Streamer.Space: A Toolkit for Prototyping Context-Aware Mobile Video Streaming Apps

Rui Pan
School of Interactive Arts and Technology
Simon Fraser University
250 – 13450 102nd Avenue
Surrey, BC, Canada, V3T 0A3
ruip@sfu.ca

Carman Neustaedter
School of Interactive Arts and Technology
Simon Fraser University
250 – 13450 102nd Avenue
Surrey, BC, Canada, V3T 0A3
carman@sfu.ca

CHI'17 Extended Abstracts, May 06-11, 2017, Denver, CO, USA
ACM 978-1-4503-4656-6/17/05.
http://dx.doi.org/10.1145/3027063.3053083

Abstract
Mobile video communication helps people connect with friends and families over distance. However, it is challenging to use existing commercial video chat systems (e.g., Skype, FaceTime) to prototype new types of futuristic experiences. Existing video communication tools are limited in granting designers access to augment video frames and control them based on contextual information. Creating video communication systems from scratch can be time intensive even with new APIs (e.g., WebRTC). We describe the design and rationale of *Streamer.Space*, a toolkit for prototyping context-aware mobile video streaming apps. With the toolkit, users can quickly and easily create mobile video experiences for research exploration.

Author Keywords
Mobile video communications; context-aware applications; video chat

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous;

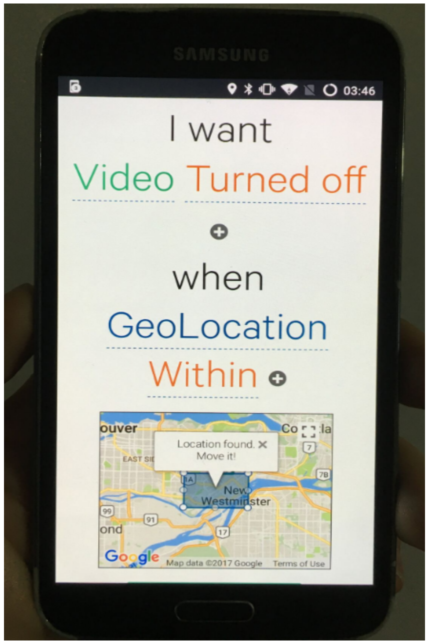


Figure 1. A simple "if-this-then that" logic is provided in Streamer.Space for creating rules for video streaming apps.

Introduction

The combination of Computer Mediated Communication (CMC) tools and smartphones provides a means for family and friends to communicate over distance via video chat [2,14]. However, existing video chat systems such as Skype, FaceTime and Google Hangouts are designed to heavily focus on 'talking heads' scenarios where people leave their face in front of the camera for a conversation [7,12]. This idea is now quite limiting as people desire to use video chat systems in a range of new usage scenarios. These include, for example, doing sports over distance, attending major life events remotely, share outdoor activities and even playing geolocation games [2,7,14,15].

What is needed are new systems to support a range of experiences where users require different capabilities, including differing means to turn on/off the video stream, new mechanisms for privacy control, and different features for staying aware of one's availability and desire to stay within a video call. Being contextual-aware could help video communication better fit these new usage scenarios [2,7,8]. For example, applications might be able to alter video feeds according to the type of activity occurring, a user's location, or even biometric conditions of the user. However, the challenge is that it's not easy to adapt existing systems to support such design requirements as commercial video chat systems do not grant designers or users access to contextual information and it is difficult to customize how video frames appear [2,7]. Different types of applications may also require different types of contextual information. This makes it challenging to explore a variety of design solutions as part of prototyping efforts. New APIs (e.g., WebRTC) make it easy to create basic video communication systems, yet they do not easily allow the integration of contextual information. Novice programmers also face learning barriers for new languages and libraries [6].

To address these problems, we describe our design and rationale of Streamer.Space, a web-based toolkit for prototyping context-aware mobile video streaming apps. Users can create their own apps through a simple "if-this-then-that" logic to set streaming rules based on contextual information (Figure 1), as rule-based and event-based systems are considered beneficial for non-experienced programmers [13]. We expect Streamer.Space could be a user-friendly platform for HCI researchers and designers to explore new prototype systems aimed at investigating ideas around awareness and privacy in emerging scenarios for video communication. It may also be valuable for end users to create personalized video streaming apps depending on their situation and goals.

Related Work

Many new usage scenarios for video communication systems have emerged while most existing commercial systems are still designed to focus on face-to-face chatting [12]. For example, a study [17] on live streaming apps found a large amount of live streaming involved indoor and outdoor activity sharing. Various situations such as cooking, playing sports, crafting and showing scenery are shared when streaming [17]. In domestic life, Long Distance Relationships (LDRs) sometimes 'live together' over distance by leaving a video chat system running, yet couples would value being able to flexibly switch audio/video channels based on their dwelling situations [8]. The study of Experiences2Go [3] revealed that it is important to give

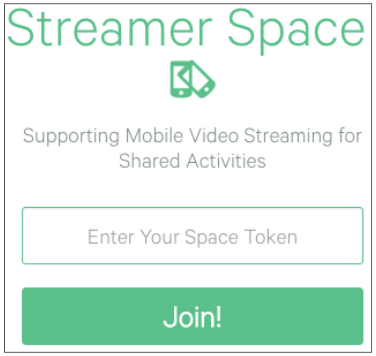


Figure 2. The welcome page of Streamer.Space. Each application is identified by unique Space token.

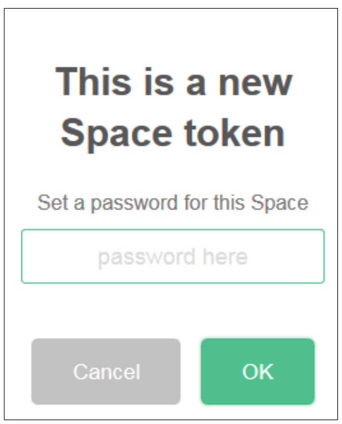


Figure 3. When creating a new app, users need to assign a password. They can share Space token and password to video communicate with each other.

users control over how the video stream appears when sharing sports activities between parents and kids. In-home video prototypes [1,11] raised new concerns of privacy and awareness in always-on video communications. All of these examples point to the value of coupling contextual information with the capabilities found in video streaming systems.

Some system designs have begun enabling contextual awareness in video communications, however, they are limited in numbers. For example, Family Portals [5] and Peek-A-Boo [9] helped users balance privacy and awareness in home situations by using video blinds. A system [16] designed for Portal game identified users' behaviors for wearable cameras based on location information. Of course, "one size doesn't fit all" and designing and developing new applications for different scenarios requires expertise and time. What is needed is a way to quickly and easily create new video streaming experiences as a means to try out new design ideas and iterate on them. Thus, a toolkit for quickly prototyping new video streaming apps would be valuable for HCI designers and researchers.

Design Goals

We first created a series of design goals based on our own design and development experiences, as well as our reading of the related literature. Below we list the three main design goals that we focused on supporting when creating our toolkit.

1. Enabling contextual information

Existing systems do not provide users/designers with the ability to control video streams through contextual information, yet this could turn out to be valuable. Thus, our first goal for the toolkit was to incorporate contextual information and allow it to regulate video streaming within an application. For example, we wanted to support the acquisition of data from mobile device sensors so applications could adjust video and audio streaming, if needed.

2. Simplicity

As many users may not have experienced development skills, we wanted a friendly user interface that incorporated easy-to-use logic. We felt this would be necessary for designers to be able to quickly adapt to the toolkit and understand how the contextual information might work with the video streams.

3. Support for Mobile Devices

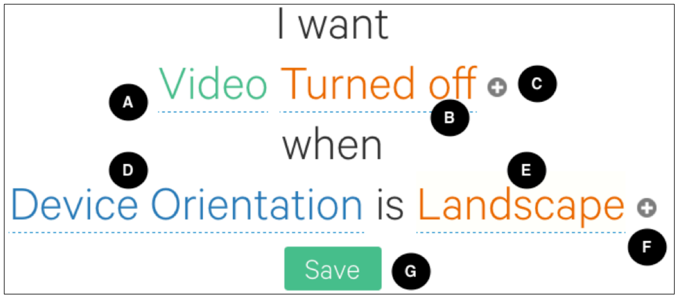
An increasing amount of video streaming usage now takes place on mobile devices [4,17]. Thus, we felt the toolkit should support video streaming from mobile devices, including phones and tablets.

The Design of Streamer.Space

Streamer.Space runs on web browsers that can be accessed on mobile and desktop devices. It allows users to share video and audio streams between devices. First, we provide two imagined usage scenarios that illustrate how the toolkit can be used to create two different types of mobile video streaming applications. Following this, we describe the specific design features.

'Bicycle with Me': Sharing Bicycling Over Distance

Brian is an enthusiastic bicyclist. He wanted to share the beautiful scenes via video link with his kids while he is bicycling in a new city. He only wanted to stream views from his phone mounted on the bicycle when he stops at some famous sight spots because the view of riding on the road is not stable and less interesting to



ure 4. The interface for customizing
eaming rules. Users could click or touch
dashed text to configure a trigger-
ion rule for the app.

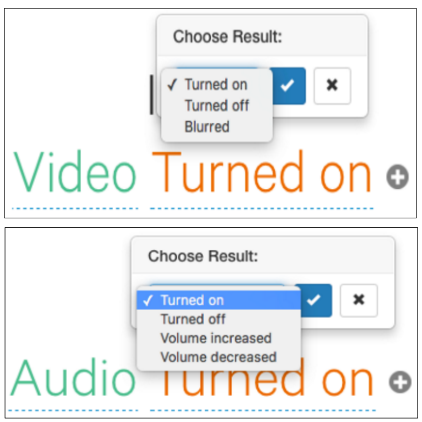


Figure 5. The options for controlling video and audio stream.

his kids. He used Streamer.Space to create a new Space called 'Bicycle with Me' and shared the token and password with his kids. He set the rule as, "I want video turned on when moving speed is less than 2 meters per second." Thus, the video stream paused while he was riding the bicycle. When Brian stopped at a famous church, panning the bicycle front slowly, the stream resumed so his kids could see the view of the church.

'I Know What to See': Privacy in Video Tour

Ann works for a high-tech company and her client wanted to have a virtual tour of the company remotely because they live in another continent. Due to the confidentiality of a new product the company was about to launch, there were several buildings restricted for visitors. Ann created a new app using Streamer.Space and set the rule as "I want video blurred when geolocation within [restricted area picked in Google Maps]." She put her mobile phone in her shirt-pocket with her camera facing outwards and guided the client in an always-on mode. The video stream automatically blurred in the restricted areas.

Design Features

We used the word *Space* to represent newly created spaces/connections. Figure 2 shows the welcome page of the toolkit. Users are required to type a Space token to continue. If it is a new Space token, they will be asked to assign a password for the new Space created (Figure 3). If a token is already registered, users need a correct password to access. Users can share the token and the password to other people for them to use the app. After users create a new Space, they will be directed to a rules customization page. Similar to a popular trigger-action programming platform called IFTTT [18,19], we added a simple "if this then that" user interface. Users can customize their application by setting rules. Figure 4 shows an example customization page for rules with parts labeled from A to G. We describe each part next.

A. Stream control source

We added individual control of video and audio streams. A study [14] pointed out that separating video and audio channels can be important when sharing activities outdoors. Users can set rules for video and audio individually.

B. Stream control options

Figure 5 shows the options for video and audio control. Users can choose to turn the selected stream on or off, or blur it. We added blurring as an option in video control because we wanted users to have some control over their privacy if streaming video. A blurred video stream could provide a general view of what is happening while not revealing complete details.

C. Stream control appending button

Users can click a "plus" button to append a more complex syntax such as "Video turned on AND audio turned off". By default, audio is on or off as the same as the video.

D. Contextual information source

This field shows the sources of contextual information. At present, we have added Device Orientation, Geolocation, Moving Speed, Time, and Lightness in our design. The selection of contextual information is described in the next section of the paper.

E. Contextual information options
This field shows the values for the selected contextual information source. For example, a map within Google Maps will appear when users define a rule for 'Geolocation'.

F. Contextual information appending button
Similar to element C, users can click this "plus" button to append a more complex syntax for contextual information.

G. Save button
Users can click this button to save the rule they have created. A unique JavaScript configuration file defining all of the created rules is generated automatically.

When a Space has been created, other users can use the Space token and password to join in the same app with others (akin to joining a chatroom). Then a FaceTime style video chat interface is enabled. Users will be asked to choose either their front or rear camera for video capture.

We selected WebRTC [20] (a framework for real-time communications in the browser) as the backbone framework for the toolkit because it is open-source and cross-platform. Advanced users could utilize the JavaScript configuration file to further modify the app by editing the code. We also added the functionality of logging and monitoring which tracks usage data including connection duration and the timing of contextual information. A backend database is setup for storing the configuration of each Space and its usage data. We imagine this could be helpful when collecting quantitative data for research purposes.

Selection of Contextual Information

We want to move beyond existing video streaming systems like Skype to provide context-enabled functions for controlling video stream. The types of contextual information we selected for Streamer.Space is based on the related literature that focuses on video conferencing applications. We included six options for users to customize: (1) device orientation, (2) acceleration, (3) moving speed, (4) lightness, (5) geolocation, and (6) time.

Device orientation is one of the basic characteristics of a device's mechanics when video streaming [4]. Users change their device's orientation in different conditions [4] and, with Streamer.Space, can have video or audio stream differently depending on the orientation of their device. Many smartphones have a built-in accelerometer that measures the device's motion. The **acceleration** of a device usually changes along with the movement of the user. Enabling the access to acceleration in Streamer.Space could be important for detecting the initiation or ending of a movement. Again, streaming options could be adjusted based on **movement speeds**. We imagine acceleration and speed could be important when users are sharing sports such as running, walking and bicycling. **Light** is an important representative of context [8]. It impacts the physiological behavior of users when video chatting. For example, LDRs care about the light in an intimate video chatting context [8]. It can also affect the fidelity of video stream dramatically. **Geolocation** are the important factor for outdoor contexts [10,14]. Users can customize video streaming apps for outdoor activities based on where a person is or how fast they are moving. **Time** is another fundamental context factor. For example, one may want to start streaming when it is an off-work time or blur video frames during meeting times.

Discussion and Conclusions

This paper describes our design of a toolkit called Streamer.Space for prototyping context-aware mobile video streaming apps. The goal is to help designers, researchers, and even end-users create video communication apps for sharing activities where contextual information might be used to control the video and audio streams. The toolkit provides a simple user interface requiring little, if any, programming background. Thus, it could even be used by researchers and designers without any programming experience. We imagine Streamer.Space could be a useful tool for helping researchers to explore mobile video communications in broad settings including both indoor and outdoor activities. Researchers could use Streamer. Space to explore which contextual information would be valuable in a particular context by conducting comparative studies. It could also be used as a research instrument to collect data for analyzing users' interest and characteristics in activity sharing via video communications. Both will help researchers and designers extend video communication systems to new usage scenarios.

While beneficial, Streamer.Space still has some limitations. Because the toolkit is running in web browsers, it is difficult to get information from external devices (head mounted cameras, step counters, etc.). Future work could investigate how to integrate more sources of contextual information into the toolkit. At present, Streamer.Space sets the same rules for both local and remote sides. It would be valuable to consider how video appears on different sides of the connection and use contextual information to adjust the video and audio streams correspondingly.

We plan to conduct user studies of new applications created with Streamer.Space for promising video communication scenarios.

Acknowledgments

This research was funded by the Natural Sciences and Engineering Research Council of Canada.

References

1. Uddipana Baishya and Carman Neustaedter. 2017. In Your Eyes: Anytime, Anywhere Video and Audio Streaming for Couples. *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, ACM, 84–97.

2. Jed R. Brubaker, Gina Venolia, and John C. Tang. 2012. Focusing on shared experiences: moving beyond the camera in video communication. *Proceedings of the Designing Interactive Systems Conference*, ACM, 96–105.

3. Kori Inkpen, Brett Taylor, Sasa Junuzovic, John Tang, and Gina Venolia. 2013. Experiences2Go: Sharing Kids' Activities Outside the Home with Remote Family Members. *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, ACM, 1329–1340.

4. Brennan Jones, Anna Witcraft, Scott Bateman, Carman Neustaedter, and Anthony Tang. 2015. Mechanics of Camera Work in Mobile Video Collaboration. *Proceedings of the 33rd Annual ACM*

Conference on Human Factors in Computing Systems, ACM, 957–966.

5. Tejinder K. Judge, Carman Neustaedter, Steve Harrison, and Andrew Blose. 2011. Family Portals: Connecting Families Through a Multifamily Media Space. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 1205–1214.

6. Andrew J. Ko, B. A. Myers, and H. H. Aung. 2004. Six Learning Barriers in End-User Programming Systems. *2004 IEEE Symposium on Visual Languages - Human Centric Computing*, 199–206.

7. Michael Massimi and Carman Neustaedter. 2014. Moving from Talking Heads to Newlyweds: Exploring Video Chat Use During Major Life Events. *Proceedings of the 2014 Conference on Designing Interactive Systems*, ACM, 43–52.

8. Carman Neustaedter and Saul Greenberg. 2012. Intimacy in Long-distance Relationships over Video Chat. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 753–762.

9. Carman Neustaedter and Tejinder K. Judge. 2010. Peek-A-Boo: The Design of a Mobile Family Media Space. *Proceedings of the 12th ACM International Conference Adjunct Papers on Ubiquitous Computing - Adjunct*, ACM, 449–450.

10. Shannon O'Brien and Florian "Floyd" Mueller. 2007. Jogging the Distance. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 523–526.

11. Erick Oduor and Carman Neustaedter. 2014. The family room: a multi-camera, multi-display family media space. ACM Press, 289–292.

12. Erick Oduor, Carman Neustaedter, Gina Venolia, and Tejinder Judge. 2013. The Future of Personal Video Communication: Moving Beyond Talking Heads to Shared Experiences. *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, ACM, 3247–3250.

13. John Pane and Brad Myers. 2000. The Influence of the Psychology of Programming on a Language Design: Project Status Report. *Institute for Software Research*.

14. Jason Procyk, Carman Neustaedter, Carolyn Pang, Anthony Tang, and Tejinder K. Judge. 2014. Exploring Video Streaming in Public Settings: Shared Geocaching over Distance Using Mobile Video Chat. *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, ACM, 2163–2172.

15. Samarth Singhal and Carman Neustaedter. 2017. BeWithMe: An Immersive Telepresence System for Distance Separated Couples. *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, ACM, 307–310.

16. T. Starner, B. Schiele, and A. Pentland. 1998. Visual contextual awareness in wearable

computing. *Digest of Papers. Second International Symposium on Wearable Computers (Cat. No.98EX215)*, 50–57.

17. John C. Tang, Gina Venolia, and Kori M. Inkpen. 2016. Meerkat and Periscope: I Stream, You Stream, Apps Stream for Live Streams. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ACM, 4770–4780.

18. Blase Ur, Melwyn Pak Yong Ho, Stephen Brawner, et al. 2016. Trigger-Action Programming in the Wild: An Analysis of 200,000 IFTTT Recipes. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ACM, 3227–3231.

19. IFTTT - Connect The Apps You Love. Retrieved from https://ifttt.com/.

20. WebRTC Home | WebRTC. Retrieved from https://webrtc.org/.