The LetsWatch Webpage

As mentioned in the related works, much existing work includes less easily accessible applications, and thus LetsWatch explores this design opportunity to make the experience more accessible. Therefore, LetsWatch is made as a web application, making it accessible through a simple query of any common browser. Users are not required to register a new account but are rather required to log in using their Youtube account.

All logged-in users have a personal *channel*; a unique URL to their own virtual room. Any other active users can be invited directly from the channel page through an invitation system (Fig 2, top left). When users participate in the same channel, they may choose to broadcast their webcam feed or microphone input to all other participants of a room (Fig 2, top right). This is done efficiently by establishing a direct peer-to-peer connection between users, such that the webcam and audio streams are streamed directly, rather than through a central server, to other users. Any user may choose to disable the webcam or audio of other users. As an alternative way of communication, users can also use the text chat to send text messages to the other users.

All logged in users have a personal *channel*; a unique URL to their own virtual room. Any other active users can be invited directly from the channel page through an invitation system (Fig 2 top left). When users participate in a common channel they may choose to broadcast their webcam feed or microphone input to all other participants of a room (Fig 3). This is done efficiently by establishing a direct peer-to-peer connection between users, such that the webcam and audio streams are streamed directly, rather than through a central server, to other users. If others choose so, they may disable the webcam or audio of other users. As an alternative way of communication, users can also use the text chat to send text messages to the other users.

A video to collectively watch can be selected directly in any channel itself. The Youtube API is utilized to query data about the users' subscriptions to display different categories of videos (Fig 2,

lower left). Two of these categories, “What all of us love” and “Friends’ interests”, directly encourages users to explore videos that participating users of the same channel have in common or only others have shown interest in, respectively. A recommender algorithm is thus employed to cross-reference Youtube subscribers of users to accordingly recommend a subset of videos from the selected subscribers.

When a user selects a video to watch, it will be added to a common *playlist* (Fig 2, lower right) of the channel. Users of a channel can thus co-curate the playlist collaboratively. When any user clicks a video in the playlist, it will be played synchronized at all users’ display. When a video ends, a selection of 4 relevant videos will be displayed on the screen which when clicked is added to the playlist.

3.2 The Synchronization Server

The synchronization server is tasked with real-time capturing of the state of all channels and ensuring the synchronicity of video, including relaying text messages between users of a channel. When a user enters a channel, they will establish a WebSocket connection to the synchronization server and retrieve the current watched video, timestamp and which other users are present.

All users can equally control the video playback, analogous to every user possessing the remote control. When playing, pausing or seeking in a video, it will thus be synchronized between all users and happened as if they clicked themselves. It was an explicit design choice we made, due to providing the power to all users, rather than having a ‘host’ situation where one person is in charge (analogous to Zoom meetings or real-world scenarios with only one remote available). It obviously only works with a smaller number of people watching together, and in settings of trust, but that was also the initial use case for the tool.

The synchronization server relays control messages between all users in order to establish the direct peer-to-peer connection for video and voice, which thus are entirely separated from the service after being established.

4 Conclusion

In the time of physical distancing due to the COVID19 pandemic, socializing around video content with friends and family from a distance is more relevant than ever. This demo shows an easy to access service where friends can synchronize their viewing, and communicate through text, voice and video. Recommendation algorithms help users find content they have in common and suggest videos based on past viewings. Due to the simplicity of the setup, it is novel compared to past and current services.

REFERENCES

1. Baillie, L., Frohlich, P. and Schatz, R., 2007, June. Exploring social TV. In *2007 29th International Conference on Information Technology Interfaces* (pp. 215-220). IEEE.

2. Ducheneaut, N., Moore, R.J., Oehlberg, L., Thornton, J.D. and Nickell, E., 2008. Social TV: Designing for distributed, sociable television viewing. *Intl. Journal of Human–Computer Interaction*, *24*(2), pp.136-154.

3. Nathan, M., Harrison, C., Yarosh, S., Terveen, L., Stead, L. and Amento, B., 2008, October. CollaboraTV: making television viewing social again. In *Proceedings of the 1st international conference on Designing interactive user experiences for TV and video* (pp. 85-94).

4. Tullio, J., Harboe, G. and Massey, N., 2008, July. Investigating the use of voice and text chat in a social television system. In *European Conference on Interactive Television* (pp. 163-167). Springer, Berlin, Heidelberg.