August 2023

Adjusting Bandwidth in a Video Conference

Stefan Lindmark

Adjusting Bandwidth in a Video Conference

Modern-day video conferencing serves as a powerful communications tool, augmenting the capabilities of traditional communication methods. Due to the visual and collaborative aspects of video conferencing, such platforms have found success in many settings, for example, application in modern workplaces (including work-related meetings), educational contexts (including online classes), social visits, global conferences, community events, or medical appointments just to name a few current applications of the technology. With only increasing functionalities, such as screen sharing, recording, live chat, etc., video conferencing increasingly enhances users' ability to share information and communicate effectively, and overall works to deliver a smooth and collaborative communications experience between users.

In terms of practical implementations, video conferencing is typically enabled through a dedicated video conferencing platform, either equipped, or with access to a video processing module and a control module. The control module may manage data and communications, performing tasks such as initiating connections, maintains session states, and regulating data transmission based on network conditions and system requirements. The video processing module may handle managing the video data itself, and perform tasks ranging from capturing video transmissions from client devices, to compression and decompression, scaling, color correction, and noise reduction, among others. Through the coordinated efforts of the control and video processing modules, the video conferencing platform can support a multitude of connected client devices, each participating in the conference.

Like any other technological implementation, modern video conferencing systems face some challenges. For instance, the process of sending processed video data from the supporting platform to the client devices, and conversely, receiving such data at the client device for playback, is intrinsically bandwidth-intensive. Video data, especially when rendered in high definition or transmitted in real-time, can consume considerable available network bandwidth. Video frame data, audio data, and ancillary pieces of data such as captions or metadata, all contribute to the total data load.

On the client-side, high-quality video streams demand a high-speed, stable internet connection, that may not always be available. Furthermore, the bidirectional nature of many video applications, where client devices are both sending and receiving video streams, can further compound the bandwidth requirements.

Compression techniques may alleviate some of the load, but the fundamental challenge continues, particularly in scenarios where numerous client devices are connected simultaneously, or where network resources are otherwise constrained. The management of bandwidth thus becomes a critical component in the design and operation of video systems, influencing the user experience in terms of quality and responsiveness, as well as the scalability and reliability of the platform at large.

Therefore, a technique is proposed for optimizing bandwidth usage by ranking video streams (e.g. prioritizing or deprioritizing certain video streams over others), and lowering video quality (e.g. by requesting lower resolutions, lower frame rates, or lower quality codec from a video conferencing platform) for lower ranked videos. In some cases, implementing such a technique may leverage an eye-tracker module (including the requisite hardware and software) to determine the video stream the user is focusing on, and may use such information to generate video stream rankings. In such a way, bandwidth usage can be reduced, thus providing relief at both the client and platform level. Such a technique may be applied to many concurrent video streams, for an aggregate effect and more significant reduction in bandwidth usage during a video conference.

To illustrate the elements used for such a technique, while a user's peripheral vision can detect motion and broad shapes area, the focus are associated with a user's eyes is inherently limited. The human eye can only focus sharply on a small area at any given time (typically around 2 degrees of visual field). This means that at any given moment, the majority of visual acuity is concentrated on a very small area. In terms of a video conference-presented UI, this means most visual elements and associated attributes may not be perceived, much less pertain to a user's visual area of focus.

For example, when looking at a video conference user interface (UI) that is laden with many visual elements and video streams, the limited area of visual focus cannot grasp, and intake all data streams. A typical video conference UI may present multiple video feeds from different participants, along with other elements such as chat windows, menus, participant lists, and status indicators. A user may only be able to focus on one element or video stream at a time, and may be in a constant state of shifting or moving their visual focus from one visual element to another (likely according to whom is speaking), and so forth.

Thus, an eye-tracker module can be used to monitor, and determine, which visual element out of the various displayed visual elements, is actively under the visual focus of a user. To begin with, an image data from a user's facial and eye profile can be gathered and processed. In some cases, illuminators (e.g. infrared emitters) emit light towards the eyes. This light reflects off the cornea and is captured by an image capture device (e.g. an infrared camera). Afterwards, the captured images are processed in real time by sophisticated image processing algorithms. Such algorithms identify key features of the eyes, such as the centers of the pupils and the reflections on the corneas. By comparing these features, the system can determine the direction and focus of the eyes. Based on such data, the system employs geometric or statistical models to translate this into a focus point, i.e., the point in space where the user is looking. The eye-track module can then output the approximate coordinates (e.g. in specific pixel coordinates of the display) that the user is currently focusing on.

With this information, a visual element ranking module can then intake the visual elements of the UI, along with the coordinates associated with user focus, and determine a ranking for each visual element of the UI, according to its relevance to the current coordinates and/or visual item of focus. For example, in some cases, each visual element of the UI may be assigned a ranking 1-3 from the ranking module. In some cases, the ranking module can include elements of a PID controller. In some cases, the ranking module can include more sophisticated methods such as neural networks, or other machine-learning algorithms. Once rankings are generated, they can be passed on to the video processing module for quality adjustment.

In some versions of the device, an optional motion prediction module can be used. Such a module may keep a history of the user's eye movement, and thus use such data to predict movement of the user's eye. In some cases the motion prediction module can intake the coordinates, or focus item of the user, and output a motion prediction vector, for a visual element ranking module to incorporate. In some cases, the motion prediction vector can be a 2-dimensional vector indicating the current trajectory. In some cases the motion prediction module can be a rules-based algorithm. In some instances more sophisticated methods such as one or more neural networks, or other machine-learning algorithms, can be used.

Since the user eye has the capacity to move so quickly, the motion prediction vector can be used to preemptively increase, or maintain, video quality, prior to the user's visual focus passing to a new location of the UI. In such a way, the motion prediction vector can help in maintaining overall imperceptibility to the user of the reduction in quality.

Once such rankings are generated, they can be received at the video processing module to be effected. According to such ranking, the video processing module can decrease either the compression quality, the resolution, or frame rate (or all three) associated with each visual element (e.g. video stream) that is being sent to a respective client device and user. Accordingly, the bandwidth used to send and receive such video data to a client device can be reduced. Accordingly, optimization can be performed, bandwidth usage can be reduced, and valuable computing and network resources can be freed, by removing quality in a way that may be imperceptible to a user of the device and video conferencing system.

Figure 1 illustrates an example process for adjusting bandwidth within a video conference. Such a process can be associated with one or more client devices (e.g. client device 120) corresponding to participants of the video conference. Such a process can make use of an eye-track module 142 and associated platform 140, and a motion prediction module 150, a visual element ranking module 162 and a video processing module 164, corresponding to an appropriate video conferencing platform 160. Such elements, platforms, and modules can be connected by a network (not shown) that can transfer communications between such components.

The process can begin at operation 1.1, where the system can identify the UI coordinates 132B of focus of a user of a client device. An eye-tracker module (e.g. module 142) can intake image data 132A, that can be used to monitor, and determine the approximate location of a user’s visual focus. As was discussed above, image data 132A may include image data of a user’s facial and eye profile. Illuminators, such as infrared emitters may emit light towards the user’s eyes, for an image sensor associated with the user device to capture image data from.

Afterwards, the captured images are processed in real time by sophisticated image processing component of the eye-track module, or component of the platform. Such algorithms can include one or more convolutional neural networks (CNNs), visual transformers (ViTs), or any other type of algorithm or process commonly used within image processing. Such image processing can identify key features of the eyes (e.g. the centers of the pupils and the reflections on the corneas), and compare such features to historical, or calibration data to extract the point of visual focus i.e., the point in the UI where the user is looking. As seen in operation 1.1 of FIG. 1, the eye-track module can then output the approximate UI coordinates 132B (e.g. in specific pixel coordinates) that the user is currently focusing on.

At operation 1.3, the system may determine, from UI coordinates 132B, rankings for the visual elements (along with which particular element is in focus) for all the visual elements of the UI. Operation 1.3, may intake UI coordinates 132B, UI data (e.g. from a UI controller) and produce a set of rankings for all visual items within a specific UI (e.g. visual element rankings 138A), which may be used downstream for adjusting video stream quality. Note that in some cases, visual element ranking module may directly receive UI coordinates 132B, and generate the visual rankings 138A from UI coordinates 132B. In some cases, the eye-track module output, instead of coordinates, the element of visual focus, that can then be used to create rankings.

By way of example, in some instances of a video conference, connected client devices (e.g. client device 120) under direction by the video conferencing platform can present a UI to a user of a respective device. Such as UI (by way of example) is visible in figure 1 as UI 122. Such a UI may can include various visual elements (as discussed above) and may be the primary mechanism by which the user engages with the video conference.

Such visual elements can include one or more windows (e.g. as shown in visual elements 101-112 of example UI 122), which may display the video streams or data streams associated with the video conference. But visual elements may include many more forms and purposes, including chat boxes (e.g. chat boxes for a user to input textual information), informational displays (such as participant lists, document viewers, etc.), as well as input elements (such as buttons, sliders, chat interfaces etc. for a user to input data) (as seen by command buttons 124 of example UI 122), or any other kind of visual element commonly associated with a UI.

A user's visual focus (as seen by visual focus indicator 126 in example UI 122), can rest on any portion, or any part of the display, and a such may be associated with any one or more visual elements.

Thus, at operation 1.3, the system may determine from UI coordinates 132B, a series of rankings for each visual elements for a number of visual elements within the UI, based on which particular element (e.g. focus element) of the UI a user is focused on.

In some cases, visual element ranking module may be a simple ranking algorithm (such as a PID controller), that can be used to generate rankings based on a visual element's proximity to the current area, or visual element, within a user's visual focus. In other cases, more sophisticated algorithms that may include one or more machine-learning, or neural networks, may be used. In some cases, (such as the one seen in example UI 122 of FIG. 1), each visual element may be assigned a visual ranking 1-3 (as seen within each window of visual elements 101-112).

In some versions of the device, an optional motion prediction process 1.2 and associated motion prediction module 150 can be used, to generate a motion prediction vector 134 for the visual element ranking module 162 to incorporate. Such a module may intake keep a history of the user's eye movement (e.g. through intaking UI coordinates 132B), and thus use such data to predict movement of the user's eye. In some cases the motion prediction module may intake such coordinates, or focus item of the user, and output a motion prediction vector, for a visual element ranking module to incorporate. In some cases, the motion prediction vector can be a 2-dimensional vector indicating the current trajectory. In some cases the motion prediction module can be a rules-based algorithm. In some instances more sophisticated methods such as a neural networks, or other machine-learning algorithm, can be used.

The motion prediction vector (illustrated by vector 128 as seen in FIG. 1) can be used to influence the ranking module 162, and visual element rankings. Video streams directly in the path of the vector, or within such a field can have their ranking and priority increased, while video streams directly out of the vector path (e.g. visual element 109 and associated video stream as seen in FIG. 1), can have their respective priority and rank decreased. In such a way, the system may preemptively increase, or maintain, video quality, prior to the user's visual focus passing to a new location of the UI. In such a way, the motion prediction vector can help in maintaining overall imperceptibility to the user of the reduction in quality.

After rankings 138A are generated, at operation 1.4, such rankings can be used to adjust the quality of video streams associated with each visual element, according to a respective rank (if such a visual element is a video stream). For example, in the case of UI 122, which is presented at example client device 120, the video streams associated with visual elements 101, 102, 111, and 112, can be reduced in quality when compared to visual elements 103, 104, 106, 107, and 108. The quality of visual elements 105, 109, and 110, can be reduced even further, and so on and so forth.

In some cases, bandwidth adjustments can be in the form of (1) reduced video resolution, which involves lowering the number of pixels that represent the video image, effectively decreasing the clarity and sharpness of the visual content, (2) reduced frame rate, which involves transferring fewer images (e.g. video data) over the network in a given time, and (3) reduced compression and decompression quality meaning that the video is represented using fewer data bits, leading to potential loss in visual and audio quality. In some cases all three may be applied to reduce bandwidth usage of a particular video stream.

Thus, quality adjustments can be made and propagated to an associated client device to optimize and reduce bandwidth, saving valuable computing and networking resources for both the client device and the server device.

ABSTRACT

A technique is proposed for optimizing the bandwidth usage of a video conferencing system. The video quality requested and received by a client device for specific video streams associated with a client device user-interface (UI) can be lowered based on user engagement, thereby lowering total bandwidth usage. An eye-tracking module can be used to identify the specific area of visual focus of a user. A ranking module can then generate rankings for every visual element (and any associated video streams) of the UI. Video quality adjustments can then be made based on such rankings. A quality adjustment could include lowering video resolution, lowering video frame rate, lowering compression quality, or any combination of such. Thus, overall bandwidth usage by a client device, and a corresponding conferencing platform, can be reduced.

Keywords: bandwidth optimization, video quality, video resolution, video frame rate, video compression, client device, video conferencing

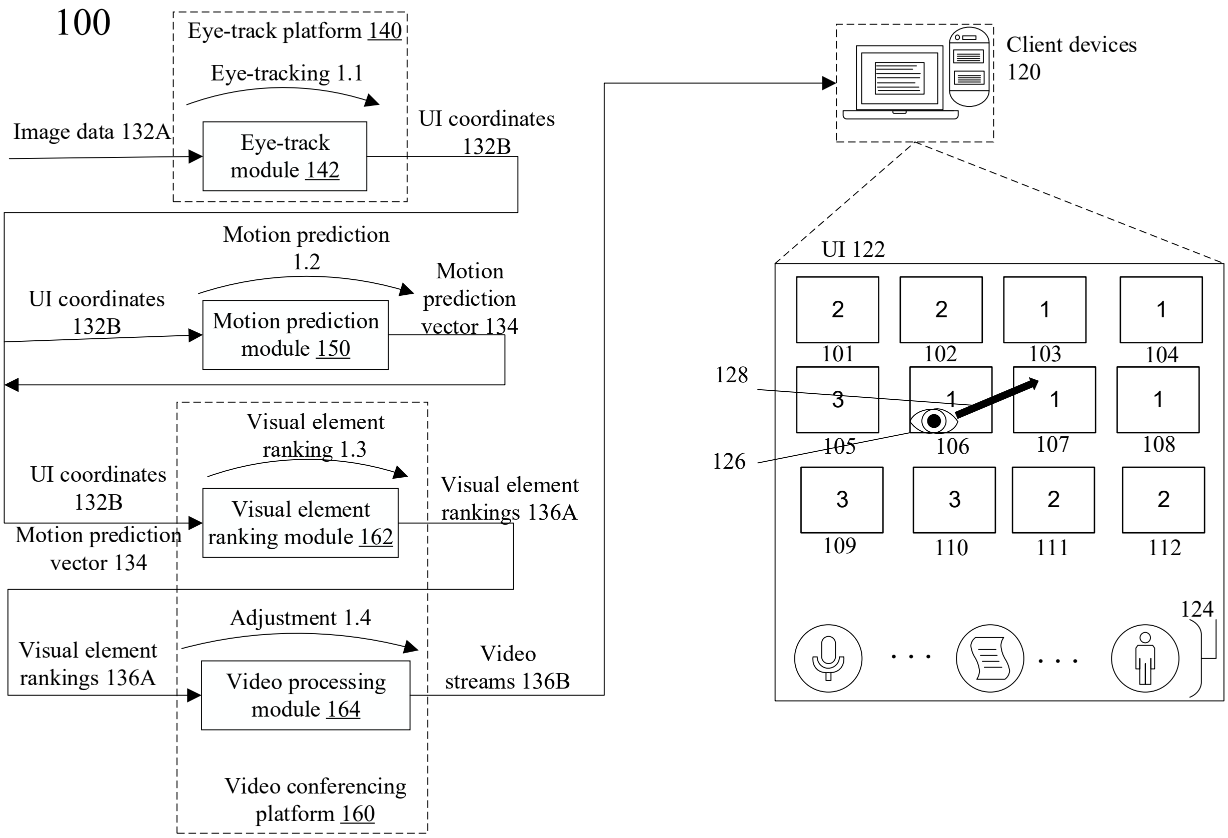
100
Eye-track platform 140
Eye-tracking 1.1
Image data 132A
Eye-track module 142
UI coordinates 132B
Motion prediction 1.2
UI coordinates 132B
Motion prediction module 150
Motion prediction vector 134
Visual element ranking 1.3
UI coordinates 132B
Motion prediction vector 134
Visual element ranking module 162
Visual element rankings 136A
Adjustment 1.4
Visual element rankings 136A
Video processing module 164
Video streams 136B
Video conferencing platform 160
Client devices 120
UI 122
2
101
2
102
1
103
1
104
128
3
105
1
106
1
107
1
108
126
3
109
3
110
2
111
2
112
124

FIG. 1