Snapstream: Snapshot-based Interaction in Live Streaming for Visual Art

Saelyne Yang[1] **Changyoon Lee**[1] **Hijung Valentina Shin**[2] **Juho Kim**[1]

[1]KAIST, Daejeon, Republic of Korea
[2]Adobe Research, USA
{saelyne, cyoon47, juhokim}@kaist.ac.kr vshin@adobe.com

ABSTRACT

Live streaming visual art such as drawing or using design software is gaining popularity. An important aspect of live streams is the direct and real-time communication between streamers and viewers. However, currently available text-based interaction limits the expressiveness of viewers as well as streamers, especially when they refer to specific moments or objects in the stream. To investigate the feasibility of using snapshots of streamed content as a way to enhance streamer-viewer interaction, we introduce Snapstream, a system that allows users to take snapshots of the live stream, annotate them, and share the annotated snapshots in the chat. Streamers can also verbally reference a specific snapshot during streaming to respond to viewers' questions or comments. Results from live deployments show that participants communicate more expressively and clearly with increased engagement using Snapstream. Participants used snapshots to reference part of the artwork, give suggestions on it, make fun images or memes, and log intermediate milestones. Our findings suggest that visual interaction enables richer experiences in live streaming.

Author Keywords

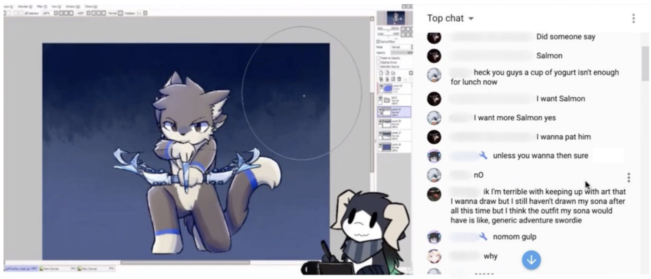
Live streaming; context sharing; online interaction; chat interaction

CCS Concepts

•**Human-centered computing** → **Interactive systems and tools;**

INTRODUCTION

As live streaming gains popularity, diverse contents from online game playing to live lectures are being streamed [4, 9, 15]. Among these, visual art is a rapidly growing genre, in which streamers create novel artifacts such as painting, drawing, design, and craft. Many artists share their creative process or stream how-to tutorials on specific design tools or techniques on platforms such as YouTube, Twitch, and Behance[2]. For instance, Twitch has a separate Art category for creative streams, which has on average 350 currently-live streams and 6,500 currently-live viewers at any moment [11]. Behance and Picarto[3] are platforms dedicated to visual art, where Behance hosts professional artists and Picarto allows novices to stream their artwork.



Did someone say

Salmon

heck you guys a cup of yogurt isn't enough for lunch now

I want Salmon

I want more Salmon yes

I wanna pat him

unless you wanna then sure

nO

ik I'm terrible with keeping up with art that I wanna draw but I still haven't drawn my sona after all this time but I think the outfit my sona would have is like, generic adventure swordie

nomom gulp

why

Figure 1. An example of visual art live streaming: The player shows the visual art the streamer is working on. Viewers interact with the streamer and other viewers via text-based chat.[1]

A key benefit of live streams that distinguishes them from recorded videos is the live interaction between the streamer and the viewers. Live interaction during the stream drives engagement and entertainment [18, 22, 24, 34], and provides unique opportunities for real-time learning [9, 11]. However, text chat, which is the most common form of communication in live streaming, is limiting for several reasons. First, both viewers and streamers struggle to make visual or temporal references, such as pointing out a specific visual content in the stream or a specific time, e.g., *"the person you drew 10 minutes ago"*. This is especially important for visual art streams where the main focus is the creation of visual content. In addition, for streams with many live viewers participating in the chat, questions or important comments pertaining to the stream can get buried in the flow of the text messages [11, 16, 25]. For these reasons, previous research suggests that live streaming can benefit from supporting richer interactions [22, 24], such as by providing separate channels for different types of chat [11] or having a dashboard of real-time user engagement metrics [34].

To improve the interaction in visual art live streaming, we first conducted a series of interviews with streamers and viewers, and observed live stream sessions to understand the current practices and challenges. We found that a recurring theme in

[1] source of video screenshot: youtu.be/9MmnZV3mmak

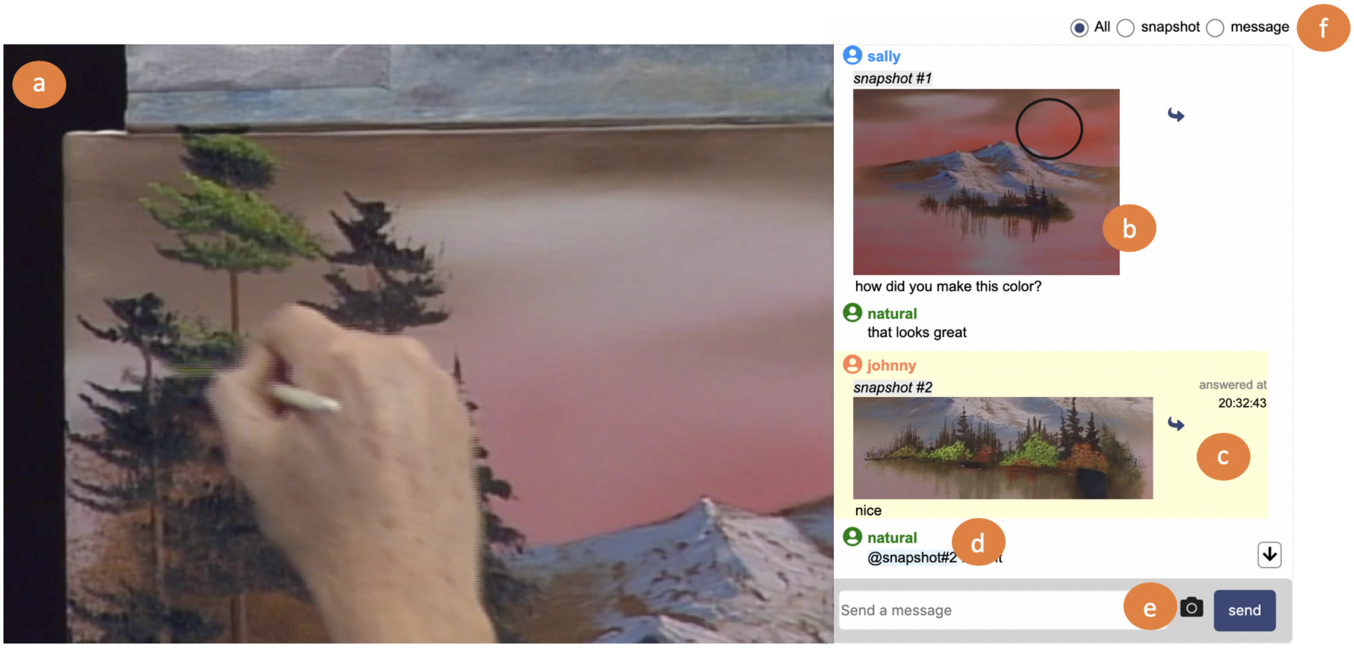
[2] youtube.com/live, twitch.tv, behance.net/live

[3] picarto.tv

CHI'20, April 25–30, 2020, Honolulu, HI, USA

ISBN 978-1-4503-6708-0/20/04. . . $15.00

DOI: https://doi.org/10.1145/3313831.3376390



sally
snapshot #1
how did you make this color?

natural
that looks great

johnny
snapshot #2
answered at
20:32:43
nice

natural
@snapshot#2 [illegible]

Figure 2. Overview of Snapstream: (a) The player. (b) Users can share a snapshot in chat. (c) Streamers can refer to and answer a specific snapshot using voice. When the streamer mentions certain snapshot by voice, it will be highlighted for all users. (d) Users can mention a specific snapshot. (e) Users can take a snapshot and (f) filter the chat.

these live streams is the need to refer to certain parts of the artwork or a specific action that the streamer performed in order to ask a question, give feedback, or make compliments. Viewers find it difficult to describe these intentions accurately using text only, and streamers often misunderstand the context, especially when a reference is made to previous parts of the stream that are no longer visible.

To alleviate these challenges, we present Snapstream (Figure 2), a snapshot-based interaction tool for visual art live streaming. Snapstream allows viewers to capture a screenshot of the current stream, which we call a ‘snapshot’. Viewers can ask questions or give feedback by making annotations on the snapshot. Streamers can reply to a specific snapshot with voice by saying *“snapshot number X”*, and direct the system to show the snapshot to all viewers. Both streamers and viewers can filter messages to see only text messages or snapshots.

We evaluate our tool in two real live streaming sessions, in which 2 streamers and 23 viewers used Snapstream. Results show that participants communicated more expressively and clearly with increased engagement using Snapstream. Viewers used snapshots in Snapstream for the expected purposes of referring to a specific part, and other creative purposes such as giving suggestions, making funny images, and logging intermediate drawings. Snapshots helped viewers to send clearer messages and to stay in sync with the streamer. Streamers also used snapshots as personal references.

The primary contributions of this paper are as follows:

- Insights into the current practices and unique challenges of referencing and communication in visual art live streaming.
- Snapstream, a snapshot-based interaction tool to improve interaction and communication between the streamer and viewers in visual art live streaming.
- Results from an in-the-wild deployment that show how users use snapshots in various ways to enrich interaction in live streaming.

RELATED WORK

As this work introduces a new interaction method in visual art live streaming by enabling reference to specific parts of streams, we review related work on improving interactions in live streaming and referencing techniques in multimedia.

Interaction in Live Streaming

Real-time interaction in live streaming is one of the key factors that motivate viewers to join the live streaming. To establish better viewer-streamer interaction, researchers have explored various forms of interaction techniques in live streaming. Weisz et al. introduce how integrating text chat into synchronous video watching can lead to a more engaging experience [33]. With the foundation of chat-based interaction, currently available tools explore the usage of Danmaku comments that overlay the video player [36]. Researchers have explored how Danmaku comments increase the interaction in live streaming [39].

In addition to the text-based interaction, researchers have further explored ways to improve viewer-streamer interaction. In game playing live streams, various attempts that encourage viewers to participate more actively in the game such as allowing them to participate as one of the characters [13] or suggest moves on the screen are introduced [21]. Researchers also explored how multimedia tools such as text, audio, image, and sketch could be used in online live learning environments [4, 17]. Twitch has third-party extensions such as Twip [32] and Toonation [30], which allow viewers to send audio and video messages that can be played by the streamer and shared to all users, and read out the message sent with the donation using Text-To-Speech (TTS).

There are other interaction techniques introduced to improve the live streaming experiences such as summarizing knowledge sharing live streams in real-time [23] and controlling the number of messages to prevent viewers from being overwhelmed [25]. Extending the previous research, our system aims to provide beyond-text interaction with snapshots to capture visual context for easy reference and communication in visual art live streaming.

Referencing Techniques in Multimedia

Previous work studied the referencing problem in the context of chats, specifically, the difficulty of sharing context and establishing common ground with other participants. Researchers have introduced a wide range of applications ranging from anchored conversations on a document [6, 26] to referencing complicated visuals such as 3D models [31, 14] and multimodal referencing methods beyond visuals such as voice, gesture, and hovering [31, 37, 19]. Video referencing techniques have also been investigated extensively. For example, studies have explored object [3], temporal [5, 35, 20], and spatio-temporal references [7, 12] in a collaborative setting. Video referencing techniques have been used in online learning [7, 10] and for video review and feedback systems [28, 27]. While these techniques explored various methods of referencing external text and multimedia content, we focus specifically on referencing in the context of live streaming. Live streaming has unique characteristics such as being volatile and non-archival, and being a one-to-many, two-way communication channel between the streamer and viewers. We focus on the particular set of challenges and opportunities that this unique setting brings, and attempt to understand and improve both viewers' and streamers' referencing experience.

FORMATIVE STUDY

To understand the current practices and challenges of streamer-viewer interaction in live streams, we conducted a series of semi-structured interviews with streamers and viewers. We recruited 5 streamers from Twitch and Behance, by contacting them via email and through the Discord platform[4]. Each streamer had a different expert domain that spanned pixel art (S1), anime-style art (S2), Adobe XD designs (S3), Photoshop and XD (S4), and Photoshop and Illustrator (S5) (Table 1). All streamers streamed 2-4 times per week. We also recruited 4 live stream viewers from our academic institution and the Discord channel of Twitch and Behance streams. The viewers' experience of watching live streams spanned drawing on Twitch (V1), painting on YouTube (V2), knitting on Instagram (V3), and Adobe Photoshop and XD tutorials on Behance (V4) (Table 2).

We also observed 6 real live stream sessions from Twitch and 5 sessions from Behance by participating as viewers. We took screenshots of the important moments of communication and artwork processes in the stream, and wrote down our observations of the situation. Twitch streams featured artists casually performing their work, while Behance streams focused on providing instructional tutorials about Adobe design software.

Key Challenges of Communication in Live Streams

From these interviews and observations, we identified several key challenges that streamers and viewers face during their communication.

Viewers often make references to parts of the artwork or certain actions performed by the streamer, but streamers often misunderstand them.

Viewers often refer to certain actions performed by the streamer or parts of the artwork the streamer is working on (Table 3). This happens both in the context of asking questions or giving feedback. For example, viewers ask questions about the techniques or tools that the streamer is using (e.g., *"How do I find the brush that XX is using?"*). Sometimes viewers ask streamers to re-do an action for clarification (e.g., *"Could you show me how you made the changes again?"*). They also ask streamers about the artwork (e.g., *"Why did you move the letters one by one instead of moving them together?"* or *"Why the last one with a head?"*). This type of referencing happens also when viewers are giving feedback (e.g., *"I like the basketball pitch picture"* or *"The arrow is still pointing right!"*).

However, viewers struggle to express the exact action or part of the artwork they are referring to via text. For example, V2 commented that *"It is hard to describe the stitch in knitting when I ask for re-do in the chat."* Viewers' incomplete descriptions of the parts make streamers struggle to understand the meaning of these references. S4 mentioned that he often needs to ask the viewer to clarify their question. For example, if a viewer asked *"How did you get that color?"*, he would ask back *"what color?"* He also said that since there are many functionalities and UI elements in design software, it is difficult to understand what the viewer is referring to. S3 mentioned that some questions and answers are not communicated clearly when a viewer is not a native English speaker.

The time delay between streamers and viewers exacerbate the communication problem.

Along with the above problem, there is usually a non-negligible time delay between the streamer and the viewers. Most live streams have a broadcast latency, which means that there is a delay between when the streamer performs an action and when the viewer sees it on the stream [38]. There is also a latency in the chat messaging itself, and since streamers are switching between working on their art and looking at the chat, there is an additional time delay between when the viewer writes a message and when streamer notices it. In visual art live streaming, the artwork is constantly undergoing change, so by the time the streamer notices the viewer's message refer-

[4] discordapp.com

Streamer	Content	Platform	Streamed for	# of viewers
S1	Pixel Art	Twitch	1 year	25
S2	Illustration	Twitch	2 years	100
S3	XD	Facebook, Twitter, Behance (YouTube)	2 years	550-2200
S4	Photoshop, XD	Facebook, Twitter, Behance (YouTube)	3 years	600-700
S5	Photoshop, InDesign, Illustrator	Facebook, Twitter, Behance (YouTube)	3 years	75-500

Table 1. Demographic information of streamers we interviewed for the formative study

Viewer	Content	Platform	Duration	Frequency
V1	Drawing	Twitch	a few hours	Daily
V2	Painting	YouTube	30 min	3-4 times / week
V3	Knitting	Instagram	30 min	Weekly
V4	Photoshop, XD tutorial	Behance	30 min	3-4 times / week

Table 2. Demographic information of viewers we interviewed for the formative study

Intent	Reference Type	Example
Question	Streamer's Action	*How do I find the brush that XX is using?* *Why did you do that instead of...?*
	Artwork	*What if I want this part to be...?*
Feedback	Streamer's Action	*Why don't you just move the letters together?*
	Artwork	*I like the basketball pitch picture.*

Table 3. Reference types in visual art live streaming

ring to a certain part of the artwork or a previous action, the artwork could have changed and the context of the message could be lost. S3 mentioned that there usually is a delay of 30 seconds between the chat and the stream so he sometimes feels the need for a reminder to what happened earlier. This echos the observation by Lessel et al. that the streamer has to think back to his previous actions to understand the meaning of the messages correctly when there is a delay [21].

Some streams have moderators to help streamers deal with the messages from the viewers. However, a human moderator is costly and not always available, and chatbots (e.g., Nightbot[5]) that are currently used as moderators can only make general comments that are pre-programmed.

Streamers have difficulty switching back and forth between artwork and chat messaging.
While viewers primarily interact via text chats, streamers can also talk to respond to the viewers' messages. In fact, since artists usually work with a physical object (e.g., mouse, touch-pen, or brush) they often prefer to respond via voice rather than typing on a keyboard. The streamers have to concentrate on their artwork, and frequently looking at the chat and typing their replies can be burdensome. In addition, there are streamers who stream to multiple platforms at the same time, which means there are multiple chat windows to look at. Streamers often set up separate screens or even separate computers for chat messaging and the artwork (S3, S4, S5). This kind of setup makes it even more cumbersome for streamers to switch between the artwork and chat messaging. S5 mentioned that he rarely types in the chat and prefers to communicate with viewers using speech or demonstration (e.g., drawing on the screen or by re-doing certain actions). S2 mentioned that he is sometimes overwhelmed when there are many chat messages, and encourages viewers to retype any questions that he missed. S5 also said that he goes through the chat again at the end of the stream to answer the questions he missed during the stream.

Design Goals

To address the above challenges and to enhance streamer-viewer interaction in live streaming, we identify the following three design goals:

- G1: Help viewers refer to specific part of the artwork or streamer's action, and help streamers understand the context of viewers' questions or feedback.
- G2: Help streamers manage chat messages with minimal disruption to the creative process.
- G3: Encourage viewers' participation and interaction with the streamer and other viewers in the live stream.

SNAPSTREAM: SNAPSHOT-BASED INTERACTION TOOL

Based on these goals, we design Snapstream (Figure 2), a snapshot-based interaction tool for live streaming communication. In Snapstream, streamers and viewers can capture the current video feed of the stream as a screenshot image, annotate on it, and share it with others in the chat. By capturing a snapshot of the stream and including it in the chat, we expect users to easily and clearly refer to parts of the stream to express their questions or comments (G1). In this way, the streamer can understand the context of the viewers' comments by looking at the snapshot. Viewers and streamers can also directly refer to a previously shared snapshot. In addition, we allow streamers to verbally highlight and respond to a snapshot (G2).

Taking and Annotating Snapshots

The main feature of Snapstream allows users to take a screenshot of the stream's video feed, annotate on it, and share it

[5] nightbot.tv

in the chat (G1). Users click on the camera icon in the chat area (Figure 2e) to take a snapshot of the current moment in the stream. After taking the snapshot, users can quickly crop and annotate the image in a new pop-up window (Figure 3). Users can add text boxes to write comments or add marks (ellipses and rectangles) to emphasize specific parts of the image. Finally, users can post the snapshot in the chat with an accompanying text message (Figure 2b). To prevent malicious users from abusing snapshots and cluttering the chat, we implement a 30-second cool-down after a user takes a snapshot before they can take another one.

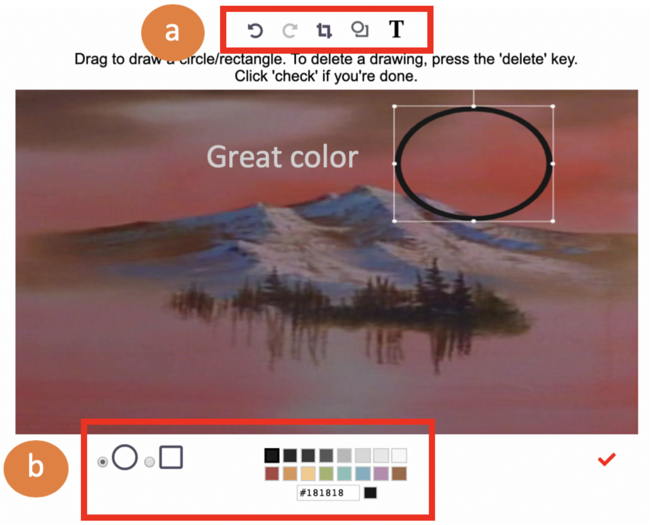

Drag to draw a circle/rectangle. To delete a drawing, press the 'delete' key.
Click 'check' if you're done.

Figure 3. (a) Users can crop and annotate the snapshot with text and shapes. (b) Users can choose the shape of the mark and its color.

Interacting with Snapshots

Once a user posts a snapshot in the chat, both the viewers and the streamer can interact with the snapshot in several ways. First, users can mention a particular snapshot in their message by clicking the reply button on the snapshot (Figure 4a), which automatically adds the tag '@snapshot#X' to the message (Figure 4b). Users can also manually tag a snapshot in a message using the snapshot ID (e.g., '@snapshot#3'). Clicking the snapshot tag (Figure 4c) scrolls the referred snapshot into the center of the chat area and highlights it for more dynamic and easy interaction with the previous snapshots (G3) (Figure 4d).

In addition, the streamer can refer to a snapshot using speech. When a streamer mentions *"snapshot number X"* at any point during the stream, that snapshot will be scrolled into view in the chat window and highlighted for all the viewers for 5 seconds (Figure 2c). This allows the streamer to share the context of the message they are responding to, without being physically disrupted (G2). Once a snapshot has been mentioned by the streamer, it is considered *answered*. Next to that snapshot, Snapstream displays a time-stamp of when the streamer answered that snapshot.

Finally, users can filter the chat to see only snapshots or only text messages (Figure 2f). The snapshot-only view can be useful, for example, when the streamer wants to find questions or comments about the artwork that is accompanied by a snapshot (G2). The *answered* time-stamp also helps the streamer to easily distinguish between comments they already responded to and those they have not yet (G2). The text-only view can be useful, for example, when there are too many snapshots, which take up large areas of the chat. Users may want to just look at the text messages to follow the overall of flow the conversations.

Implementation

We implemented Snapstream as a web interface using HTML, CSS and jQuery. We used socket.io [29] for the real-time chat. To integrate Snapstream with the existing workflow of streamers, Snapstream embeds the streams of Twitch and YouTube Live through their official APIs. Streamers can stream on their usual platforms while also streaming through Snapstream. To take the snapshots, the URL of the stream video in m3u8 format is retrieved by using the twitch-m3u8 library [8] for Twitch and parsing the video information for YouTube Live. We used the JavaScript Web Speech API to recognize the streamer's voice when referencing a certain snapshot.

USER EVALUATION

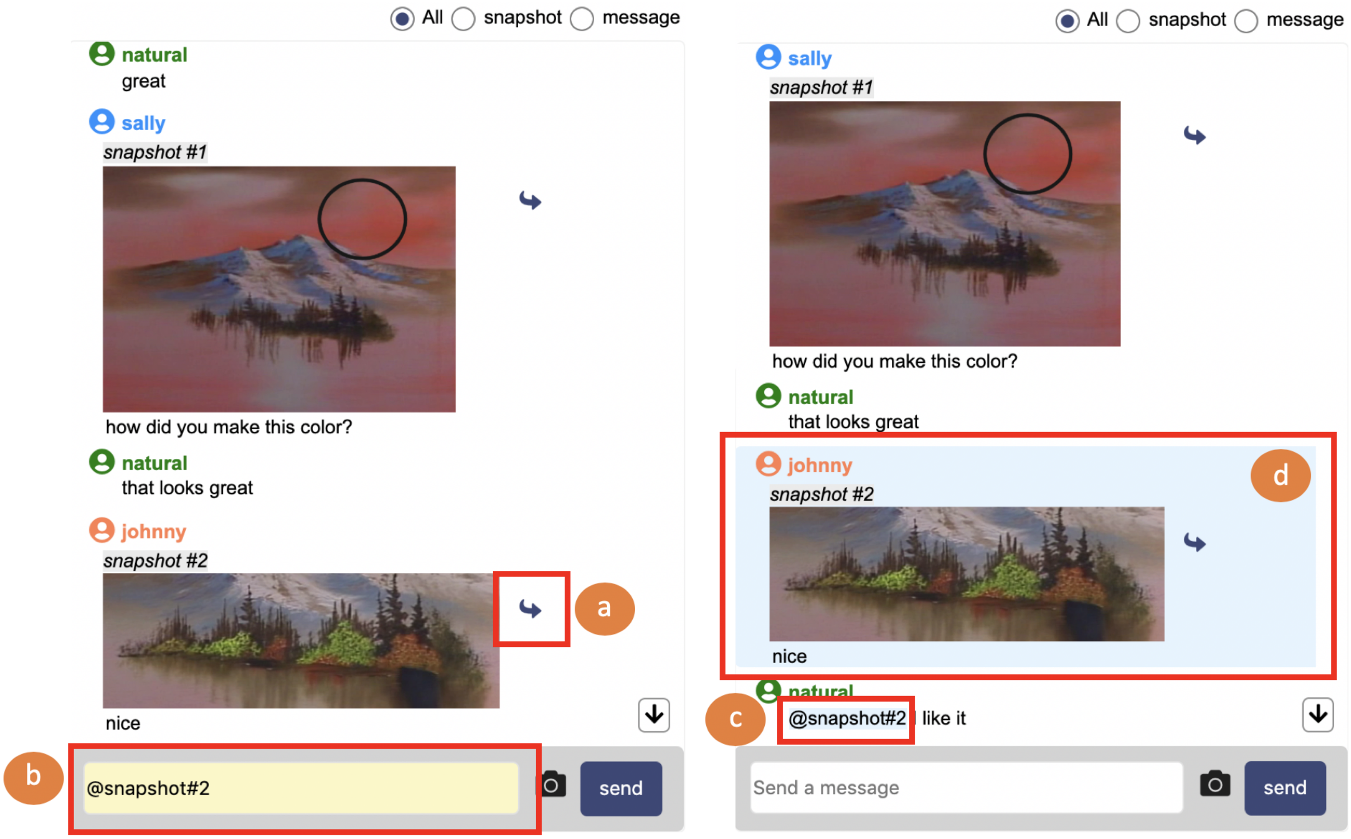
For evaluation, we explore the role of snapshots as a means to enhance streamer-viewer interaction. We deployed Snapstream in 2 real live stream sessions and investigated how streamers and viewers interact through snapshots. The main goals of this qualitative study were to see (1) how streamers and viewers use snapshots to interact with each other and (2) how Snapstream affects their experience in live streaming.

Participants

We reached out to visual art streamers on Twitch and YouTube via email and through the Discord platform[4]. Two Twitch streamers participated in the study. Streamers used Snapstream in one of their actual streaming sessions, while simultaneously streaming on their usual platforms. Through their own social media, they invited their regular viewers to optionally join the streaming session using Snapstream instead of Twitch. To ensure we had enough users to simulate a realistic live stream session, we recruited additional viewers on an online community at our academic institution (for the first session) and via Amazon Mechanical Turk (for the second session). To reliably compare their experience watching live streams with and without Snapstream, we only recruited viewers with prior experience watching visual art live streaming.

The first streamer (S1) streams comic art. S1 has been streaming on Twitch for a year. He has about 10 regular viewers and 291 followers on Twitch. To engage his viewers, in addition to the chat messaging in Twitch, he also uses Hangouts[6] and asks viewers to share their own drawings. S1 engages with the viewers by answering questions and explaining his creative process. 8 regular viewers from his stream and 6 additional recruited viewers participated in the study session. Among these, 3 regular viewers and all of the recruited viewers responded to the study survey (Table 4).

[6] hangouts.google.com



● All ○ snapshot ○ message

natural
great

sally
snapshot #1
how did you make this color?

natural
that looks great

johnny
snapshot #2
nice

@snapshot#2

send

● All ○ snapshot ○ message

sally
snapshot #1
how did you make this color?

natural
that looks great

johnny
snapshot #2
nice

natural
@snapshot#2 I like it

Send a message

send

Figure 4. To mention a snapshot, users can either manually tag the snapshot in the message or (a) click the reply button next to the snapshot (b) which automatically adds the tag '@snapshot#X' to the message. Then, (c) users can click on the snapshot tag (d) which scrolls the snapshot into view and highlights it.

The second streamer (S2) uses Adobe Photoshop in streams to create emojis. S2 has been streaming for 2 years on Twitch. She has about 8 regular viewers and 615 followers on Twitch. To engage her viewers, S2 usually invites her friend to join the stream and broadcasts their verbal communication through Discord, a voice and text chat application. She also talks about her art and answers questions from the viewers. 3 regular viewers and 6 additional recruited viewers participated in the study. All participants responded to the study survey (Table 4).

Procedures

First, we gave a tutorial of Snapstream to the streamers before the session via video call. Recruited viewers for the first session received an in-person tutorial from one researcher, and recruited viewers from the second session and regular viewers from both sessions were given a link to the tutorial slides. Streamers were then asked to stream just as they would normally for approximately an hour, but using Snapstream in addition to their regular platform. Viewers were asked to use Snapstream for at least 30 minutes before answering an online survey. The live streams lasted for about 80 minutes for S1's live stream (L1) and 45 minutes for S2's live stream (L2). We recorded the chat logs including the snapshots. After the stream session, we conducted an online survey with the viewers and a semi-structured interview with the streamers. The interview lasted for 30 minutes. Viewers were compensated with $15 and streamers were compensated with $100 for their participation.

Methods for Analyzing Snapshots

To understand how viewers use snapshots, two of the authors analyzed all the snapshots shared in the chat during the two live stream sessions. We conducted a thematic analysis to categorize all the snapshots. First, we conceptually distinguished between two top-level categories: 'References' (Table 6(a)), which was the main use case identified from the formative study, and 'Engagement', which was another frequently observed use case during the session. Then, we identified three sub-themes within the 'Engagement' category: (Table 6(b)) drawing suggestions, (Table 6(c)) making funny images, and (Table 6(d)) logging intermediate milestones. Two researchers independently coded the snapshots into these four categories and resolved conflicts through discussion. For example, some snapshots could be seen as either a suggestion (b) or a joke (c). In these cases, we decided to classify snapshots that contain meme-like text as jokes, and others as suggestions.

	Regular Viewers		Recruited Viewers	Total Viewers
	w/ survey	w/o survey		
L1	3	5	6	14
L2	3	-	6	9

Table 4. The number of viewers that participated in the study

RESULTS

We discuss the main use cases of snapshots in visual art live streaming and how Snapstream affected the interaction between the streamers and viewers.

Viewers use snapshots to achieve different purposes

In total, 57 snapshots were shared in the chat. In the two streams, viewers shared 25 (7.2% of total messages) and 32 (24.8%) snapshots, respectively (Table 5). Among 57 snapshots, 20 were used for making references (G1) while 37 were used for engagement (G3).

Referencing part of artwork (N=20, Table 6(a)). Viewers annotated on the snapshots to reference specific parts of the artwork and express their opinion. They used the crop tool or added marks to highlight the region they were talking about. These types of snapshots were used to either express passing opinions (e.g., *“Looks like characters from MapleStory”*), or give feedback about the artwork (e.g., *“Make him more lively”*).

Adding suggestions on the artwork (N=6, Table 6(b)). Viewers also used snapshots to make suggestions on the work. Unlike the marks in the previous use case, which were used to emphasize specific parts of the artwork, here, users used the marks to draw suggestions they wanted the streamers to add on their artwork. For instance, one viewer used the shape tool to draw glasses on a character, and another user drew a background object.

Making jokes with funny images (N=22, Table 6(c)). The most frequent use case of snapshots was for humor. Users made jokes by taking snapshots and making funny images. For example, a viewer added an antenna on the streamers drawing of a character’s head and wrote *“Teletubby?”* Another viewer added text to the snapshot of a funny character with a message *“pls help – seth.”*

Logging intermediate milestones (N=9, Table 6(d)). Viewers also used snapshots to record notable intermediate steps in the artwork process. For example, S1 drew multiple characters in the stream and some viewers used snapshots to record when each character was finished. In the survey, viewers also mentioned they wanted to download and save these snapshots for later review.

Viewers’ Experience with Snapstream

In the online survey, we asked viewers how helpful each feature in Snapstream was and how it could be improved. We also asked viewers to compare their interaction using Snapstream with their usual interactions on visual art live streams. We concluded the survey with a SUS usability test [2] to evaluate the feasibility of Snapstream for live deployment. Figure 5 summarizes the viewers’ responses. Below, we summarize our findings from the survey.

Snapstream helped viewers communicate effectively. Viewers felt that the snapshots helped them to express their intentions more easily and comprehensively, especially when they were referring to parts of the artwork that were hard to describe using text. V12 said, *“I was able to point out specific things on the stream I wanted to talk about.”* V7 mentioned *“It was helpful to refer to a specific part when the communication with the streamer was not clear.”* V2 said, *“It became easier to ask questions that used to be hard to describe.”* Overall, viewers were able to express themselves more clearly and also understand each other better (Figure 5-(6),(7),(8)).

Snapstream helped viewers feel more in sync with the streamers. The highlight functionality in Snapstream, whereby the streamer could share a snapshot, allowed viewers to feel more connected with the streamer. Usually in live streaming sessions, when the streamer responds to a particular viewer’s message, other viewers have to figure out which message the streamer is talking about. This can be difficult especially when the streamer does not give the full context of the original message (e.g., by calling out the user name and reading out loud the part of the message). In Snapstream, when the streamer mentioned a snapshot, it was automatically highlighted and scrolled into view for all the viewers to follow. Referring to this feature, V5 said, *“It was good to know which message or image [the] streamer was talking about when he was responding to viewers.”* V6 said, *“I could see what the streamer was referring to right away and it was easier for me to understand with the image than the text.”* The reply timestamps also helped viewers to understand the flow of the chat. V14 said, *“It showed what had already been under discussion which helped when browsing the conversation.”* Overall, by sharing the context with the streamer, viewers understood the streamer better and also felt they were *“interacting more with the streamer (V4)”* (Figure 5-(10)).

Snapstream made the live stream more enjoyable for viewers. Snapstream also made the live stream sessions more entertaining. As mentioned above, humor or making funny images was the most frequent use case of snapshots (Table 6(c)). Due to the nature of live streaming, viewers cannot skip parts of the video. Especially for visual art, the streams can take many hours while the streamer works on the project, and visual progress or change can be rather slow. Viewers actively used snapshots to fill such void by making memes and jokes. V16 said, *“[Snapstream] made interacting with other people on the chat more fun and active.”* In addition to creating their own snapshots, viewers also found looking at snapshots that others shared entertaining. V12 said, *“It was fun to see other people’s creation and what jokes they came up with.”*

Overall, Snapstream helped viewers interact with each other with increased engagement (Figure 5-(3),(4)). V1 commented that *“I felt like I had more interaction with other viewers and the streamer compared to other live streaming platforms.”* Snapstream also expanded the way of interactions for viewers, for example by allowing viewers to draw directly on top of the streamer’s artwork. V18 said, *“It was definitely nice to give*

	# of snapshots	# of messages	Total	% of snapshots	# of tags	# of mentions
Session 1 (L1)	25	322	347	7.2%	4	17
Session 2 (L2)	33	97	130	25.4%	6	19

Table 5. The number of messages for each category written by participants

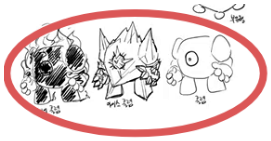
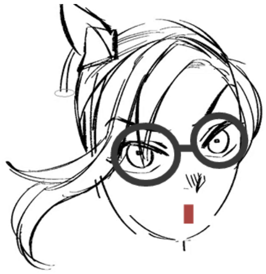
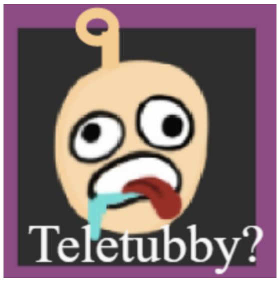
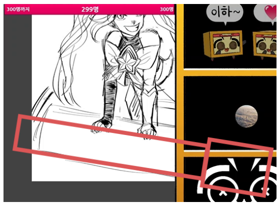
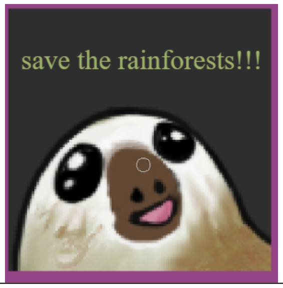
	References	**Engagement**		
Category	**(a) Referring to part of the artwork**	**(b) Drawing suggestions on the artwork**	**(c) Making funny images**	**(d) Logging intermediate milestones**
Snapshot				
Message	“Looks like characters from MapleStory”	“yay”	“...”	“This is so awesome”
Snapshot				
Message	“Make him more lively”	“complete”	“pls help – seth”	“Oh god this looks so cute uwu”

Table 6. Different use cases of snapshots in the studies. (a)-top: a user annotated on the image with the red circle (snapshots cropped to fit the table). (a)-bottom: a user cropped the image. (b)-top: a user added glasses and mouth. (b)-bottom: a user added a background object with red rectangles (snapshots cropped to fit the table). (c)-top: a user added an antenna. (c)-bottom: a user added the text. (d)-top: a user cropped the image. (d)-bottom: a user cropped the image.

the streamer an idea to what they can do like adding stuff to the drawing and the streamer takes inspiration from it.” Viewers felt they had a richer interaction with the streamer and other viewers (Figure 5-(11),(12)). The SUS usability score was 77.92 out of 100, which indicates that Snapstream is easy to use for many users [1]. There were no statistically significant differences between the recruited viewers and regular viewers for all data: the average number of snapshots taken per participant (3 for recruited, 2.67 for regular), the distribution of the snapshot uses (Chi-squared test, χ^2 (3, N=52) = 0.98, p = 0.81), survey score (Figure 5, Mann-Whitney U Test, U=23.5, p=0.26), and SUS scores (t-test, t(16) = 0.12, p = 0.91).

Streamers’ Experience with Snapstream

In the post session interview, we asked streamers about their interaction with viewers using Snapstream, their opinions on the features of Snapstream, difficulties they faced using the system, and general thoughts and feedback on Snapstream. We summarize the major findings.

Streamers enjoyed active viewer participation in Snapstream. Streamers appreciated that viewers participated more actively in the stream with Snapstream. In visual art live streams, the progress is slow and the viewers usually don’t have many ways to participate. For example, S1 uses Google Hangouts to allow viewers to share their drawings and discuss them, but only a few viewers participate. Snapstream provided a new and fun means of participation and created more contents for viewers to engage in. S1 commented that for art streams, it is easy to lose the viewers’ attention, but with Snapstream, viewers’ participation increased. Similarly, S2 enjoyed watching the viewers become more interactive with creative snapshots.

Streamers used snapshots as personal references. Even though neither streamer created snapshots during their live session, they felt it was helpful to see the snapshots generated by their viewers, and both streamers used the snapshots as personal references to review their own stream. S2 commented that she could look back at her artwork process, and could change and compare her artwork based on the snapshots. S1 also commented that he could reflect on his work by looking at the viewers' snapshots on the drawing and reviewing which parts viewers liked most.

DISCUSSION

In this paper, we attempt to understand how snapshots can enhance the interaction in visual work live streaming. Snapshots help users to communicate more expressively and clearly with increased engagement. We discuss the possible ways to expand the use of snapshots for different purposes and across multiple time spans. Then, we discuss the limitations of our work.

Beyond making references: Originally, we anticipated that viewers would use snapshots to make references for asking questions or giving feedback about the artwork, a use case that came up often in our formative study. However, even with the limited editing features in Snapstream, viewers used snapshots for various creative reasons beyond information seeking such as drawing suggestions, making jokes or logging milestones. Snapshots serve as a flexible medium that can be adapted for a wide range of purposes. For instance, users can create their own *fork* of the streamer's artwork by augmenting it via snapshots, or create a highlight reel of the streamer's creative process.

Snapshots as a learning medium: We tested Snapstream on recreational live streams, where the main purpose of the stream is enjoyment (through watching the streamers' artwork). However, we think Snapstream or snapshots in general can be used as an effective instructional aid. Snapshots allow viewers to go beyond passive watching and participate actively. For example, in our study sessions, viewers used snapshots creatively to augment the streamer's artwork or to give concrete suggestions. This reveals opportunities to use snapshot as a learning medium. For example, in instructional live streams, streamers can promote viewers to actively try techniques by asking them to edit or draw on the artwork and to share their input through snapshots. Going beyond live streams, exploring the use of snapshots in academic settings, such as flipped classrooms, is an interesting extension of our research. We expect snapshots can be useful to support various live activities, as well as serve as learning artifacts to assist peer learning and reviewing.

Expanding the live experience across time: Snapstream can also reinvent the replay experience of live streams. One problem that viewers experience when they watch streams after-the-fact is that the video is very long and it is hard to get a quick overview or find interesting moments. Snapshots can provide a visual summary for viewers to get a quick glance of the stream. Furthermore, since viewers can see the timestamp for snapshots that streamers responded to, they can easily find parts of the stream where the streamer addressed questions or comments that they are interested in.

Finally, Snapstream can help extend the live experience by creating a bridge between live and asynchronous settings. For example, as one of the viewers (V3) suggested, by allowing users to download and save snapshots, viewers can use them as artifacts to share within the community. Streamers can also address unresolved questions and share them in an asynchronous platform. In this way, snapshots can enrich the stream community even after the streams have ended.

Beyond visual art: We believe snapshot-based interaction can be applied to domains beyond visual art. Live streaming where visual elements have high importance, such as cooking, origami, or make-up can each benefit from snapshot-based interaction. Game play is another area that participants frequently mentioned. For example, viewers can use snapshots to alert streamers about objects or characters in the screen. They can also share interesting in-game moments in the chat. However, in domains such as games or sports where the visuals change fast, we need to support snapshot taking with minimal delay so that viewers can capture important moments more precisely. Snapshots can also extend to streams that have more audio component such as music performance, if users can take snapshots of *intervals* in live streams.

Limitations

Expressiveness with snapshots: There is a few seconds of delay between the time the user clicks on the snapshot button and the time the actual stream is captured. Although most visual art live streams move slowly, when there is a quick change of scene or window, users might miss the exact moment that they want to capture. Also, allowing the users to edit the snapshot more freely beyond just annotating with marks or adding text can enrich their possibility.

More domains: Our live deployment only looked at recreational visual art live streams, where streamers showcase their artwork and viewers watch it for entertainment. Thus, only a few users used snapshots for informative purposes, such as asking questions. We plan to evaluate how Snapstream is used differently in instructional live streams, such as ones that feature how-to-tutorials on design software.

Scaling to larger streams: Our deployment studies with 14 and 9 viewers were helpful for us to identify how people use snapshots and how snapshots affect their interaction. However, in a live stream session that involves more viewers (100s or 1000s), different behaviors and additional challenges may emerge. For example, as more people use snapshots, it may become challenging for the streamer to keep up with the messages. Potential abuse of snapshots can also be a problem. More investigation is needed to scale Snapstream and to prevent snapshot spamming.

CONCLUSION AND FUTURE WORK

This paper explores using snapshots in visual art live streaming as a way to improve streamer-viewer interaction. We built Snapstream, an interactive tool where users can take snapshots of the live stream, make annotations, and share the annotated snapshot in the chat. Our deployment with real streamers and viewers show that users interact more actively using snapshots by making references, adding suggestions on the artwork, sharing funny images, and logging intermediate milestones. Users were able to communicate more expressively and clearly with increased engagement. We discuss the possible extensions of

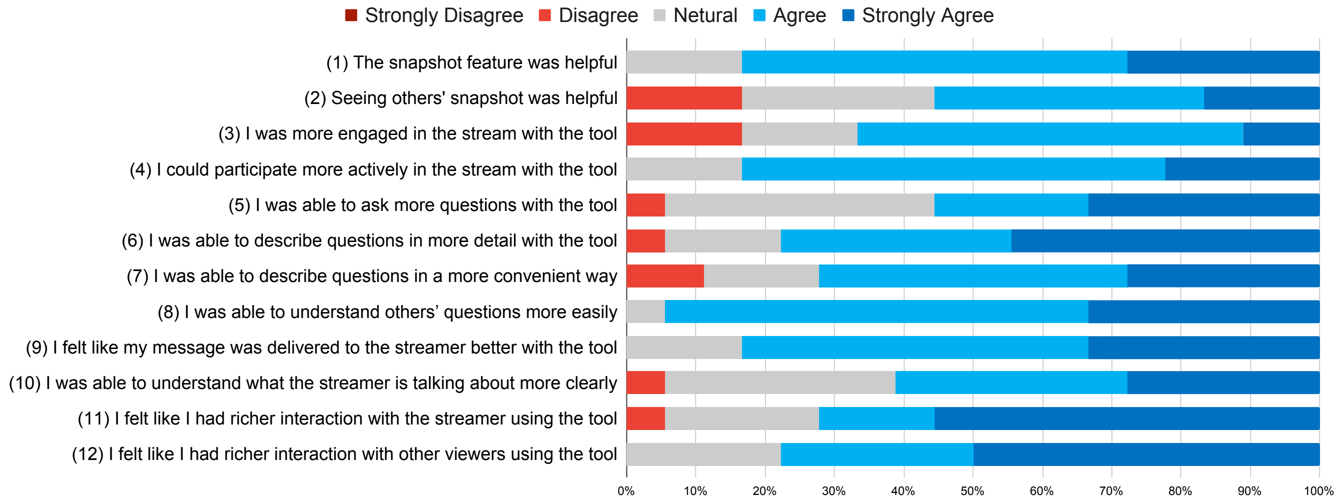


Figure 5. Survey responses about the experience using Snapstream

snapshots that go beyond live streaming, such as using snapshots as a learning medium or as a new way to experience asynchronous watching.

In the future, we plan to deploy Snapstream as a live system and study the interaction in different domains of streams with varying number of participants. We will also explore ways to use snapshots for replays. Examining and expanding the role of snapshots is a promising step towards enriching interaction in live streaming.

ACKNOWLEDGMENTS

The authors would like to thank Howard Pinsky, Terry White, and Paul Trani at Adobe for their kind advice and discussions. This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science and ICT (2017M3C4A7065960). Saelyne Yang is supported by the Kwanjeong Educational Foundation Scholarship.

REFERENCES

[1] Aaron Bangor, Philip T Kortum, and James T Miller. 2008. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction* 24, 6 (2008), 574–594.

[2] John Brooke and others. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.

[3] Yuan-Chia Chang, Hao-Chuan Wang, Hung-kuo Chu, Shung-Ying Lin, and Shuo-Ping Wang. 2017. AlphaRead: Support Unambiguous Referencing in Remote Collaboration with Readable Object Annotation. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW '17)*. Association for Computing Machinery, New York, NY, USA, 2246–2259. DOI: http://dx.doi.org/10.1145/2998181.2998258

[4] Di (Laura) Chen, Dustin Freeman, and Ravin Balakrishnan. 2019. Integrating Multimedia Tools to Enrich Interactions in Live Streaming for Language Learning. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Article 438, 14 pages. DOI: http://dx.doi.org/10.1145/3290605.3300668

[5] Soon Hau Chua, Toni-Jan Keith Palma Monserrat, Dongwook Yoon, Juho Kim, and Shengdong Zhao. 2017. Korero: Facilitating Complex Referencing of Visual Materials in Asynchronous Discussion Interface. *Proc. ACM Hum.-Comput. Interact.* 1, CSCW, Article 34 (Dec. 2017), 19 pages. DOI: http://dx.doi.org/10.1145/3134669

[6] Elizabeth F Churchill, Jonathan Trevor, Sara Bly, Les Nelson, and Davor Cubranic. 2000. Anchored conversations: chatting in the context of a document. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 454–461.

[7] Brian Dorn, Larissa B. Schroeder, and Adam Stankiewicz. 2015. Piloting TrACE: Exploring Spatiotemporal Anchored Collaboration in Asynchronous Learning. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15)*. ACM, New York, NY, USA, 393–403. DOI: http://dx.doi.org/10.1145/2675133.2675178

[8] Samuel Dudik. 2019 (accessed September 20, 2019). *twitch-m3u8*. https://www.npmjs.com/package/twitch-m3u8/

[9] Travis Faas, Lynn Dombrowski, Alyson Young, and Andrew D. Miller. 2018. Watch Me Code: Programming Mentorship Communities on Twitch.Tv. *Proc. ACM*

Hum.-Comput. Interact. 2, CSCW, Article 50 (Nov. 2018), 18 pages. DOI: http://dx.doi.org/10.1145/3274319

[10] Matthew Fong, Samuel Dodson, Xueqin Zhang, Ido Roll, and Sidney Fels. 2018. ViDeX: A Platform for Personalizing Educational Videos. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries (JCDL '18)*. Association for Computing Machinery, New York, NY, USA, 331–332. DOI: http://dx.doi.org/10.1145/3197026.3203865

[11] C. Ailie Fraser, Joy O. Kim, Alison Thornsberry, Scott Klemmer, and Mira Dontcheva. 2019. Sharing the Studio: How Creative Livestreaming Can Inspire, Educate, and Engage. In *Proceedings of the 2019 on Creativity and Cognition (C&C '19)*. ACM, New York, NY, USA, 144–155. DOI: http://dx.doi.org/10.1145/3325480.3325485

[12] Elena L. Glassman, Juho Kim, Andrés Monroy-Hernández, and Meredith Ringel Morris. 2015. Mudslide: A Spatially Anchored Census of Student Confusion for Online Lecture Videos. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1555–1564. DOI: http://dx.doi.org/10.1145/2702123.2702304

[13] Seth Glickman, Nathan McKenzie, Joseph Seering, Rachel Moeller, and Jessica Hammer. 2018. Design Challenges for Livestreamed Audience Participation Games. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '18)*. ACM, New York, NY, USA, 187–199. DOI: http://dx.doi.org/10.1145/3242671.3242708

[14] David S. Goodsell and Jodie Jenkinson. 2018. Molecular Illustration in Research and Education: Past, Present, and Future. *Journal of Molecular Biology* 430, 21 (2018), 3969 – 3981. DOI:http://dx.doi.org/https://doi.org/10.1016/j.jmb.2018.04.043

[15] Lassi Haaranen. 2017. Programming As a Performance: Live-streaming and Its Implications for Computer Science Education. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '17)*. ACM, New York, NY, USA, 353–358. DOI: http://dx.doi.org/10.1145/3059009.3059035

[16] William A. Hamilton, Oliver Garretson, and Andruid Kerne. 2014. Streaming on Twitch: Fostering Participatory Communities of Play within Live Mixed Media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. Association for Computing Machinery, New York, NY, USA, 1315–1324. DOI: http://dx.doi.org/10.1145/2556288.2557048

[17] William A. Hamilton, Nic Lupfer, Nicolas Botello, Tyler Tesch, Alex Stacy, Jeremy Merrill, Blake Williford, Frank R. Bentley, and Andruid Kerne. 2018. Collaborative Live Media Curation: Shared Context for Participation in Online Learning. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 555, 14 pages. DOI: http://dx.doi.org/10.1145/3173574.3174129

[18] Zorah Hilvert-Bruce, James T. Neill, Max SjÃűblom, and Juho Hamari. 2018. Social motivations of live-streaming viewer engagement on Twitch. *Computers in Human Behavior* 84 (2018), 58 – 67. DOI: http://dx.doi.org/https://doi.org/10.1016/j.chb.2018.02.013

[19] Chelhwon Kim, Patrick Chiu, Joseph de la Pena, Laurent Denoue, Jun Shingu, and Yulius Tjahjadi. 2019. Toward Long Distance Tabletop Hand-Document Telepresence. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces (ISS '19)*. Association for Computing Machinery, New York, NY, USA, 391–396. DOI: http://dx.doi.org/10.1145/3343055.3360757

[20] Scott Leetiernan and Jonathan Grudin. 2001. Fostering Engagement in Asynchronous Learning Through Collaborative Multimedia Annotation. (03 2001).

[21] Pascal Lessel, Alexander Vielhauer, and Antonio Krüger. 2017. Expanding Video Game Live-Streams with Enhanced Communication Channels: A Case Study. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 1571–1576. DOI: http://dx.doi.org/10.1145/3025453.3025708

[22] Zhicong Lu, Michelle Annett, Mingming Fan, and Daniel Wigdor. 2019. "I Feel It is My Responsibility to Stream": Streaming and Engaging with Intangible Cultural Heritage Through Livestreaming. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Article 229, 14 pages. DOI: http://dx.doi.org/10.1145/3290605.3300459

[23] Zhicong Lu, Seongkook Heo, and Daniel J. Wigdor. 2018a. StreamWiki: Enabling Viewers of Knowledge Sharing Live Streams to Collaboratively Generate Archival Documentation for Effective In-Stream and Post Hoc Learning. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 112 (Nov. 2018), 26 pages. DOI: http://dx.doi.org/10.1145/3274381

[24] Zhicong Lu, Haijun Xia, Seongkook Heo, and Daniel Wigdor. 2018b. You Watch, You Give, and You Engage: A Study of Live Streaming Practices in China. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 466, 13 pages. DOI: http://dx.doi.org/10.1145/3173574.3174040

[25] Matthew K. Miller, John C. Tang, Gina Venolia, Gerard Wilkinson, and Kori Inkpen. 2017. Conversational Chat Circles: Being All Here Without Having to Hear It All. In *Proceedings of the 2017 CHI Conference on Human*

Factors in Computing Systems (CHI '17). ACM, New York, NY, USA, 2394–2404. DOI: http://dx.doi.org/10.1145/3025453.3025621

[26] Martin Mühlpfordt and Martin Wessner. 2005. Explicit Referencing in Chat Supports Collaborative Learning. In *Proceedings of Th 2005 Conference on Computer Support for Collaborative Learning: Learning 2005: The next 10 Years! (CSCL '05)*. International Society of the Learning Sciences, 460–469.

[27] Amy Pavel, Dan B. Goldman, Björn Hartmann, and Maneesh Agrawala. 2016. VidCrit: Video-Based Asynchronous Video Review. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. Association for Computing Machinery, New York, NY, USA, 517–528. DOI: http://dx.doi.org/10.1145/2984511.2984552

[28] Vikash Singh, Celine Latulipe, Erin Carroll, and Danielle Lottridge. 2011. The Choreographer's Notebook: A Video Annotation System for Dancers and Choreographers. In *Proceedings of the 8th ACM Conference on Creativity and Cognition (C&C '11)*. Association for Computing Machinery, New York, NY, USA, 197–206. DOI: http://dx.doi.org/10.1145/2069618.2069653

[29] Socket.io. 2019 (accessed September 20, 2019). *Socket.IO*. https://socket.io/

[30] Toonation. 2019 (accessed September 20, 2019). *Donation and broadcasting support platform*. https://toon.at/ko/index/

[31] Michael Tsang, George Fitzmaurice, Gordon Kurtenbach, Azam Khan, and William Buxton. 2003. Boom chameleon: simultaneous capture of 3D viewpoint, voice and gesture annotations on a spatially-aware display. *ACM Trans. Graph.* 22 (01 2003), 698. DOI: http://dx.doi.org/10.1145/571985.572001

[32] Twip. 2019 (accessed September 20, 2019). *Twip - Overlay tool for Twitch streamers*. https://twip.kr/

[33] Justin D. Weisz, Sara Kiesler, Hui Zhang, Yuqing Ren, Robert E. Kraut, and Joseph A. Konstan. 2007. Watching Together: Integrating Text Chat with Video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 877–886. DOI: http://dx.doi.org/10.1145/1240624.1240756

[34] Donghee Yvette Wohn, Guo Freeman, and Caitlin McLaughlin. 2018. Explaining Viewers' Emotional, Instrumental, and Financial Support Provision for Live Streamers. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 474, 13 pages. DOI: http://dx.doi.org/10.1145/3173574.3174048

[35] W Wong and HP Pauline. 2010. Teaching developmental psychology using an interactive online video platform. In *Proceedings of the 2010 Conference of the Australasian Society for Computers in Learning in Tertiary Education (ASCILITE)*.

[36] Qunfang Wu, Yisi Sang, Shan Zhang, and Yun Huang. 2018. Danmaku vs. Forum Comments: Understanding User Participation and Knowledge Sharing in Online Videos. In *Proceedings of the 2018 ACM Conference on Supporting Groupwork (GROUP '18)*. ACM, New York, NY, USA, 209–218. DOI: http://dx.doi.org/10.1145/3148330.3148344

[37] Dongwook Yoon, Nicholas Chen, Bernie Randles, Amy Cheatle, Corinna E. Löckenhoff, Steven J. Jackson, Abigail Sellen, and François Guimbretière. 2016. RichReview++: Deployment of a Collaborative Multi-Modal Annotation System for Instructor Feedback and Peer Discussion. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work Social Computing (CSCW '16)*. Association for Computing Machinery, New York, NY, USA, 195–205. DOI:http://dx.doi.org/10.1145/2818048.2819951

[38] Cong Zhang and Jiangchuan Liu. 2015. On Crowdsourced Interactive Live Streaming: A Twitch.Tv-based Measurement Study. In *Proceedings of the 25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '15)*. ACM, New York, NY, USA, 55–60. DOI: http://dx.doi.org/10.1145/2736084.2736091

[39] Jilei Zhou, Jing Zhou, Ying Ding, and Hansheng Wang. 2019. The magic of danmaku: A social interaction perspective of gift sending on live streaming platforms. *Electronic Commerce Research and Applications* 34 (2019), 100815. DOI:http://dx.doi.org/https://doi.org/10.1016/j.elerap.2018.11.002