

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

PELOTON INTERACTIVE, INC.,
Petitioner

v.

NEC CORPORATION,
Patent Owner

IPR Case No. IPR2023-01239
U.S. Patent No. 8,752,101

**SUPPLEMENTAL DECLARATION OF CLIFFORD READER, PH.D. IN
SUPPORT OF PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 8,752,101**

I. Introduction

1. I fully incorporate by reference in this supplemental declaration the analysis and opinions contained in my previous declaration (Ex. 1006) in this matter, dated December 2, 2023. I have reviewed Patent Owner's Response to the Petition, as well as the supporting declaration of Dr. Cooperstock (Ex. 2012). I have been asked to provide this supplemental declaration in order to respond to the arguments made by Patent Owner and Dr. Cooperstock regarding my analysis of the prior art Chou reference ("Chou") (Ex. 1004) in the context of claim limitation 1(d) of the '101 Patent.

II. Claim Construction

2. As I previously explained, in providing my analysis regarding the claims of the '101 Patent at issue in this proceeding, I have adopted the plain and ordinary meaning of the claims. Ex. 1004, ¶¶106-107.

3. I disagree with Dr. Cooperstock's opinion regarding the scope of the claims of the '101 Patent. *See* Ex. 2012, ¶¶55-64.

4. I have reviewed the claims, specification and file history of the '101 Patent. In view of that intrinsic evidence, it is my opinion that at the time of the

alleged invention of the '101 Patent, a POSITA would not have understood that “***based on***” implies that “available time” and “remaining time” must be calculated values that are directly used as an input when calculating the code rate.¹ Nor does it necessarily require a single mathematical equation that explicitly uses the two values directly to calculate a code rate. Based on the intrinsic evidence, a POSITA would not conclude that the values of “available time” and “remaining time” must be first “determined, acquired or examined” before the code rate is determined.

III. Chou Discloses Limitation 1(D) of '101 Patent

A. Chou Determines Code Rate Based on “Available Time”

1. Chou Discloses Calculating Code Rate Based on “Client Buffer Duration”

5. As I previously explained, Chou discloses a *client buffer duration* for a frame that is the difference between the playback deadline and the arrival time of the frame. Ex. 1004, ¶¶90-95, 149-155. The client buffer duration ($t_d(n) - t_a(n)$) corresponds to the claimed “available time” and Chou’s technique to determine code rate is based on the client buffer duration. *Id.*, ¶¶150-153.

2. Dr. Cooperstock’s Reliance on Chou’s Equation 18 is Misplaced

¹ I use “available time” and “remaining time” as a shorthand for the terms “available reproduction time . . . storage device of the reception device” and “remaining time before . . . from a present moment” respectively as they are used in the claims of the '101 Patent.

6. Dr. Cooperstock disputes my mapping of the Limitation 1(d) by focusing on Equation 18 of Chou in order to argue that “[a] value of *Chou*’s ‘gap’ between the ‘arrival schedule’ and the ‘playback deadline’ is never used to ‘determine the code rate.’” Ex. 2012 (Cooperstock Dec.), ¶¶97; *see also id.*, ¶¶72. In particular, Dr. Cooperstock disputes that my identification of “available time” is based on buffer duration. *Id.*, ¶¶87-110. In my opinion, Dr. Cooperstock’s conclusions regarding Equation 18 and related equations are incorrect. I demonstrate this by providing additional context regarding Equation 18 in Chou and how that equation is employed to determine code rate.

7. Through Equations 10 to 17, Chou derives Equations 18 and 19 that are used to ***calculate coding rate $r_c(n)$ for frame (n)***. Ex. 1004, ¶¶66-70. Equations 18 and 19 are excerpted below:

$$t_b(n) = t_b(n-1) + \frac{r_c(n)}{f(n)\tilde{r}_a(n)} + w(n-1), \text{ where} \quad (18)$$

$$w(n-1) = \frac{g(n-1)}{\tilde{r}_a(n)} - \frac{g(n-1)}{\tilde{r}_a(n-1)} + v(n). \quad (19)$$

In Equation 18, $t_b(n)$ and $t_b(n-1)$ are the upper bound arrival times for frames (n) and (n-1) respectively. Chou mentions that coding rate $r_c(n)$ for frame (n) can be controlled so that the upper bound on the arrival time $t_b(n)$ reaches a desired value. Ex. 1004, [0069]. Chou characterizes Equation 18 as “the ***state transition equation***

of feedback control system” and concludes that a control-theoretic approach can be employed to regulate coding bit rate. *Id.*

8. Chou mentions that the upper bound $t_b(n-1)$ for frame (n-1) can be calculated using Equation 16. *Id.* Equation 16, when applied to frame (n-1), yields the following:

$$t_b(n-1) = t_a(n-1) + \frac{g(n-1)}{\tilde{r}_a(n-1)}$$

9. Notably, both modified Equation 16 and Equation 19 rely upon a variable referred to as the **gap $g(n-1)$** . As discussed below, **gap $g(n-1)$** represents the *data in the decoder buffer* when frame (n-1) arrives at time $t_a(n-1)$. Ex. 1004, ¶¶60, 66. It also represents the corresponding playback duration for the data in the decoder buffer. I provide a brief overview of Chou’s leaky bucket model to illustrate how Chou defines **gap $g(n-1)$** and how Chou determines code rate for frame (n) based on gap $g(n-1)$.

10. In Figure 2, Chou depicts the communication channel for its disclosed system, which is a “block diagram of a simplified streaming media communications pipeline,” where points A, B, C, and D represent different “points” in the pipeline. Ex. 1004, ¶36.

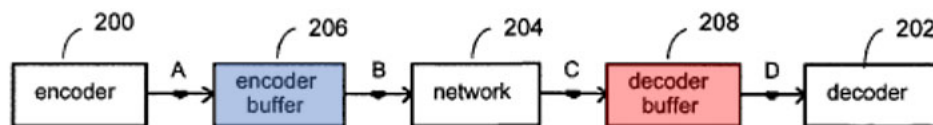


FIG. 2

In Figure 3, which I have annotated below, the gap between Schedule A and Schedule B (shaded blue) represents the size of the encoder buffer (in bits) at any point in time. *Id.*, ¶60. Likewise, the gap between Schedule C and Schedule D (shaded red) represents the size of the decoder buffer (in bits) at any point in time.

*Id.*²

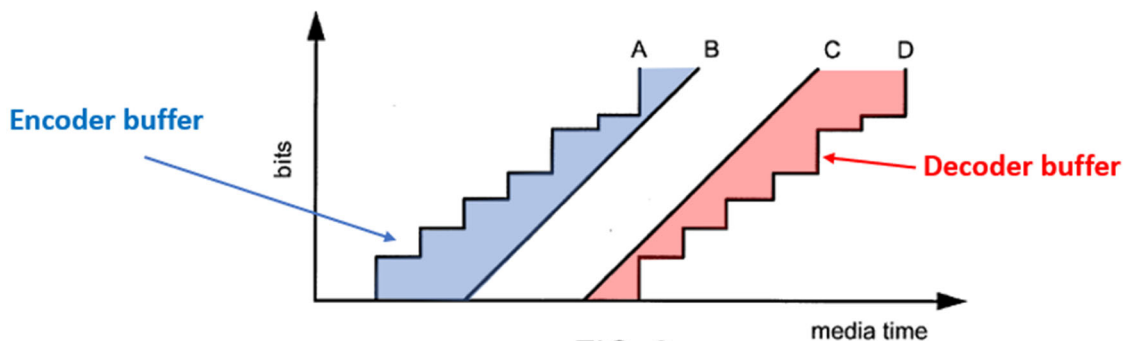
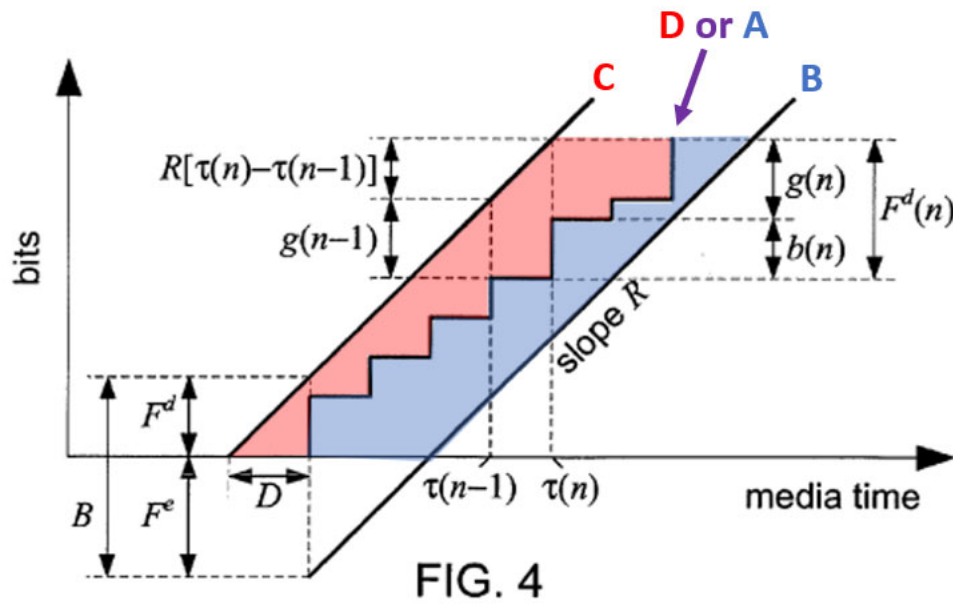


FIG. 3

11. Importantly, Chou states that “[t]he encoder and decoder buffer sizes are complementary,” and therefore “the coding schedule (either A or D) can be contained within a buffer tube,” as illustrated in Figure 4 below, which I have annotated. *Id.*, ¶60. When comparing Figures 3 and 4, the staircase pattern in

² Following well-established practice in the art, the vertical steps in the figure model the processes for encoding, decoding and the corresponding addition and removal of bits to and from the buffer as instantaneous.

Figure 4 corresponds to either Schedule A or D of Figure 3. Because Schedule C and Schedule D are complementary, they mirror each other. Similarly, schedule of the left of the staircase pattern in Figure 4 is Schedule C (upper bound) and the schedule on the right is Schedule D (coding schedule). *Id.*



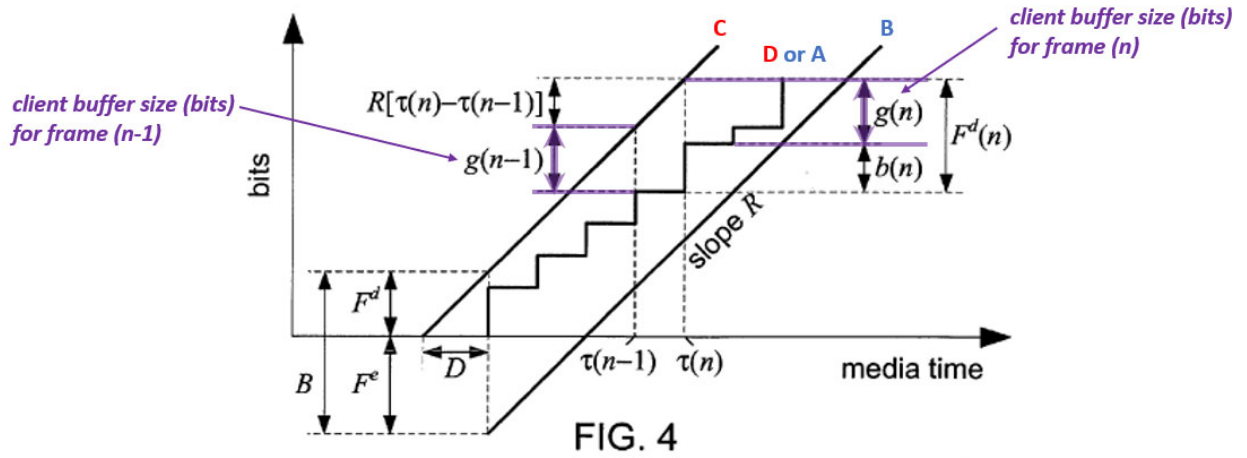
12. In Figure 4, **gap $g(n)$** is the difference between the buffer tube upper bound (i.e., Schedule C) and the coding schedule (i.e., Schedule D). Ex. 1004, ¶¶60, 66. In other words, **gap $g(n)$** represents data in the decoder buffer after frame (n) is decoded at the decoder.³ Likewise, **gap $g(n-1)$** represents the number of bits

³ Hereinafter, I use “time τ ” to refer time media time, and “time t ” to refer to client time, consistent with Chou’s “temporal coordinate systems.” See Ex. 1004, [0059].

in the decoder buffer for frame $(n-1)$. In Figure 4, $b(n)$ represents is the number of bits for frame (n) that were added to the decoder buffer at time $\tau(n)$. *Id.*, ¶61, 66.

As shown in Figure 4 and in Chou's Equation 10, decoder buffer fullness $F^d(n)$ is the sum of (i) bits that comprise the frame $b(n)$ and (ii) bits there were present in the buffer $g(n)$ when frame (n) arrived. *Id.*, ¶66, Figure 4.

13. Gaps $g(n)$ and $g(n-1)$ are depicted in annotated Figure 4 below:



In the figure above, **gaps $g(n)$ and $g(n-1)$** are denoted in bits on the y-axis.

However, a POSITA would be able to easily convert the buffer size in bits into the corresponding playback duration in seconds providing the coding rate r_c is known.

Thus, Equation 18 in Chou can be used to calculate the coding rate $r_c(n)$ for frame (n) based on the size of the decoder buffer (**gap $g(n-1)$** either in bits or in seconds).

14. Chou extends the model depicted in Figure 4 to include a margin of safety (for variations in transmission bit rate) by incorporating a playback deadline

t_d and a target schedule t_T , as shown in Figure 5. In addition, Chou derives Figure 5 by dividing the vertical scale of the schedules of Figure 4 by the arrival rate r_a such that “the schedules are obtained in *client time*, rather than *bits*.” Ex. 1004, ¶67. As a result, *coding* schedule (Figure 4) yields *arrival* schedule (Figure 5), which provides the arrival time $t_a(n)$ at the client and the buffer tube upper bound (*in bits*) (Figure 4) yields the buffer tube upper bound $t_b(n)$ (*in time*) (Figure 5). *Id.* Notably, Chou only discusses the “target schedule” *after* presenting Equation 18. Ex. 1004, ¶70. Accordingly, a POSITA would have understood that when the Equation 18 accounts for the “playback deadline” and “target schedule” of Figure 5, it would factor in the client buffer size according to this model. *Id.*, ¶67.

15. As I mentioned earlier, Chou discloses a client buffer duration for frame (n), which is the difference between the playback deadline $t_d(n)$ for frame (n) and time of arrival $t_a(n)$ for frame (n). Ex. 1006, ¶¶149-155. Mirroring that opinion, the annotated Figure 5 denotes the ***client buffer duration*** (or playback duration) for frame (n-1) (dotted red arrow on left), which is difference between the playback deadline $t_d(n-1)$ and the arrival time $t_a(n-1)$. Likewise, the figure (red arrow on right) denotes the ***client buffer duration*** associated with frame (n), which is the difference between the playback deadline $t_d(n)$ and the arrival time $t_a(n)$.

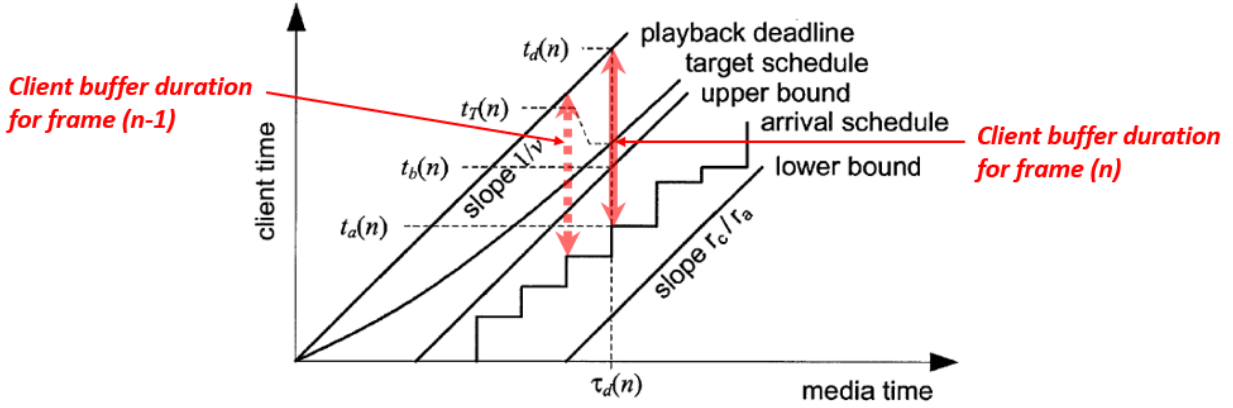


FIG. 5

3. Chou Takes One-Frame Delay Into Account

16. Chou mentions that the calculation of frame rate must account for transmission delay when providing feedback from the client to the server. Ex. 1004, ¶¶69, 81. Chou mentions that if the frame (n) arrives at time $t_a(n)$, the earliest frame for which the controller can determine code rate is frame (n+2) because the controller has already started to transmit frame (n+1). *Id.* When frame (n-2) arrives at the decoder, the **client buffer duration** will be the difference between playback deadline $t_d(n-2)$ and arrival schedule $t_a(n-2)$, *Id.*, ¶67, as depicted below:

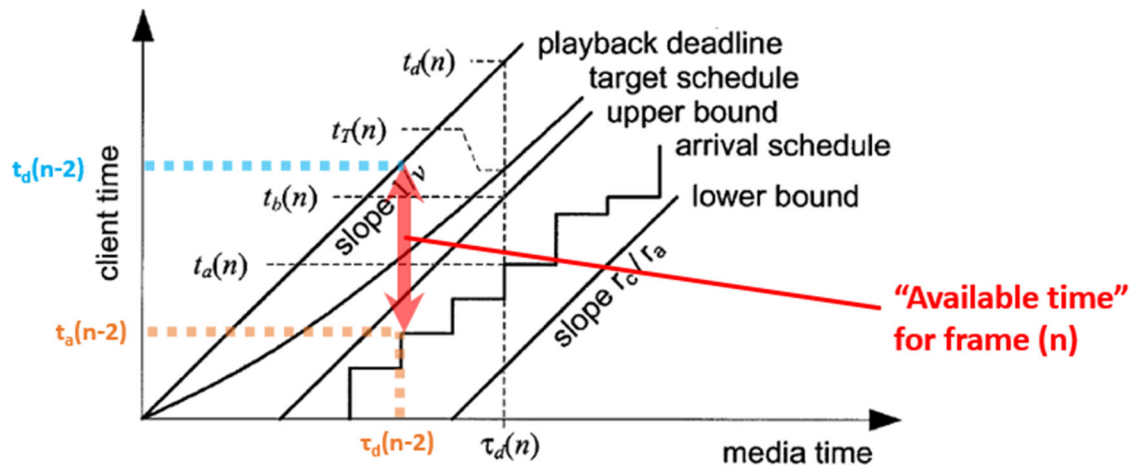


FIG. 5

A POSITA would have been able to easily modify Equations 18 and 19 (and other equations leading up to Equation 18) to account for one-frame delay referenced by Chou. As I mentioned earlier, the client buffer duration (difference between playback deadline and arrival schedule) corresponds to the claimed “available time.” Ex. 1006, ¶153.

17. It must be noted that when determining code rate for frame (n), Chou relies upon the buffer duration for prior frames. Buffer duration for frame (n) (i.e., difference between $t_d(n)$ and $t_a(n)$) cannot be determined because the arrival time $t_a(n)$ for frame (n) is not yet known. As I testified during my deposition, frame (n) is yet to be transmitted and its arrival time can only be estimated. In order to overcome this issue, as shown above, Chou relies on buffer durations associated with prior frames, e.g., frame (n-1) and (n-2), because arrival times for those

frames are known to the controller via the *feedback mechanism* of Chou when the controller is determining the coding rate $r_c(n)$ for frame (n). Ex. 1004, ¶¶69, 81.

18. The process for determining the code rate in the '101 Patent operates in a similar manner. As shown in Figures 3 and 6 of the '101 Patent, client devices 20a and 20b periodically acquire and transmit “available time” to the server 10. Ex. 1001, Fig. 3 (steps S107 and S108) and Fig. 6 (S305 and S306). Both figures depicts that some time elapses before the server receives the available time transmitted by the clients and employs that information to determine the code rate. *Id.*, Figures 3, 6 and 10. Thus, the server 10 employs the acquired available time that represents the buffer status of the client when the client acquired the available time a while ago. It is not possible to determine the status of the buffer in the clients 20a and 20b when the server 10 is determining the code rate at “a present moment.” As a result, the claims of the '101 Patent do not impose any temporal relationship between the time when the code rate is determined and the time at which available time was acquired.

B. Chou Determines Coding Rate Based on Remaining Time

1. Dr Cooperstock Mischaracterizes my Opinion Regarding “Remaining Time”

19. Dr. Cooperstock’s critique of my opinion regarding “remaining time” stems from his incorrect understanding of my opening declaration. Ex. 2012

(Cooperstock Dec.), ¶¶ 111-112. Specifically, Dr. Cooperstock states that I equated “target buffer duration” disclosed in Chou to the claimed “remaining time.” *Id.* Dr. Cooperstock is incorrect.

20. In my declaration, I stated that in Chou’s system, the **gap** from the **time at which the system is determining the code rate** for a frame to the **playback deadline $t_d(n)$** for that frame constitutes the **remaining time** for frame (n). Ex. 1006, ¶147. I even clarified during my deposition that the claimed remaining time for a frame corresponds to the duration between the time when Chou determines the code rate to the playback deadline for that frame. Ex. 2015 (Reader dep. tr.), p. 116-117. Dr. Cooperstock does not list my deposition transcript in the section that lists the materials he reviewed. Ex. 2012 (Cooperstock Dec.), ¶27. Contrary to Dr. Cooperstock’s assertion, I did not equate Chou’s “target buffer duration” to the claimed remaining time.

2. Chou Discloses Remaining Time

21. As discussed above, Chou mentions that when frame ($n-2$) arrives, then the earliest frame for which the controller can determine the coding rate is frame (n). Ex. 1004, ¶81. As shown in Figure 5 below, frame ($n-2$) arrives at time $t_a(n-2)$. Because the coding rate for frame (n) is being determined when frame ($n-2$) arrives, the arrival time $t_a(n-2)$ of frame ($n-2$) constitutes the **present moment** (which I will refer to using the shorthand $t_{present}$) for frame (n). The playback

deadline $t_d(n)$ for frame (n) is the claimed “reproduction start time.” Thus, the remaining time for frame (n) is the difference between the playback deadline $t_d(n)$ and the present moment $t_a(n)$ (or the arrival time $t_a(n-2)$ for frame (n-2)), as shown in with the green arrow below.

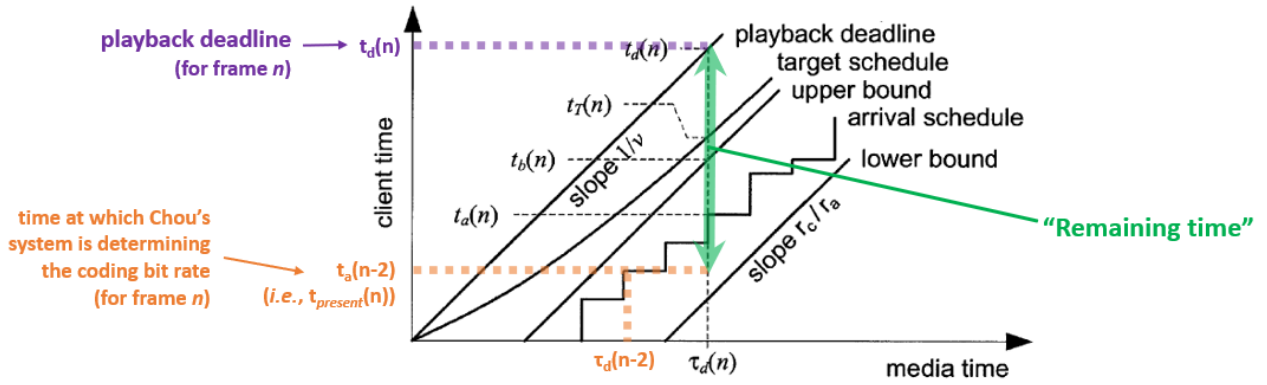


FIG. 5

3. Chou's Equations Calculate Coding Rate based on Remaining Time

22. Chou's technique “attempt[s] to control the buffer tube upper bound” $t_b(n)$ for frame (n) “so that it tracks the target schedule” time $t_T(n)$ for that frame. Ex. 1004, ¶70, Equation 20. Chou mentions that the upper bound $t_b(n)$ is based on arrival time $t_a(n)$. *Id.*, Equation 16. Chou identifies Equation 20 as a cost function that aims to control the upper bound to the target schedule and to minimize quality variations due to frequent changes coding rate. *Id.*, ¶¶ 71-72. As depicted below, the first term in Equation 20 penalizes any deviations between the upper bound $t_b(n)$ and the target schedule $t_T(n)$:

$$I = \sum_{n=0}^N \left((t_b(n) - t_T(n))^2 + \sigma \left(\frac{r_c(n+1) - r_c(n)}{\tilde{r}_a(n)} \right)^2 \right), \quad (20)$$

23. Using Equation 18, Chou determines coding rate $r_c(n)$ for frame (n) when frame (n-2) has arrived at time $t_T(n)$. Thus, for frame (n), the arrival time of frame (n-2) corresponds to the “present moment (or $t_{present}$). When Chou attempts to ascertain a value for the upper bound $t_b(n)$ from the present moment ($t_{present}$), Chou considers the time duration from the present moment ($t_{present}$) to the upper bound $t_b(n)$.

24. In addition, Chou expressly derives the target schedule t_T from the playback deadline t_d . Ex. 1004, ¶¶76-77 (disclosing two exemplary methods for choosing the target schedule as a function of playback deadline). Thus, Chou specifies the delay between $t_T(n)$ and $t_d(n)$ for frame (n). Because Chou attempts to determine a coding rate $r_c(n)$ for frame (n) that allows $t_b(n)$ to track $t_T(n)$, Chou also considers the duration between the target schedule $t_T(n)$ and the playback deadline $t_d(n)$. Thus, Chou determines the that coding rate is based on the gap between $t_{present}$ for frame (n) and playback deadline $t_d(n)$. As I have mentioned earlier, this difference corresponds to the claimed “remaining time.” Ex.1006, ¶147.

IV. MISC. ISSUES

A. Dr. Cooperstock’s Analysis for Limitation 1(d) Supports my Conclusions

25. I have reproduced below Dr. Cooperstock's table in which he attempts to characterize "remaining time" and "available time." Ex. 2012 (Cooperstock Dec.), at 56.

Frame No.	Playback Deadline (Media Time)	Playback Deadline (Client Time)	Arrival Time (Media Time)	Arrival Time (Client Time)
N-3	0s	1:01:30 PM	-5s	1:01:25 PM
N-2	30s	1:02:00 PM	15s	1:01:45 PM
N-1	60s	1:02:30 PM	Unknown	Unknown
N	90s	1:03:00 PM	Unknown	Unknown
N+1	120s	1:03:30 PM	Unknown	Unknown

Frame **N-2** has arrived, the system is calculating the code rate for frame **N**. Therefore, the "current time" (present moment for determining code rate of frame **N**) is the arrival time of frame **N-2**, which is **15s (media time)** or **1:01:45 PM (client time)**.

"Remaining Time" (for frame **N**) = **75s**
(e.g., "90s – 15s" in media time; or 1:03:00PM – 1:01:45PM)
Thus, **playback deadline (90s)** is **not equal to remaining time (75s)**.

"Available Reproduction Time" = **45s**
(e.g., frame **N-2** will play for 30s in addition to the 15s left of playback for **N-2**)
Thus, "playback deadline" minus "arrival time" (**30s – 15s = 15s**) is **not equal to available reproduction time (45s)**.

Ex. 2012 (Cooperstock Dec.), at 56.

26. Once again, Dr. Cooperstock mischaracterizes my opinion. In the table above, Dr. Cooperstock apparently assumes that I equated the claimed "remaining time" to the playback deadline in Chou. That is incorrect.

27. The sequence shown in Dr. Cooperstock's table actually supports my analysis when viewed under correct interpretation of Chou. With respect to "Remaining Time," the sequence shows that the present moment (or $t_{present}$) is 15 seconds, i.e., the arrival time $t_a(n-2)$ for frame (n-2). The playback deadline $t_d(n)$ for frame (n) is 90 seconds. In my prior declaration, I stated that the "remaining time" when determining code rate $r_c(n)$ for frame (n) is the difference of $t_d(n)$ and $t_{present}$ (or arrival time $t_a(n-2)$). Ex. 1006, ¶147. Thus, the "Remaining Time" for frame (n) is 90 seconds minus 15 seconds, which yields 75 seconds. Contrary to the suggestion in the table, there is no discrepancy for "Remaining Time."

28. With respect to "Available Time" also, Dr. Cooperstock's table likewise supports my analysis when viewed in correct context of Chou's disclosure. As I explained above, when determining the code rate for frame (n), the Available Time is the difference of playback deadline $t_d(n-2)$ and the arrival time $t_a(n-2)$. See Section III.A above. In the table above, $t_d(n-2)$ is 30 seconds and $t_a(n-2)$ is 15 seconds. Thus, the "Available Time" when frame (n-2) arrives ($t_a(n-2)$) is 15 seconds, and not 45 seconds as Dr. Cooperstock concludes.

29. Dr. Cooperstock arrives at a value of 45 seconds for "Available Time" because to calculate "Available Time," he adds the duration of frame (n-2) (i.e., 30 seconds) and the portion of frame (n-3) that has not yet been reproduced (i.e., 15

seconds). Dr. Cooperstock's conclusion is also based on a misinterpretation of Chou. As I explained above, Chou describes decoder buffer fullness $F^d(n)$ as a sum of $g(n)$ and $b(n)$. Ex. 1004, ¶66. Per Equation 10, Chou explains that $g(n)$ denotes bits in the decoder buffer when frame (n) arrives), whereas $b(n)$ denotes the bits that represent frame (n). *Id.* Thus, the number of bits in the buffer $g(n)$ does not include the bits $b(n)$ for the frame that just arrived. In other words, when calculating the *client buffer duration*, Chou does not include the bits of the recently arrived frame $b(n)$. Thus, Dr. Cooperstock is mistaken when he includes the duration (30 seconds) for frame (n-2) when calculating the "Available Time" frame (n). The correct value is 15 seconds that corresponds to **gap $g(n)$** in Chou's framework (i.e., data in the decoder buffer) and gap $g(n)$ does not include $b(n)$.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Executed on August 27, 2024, in Escondido, California.

A handwritten signature in cursive script, appearing to read "Clifford Reader", written in black ink.

Clifford Reader, Ph.D.