

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC

Petitioner

v.

ART RESEARCH AND TECHNOLOGY LLC

(record) Patent Owner

IPR2025-01405

Patent No. 10,681,103

**PETITION FOR INTER PARTES REVIEW
UNDER 35 U.S.C. §§ 311-319 AND 37 C.F.R. § 42.100 ET. SEQ**

TABLE OF CONTENTS

I.	INTRODUCTION	7
A.	Summary of the '103 Patent.....	7
B.	Overview of Prosecution History	8
II.	CLAIM CONSTRUCTION	8
III.	DETAILED EXPLANATION OF THE REASONS FOR UNPATENTABILITY	9
	Ground 1. Claims 1-10 and 13-22 are obvious over Hedinsson in view of Ford 9	
A.	Overview of the Ground.....	9
1.	Overview of Hedinsson.....	10
2.	Overview of Ford.....	13
B.	Rationale (Motivation) Supporting Obviousness.....	14
C.	Reasonable Expectation of Success	16
D.	Graham Factors	17
E.	Analogous Art	17
F.	Claim Mapping.....	18
	CLAIM 1	18
	CLAIM 2	28
	CLAIM 3	29
	CLAIM 4	38
	CLAIM 5	39
	CLAIM 6	39

CLAIM 7	39
CLAIMS 8-10.....	46
CLAIM 13	46
CLAIMS 14-18.....	47
CLAIM 19	48
CLAIMS 20-22.....	48
Ground 2. Claims 11-12 and 23-24 are obvious over Hedinsso in view of Ford and Kostello	49
A. Overview of the Ground.....	49
1. Overview of Kostello	49
B. Rationale (Motivation) Supporting Obviousness.....	52
C. Reasonable Expectation of Success	56
D. Analogous Art	56
E. Claim Mapping.....	57
CLAIM 11	57
CLAIM 12	66
CLAIM 23	72
CLAIM 24	72
IV. CONCLUSION.....	73
CLAIM LISTING APPENDIX	74

TABLE OF EXHIBITS

Exhibit No.	Description
1001	U.S. Patent No. 10,681,103 B2 (“ ’103 patent ”).
1002	Declaration of Dr. Henry Houh.
1003	C.V. of Dr. Henry Houh.
1004	U.S. Patent No. 8,793,282 B2 (“ Hedinsson ”).
1005	Excerpts of J.L. Ford, <i>Programming for the Absolute Beginner</i> (2007) (“ Ford ”).
1006	U.S. Patent No. 8,745,500 B1 (“ Kostello ”).
1007	Prosecution History of U.S. Application No. 16/134,541 (which issued as the ’103 patent).
1008	U.S. Patent Appl. Pub. No. 2005/0084232 A1.
1009	Declaration of June Ann Munford.
1010	U.S. Patent Appl. Pub. No. 2012/0284343 A1.
1011	U.S. Patent Appl. Pub. No. 2014/0229866 A1.
1012	U.S. Patent Appl. Pub. No. 2015/0185965 A1.
1013	Redline Versions of Claims 4, 5, 8-10, and 13-24 of U.S. Patent No. 10,681,103.

Petitioner respectfully requests *inter partes* review under 35 U.S.C. § 311 of claims 1-24 of U.S. Patent No. 10,681,103 B2 (“the ’103 patent”).

NOTICE OF LEAD AND BACKUP COUNSEL

Lead Counsel Matthew A. Smith Reg. No. 49,003 SMITH BALUCH LLP 376 Boylston Street Suite 401 Boston, MA 02116 (202) 669-6207 smith@smithbaluch.com	Backup Counsel Elizabeth Laughton Reg. No. 70,484 SMITH BALUCH LLP 376 Boylston Street Suite 401 Boston, MA 02116 (703) 585-8839 laughton@smithbaluch.com
	Backup Counsel Andrew S. Baluch Reg. No. 57,503 SMITH BALUCH LLP 376 Boylston Street Suite 401 Boston, MA 02116 (202) 880-2397 baluch@smithbaluch.com

NOTICE OF THE REAL-PARTIES-IN-INTEREST

The real-parties-in-interest for this petition are Google LLC (“Google”)¹ and YouTube, LLC. YouTube, LLC is a subsidiary of Google.

¹ Google LLC is a subsidiary of XXVI Holdings Inc., which is a subsidiary of Alphabet Inc. XXVI Holdings Inc. and Alphabet Inc. are not real parties-in-interest to this proceeding.

NOTICE OF RELATED MATTERS

Petitioner is aware of the following matters that may affect, or be affected by, a decision in this proceeding.

The '103 patent is asserted in *ART Research and Technology v. Google, LLC, et al.*, Civ. Action No. 3:24-cv-04898 (N.D. Cal), filed August 8, 2024. In that case, Google (the Petitioner here) and YouTube, LLC filed a motion to dismiss under FRCP 12(b)(6), alleging that all asserted patents claim ineligible subject matter under 35 U.S.C. § 101. The Court has not yet ruled on that motion, held an initial case management conference, or entered a scheduling order. In addition, the parties have not served infringement or invalidity contentions.

In addition, Petitioner is concurrently requesting *inter partes* review of the other three patents at issue in that litigation: U.S. Patent No. 9,451,001 (IPR2025-01404), U.S. Patent No. 10,084,840 (IPR2025-01403), and U.S. Patent No. 10,609,442 (IPR2025-01406). Each of these patents is related to the '103 patent.

NOTICE OF SERVICE INFORMATION

Please address all correspondence to the lead counsel at the addresses shown above. Petitioner consents to electronic service by email at: smith@smithbaluch.com, laughton@smithbaluch.com, and baluch@smithbaluch.com.

GROUND FOR STANDING

Petitioner hereby certifies that the '103 patent is available for *inter partes* review, and that Petitioner is not barred or estopped from requesting an *inter partes* review on the grounds identified in the petition.

THRESHOLD REQUIREMENT FOR INTER PARTES REVIEW

This petition presents “a reasonable likelihood that the Petitioners would prevail with respect to at least one of the claims challenged in the petition,” 35 U.S.C. § 314(a), as shown in the Grounds below.

STATEMENT OF PRECISE RELIEF REQUESTED

Petitioner respectfully requests that claims 1-24 of the '103 patent be canceled based on the following grounds:

Ground 1: Claims 1-10 and 13-22 are obvious over Hedinsson in view of Ford.

Ground 2: Claims 11-12 and 23-24 are obvious over Hedinsson in view of Ford and Kostello.

I. INTRODUCTION

A. Summary of the '103 Patent

The claims of the '103 patent recite pseudocode that combines two known concepts: (1) a **programming loop** and (2) a **composite virtual clip** that includes separate **virtual clips** (e.g., video clips) that are each identified using a **pointer** or **link**. (Ex-1002, ¶¶27-29). The claimed programming loop executes once and

repeats **until** a *counting* variable (**M** or **P**, which is initially set to one) equals the *number* of times the loop should execute (**N**). This was known as a do-until loop, and is shown in Ford. (Ex-1002, ¶27). The claims also use pointers or links to identify the virtual clips in a larger composition, but those aspects of the claim were also known in the art, as shown by Hedinson. (Ex-1002, ¶27).

The Specification provides little explanation of the claimed subject matter. It includes relevant flow charts, but the disclosure largely mirrors the claim language. (See Ex-1001, 11:1-14:29, Figs. 6-13).

B. Overview of Prosecution History

The '103 patent enjoyed a short and uneventful prosecution history. Only one office action was mailed, and it included no substantive rejections. (Ex-1007, 63-69). The applicants made minor clarifying revisions to the claims and Specification (Ex-1007, 41-59), and the Examiner allowed all claims (Ex-1007, 18-24).

The '103 patent claims priority to earlier-filed applications, but those applications received a similarly cursory examination. None of that prosecution history is germane to this proceeding.

II. CLAIM CONSTRUCTION

Petitioner does not believe that the Board needs to construe the claims in order to apply the prior art herein. Petitioner reserves the right to propose constructions or contend that the patent claims are indefinite in the co-pending litigation.

III. DETAILED EXPLANATION OF THE REASONS FOR UNPATENTABILITY

Ground 1. Claims 1-10 and 13-22 are obvious over Hedinsson in view of Ford

Claims 1-10 and 13-22 would have been obvious under pre-AIA or AIA 35 U.S.C. § 103 over U.S. Patent No. 8,793,282 B2 (“**Hedinsson**”) (Ex-1004) in view of “Programming for the Absolute Beginner” by Jerry Lee Ford, Jr. (“**Ford**”) (Ex-1005).

Both Hedinsson and Ford are prior art. Hedinsson is a U.S. patent filed on April 14, 2009, and issued on July 29, 2014. (Ex-1004). Ford is a book published in 2007. (Ex-1005). Ford is a printed publication that was publicly accessible before the critical date: it has conventional markers of publication in 2007 (Ex-1005, 3), it was catalogued and shelved in a library by June 2, 2007 (Ex-1009, ¶¶8-9), and a PHOSITA could have located it with reasonable diligence (Ex-1002, ¶37).

Neither Hedinsson nor Ford was cited during prosecution of any application leading to the ’103 patent. (Ex-1002, ¶38).

A. Overview of the Ground

Hedinsson describes a **composite virtual clip** that includes separate **virtual clips** (video clips) identified via **pointers** or **links**. Hedinsson’s “media presentation” includes a plurality of video clips that are identified by corresponding “metadata clips.” A PHOSITA would have been motivated to implement Hedinsson using well-known programming concepts, including **programming loops**, which

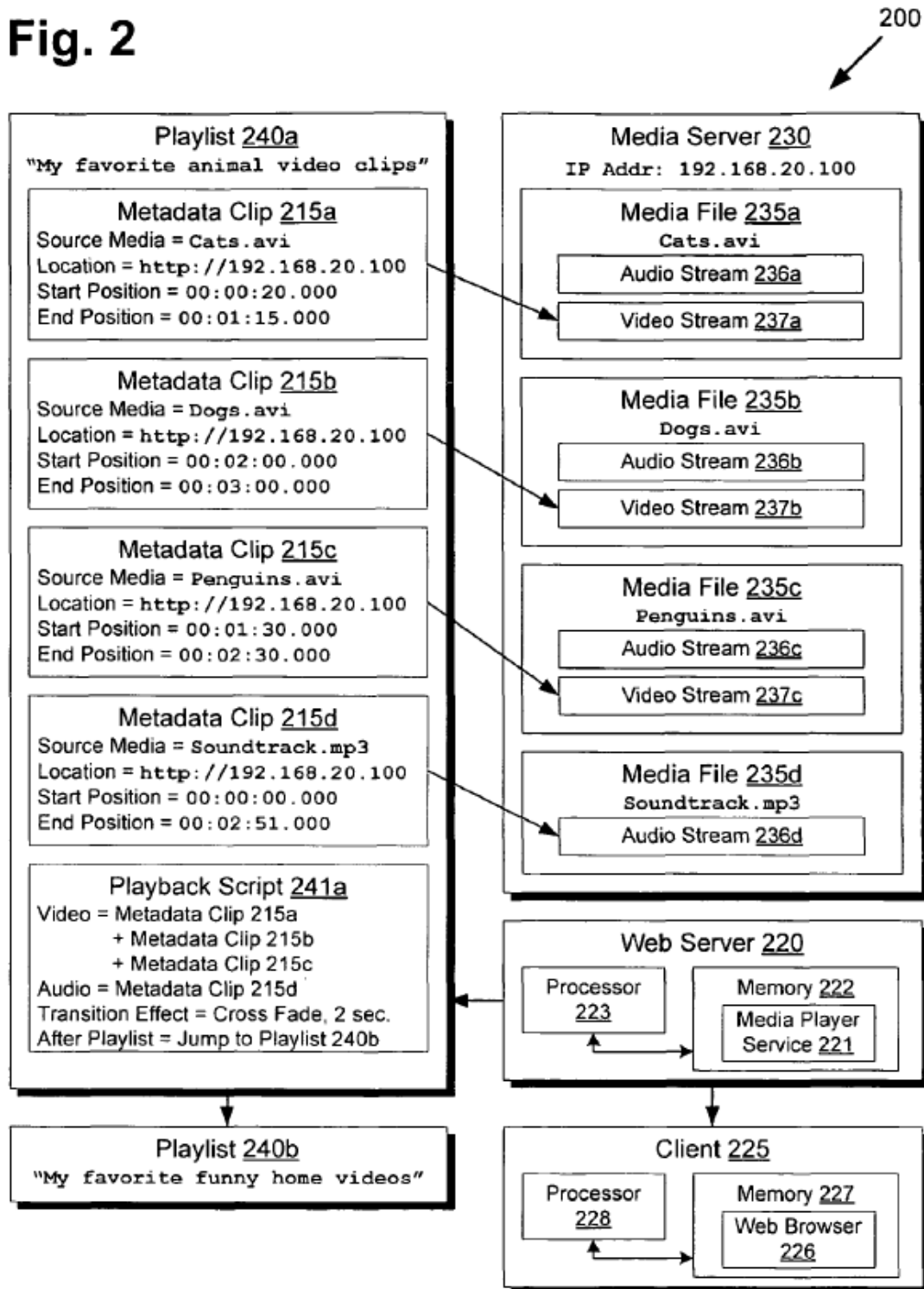
are described in Ford. Specifically, a PHOSITA would have been motivated to make or display Hedinsson's composite virtual clip by stepping through each of the video clips using a do-until loop, and the resulting combination renders obvious the challenged claims.

1. Overview of Hedinsson

Hedinsson streams selected portions of different source media files as “one continuous sequential media stream” using a “playlist” that contains “metadata clips.” (Ex-1004, 5:37-47, Abstract). Each metadata clip “only provide[s] a reference to the actual media file[,]” which allows Hedinsson's system to combine media file segments without generating a new media file. (Ex-1004, 4:11-14, 4:19-31, 4:59-67).

Figure 2 (below) illustrates the system.

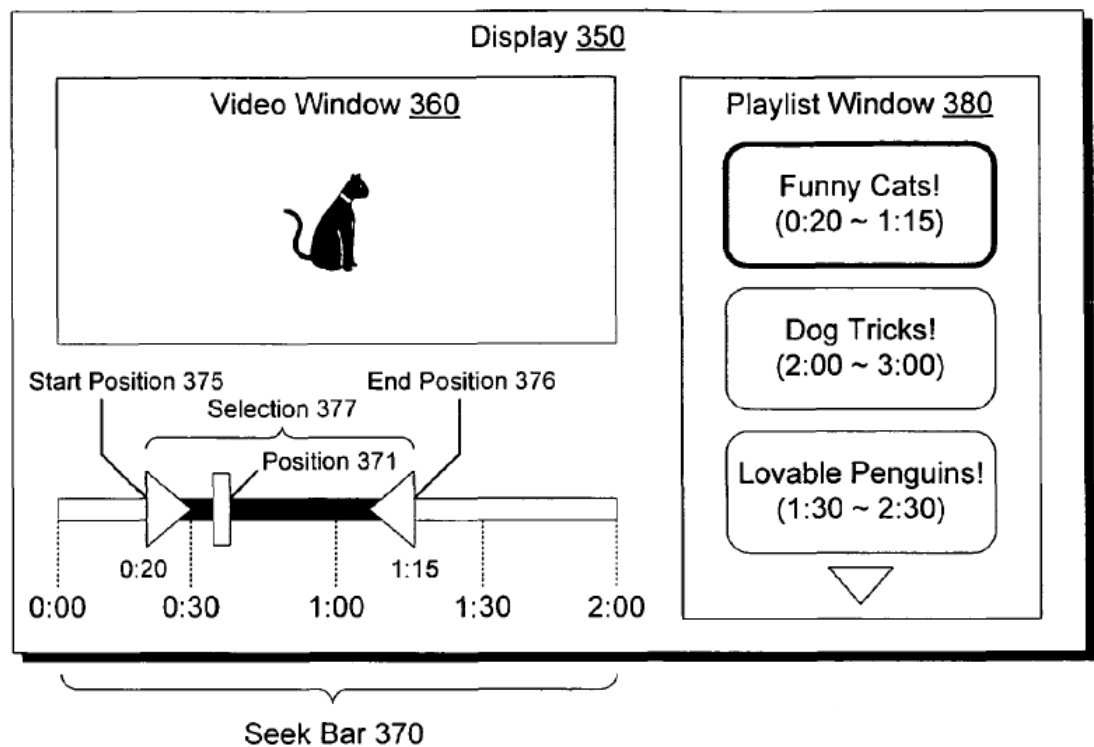
Fig. 2



A metadata clip identifies a specific excerpt of a media stream using: (1) a filename, (2) "a uniform resource locator (URL) pointing to media server 130a via

an Internet Protocol (IP) address,” and (3) a start position and an end position. (Ex-1004, 4:33-42, 4:49-55, Fig. 2). In operation, Hedinsson uses playlist 240a to sequentially stream “the referenced video from metadata clips 215a, 215b, and 215c, in order, accompanied by the referenced audio from metadata clip 215d” to client 225 for playback on display 350. (Ex-1004, 5:37-47, 9:37-67).

An example of such playback is shown in Figure 3, which includes an exemplary “simplified player interface” (below) for displaying the presentation. (Ex-1004, 7:39-41).



Video window 360 (top left) displays a video, and playlist window 380 (right) shows “a sequential listing of metadata clips contained in the currently playing playlist, corresponding to playlist 240a of FIG. 2.” (Ex-1004, 7:41-50).

Hedinsson explains that “the player interface depicted in display 350” can support both “playlist playback” and “the creation and editing of playlists and metadata clips.” (Ex-1004, 8:14-19). Hedinsson specifically states that a user can: (1) create a playlist using new or existing clips, (2) add a clip to a playlist, (3) adjust the start and/or end positions of a clip in a playlist, and (4) remove a clip from a playlist. (Ex-1004, 8:63-9:34, 8:34-45, Abstract).

2. Overview of Ford

Ford describes “fundamentals of computer programming” that are applicable to “all programming languages.” (Ex-1005, 11).²

Ford describes “loops,” which “are code blocks that repeat a series of programming statements over and over again.” (Ex-1005, 18). As Ford notes, there are several known types of loops, including “do...until” loops. (Ex-1005, 22-23). A do-until loop “executes at least once and repeats until a specified condition is true.” (Ex-1005, 22-23). Ford provides an example:

² Petitioner refers to the page of the exhibit, shown in the bottom center of the page, rather than Ford’s native pagination.

```
i = 1

do

    print i
    i = i + 1

loop until i > 10
```

(Ex-1005, 26). The variable “i” is used to track the number of loops; each iteration of the loop prints the value of “i”; and the loop repeats until “i” is greater than 10. (Ex-1005, 25-26). Ford explains that “i > 10” is the loop’s *condition*, and its *statements* are “print i” and “i = i + 1.” (Ex-1005, 25). By changing “10” to a different number, the loop can repeat any number of times “without adding a single line of code.” (Ex-1005, 22, 26; Ex-1002, ¶45).

B. Rationale (Motivation) Supporting Obviousness

As explained in the “Overview of Hedinsson,” Hedinsson uses a playlist of metadata clips to display a composite presentation that includes different video clips.³ Hedinsson shows an example playlist, displays a presentation using that playlist, and states that a user can create new playlists. (Ex-1004, 5:37-47, 9:25-67,

³ Petitioner uses “metadata clip” to refer to the data structure in Hedinsson’s playlist, and “audio clip” and “video clip” to refer to the referenced audio or video.

8:16-19, 8:63-9:24, 8:34-51, 9:32-34, Figs. 2-3). Hedinsson does not include pseudocode (or otherwise specify a similarly precise manner of operation); however, these implementation details were well within a PHOSITA's skill. (Ex-1004, 2:26-28, 8:55-57, 7:39, 8:45-51; Ex-1002, ¶46).

A PHOSITA would have been motivated to implement Hedinsson using code, and specifically would have been motivated to use programming loops both: (1) to create a new playlist that points to desired video clips and (2) to stream the video clips identified by an existing playlist. (Ex-1002, ¶47). A PHOSITA would have been motivated to use loops because these operations (i.e., creating a new playlist and streaming an existing one) both required performance of the same steps for each of the playlist's video clips. (Ex-1002, ¶47). Loops repeat a series of programming statements, and they and their benefits were well-known to PHOSITAs, as shown by Ford. (Ex-1002, ¶47; Ex-1005, 11, 18, 21-22).

Also, a PHOSITA would have been motivated to implement Hedinsson using loops because this would facilitate playlists with different numbers of metadata clips, as contemplated by Hedinsson. (Ex-1004, 3:1-6, 9:1-13, 9:25-29, 9:67-10:2, claim 2; Ex-1002, ¶48). A PHOSITA would have been motivated to add or access one metadata clip during each iteration of the loop because this would reduce repetitive code and increase the ability to accommodate different numbers of clips. (Ex-1002, ¶48).

Ford also explains that there were known types of loops. (Ex-1005, 22-23). A PHOSITA would have understood these to be a finite number of identified, predictable solutions to the problem of writing code that repeats programming statements. (Ex-1005, 22-23; Ex-1002, ¶49). Implementing each type of loop was within a PHOSITA's skill. (Ex-1002, ¶49). Also, a PHOSITA would have been motivated to use a do-until loop when implementing Hedinsson because this type of loop is executed at least once and repeats until a specified condition is true, where Hedinsson's playlist includes at least one metadata clip and looping should continue until all metadata clips have been processed. (Ex-1005, 23; Ex-1002, ¶¶49-50).

Specifically, a PHOSITA would have been motivated to create a new playlist, as contemplated by Hedinsson, using a do-until loop that adds one metadata clip (pointing to a video clip) and repeats until all desired clips have been added. (Ex-1002, ¶50). Also, a PHOSITA would have been motivated to stream Hedinsson's presentation using a do-until loop that accesses a metadata clip (to stream the linked video clip) and repeats until all clips have been streamed. (Ex-1002, ¶50).

C. Reasonable Expectation of Success

A PHOSITA would have had a reasonable expectation of success in making the proposed combination. (Ex-1002, ¶51).

The art in the relevant timeframe was predictable. (Ex-1002, ¶52). Specifically, in the relevant timeframe, the web server described by Hedinsson and

the basic programming concepts explained in Ford were well-known in the art. (Ex-1002, ¶52). Indeed, programming loops and their types were fundamental concepts taught in introductory computer programming books and classes. (Ex-1002, ¶52; Ex-1005). In addition, the '103 patent provides no elaboration beyond the requirements recited in the claims, which further supports that these steps are within the exercise of ordinary skill. (Ex-1001, 11:1-14:29).

D. Graham Factors

The **level of ordinary skill** in the art was equal to an undergraduate degree in electrical engineering, computer engineering, or computer science and two years of computer programming experience and experience with web-based digital media. (Ex-1002, ¶35). Relevant practical experience could offset less or different education and vice versa. (Ex-1002, ¶35).

The **scope and content of the prior art** are discussed throughout the Ground.

The **differences between the prior art and the claims** are discussed throughout the Ground.

Petitioner is not aware of any **secondary considerations** that would make an inference of non-obviousness more likely.

This discussion of Graham factors applies to all Grounds.

E. Analogous Art

Hedinsson is analogous art because it is in the same field as the '103 patent

(“managing information ... and more particularly ... sharing and annotating video data”), and it would have been reasonably pertinent to the video-sharing problems facing the named inventors. (Ex-1004, Title, Abstract, 1:6-53; Ex-1001, Title, Abstract, 1:18-22; Ex-1002, ¶58). Ford is analogous art because it describes basic computer programming concepts, which were known to and used by PHOSITAs. (Ex-1005, 11; Ex-1002, ¶58).

F. Claim Mapping

This section maps the challenged claims to the relevant disclosures of Hedinsson and Ford, where the claim text appears in bold-italics, and the relevant mapping follows.

CLAIM 1

“1[a]. A method to make a composite virtual clip, comprising:”

To the extent the preamble is limiting, Hedinsson **makes** a “sequential media stream” of audio and video clips from different source media files by creating a new playlist. (Ex-1004, 5:37-47, 8:16-19, 9:37-67, Abstract, Figs. 2, 4). The media stream and/or its playlist are **a composite virtual clip**. (Ex-1002, ¶60).

Hedinsson discloses that a user may create a new playlist. (Ex-1004, Abstract; Ex-1002, ¶61). “[T]he player interface depicted in display 350” can support “the creation and editing of playlists” “via user input read from input device 329.” (Ex-1004, 8:14-24, 8:40-45). Hedinsson explains that a user can “assemble playlist 340”

by creating new metadata clips or adding existing ones (Ex-1004, 9:1-24), and after generating metadata clips, the web server “determin[es] [a] playlist” (Ex-1004, 9:25-36). Also, Hedinsson states that a user may save a playlist for later access by others. (Ex-1004, 9:32-34).

Hedinsson’s playlist includes multiple metadata clips that each point to an audio or video clip. (Ex-1004, Fig. 2, 4:11-58, 5:31-47, 9:37-53). For example, playlist 240a includes one metadata clip (215d) pointing to an audio clip and three metadata clips (215a-c) pointing to video clips. (Ex-1004, Fig. 2). Also, Hedinsson teaches that a user may create a new playlist that includes multiple metadata clips. (Ex-1004, 8:63-9:24; Ex-1002, ¶62).

“[1b] selecting (N) saved virtual clips and an order of presentation for each virtual clip;”

Hedinsson expressly or inherently teaches **selecting** a particular number of saved video clips ((N) **saved virtual clips**) for inclusion in the new playlist. (Ex-1002, ¶63). As explained in element [1a], Hedinsson discloses that a user may create a new playlist that includes multiple metadata clips each pointing to a video clip. (E.g., Ex-1004, Abstract, 8:63-9:24). From this, a PHOSITA would have immediately envisaged that Hedinsson’s user (and its system) selects the video clips for this new playlist. (Ex-1002, ¶63). Also, in order for a user to create a new playlist of video clips, the video clips must necessarily first be **selected** by the user

and/or by the system. (Ex-1002, ¶63).

In either event, the number of selected video clips teaches (**N**), which can be any number. (Ex-1002, ¶64). For the creation of playlist 240a, (**N**) would equal 3, and the **virtual clips** would be the video clips referenced by metadata clips 215a-c. (Ex-1004, 5:37-47, 9:44-53, Fig. 2). Also, Hedinsson's video clips are **saved** on media server 230. (Ex-1004, 5:26-31, 2:46-50, Figs. 1-2).

Hedinsson expressly or inherently teaches **selecting an order of presentation for each virtual clip**. (Ex-1002, ¶65). As explained in element [1a], Hedinsson discloses that a user may create a new playlist to display multiple video clips. Also, Hedinsson's playlist specifies the order in which a plurality of video clips should be played: Hedinsson uses playlist 240a to stream "the referenced video from metadata clips 215a, 215b, and 215c, in order." (Ex-1004, 5:41-47, 9:37-53). From this, a PHOSITA would have immediately envisaged that the system selects the order in which the video clips should be played. (Ex-1002, ¶65). Also, in order to create a playlist that plays video clips in a given order, the system must necessarily first select an order of presentation of each video clip. (Ex-1002, ¶65).

Alternatively, element [1b] would have been obvious over Hedinsson. The above-referenced discussion of creating a playlist at least suggests element [1b]. (Ex-1002, ¶66). In addition, a PHOSITA would have been motivated to allow a user to **select** the video clips and their order of presentation when creating a new playlist,

because this would increase user-friendliness and because Hedinsson seeks to provide users with the ability to “participate in the content creation process,” “tailor[] content,” and “[r]emix[] and juxtapos[e] clips.” (Ex-1004, 1:32-39; Ex-1002, ¶66).

“[1c] initially setting (M)=1;”

Ford describes an exemplary do-until loop that begins with “i = 1.” (Ex-1005, 26). This teaches **initially setting (M)=1**, where **(M)** is taught by Ford’s “i,” because both “i” and “(M)” are arbitrary names for a variable (a counter). (Ex-1002, ¶67). As explained in the “Rationale Supporting Obviousness” and elements [1d] and [1e], a PHOSITA would have been motivated to create Hedinsson’s new playlist using Ford’s do-until loop, which counts the iterations of the loop by first setting a variable to 1. (Ex-1002, ¶67).

“[1d] configuring a (M)th pointer to indicate a storage location for a (M)th saved virtual clip, wherein a (M)th virtual clip comprises a (M)th content from one or more designated sources and (M) is greater than or equal to 1 and less than or equal to (N); saving said (M)th pointer;”

Hedinsson teaches that each metadata clip includes **a pointer to indicate a storage location** of a corresponding video clip (**saved virtual clip**) or audio clip. (Ex-1004, 4:33-64, Fig. 2; Ex-1002, ¶68). Hedinsson’s metadata clip includes (1) “information for media player service 121 to locate the actual referenced source

media file,” (2) “information for retrieving the actual referenced source media file from a network, in the form of a uniform resource locator (URL) *pointing to* media server 130a via an Internet Protocol (IP) address,” and (3) “a referenced start position” and “a referenced end position” identifying the relevant “portion of [the] media file.” (Ex-1004, 4:33-64 (emphasis added), Fig. 2). The **pointer** is taught by: (1) the URL (which includes an IP address in Hedinsson’s example) and source media filename, or (2) the URL, source media filename, and the start and end positions. (Ex-1002, ¶68; Ex-1010, ¶47; Ex-1011, ¶107).

To the extent that the term **pointer** requires more, this is rendered obvious by Hedinsson’s disclosure. A PHOSITA would have understood that a pointer was a known and predictable way to reference data. (Ex-1002, ¶69). Indeed, outside of the claims, the ’103 patent uses the term “pointer” only once, in passing, showing that a PHOSITA was familiar with pointers and would have known how to use them. (Ex-1001, 8:39-41; Ex-1002, ¶69).

Hedinsson discloses or renders obvious **saving said pointer**. (Ex-1002, ¶70). Hedinsson discloses that a user can create new metadata clips and “store the newly created metadata clips in playlist 340,” and as explained above, each metadata clip includes a **pointer**. (Ex-1004, 8:63-9:16, Fig. 4 (step 410)). Thus, a PHOSITA would have understood that Hedinsson discloses **saving** the new metadata clips (and their **pointers**) in the playlist. (Ex-1002, ¶70). In addition, a PHOSITA would have

been motivated to **save** the newly created metadata clips and the playlist because this would allow later use of those metadata clips and that playlist, as contemplated by Hedinsson. (Ex-1004, 9:14-18, 9:9-13, 9:32-34, 7:15-18, Fig. 3; Ex-1002, ¶70).

Hedinsson also discloses or renders obvious **configuring said pointer**. (Ex-1002, ¶71). A PHOSITA would have understood that creating and saving a new metadata clip (with a new **pointer**) involves **configuring the pointer**. (Ex-1002, ¶71). As Dr. Houh explains, the specific configuration steps vary by programming language, but involve adding the pointer's data (i.e., the information specifying the destination referenced by the pointer). (Ex-1002, ¶71). Also, a PHOSITA would have been motivated to **configure the pointer** before it is saved given Hedinsson's disclosure of saving metadata clips. (Ex-1002, ¶71).

Hedinsson discloses or renders obvious configuring and saving an **(M)th pointer** for an **(M)th saved virtual clip**. (Ex-1002, ¶72). Initially, in the combination, **(M)** equals 1 (element [1c]), and as explained in the preceding paragraphs, Hedinsson at least discloses or renders obvious configuring and saving a first pointer for a first saved video clip. Also, to the extent required, Hedinsson discloses or renders obvious configuring and saving a pointer for *each* of the N video clips (element [1b]) because Hedinsson seeks to create a new playlist that includes a set of new and/or existing metadata clips. (Ex-1002, ¶72). As noted in element [1b], N can be any number.

Also, Hedinsson teaches that each video clip **((M)th virtual clip) comprises** corresponding content **((M)th content) from** a particular media file 235 in a particular media server 230 **(one or more designated sources)**. (Ex-1004, 9:37-67, 4:11-22, 5:26-36, 5:58-67, Fig. 2; Ex-1002, ¶73). In addition, in the combination, **(M) is greater than or equal to 1 and less than or equal to (N)**. As discussed in elements [1b] and [1e], M initially equals 1, and M is incremented until N video clips are added to the new playlist. (Ex-1002, ¶73).

“[1e] determining if (M) equals (N); when (M) is less than (N), setting (M) equal to (M)+1; and repeating said configuring, saving, and determining until (M) equals (N).”

As explained in elements [1a], [1b], and [1d], Hedinsson discloses or renders obvious creating a new playlist that includes N metadata clips pointing to corresponding video clips. (E.g., Ex-1004, Abstract, 8:16-19, 8:63-9:24). As explained in the section entitled “Rationale Supporting Obviousness,” a PHOSITA would have been motivated to create this new playlist using a do-until loop, as taught by Ford. (Ex-1002, ¶74). In particular, a PHOSITA would have been motivated to use a do-until loop that configures and saves one metadata clip (including its pointer) during each iteration and repeats until all N metadata clips have been configured and saved, so that the playlist includes pointers to each of the N selected video clips

(from element [1b]).⁴ (Ex-1002, ¶74).

This proposed Hedinsson-Ford combination discloses or renders obvious **determining if (M) equals (N)**. (Ex-1002, ¶76). Ford’s do-until loop “evaluate[s]” whether a *condition* is true (**determining if**) (Ex-1005, 25-26), and in the proposed Hedinsson-Ford combination, a PHOSITA would have been motivated to use **(M) equals (N)** as the *condition*. (Ex-1002, ¶76). Ford initially sets “i” to 1, increments “i” in the loop, and repeats the loop until “i” exceeds a particular value. (Ex-1005, 21-22, 26). As explained in element [1c], Ford’s “i” teaches **(M)**—both count the number of loop iterations. Based on this teaching, a PHOSITA would have been

⁴ As Dr. Houh explains, it is unclear whether the claim requires saving N or N-1 pointers. (Ex-1002, ¶75). The Board need not address this ambiguity because, under either interpretation, the claim is rendered obvious by Hedinsson-Ford. In particular, if the latter interpretation (N-1) applies, the first N-1 iterations of the Hedinsson-Ford loop render claim 1 obvious because the claim does not prohibit additional loop iterations. *See Crystal Semiconductor Corp. v. TriTech Microelectronics Int’l, Inc.*, 246 F.3d 1336, 1348 (Fed. Cir. 2001) (There is a presumption that a “comprising” claim “does not exclude additional, unrecited elements.”); *see also Intel Corp. v. Qualcomm Inc.*, 21 F.4th 801, 813 (Fed. Cir. 2021) (the Board can rule on patentability despite indefiniteness).

motivated to use, as the loop's condition, a comparison of M with a number indicative of the number of times the loop should repeat. (Ex-1002, ¶76). In other words, a PHOSITA would have been motivated to use **(M) equals (N)** as the condition because M is used to count the number of loops, M begins at 1, and the loop should repeat N times. (Ex-1002, ¶76).

Ford teaches **setting (M) equal to (M)+1** because Ford increments “i” in the loop (i.e., “i = i + 1”). (Ex-1005, 26). Also, the combination discloses or renders obvious setting (M) equal to (M)+1 **when (M) is less than (N)**. (Ex-1002, ¶77). Because the loop increments the value of M during *each* iteration, M will be incremented whether or not this condition is satisfied (and, thus, will be incremented when this condition is satisfied). Moreover, if required by the claim, a PHOSITA would also have been motivated to increment M *only when* M is less than N to avoid an unnecessary step (i.e., incrementing M even after the loop has been executed the desired number of times). (Ex-1002, ¶77).

The Hedinsson-Ford combination discloses or renders obvious **repeating said configuring, saving, and determining until (M) equals (N)**. (Ex-1002, ¶78). Ford's “*statements* ... are executed each time the loop repeats,” and the “*condition* ... is evaluated at the end of each iteration of the [do-until] loop.” (Ex-1005, 26). The do-until loop “repeats until [the] specified condition is true.” (Ex-1005, 23). Thus, Ford teaches **repeating** both (1) the *statements* inside the loop and (2) the

determining of the *condition*, **until** the *condition* is true. As explained above, a PHOSITA would have been motivated to use **(M) equals (N)** as the *condition*. Also, a PHOSITA would have been motivated to include the **configuring** and **saving** steps (i.e., element [1d]) in the loop's *statements* so that each metadata clip (and its pointer) is configured and saved in the playlist. (Ex-1002, ¶78). In particular, as explained in element [1d], a pointer corresponding to a given metadata clip is **configured** and **saved** when adding the metadata clip to the playlist, and a PHOSITA would have been motivated to repeat this for each metadata clip to be added to the playlist so that the playlist includes all of the desired information. (Ex-1004, 5:37-47, 9:37-67; Ex-1002, ¶78).

Also, to the extent required by the claim, the combination discloses or renders obvious element [1d] upon **repeating** because, as explained in element [1d], Hedinsson discloses or renders obvious performing the steps recited in element [1d] for *each* of N video clips. (Ex-1002, ¶79).

Finally, to the extent that the combination of elements [1d] and [1e] requires the pointers to be saved in sequential order, this is disclosed or rendered obvious by Hedinsson. (Ex-1002, ¶80). Hedinsson discloses processing the video clips in sequential order and depicts playlist 240a in the same order. (Ex-1004, 9:37-67, Fig. 2). Although Hedinsson's playback script 241a would allow the video clips to be easily reordered after playlist creation (*see* Ex-1004, 5:37-47), a PHOSITA would

have understood Hedinsson to create its playlist 240a with metadata clips 215 in the original sequential order (Ex-1002, ¶80). Also, this ordering would have been obvious over Hedinsson because a PHOSITA would have been motivated to sequentially store the metadata clips in the playlist when originally creating the playlist. (Ex-1002, ¶80). Using this ordering would have been a very straightforward way to write the code, and it would have been one of a finite number of identified, predictable solutions to the problem of storing data that were well within the skill of a PHOSITA. (Ex-1002, ¶80).

CLAIM 2

“2[a]. The method of claim 1, wherein said composite virtual clip further comprises one or more first type virtual clips, wherein the one or more first type virtual clips comprise content from a Media File, playable or static, from a beginning to a first designated end point,”

See claim 1. As explained in element [1a], Hedinsson makes a **composite virtual clip** that **comprises** audio and video clips (either of which, in the alternative, teaches the **one or more first type virtual clips**). (Ex-1002, ¶¶81-82). The audio and video clips each **comprise content from a Media File** (one of the media files 235a-d) **playable from a beginning** (the metadata clip’s start position) **to a first designated end point** (the metadata clip’s end position). (Ex-1004, 4:11-22, 4:52-58, 9:37-53, Figs. 1-2).

For example, playlist 240a includes metadata clip 215d, which causes the

system to play an audio clip—“Soundtrack.mp3” from “00:00:00” to “00:02:51.” (Ex-1004, Fig. 2, 9:43-44, 5:46-47). Also, playlist 240a includes metadata clip 215a, which causes the system to play a video clip—“Cats.avi” from “00:00:20” to “00:01:15.” (Ex-1004, Fig. 2, 9:44-48).

“[2b] wherein the Media File, playable or static, is selected from the group consisting of AVI file, MP3 file, MP4 file, WMA file, WAV file, FLASH, MPEG file, an image file, a PDF file, a text file, a VISIO file, a .ppt file, a .key file, a spreadsheet file, 3D media file which provides a Stereoscopic visual display that can be viewed through stereoscopic eyewear or played on 3D display technology, and a Virtual Reality/Augmented Reality file that can be viewed through Virtual Reality devices.”

Hedinsson discloses element [2b] under either alternative mapping of the **first type virtual clip** (*i.e.*, audio or video). Hedinsson’s audio clip comprises content from an MP3 file (Ex-1004, 9:43-44, Fig. 2), and Hedinsson’s video clip comprises content from an AVI file (Ex-1004, 9:46-48, Fig. 2). Any of the alternatives recited in the claimed Markush group is sufficient to satisfy the claim language. *See Abbott Lab’ys v. Baxter Pharm. Prods., Inc.*, 334 F.3d 1274, 1280 (Fed. Cir. 2003).

CLAIM 3

Petitioner reorders the elements of this claim to address the limitations pertaining to a **first interactable graphical object** (elements [3b] and [3d]) before addressing a **second interactable graphical object** (elements [3c] and [3e]).

“3[a]. The method of claim 2, further comprising:”

See claim 2.

“[3b] displaying a first interactable graphical object, wherein the first interactable graphical object comprises a Cancel Clip link;”

“[3d] when activating the first interactable graphical object, a begin time location, end time location, and first media duration of the first type virtual clip are not stored; and”

As discussed in claim 2, either the video clip(s) or audio clip(s) teach the **first type virtual clip**. Petitioner addresses each alternative separately.

Video Alternative:

When the “first type virtual clip” is a video clip, Hedinsson discloses or renders obvious these elements.

First, Hedinsson teaches that a user may edit playlists using display 350. (Ex-1004, 7:39-41, 8:16-19, Fig. 3). Specifically, “items from playlist window 380 might be clicked and dragged into a trashcan icon using input device 329 to remove items from playlist.” (Ex-1004, 8:42-45). The trashcan icon is the **first interactable graphical object** that **comprises a Cancel Clip link**, and when it is **activated** (by a user dragging a video clip from playlist window 380 to the trashcan icon), the corresponding video clip is removed from the playlist. As a result, the clip’s **begin time location, end time location, and first media duration are not stored**. (Ex-1002, ¶88).

Second, it would have been obvious to add buttons to Hedinsson's display 350 that allow a user to "save" or "cancel" a particular clip, given Hedinsson's disclosure and the knowledge of a PHOSITA. (Ex-1002, ¶¶89-92).

Hedinsson teaches that a user can "generat[e] ... metadata clips" from "media files" by setting "start position 375 and end position 376." (Ex-1004, 8:63-9:9, 8:34-40, Fig. 3). Hedinsson states that "the newly created metadata clips" can be "store[d]" in the playlist, and Hedinsson indicates that those clips can be saved for future use by others. (Ex-1004, 9:9-24). Hedinsson's "simplified player interface example" in Figure 3 does not show how a user would specify whether to save a new clip (or not), but Hedinsson notes that "other editing and creation facilities" may be added. (Ex-1004, 7:39, 8:45-48).

Given Hedinsson's disclosure, a PHOSITA would have been motivated to add a save-clip button and a cancel-clip button to the display: the former would save a user's edits to a particular video clip, and the latter would discard any edits. (Ex-1002, ¶¶90-91). A PHOSITA would have known that user interfaces frequently provided such buttons to allow a user to specify whether to save or discard changes. (Ex-1002, ¶91; Ex-1012, ¶34, Fig. 2). In addition, adding these buttons to a user interface was a predictable design choice within the skill of a PHOSITA, and a PHOSITA would have known that it would increase the user-friendliness of Hedinsson's display. (Ex-1002, ¶91).

Thus, Hedinsson renders obvious a **first interactable graphical object** (the cancel-clip button) that **comprises a Cancel Clip link** that, when **activated** by user input, does **not store the begin time location, end time location, and first media duration** of the video clip. (Ex-1002, ¶92).

Audio Alternative:

Although not specifically shown in Figure 3, given Hedinsson's disclosure and the knowledge of a PHOSITA, it would have been obvious to implement Hedinsson so that audio clip(s) are created, added, changed, and/or removed using the same (or substantially the same) **interactable graphical objects** discussed in the video alternative for claim 3.

Hedinsson suggests that the same (or substantially the same) **interactable graphical objects** should be used for both video and audio clips. (Ex-1002, ¶94). Hedinsson uses a video clip to illustrate the functionality, but Hedinsson refers to creating, adding, changing, and removing a playlist's "metadata clips" generally, and some metadata clips point to audio clips. (Ex-1004, 8:16-51, 8:63-9:24, 5:46-47, 4:59-64, Fig. 2 (metadata clip 215d)). Also, Hedinsson indicates that a user can select which audio clip should be included in a playlist. (Ex-1004, 1:34-37, 5:26-31, 3:34-38, 3:44-48, 5:46-57, Fig. 2; Ex-1002, ¶94). In fact, although Hedinsson's example playlist identifies only one audio clip ("Soundtrack.mp3"), Hedinsson states that "[m]ultiple audio tracks might be specified" in a playlist. (Ex-1004, 5:54-

57). In addition, Hedinsson recognizes that its display 350 is a “simplified player interface example,” and it suggests using additional “interface elements” to support “referencing outside audio sources.” (Ex-1004, 7:39, 8:45-48).

A PHOSITA would have been motivated to implement Hedinsson in a manner that allows a user to create, add, change, and remove audio clip(s) using the same (or substantially similar) graphical objects discussed in the video alternative to provide the user with the flexibility to define a playlist’s audio clip. (Ex-1002, ¶¶94-95). Also, using the same (or substantially similar) graphical objects for both the video and audio clips would increase user-friendliness. (Ex-1002, ¶95).

As a result, when the “first type virtual clip” is an audio clip, Hedinsson renders elements [3b] and [3d] obvious.

First, given Hedinsson’s disclosure of dragging metadata clips to a trashcan icon to remove them from the playlist (Ex-1004, 8:42-45), it would have been obvious to use the same process to remove an audio clip from the playlist. That trashcan icon is the **first interactable graphical object** that **comprises a Cancel Clip link**, and when it is **activated** by dragging an audio clip to the trashcan, the **begin time location, end time location, and first media duration** of the corresponding audio clip are deleted from the playlist and **are not stored**. (Ex-1002, ¶97).

Second, for the reasons explained above with respect to video clips,

Hedinsson's disclosure would have motivated a PHOSITA to add buttons that allow a user to either "save" or "cancel" a new audio clip. (Ex-1002, ¶98). Thus, Hedinsson renders obvious a **first interactable graphical object** (cancel-clip button) that **comprises a Cancel Clip link** that, when **activated** by user input, does **not store the begin time location, end time location, and first media duration** of the audio clip. (Ex-1002, ¶98).

"[3c] simultaneously displaying a second interactable graphical object, wherein the second interactable graphical object comprises an End Clip link;"

"[3e] when activating the second interactable graphical object, the begin time location, end time location, and first media during of the first type virtual clip are stored."

Video Alternative:

When the "first type virtual clip" is a video clip, Hedinsson teaches or renders obvious these elements.

First, Hedinsson teaches that a user can add a clip to a playlist using display 350. (Ex-1004, 7:39-41, 8:16-19, Fig. 3). "Video window 360 might be clicked and dragged into playlist window 380 using input device 329 to add the presently playing metadata clip to playlist 340." (Ex-1004, 8:40-42). The playlist window 380 is the **second interactable graphical object** that **comprises an End Clip link**; when it is **activated** (by a user dragging a video clip from video window

360 to playlist window 380), a metadata clip associated with the video clip is stored in the playlist. As a result, the video clip's **begin time location** and **end time location are stored** in the playlist. (Ex-1002, ¶100).

Second, as explained in connection with the second video alternative for elements [3b] and [3d], a PHOSITA would have been motivated to implement Hedinsson's display 350 to include save-clip and cancel-clip buttons. (Ex-1002, ¶101). For those reasons, Hedinsson renders obvious a **second interactable graphical object** (the save-clip button) that, **when activated** by user input, **stores** the **begin time location** and the **end time location** of the video clip. (Ex-1002, ¶101).

Audio Alternative:

When the "second type virtual clip" is an audio clip, Hedinsson renders these elements obvious because, as explained in connection with the audio alternative for elements [3b] and [3d], a PHOSITA would have been motivated to implement Hedinsson so that audio clip(s) are created, added, changed, and/or removed using the same (or substantially the same) **interactable graphical objects** discussed in the video alternative. (Ex-1002, ¶102).

First, given Hedinsson's disclosure of dragging clips to playlist window 380 to add them to the playlist (Ex-1004, 8:40-42), it would have been obvious to similarly use a playlist window to add an audio clip to the playlist. That playlist

window teaches the **second interactable graphical object** that **comprises an End Clip link**, and when **activated** by dragging an audio clip to it, the **begin time location** and **end time location** of the audio clip **are stored** in the playlist. (Ex-1002, ¶103).

Second, for the reasons explained in elements [3b] and [3d] with respect to video clips, Hedinsson's disclosure would have motivated a PHOSITA to add buttons that allow a user to either "save" or "cancel" a new audio clip. (Ex-1002, ¶104). Thus, Hedinsson renders obvious **a second interactable graphical object** (save-clip button) that, **when activated** by user input, **stores** the **begin time location** and the **end time location** of an audio clip. (Ex-1002, ¶104).

Analysis for Both Alternatives:

In each alternative, the **second interactable graphical object** (i.e., the playlist window or save-clip button) is **displayed simultaneously** with the first interactable graphical object (i.e., the trashcan icon or cancel-clip button) because these graphical objects are shown at the same time in the browser window that includes display 350 (Ex-1004, Fig. 3), and because a PHOSITA would have been motivated to display these objects at the same time in the proposed combination. (Ex-1002, ¶105). A PHOSITA would have been motivated to show a consistent user interface for creating, adding, changing, and removing video and audio clips to increase ease of use. (Ex-1002, ¶105). In addition, the decision to show these

objects simultaneously (or to instead display them on different screens) was a standard design choice with well-understood and predictable advantages and disadvantages. (Ex-1002, ¶105). Moreover, a PHOSITA would have been specifically motivated to simultaneously display the objects that allow a user to select whether to add or remove particular information, in order to reduce user confusion and frustration. (Ex-1002, ¶105; Ex-1012, Fig. 2).

Finally, a PHOSITA would have found the phrase “and first media during” in element [3d] to be unclear. (Ex-1002, ¶106). If the claim is interpreted to require storage of the clip’s *duration*, this would have been an obvious variation of Hedinsson. (Ex-1002, ¶106). Hedinsson states that, instead of an end position, metadata clip may “specify a length or duration of playback, from which the referenced end position can be derived.” (Ex-1004, 4:55-58). Storing both an end position and duration would have been a trivial variation of Hedinsson’s teachings. (Ex-1002, ¶106). A PHOSITA would have been motivated to store both pieces of data in an implementation that showed the total length of the composite presentation in display 350 because it would reduce computational steps required to display both the total duration and the end position of each clip. (Ex-1002, ¶106). In addition, a PHOSITA would have been motivated to store both pieces of data in the metadata clip because it required a minimal amount of additional storage and increased the compatibility of the metadata clip’s format with other (or future) systems. (Ex-1002,

¶106).

CLAIM 4

Dependent claim 4 is substantially the same as claim 2, except: (1) it recites a **second type virtual clip** (rather than “a first type virtual clip”), (2) it recites a **designated beginning** (rather than “a beginning”), and (3) it recites a **second designated end point** (rather than “a designated endpoint”). (Ex-1013, 2). Claim 4 is obvious for the reasons provided for independent claim 1 and dependent claim 2. (Ex-1002, ¶107).

As explained in element [2a], Hedinsson’s **composite virtual clip** comprises audio and/or video clips (either of which teaches the **one or more second type virtual clips**). To the extent the “first type virtual clip” (recited in claim 2) must be different than the **second type virtual clip** (recited in this claim), these elements are taught by the audio and video clips, respectively, or vice versa.

To the extent the **designated beginning** must specify a particular starting location within the media file, Hedinsson teaches this for both video and audio clips. (Ex-1004, Fig. 2, 9:48-51, 6:37-41 (start position of audio stream 236d is 00:02:51); Ex-1002, ¶109). Indeed, as explained in elements [3c] and [3e], Hedinsson teaches that a user can create new metadata clips and can select the start and end positions for those clips. (*E.g.*, Ex-1004, 9:1-13). Also, to the extent not disclosed in Hedinsson, a PHOSITA would have been motivated by Hedinsson’s disclosure to

provide a user with the flexibility to specify a particular beginning for the selected audio clip(s). (Ex-1004, Fig. 2, 4:22-31, 1:32-37, 4:59-64; Ex-1002, ¶109). *See* elements [3b] and [3d].

CLAIM 5

Dependent claim 5 is substantially the same as claim 3, except: (1) it depends from claim 4 rather than claim 2, and (2) it recites **the second type virtual clip** (introduced in claim 4) rather than the first type virtual clip (introduced in claim 2). (Ex-1013, 2). Claim 5 is obvious for the reasons provided for independent claim 1 and dependent claims 3 and 4. (Ex-1002, ¶110).

CLAIM 6

“6. The method of claim 1, wherein said composite virtual clip further comprises one or more first type virtual clips and one or more second type virtual clips.”

See claim 1 and elements [2a] and [4a]. The video clips are the **first type virtual clips**, and the audio clips are the **second type virtual clips**, or vice versa.

CLAIM 7

“7[a]. A method to display a composite virtual clip, comprising:”

To the extent the preamble is limiting, Hedinsson **displays** a “sequential media stream” of video clips from different source media files using a playlist, e.g., playlist 240a. (Ex-1004, 5:37-47, 9:37-67, 8:16-19, Figs. 2-4). As explained in element [1a], Hedinsson teaches that its playlist includes multiple metadata clips that

each point to a specific audio or video clip. (E.g., Ex-1004, Fig. 2, 4:11-58, 5:31-47). The media stream and/or its playlist are **a composite virtual clip**.

“[7b] providing a storage location for a composite virtual clip stored in a composite virtual clip file, wherein the composite virtual clip is configured to access (N) saved virtual clips;”

Hedinsson discloses **providing a storage location for** the playlist (**a composite virtual clip stored in a composite virtual clip file**). (Ex-1002, ¶113). Hedinsson’s “[e]nvironment 200 includes web server 220, client 225, playlists 240a-240b, and media server,” and a PHOSITA would understand Hedinsson’s playlist to be stored. (Ex-1004, 5:21-23, 7:15-17; Ex-1002, ¶113). Also, Hedinsson discloses that a user can create metadata clips and “store the newly created metadata clips in playlist 340,” which also indicates that the playlist is stored. (Ex-1004, 8:63-9:13, Fig. 4 (step 410)). In addition, given the playlist’s structure, function, and contents, a PHOSITA would understand Hedinsson to disclose that its playlist is stored as a **file**. (Ex-1004, Figs. 2-3; Ex-1002, ¶113).

Alternatively, it would have been obvious to **store** Hedinsson’s playlist as a **file** because Hedinsson states that the playlist renders a new “*media* file” unnecessary, suggesting that the playlist is a different type of file. (Ex-1004, 4:27-31, 5:63-67; Ex-1002, ¶114). Also, Hedinsson indicates that the playlist can later be retrieved by the user or others, which would encourage a PHOSITA to use a file to store the playlist. (Ex-1004, 9:32-34; Ex-1002, ¶114). Moreover, storing the playlist

in a file would have been an obvious implementation of Hedinsson because a file is a basic unit of storage, would have simplified access to the playlist, and was well-known and predictable. (Ex-1002, ¶114).

Hedinsson discloses or renders obvious that **the composite virtual clip is configured to access (N) saved virtual clips**. (Ex-1002, ¶115). As explained in elements [1a] and [1b], Hedinsson's playlist includes multiple metadata clips that are used to sequentially stream a number of saved video clips, **(N)**, which can be any number. For example, for playlist 240a, **(N)** equals 3, and the **virtual clips** are the video clips referenced by metadata clips 215a-c. (Ex-1004, 5:37-47, 9:37-53, Fig. 2). Also, Hedinsson teaches video clips **saved** on media server 230. (Ex-1004, 5:26-31, 9:1-4, 2:46-50, Fig. 2).

Although the claim language is unclear,⁵ to the extent the claim requires that the **composite virtual clip** be different than the **composite virtual clip file**: Hedinsson's playlist (the **composite virtual clip**) is stored in a file (the **composite virtual clip file**); the playlist is configured to access the video clips; and as required by element [7d], links to those clips are stored in the playlist's file. (Ex-1002, ¶116).

⁵ The corresponding portion of the specification states: "provid[ing] a storage location for a composite virtual clip file configured to access [the] saved clips." (Ex-1001, 12:61-64).

Petitioner uses the term “playlist” to refer both to the information itself as well as the file in which it is located.

“[7c] initially setting (P)=1;”

Ford describes an exemplary do-until loop that begins with “i = 1.” (Ex-1005, 26). This teaches **initially setting (P)=1**, where **(P)** is taught by Ford’s “i,” because both “i” and “(P)” are arbitrary names for a variable (a counter). (Ex-1002, ¶117). As explained in the “Rationale Supporting Obviousness” and elements [7d] and [7e], a PHOSITA would have been motivated to display Hedinsson’s playlist using Ford’s do-until loop, which counts the iterations of the loop by first setting a variable to 1. (Ex-1002, ¶117).

“[7d] activating a (P)th link encoded in the composite virtual clip file to stream a (P)th saved virtual clip to a user’s device;”

As explained in element [1d], Hedinsson’s metadata clip points to a particular video clip (**saved virtual clip**) using a source media filename, URL, and start and end positions (individually or collectively **a link**).⁶ Also, as explained in element [7b], Hedinsson teaches or suggests that the metadata clip (including the **link**) is

⁶ “The link may be in the form of a pointer, electronic bookmark, hyperlink, or any other type of link.” (Ex-1001, 8:39-41).

stored in the playlist's file (**encoded in the composite virtual clip file**). (E.g., Ex-1004, 8:63-9:13; Ex-1002, ¶118).

Hedinsson expressly or inherently teaches **activating a link** in a metadata clip **to stream a saved virtual clip** (the corresponding video clip) **to a user's device** (client 225 for playback on display 350). (Ex-1004, 9:37-67, 5:37-47, 2:51-52, 4:28-31; Ex-1002, ¶119). Hedinsson uses a metadata clip's source media filename, URL (e.g., IP address), and start and end positions to "locate[]," "retrieve[]," and "stream[]" the corresponding video clip. (Ex-1004, 4:32-64, 5:37-47, 9:37-67). A PHOSITA would have understood using a link to locate, retrieve, and stream a video clip to expressly or inherently disclose **activating the link**. (Ex-1002, ¶119). Alternatively, this would have been an obvious implementation of Hedinsson given Hedinsson's disclosure and because a PHOSITA would have been motivated to activate the link to facilitate access to the video clip. (Ex-1002, ¶119).

Hedinsson discloses or renders obvious activating a **(P)th link to stream a (P)th saved virtual clip**. (Ex-1002, ¶120). Initially, **(P)** equals 1 (element [7c]), and as explained in the preceding paragraphs, Hedinsson discloses or renders obvious activating a first link (e.g., in metadata clip 215a) to stream a first saved video clip (in stream 237a). Also, to the extent required, Hedinsson discloses or renders obvious activating a link for *each* of the N video clips (element [7b]) because Hedinsson displays a playlist that includes a set of metadata clips. (Ex-1004, 9:37-

67, 4:33-64, claim 1; Ex-1002, ¶120). As noted in element [7b], N can be any number.

“[7e] determining if (P) equals (N); when (P) is less than (N), setting (P) equals (P)+1; and repeating said activating and determining until (P) equals (N).”

As explained in elements [7a], [7b], and [7d], Hedinsson displays a sequential media stream of video clips using a playlist that includes N metadata clips that each include a link to a corresponding video clip. (E.g., Ex-1004, 9:37-67). As explained in the “Rationale Supporting Obviousness,” a PHOSITA would have been motivated to use a do-until loop, taught by Ford, to display the media stream using the playlist. (Ex-1002, ¶121). In particular, a PHOSITA would have been motivated to use a do-until loop that uses one metadata clip to stream a video clip (by activating its link) during each iteration and repeats until all N metadata clips have been used, so that the media stream includes each of the N video clips (from element [7b]).⁷ (Ex-1002, ¶121).

⁷ As Dr. Houh explains, it is unclear whether the claim requires activating N or N-1 links. (Ex-1002, ¶122). The Board need not address this ambiguity because, under either interpretation, the claim is rendered obvious by Hedinsson-Ford. *See supra* n.4.

The proposed Hedinsson-Ford combination discloses or renders obvious element [7e] for substantially the same reasons as those explained for element [1e]. (Ex-1002, ¶123). Although claim 7 uses P to count the number of loops (element [7c]), where claim 1 uses M, the combination applies equally no matter which name is given to this variable. (Ex-1002, ¶123). A PHOSITA would have been motivated to use **(P) equals (N)** as the *condition* for this loop because P is used to count the number of loops, P begins at 1, and the loop should repeat N times. (Ex-1002, ¶123). In addition, a PHOSITA would have been motivated to include the **activating** step (i.e., element [7d]) in the loop's *statements* so that each video clip is streamed (by activating the link in its corresponding metadata clip). (Ex-1004, 9:37-67, 4:32-58; Ex-1002, ¶123).

Also, to the extent required by the claim, the combination discloses or renders obvious element [7d] upon **repeating** because, as explained in element [7d], Hedinsson discloses or renders obvious performing the steps recited in element [7d] for *each* of N video clips. (Ex-1002, ¶124).

Finally, to the extent that the combination of elements [7d] and [7e] requires the pointers to be activated in sequential order, this is disclosed or rendered obvious by Hedinsson. (Ex-1002, ¶125). Hedinsson processes the video clips in sequential order and depicts playlist 240a in the same order. (Ex-1004, 9:37-67, Fig. 2). Alternatively, this ordering would have been obvious in view of Hedinsson because

a PHOSITA would have been motivated to activate the links associated with the video clips in sequential order to facilitate Hedinsson's sequential play back. (Ex-1004, 5:37-47, 7:45-47, 9:44-53, 9:61-67; Ex-1002, ¶125).

CLAIMS 8-10

Dependent claims 8-10 are substantially the same as claims 2, 4, and 6, respectively, except for their dependence from independent claim 7. (Ex-1013, 3). They are obvious for the reasons set forth under independent claim 7 and dependent claims 2, 4, and 6. (Ex-1002, ¶¶126-128).

CLAIM 13

Element [13a] recites “[a] **non-transitory computer usable medium encoded with a computer program product to make a composite virtual clip and usable with programmable computer processor disposed within a controller.**” To the degree this is limiting, it is disclosed by Hedinsson. Hedinsson teaches that “media player service 321 executing on processor 323 of web server 320” creates and updates a playlist and associated metadata clips (Ex-1004, 9:9-13, 8:14-33), and that “[w]eb server 320 includes processor 323 and memory 322 storing media player service 321” (Ex-1004, 7:17-18). Thus, Hedinsson teaches a **non-transitory computer usable medium** (memory 322) **encoded with a computer program product** (media player service 321) **to make a composite virtual clip** (a new playlist and/or the corresponding media presentation, *see* element [1a]) **and**

usable with programmable computer processor disposed within a controller (processor 323 in web server 320). (*See* Ex-1004, 7:15-31). Moreover, to the extent required, Hedinsson discloses or renders obvious a non-volatile (persistent) memory for storing media player service 321, given Hedinsson's disclosure of "memory 322 storing media player service 321," a PHOSITA's knowledge that software programs are typically stored in a non-volatile memory, and a PHOSITA's desire to maintain the software even if the web server is rebooted and/or loses power. (Ex-1004, 7:17-18; Ex-1002, ¶129).

The remaining claim elements (i.e., elements [13b] - [13e]) are the same as the corresponding elements in claim 1 except that they are directed to **computer readable program code which causes the programmable computer processor** to perform the recited method steps. (Ex-1013, 4). They are obvious for the reasons set forth under claim 1. (Ex-1002, ¶130).

CLAIMS 14-18

Dependent claims 14-18 are substantially the same as dependent claims 2-6, respectively, except that they depend from claim 13 and are directed to **computer readable program code which causes the programmable computer processor** to perform the recited method steps. (Ex-1013, 5-7). These claims are rendered obvious for the reasons set forth under claims 2-6 and further under claim 13. (Ex-1002, ¶¶131-135).

CLAIM 19

Similar to element [13a], element [19a] recites “[a] **non-transitory computer usable medium encoded with a computer program product ... usable with programmable computer processor disposed within a controller.**” Similar to element [7a], element [19a] recites “**to display a composite virtual clip.**” Thus, to the extent it is limiting, Hedinsson discloses or renders obvious element [19a] for substantially the same reasons explained for elements [13a] and [7a]. (Ex-1002, ¶136). Hedinsson teaches that media player service 321 (**computer program product**) uses the playlist to stream video clips. (Ex-1004, 5:41-47, 9:37-67).

The remaining claim elements (i.e., elements [19b] - [19e]) are the same as the corresponding elements in claim 7 except that they are directed to **computer readable program code which causes the programmable computer processor** to perform the recited method steps. (Ex-1013, 7). They are obvious for the reasons set forth under claim 7. (Ex-1002, ¶137).

CLAIMS 20-22

Dependent claims 20-22 are substantially the same as dependent claims 8-10 except that they depend from claim 19 and are directed to **computer readable program code which causes the programmable computer processor** to perform the recited method steps. (Ex-1013, 8). These claims are rendered obvious for the reasons set forth under claims 8-10 and 19. (Ex-1002, ¶¶138-140).

Ground 2. Claims 11-12 and 23-24 are obvious over Hedinsson in view of Ford and Kostello

Claims 11-12 and 23-24 would have been obvious under pre-AIA or AIA 35 U.S.C. § 103 over Hedinsson and Ford as discussed under Ground 1 and incorporated herein, further in view of U.S. Patent No. 8,745,500 B1 (“**Kostello**”) (Ex-1006).

Kostello is prior art. It is a U.S. Patent filed on December 10, 2012, and issued on June 3, 2014. (Ex-1006).

Kostello was not cited during prosecution of any application leading to the ’103 patent. (Ex-1002, ¶142).

A. Overview of the Ground

Dependent claims 11, 12, 23, and 24 recite pseudocode that uses a do-until loop to recursively add **transition effect links** (activated between a pair of virtual clip links) or **effect links** (activated simultaneously with a virtual clip link) to the **composite virtual clip file**. (Ex-1002, ¶143). Hedinsson contemplates applying effects to video clips (including both image and transition effects), but it does not expressly contemplate allowing *different* effects for *different* video clips. However, a PHOSITA would have been motivated to implement Hedinsson-Ford to provide this functionality, which is disclosed in Kostello.

1. Overview of Kostello

Kostello describes the creation of a seamless composite presentation that

includes segments of different video clips. (Ex-1006, 5:57-6:3, 8:11-23). Kostello's editing interfaces allow a user to highlight segments of video clips for inclusion in the presentation (Ex-1006, 6:4-49, 8:44-48, Fig. 4) and to specify video, transition, and music effects (Ex-1006, 6:57-8:10, 8:48-53, Figs. 5-7).

In Figure 5 (below), Kostello shows an interface that allows a user to add effects to a presentation's segments. (Ex-1006, 3:28-30).

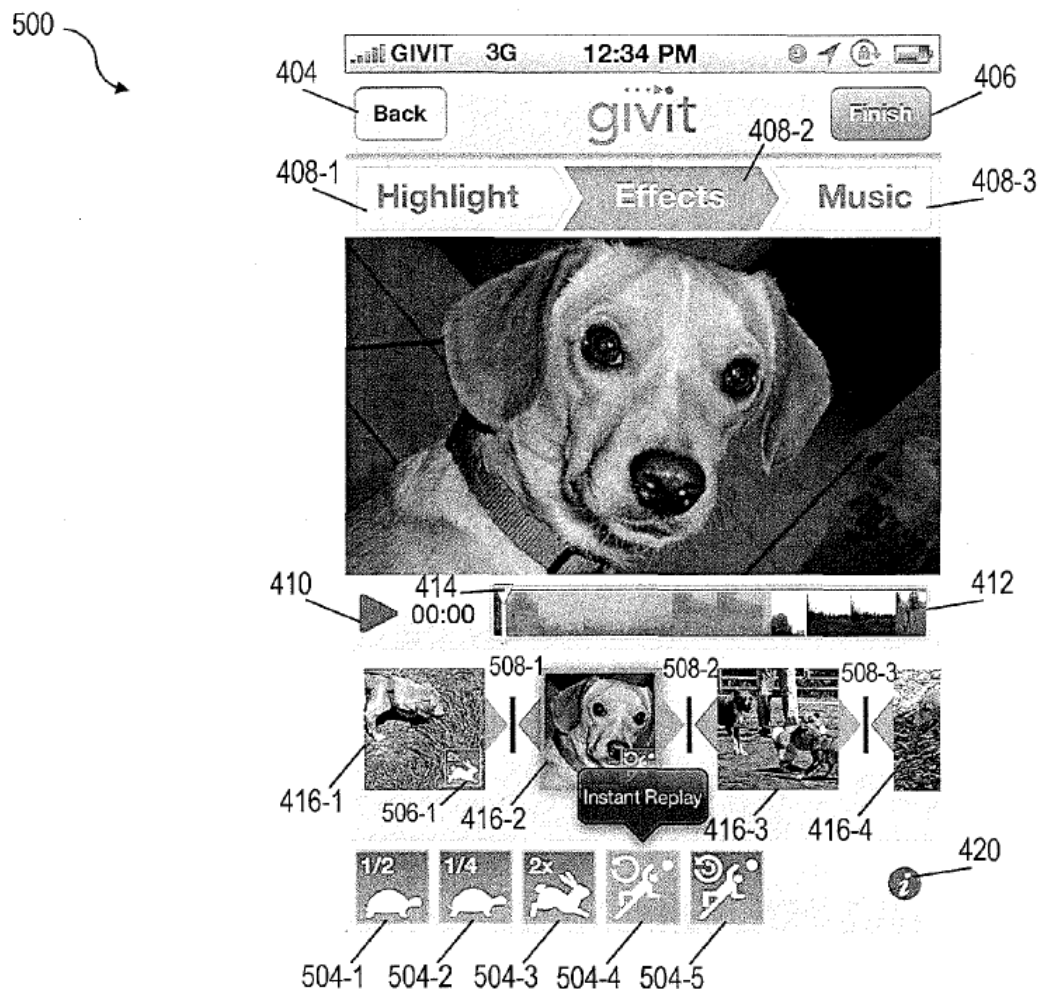
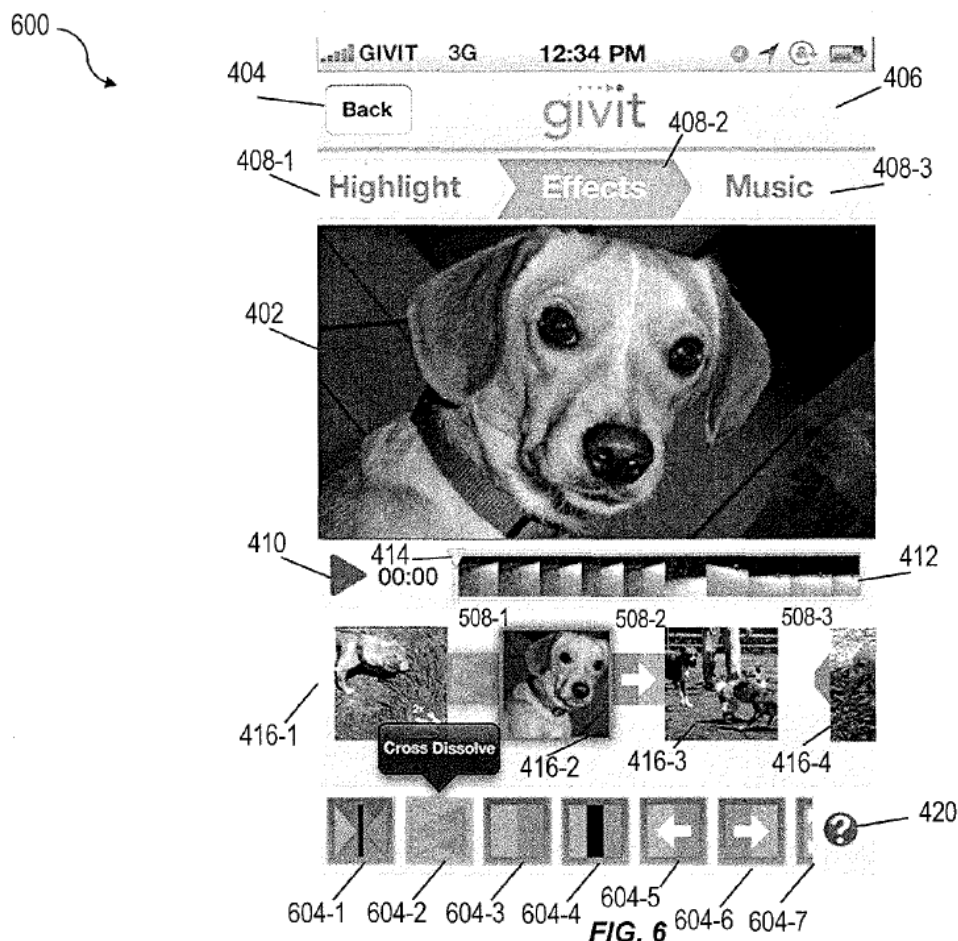


FIG. 5

A user applies an effect to a particular segment “by activating both the corresponding thumbnail 416 and a corresponding effects element 504.” (Ex-1006, 7:8-11, 6:58-

63). “For example, a first effect element 504-1 can be used for playing back the corresponding video segment at half speed.” (Ex-1006, 6:63-65). Effects can also include a “visual overlay” or “change of color of the segment.” (Ex-1006, 2:6-10, claim 9).

Figure 6 (below) shows an interface for selecting a transition effect between any two consecutive segments. (Ex-1006, 3:31-33).



Transition effects elements 604 are displayed when a user activates “a space between two thumbnails 416.” (Ex-1006, 7:15-24). Once the user selects a particular transition element 604, the presentation will display the corresponding effect when

transitioning between the video clip segments associated with the adjacent thumbnails 416. (Ex-1006, 7:22-60).

B. Rationale (Motivation) Supporting Obviousness

The rationale supporting the combination of Hedinsson and Ford is explained in Ground 1, and for the reasons explained below, a PHOSITA would have been motivated to implement Hedinsson's effects by allowing a user to select different effects for different video clips, as taught by Kostello. (Ex-1002, ¶148).

Hedinsson teaches that its playlist may include other details, such as "transition effects" between video clips and "image overlays" for video clips. (Ex-1004, 6:2-8). Hedinsson's exemplary playlist 240a specifies that "a two second cross fading effect" should be applied between video clips. (Ex-1004, 6:2-5, Fig. 2). Also, Hedinsson indicates that a playlist (singular) may specify "transition effects" and "image overlays" (plural). (Ex-1004, 6:2-5, claims 1, 12). Hedinsson's disclosure would have motivated a PHOSITA to implement Hedinsson's playlist in a way that allows a user to select effects (e.g., image and transition effects) for the video clips, and it would have motivated a PHOSITA to allow a user to select multiple image and/or transition effects in a playlist. (Ex-1002, ¶149).

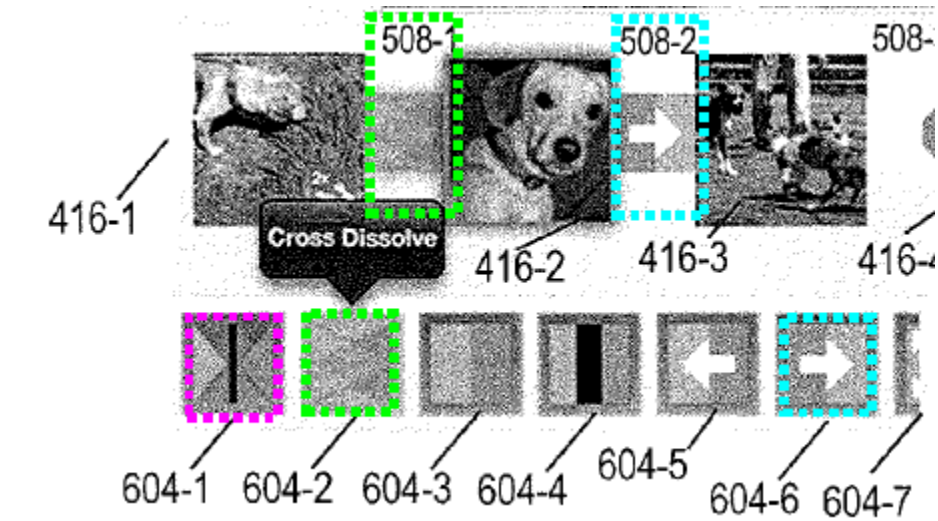
For those implementation details, a PHOSITA would have been motivated to look to Kostello, which describes a system similar to Hedinsson's system. (Ex-1002, ¶150). Kostello also allows a user to create a seamless composite presentation from

individual video clips. (Ex-1006, 5:57-6:3, 8:11-23, 8:53-56, Figs. 3-4). In fact, Kostello contemplates using a metadata file (separate from the video file) to store Kostello's enhancements, including the image and transition effects (Ex-1006, 8:11-23, 5:24-29, 8:53-56, 2:45-58).

As explained in the "Overview of Kostello," Kostello allows a user to select an image effect for each video clip (via effect element 504), and a transition effect for each transition between consecutive video clips (via transition effects element 604). (Ex-1006, 6:57-7:60, Figs. 5-6). For example, Kostello states that "effects elements 504 can correspond to different effects to be applied to one or more of the segments associated with the thumbnails." (Ex-1006, 6:61-63). Also, Kostello shows an example where the user has selected *different* transition effects between *different* video clips: for transitions 508-1 and 508-2, a user has replaced a fade to black transition (604-1) with a cross dissolve effect (604-2) and a slide right effect (604-6), respectively, as illustrated in the following annotated excerpts of Figures 5 and 6. (Ex-1002, ¶151).



Annotated Excerpt of Figure 5



Annotated Excerpt of Figure 6

A PHOSITA would have been motivated to implement Hedinsson-Ford to allow different image and transition effects for different videos, as contemplated by Kostello, because this would provide the user with additional flexibility in playlist creation. (Ex-1002, ¶152). Indeed, Hedinsson seeks to allow users to flexibly create composite presentations that incorporate creative input from users. (Ex-1004, 1:32-53, 6:49-67, 8:45-48). Also, Hedinsson recognizes the benefits of “unif[ying]” “disparate video clips” in the presentation (Ex-1004, 5:47-50), and a PHOSITA would have understood that image and transition effects were known techniques to unify disparate video clips (Ex-1002, ¶152; Ex-1008, ¶35 (“As is well known to

those of ordinary skill in the art, video transition effects are used to soften abrupt scene changes between successive video clips.”)).

In addition, a PHOSITA would have been motivated to combine this feature of Kostello with Hedinsson-Ford because a PHOSITA would have understood that existing composite video clips that included separate videos also often included different image effects for different video clips and different transition effects between videos. (Ex-1002, ¶153; Ex-1008, ¶35, Figs. 4-5). A PHOSITA would have known that users prefer to have the flexibility to select different effects and transitions in a presentation, as this is a longstanding concept in movies and other video creations. (Ex-1002, ¶153). Indeed, as the ’103 patent acknowledges, transition and image effects were known to PHOSITAs. (Ex-1001, 13:12-17, 13:49-52).

In the proposed Hedinsson-Ford-Kostello combination, a PHOSITA would have been motivated to use a programming loop to add these effects to Hedinsson’s playlist. (Ex-1002, ¶154). In particular, a PHOSITA would have been motivated to add, to the playlist, each of the image or transition effects, and a PHOSITA would have understood that this would require the repetition of programming statements (Ex-1002, ¶154; Ex-1005, 21-22).

Also, for the reasons explained in Ground 1, a PHOSITA would have been motivated to use a do-until loop to add these effects. (Ex-1002, ¶155). A PHOSITA

would have known that a do-until loop would be beneficial because at least one effect should be added to the playlist and because looping should continue until all effects have been added. (Ex-1005, 23; Ex-1002, ¶155). Also, a do-until loop would have been obvious because there were a finite number of different types of loops, each of which were predictable, well-known, and within a PHOSITA's skill. (Ex-1002, ¶155). In addition, a PHOSITA would have been motivated to use a do-until loop for adding effects because a PHOSITA would have been motivated to dupe and revise code when possible and, as explained in Ground 1, the PHOSITA would have been motivated to use this type of loop to create a new playlist and to display an existing one. (Ex-1002, ¶155).

C. Reasonable Expectation of Success

A PHOSITA would have had a reasonable expectation of success in making the proposed combination for the reasons explained in connection with Ground 1. (Ex-1002, ¶156). In addition, in the relevant timeframe, transition effects and image effects were known and understood by PHOSITAs. (Ex-1001, 13:12-17, 13:49-52). In addition, the use of these effects was predictable and within the skill of a PHOSITA. (Ex-1002, ¶156).

D. Analogous Art

Kostello is analogous art because it is in the same field as the '103 patent, and it would have been reasonably pertinent to the video-sharing problems facing the

named inventors. (Ex-1006, Title, Abstract, 1:5-22, 3:49-50, 4:13-16; Ex-1001, Title, Abstract, 1:18-22; Ex-1002, ¶158).

E. Claim Mapping

CLAIM 11

“11[a]. The method of claim 7, further comprising adding a transition effect to the composite virtual clip by:”

See Ground 1, claims 1 and 7. The proposed combination **adds a transition effect to the composite virtual clip** (the media stream and/or its playlist). (Ex-1002, ¶160). Kostello adds transition effects between consecutive video clips using transition effects elements 604. (Ex-1006, 7:15-60, Fig. 6). As explained in Ground 2’s “Rationale Supporting Obviousness,” it would have been obvious to add this functionality to the Hedinsson-Ford combination.

“[11b] selecting the composite virtual clip, wherein the composite virtual clip is defined by (N) saved virtual clips and an order of presentation of each virtual clip from 1 to N;”

The proposed combination discloses or renders obvious **selecting the composite virtual clip** (the media stream and/or its playlist). In order to add Kostello’s transition effects to a particular playlist, a PHOSITA would have been motivated to first select the playlist (and, by extension, its media stream) to facilitate the addition of those transition effects and to ensure they are added to the correct playlist. (Ex-1002, ¶161).

For the reasons explained in elements [1b] and [7b], Hedinsson discloses or renders obvious that **the composite virtual clip** (the media stream and/or its playlist) **is defined by (N) saved virtual clips** (video clips saved on a media server) **and an order of presentation of each virtual clip from 1 to N.** (Ex-1002, ¶162). Hedinsson's playlist specifies the order in which a plurality of video clips should be played (Ex-1004, 5:37-47, 9:37-53, Fig. 2), and the number of video clips in the playlist teaches (N), which can be any number.

“[11c] selecting a transition effect having a known storage location;”

Kostello teaches **selecting a transition effect.** (Ex-1006, 7:15-60, Fig. 6). For example, as shown in Figure 6, the user has selected a cross dissolve transition effect 604-2 to be applied to transition element 508-1, which occurs between the first video clip segment (represented by thumbnail 416-1) and the second segment (represented by thumbnail 416-2). (Ex-1006, Fig. 6). As explained in Ground 2's "Rationale Supporting Obviousness," Kostello also discloses or renders obvious selecting a transition effect 604 for each transition between consecutive segments. (Ex-1006, 7:15-24, 5:57-60, Fig. 6; Ex-1002, ¶163).

The combination discloses or renders obvious that each transition effect **has a known storage location.** (Ex-1002, ¶164). In the interface shown in Figure 6, Kostello displays several different transition effects elements 604 that may be

selected by a user, and Kostello specifies that its software can be downloaded to a mobile device or run on another device. (Ex-1006, 3:64-4:5, 9:47-65; Ex-1002, ¶164). Also, Hedinsson teaches that its playlist “specifie[s]” transition effects. (Ex-1004, 6:2-5, Fig. 2). From this, a PHOSITA would have understood each reference to teach, or at least suggest, that each transition effect has a known storage location. (Ex-1002, ¶164). In addition, a PHOSITA would have been motivated to store each transition effect in a known location to facilitate its use in the proposed combination, as explained below in element [11e]. (Ex-1002, ¶164).

“[11d] initially setting (i)=1;”

Ford describes an exemplary do-until loop that begins with “i = 1.” (Ex-1005, 26). This teaches **initially setting (i)=1**, where **(i)** is taught by Ford’s “i.” (Ex-1002, ¶165). As explained in Ground 2’s “Rationale Supporting Obviousness” and elements [11e] and [11f], a PHOSITA would have been motivated to add transition effects to Hedinsson’s playlist using Ford’s do-until loop, which counts the iterations of the loop by first setting a variable to 1. (Ex-1002, ¶165).

“[11e] configuring an (i)th transition effect link pointing to a known (i)th transition effect storage location; configuring the (i)th transition effect link to be activated after activating an (i)th virtual clip link and before activation of an (i+1)th virtual clip link, wherein (i) is between 1 and (N)-1; updating the composite virtual clip file to include the (i)th transition effect link;”

As explained in element [11c], the combination discloses or renders obvious **a known storage location** for each selected **transition effect**.

It would have been obvious to add a transition effect to the playlist by **configuring a transition effect link pointing to** the transition effect's **known storage location** and by **updating the composite virtual clip file** (the playlist) **to include** that **transition effect link**. A PHOSITA would have been motivated to add some sort of reference to the transition effect (**a transition effect link**) in the playlist (**the composite virtual clip file**) to identify the transition effect selected by the user for that transition. (Ex-1002, ¶167; Ex-1001, 8:39-41 (broadly defining “link”). Hedinson states that its playlist can specify transition effects, and a PHOSITA would have understood this to mean that the playlist includes some type of reference to the desired transition effects. (Ex-1004, 6:1-8, Fig. 2; Ex-1002, ¶167). Stated differently, a PHOSITA would have understood Hedinson to disclose that its playlist includes a reference to a transition effect, rather than the transition effect itself, because Hedinson's playlist includes links to video clips, rather than a new composite media file itself, as explained in elements [1d] and [7d]. (Ex-1002, ¶167). A PHOSITA would have understood using a link to be an efficient way to identify

transition effects. (Ex-1002, ¶167). Moreover, a PHOSITA would have been motivated to include a link in the playlist to facilitate a reference to the software code required to implement the desired transition effect. (Ex-1002, ¶167). For example, Kostello contemplates “a cross dissolve effect” between two segments (Ex-1006, 7:27-30), and a PHOSITA would have understood that displaying this effect would require the system to execute code that blends, on a frame by frame basis, the end of one video and the beginning of the next video. (Ex-1002, ¶167).

The combination discloses or renders obvious configuring an **(i)th** transition effect link pointing to a known **(i)th** transition effect storage location and updating the composite virtual file to include the **(i)th** transition effect link. (Ex-1002, ¶168). Initially, **(i)** equals 1 (element [11d]), and as explained in the preceding paragraphs, the combination discloses or renders obvious configuring a first transition effect link pointing to a first location associated with a first transition and updating the playlist to include that link. In addition, **(i)** is between 1 and **(N)-1**. (Ex-1002, ¶168). As discussed in elements [11d] and [11f], **(i)** initially equals 1 and is incremented. Also, as discussed in element [11b], **(N)** is the number of video clips selected for inclusion in the playlist (i.e., a positive integer), and so **(N)-1** equals the number of transitions between those video clips. (Ex-1002, ¶168). The example shown in Kostello’s Figure 6 includes four video clips (corresponding to thumbnails 416-1 to 416-4) and three transitions between those video clips (508-1 to 508-3); in this example, **(N)**

equals 4, and (N)-1 equals 3. Also, to the extent required, the combination discloses or renders obvious selecting, configuring, and adding a link for *each* of the N-1 transitions (*see* element [11b]) because Kostello allows users to add different effects to a set of transitions (*see* element [11c]). (Ex-1006, 7:15-24, Figs. 5-6; Ex-1002, ¶168). As noted in element [11b], N can be any number.

The combination discloses or renders obvious **configuring the i(th) transition effect to be activated after activating an (i)th virtual clip link and before activation of an (i+1)th virtual clip link**. A PHOSITA would have understood the combination to activate the transition effect link **after** the activation of preceding video clip link and **before** the activation of the subsequent video clip link, because Kostello's transition effects are applied *between* two video clips (i.e., the preceding and subsequent video clips). (Ex-1006, 7:22-24; Ex-1002, ¶169). In addition, a PHOSITA would have been motivated to implement the combination in this manner given Kostello's disclosure and because transition effects were known in the art and contemplated by Hedinsson. (*E.g.*, Ex-1006, 7:22-24; Ex-1004, 6:2-5; Ex-1002, ¶169). This would have been a logical and desirable way to implement the proposed combination because a PHOSITA would have been motivated to activate links in the order in which the corresponding clips or effects are to be displayed. (Ex-1002, ¶169). Further, a PHOSITA would have been motivated to implement the combination in this way because it activates links to video clips and

transition effects in sequential order, as indicated by Hedinsson. (Ex-1004, 9:44-46; Ex-1002, ¶169).

“[11f] determining if (i) equals (N); when (i) is less than (N), setting (i) equals (i)+(1); and repeating selecting and configuring to add the (i)th transition effect to the composite virtual clip until (i) equals (N)-1.”

As explained in Ground 2’s “Rationale Supporting Obviousness,” a PHOSITA would have been motivated to add each of the transition effect links to the playlist using a programming loop, and it would have been obvious to use a do-until loop, as taught by Ford, for this purpose. (Ex-1002, ¶170).

In particular, given Ford and Kostello, a PHOSITA would have been motivated to implement Hedinsson using a do-until loop that selects, configures, and adds one transition effect link during each iteration, and repeats until N-1 transition effects have been added to the playlist.⁸ (Ex-1002, ¶171). This would allow the user to select a different transition effect between each video clip, as explained in element [11e].

⁸ As Dr. Houh explains, it is unclear whether the claim requires N, N-1, or N-2 iterations. (Ex-1002, ¶172). The Board need not address this ambiguity because, under any of these interpretations, the claim is rendered obvious by Hedinsson-Ford-Kostello, as explained in this section and *supra* n.4.

In addition, if the claim requires configuring and saving N transition effect links, it would still have been obvious because a PHOSITA would have been motivated to include an additional loop iteration (for a total of N iterations) so that a user can select a transition effect to appear at the end of the last video clip in the playlist, in order to provide additional creative flexibility to the user.⁹ (Ex-1002, ¶173). Indeed, Hedinsson contemplates displaying a different playlist immediately following the instant playlist (Ex-1004, 6:13-20), which would further have motivated a PHOSITA to include this functionality. (Ex-1002, ¶173).

For the reasons explained in elements [1e] and [7e], the proposed combination discloses or renders obvious **determining if (i) equals (N); when (i) is less than (N), setting (i) equals (i)+(1); and repeating** the *statements* within the loop **until** a *condition* is satisfied. (Ex-1002, ¶174). Although claims 1 and 7 use the variables M and P, respectively, to count the number of loops, where this claim uses “i” (element [11d]), the combination applies equally no matter what name is given to the variable. (Ex-1002, ¶174).

In the context of this loop, a PHOSITA would have been motivated to include both the **selecting** and **configuring** steps in the loop’s *statements* because a user may

⁹ Petitioner need not show which of these options was most preferred. *Honeywell Int’l, Inc. v. 3G Licensing, S.A.*, 124 F.4th 1345, 1355-56 (Fed. Cir. 2025).

choose to **select** different transition effects between different pairs of video clips, and so that the appropriate link is **configured** and added to the playlist, as discussed in element [11e]. (Ex-1002, ¶175).

A PHOSITA would have been motivated to set, as the loop's *condition*, either: (1) **(i) equals (N)-1** so that the loop repeats N-1 times, once for each transition between video clips in the playlist, or (2) **(i) equals (N)**, in order to allow the user to also select a transition to occur after the playlist. (Ex-1002, ¶176).

In addition, a PHOSITA would have been motivated to **determine if “i” equals N**—even if this is not the loop's condition—to check for errors in the loop's code. (Ex-1002, ¶177). As Ford teaches, when working with loops, programmers “must constantly be on the look out for ... endless loops,” i.e., a loop that never ends. (Ex-1005, 30). A PHOSITA would have understood that, if the loop's condition is **(i)=(N)-1**, then (i) should never equal (N) inside the loop. (Ex-1002, ¶177). As a result, a PHOSITA would have been motivated to add a few lines of code (i.e., to **determine if (i) equals (N)**, and if so, to exit the loop) for error checking, as contemplated by Ford. (Ex-1002, ¶177; Ex-1005, 31-32).

Also, to the extent required by the claim, the proposed combination discloses or renders obvious element [11e] even after (i) is incremented and this step is repeated because, as explained in element [11e], it would have been obvious to select, configure, and add *each* transition effect link. (Ex-1002, ¶178).

CLAIM 12

“12[a]. The method of claim 7, further comprising adding an effect to the composite virtual clip, wherein the effect includes at least one of a lensing effect and a second effect, by:”

See Ground 1, claims 1 and 7. The proposed combination **adds an effect to the composite virtual clip** (the media stream and/or its playlist). (Ex-1002, ¶179). Kostello adds effects to video clips using effects elements 504, which teach a **lensing effect**¹⁰ and/or a **second effect**. (Ex-1006, 6:57-7:14, Fig. 5). Kostello also contemplates adding a “visual overlay” or “change of color of a segment.” (Ex-1006, 2:6-10, claim 9). As explained in Ground 2’s “Rationale Supporting Obviousness,” it would have been obvious to add this functionality to the Hedinsson-Ford combination.

“[12b] selecting the composite virtual clip, wherein the composite virtual clip defined by (N) saved virtual clips and an order of presentation of each virtual clip from 1 to N;”

See element [11b]. (Ex-1002, ¶180).

¹⁰ Although the term “lensing effect” is unclear, the Specification states that it at least includes “overlay of one or more color filters, image distortions, and annotations.” (Ex-1001, 13:49-52).

“[12c] selecting an effect having a known storage location;”

Kostello teaches **selecting an effect**. (Ex-1006, 6:57-7:14, Fig. 5). For example, as shown in Figure 5, the user has selected an instant replay effect 504-4 for the video clip associated with thumbnail 416-2. (Ex-1006, Fig. 5). Kostello also contemplates adding a “visual overlay” or “change of color of a segment.” (Ex-1006, 2:6-10, claim 9). As explained in Ground 2’s “Rationale Supporting Obviousness,” Kostello also discloses or renders obvious selecting an effect 504 for each video clip segment. (Ex-1006, 6:61-63, 7:8-10, 5:57-60, Fig. 5; Ex-1002, ¶181).

For substantially the same reasons as explained in element [11c], the combination discloses or renders obvious that each effect **has a known storage location**. (Ex-1002, ¶182). Indeed, Kostello’s Figure 5 shows the available options for image effects 504 (Ex-1006, 6:57-7:14), and Hedinsson teaches that playlist 240 can include “[r]eferences to captions, subtitles, image overlays, commentary, external web links, and other related content” (Ex-1004, 6:5-8).

“[12d] setting (k) initially equals to 1;”

Ford describes an exemplary do-until loop that begins with “i = 1.” (Ex-1005, 26). This teaches **setting (k) initially equals to 1**, where (k) is taught by Ford’s “i.” (Ex-1002, ¶183). As explained in Ground 2’s “Rationale Supporting Obviousness”

and elements [12e] and [12f], a PHOSITA would have been motivated to add effects to Hedinsson's playlist using Ford's do-until loop, which counts the iterations of the loop by first setting a variable to 1. (Ex-1002, ¶183).

“[12e] configuring an (k)th effect link pointing to a known (k)th storage location; configuring the (k)th effect link to be activated simultaneously with an (k)th virtual clip, wherein (k) is between 1 and (N); updating the composite virtual clip to include the (k)th effect link;”

As explained in element [12c], the combination discloses or renders obvious **a known storage location** for each selected effect.

For the reasons explained in element [11e], it would have been obvious to add an effect to the playlist by **configuring an effect link pointing to** the effect's **known storage location** and by **updating the composite virtual clip file** (the playlist) to **include** that **effect link**. A PHOSITA would have been motivated to add some sort of reference to the effect (**an effect link**) in the playlist (**the composite virtual clip file**) to identify the effect selected by a user for that video clip. (Ex-1002, ¶185; Ex-1001, 8:39-41 (broadly defining “link”)). Indeed, Kostello describes effect 504-1, which “can be used for playing back the corresponding video segment at half speed” (Ex-1006, 6:63-65), and a PHOSITA would have understood that the system would alter the normal display of the clip throughout the duration of the clip. (Ex-1002, ¶185).

The combination discloses or renders obvious configuring a **(k)th** effect link

pointing to a known **(k)th** storage location and updating the composite virtual file to include the **(k)th** effect link. (Ex-1002, ¶186). Initially, **(k)** equals 1 (element [12d]), and as explained in the preceding paragraphs, the combination discloses or renders obvious configuring a first effect link pointing to a first location associated with a first effect and updating the playlist to include that link. In addition, **(k)** is **between 1 and (N)** because **(k)** initially equals 1 (element [12d]) and **(k)** is incremented until it equals (N) (element [12f]), which is the number of video clips (element [12b]). Also, to the extent required, the combination discloses or renders obvious selecting, configuring, and adding a link for *each* of the N video clips (element [11b]) because Kostello allows users to add different effects to a set of video clips (*see* element [11c]). (Ex-1006, 6:61-63, 7:8-10, Fig. 5; Ex-1002, ¶186). As noted in element [11b] (referenced by [12b]), N can be any number.

The combination discloses or renders obvious **configuring the k(th) effect link to be activated simultaneously with an (k)th virtual clip**. A PHOSITA would have understood the combination to activate the effect link **simultaneously** with the corresponding video clip because Kostello's effects are applied to video clip segments. (Ex-1006, 6:61-63; Ex-1002, ¶187). In addition, a PHOSITA would have been motivated to implement the combination in this manner given Kostello's disclosure and because effects were known in the art and contemplated by Hedinsson. (*E.g.*, Ex-1006, 6:61-63; Ex-1004, 6:5-8; Ex-1002, ¶187). This would

have been a logical and desirable way to implement the proposed combination because it facilitates applying the effect to the entire video clip. (Ex-1002, ¶187). For example, some of Kostello’s effects are applied to the entire video clip (e.g., half-speed playback, “visual overlay,” and color change) (Ex-1006, 6:63-65, 2:6-10), and a PHOSITA would have been motivated to activate the effect link with the corresponding video clip to facilitate execution of the appropriate code at the appropriate time. (Ex-1002, ¶187).

“[12f] determining if (k) equals (N); when (k) is less than (N), setting (k) equals (k)+(1); and repeating selecting and configuring to add the (k)th effect to the composite virtual clip until (k) equals (N).”

As explained in Ground 2’s “Rationale Supporting Obviousness,” a PHOSITA would have been motivated to add each of the effects links to the playlist using a programming loop, and it would have been obvious to use a do-until loop, as taught by Ford, for this purpose. (Ex-1002, ¶188). In particular, given Ford and Kostello, a PHOSITA would have been motivated to implement Hedinsson using a do-until loop that selects, configures, and adds one effect link during each iteration, and repeats until N links have been added to the playlist.¹¹ (Ex-1002, ¶188). This

¹¹ As Dr. Houh explains, it is unclear whether the claim requires N or N-1 iterations. (Ex-1002, ¶189). The Board need not address this ambiguity because,

would allow the user to select a different effect for each video clip, as explained in element [12e].

For the reasons explained in elements [1e] and [7e], the proposed combination discloses or renders obvious **determining if (k) equals (N); when (k) is less than (N), setting (k) equals (k)+(1); and repeating** the *statements* within the loop **until (k) equals (N)**. (Ex-1002, ¶190). Although claims 1 and 7 use the variables M and P, respectively, to count the number of loops, where this claim uses “k” (element [12d]), the combination applies equally no matter what name is given to the variable (element [12c]). (Ex-1002, ¶190). In addition, in the context of this loop, a PHOSITA would have been motivated to include both the **selecting** and **configuring** steps in the loop’s *statements* because a user may choose to **select** different effects for each video clip in the playlist, and so that the appropriate link is **configured** and added to the playlist, as discussed in element [12e]. (Ex-1002, ¶190).

Also, to the extent required by the claim, the proposed combination discloses or renders obvious element [12e] even after (k) is incremented and this step is repeated because, as explained in element [11e], it would have been obvious to select, configure, and add *each* effect link. (Ex-1002, ¶191).

under either interpretation, the claim is rendered obvious by Hedinsson-Ford. *See supra* n.4.

CLAIM 23

Dependent claim 23 is substantially the same as dependent claim 11 except that claim 23: (1) depends from claim 19, (2) is directed to **computer readable program code which causes the programmable computer processor** to perform the recited method steps, and (3) recites “(N)” at the end of element [23e] (instead of “(N)-1”). (Ex-1013, 9). Claim 23 is obvious for the reasons provided for claims 11 and 19. (Ex-1002, ¶192).

In particular, as explained in element [11e], it would have been obvious to implement Hedinsson to iterate the loop N times so that the user can select a transition effect to occur following the end of a playlist. (Ex-1002, ¶193).

CLAIM 24

Dependent claim 24 is substantially the same as dependent claim 12 except that claim 24: (1) depends from claim 19, (2) is directed to **computer readable program code which causes the programmable computer processor** to perform the recited method steps, (3) recites “**at least one of a lensing effect and a sound effect**” in the preamble, and (4) specifies that “**the composite virtual clip includes (N) saved virtual clips.**”¹² (Ex-1013, 10). Claim 24 is obvious for the reasons

¹² As Dr. Houh explains, claim 24 is unclear (Ex-1002, ¶195); however, under any reasonable interpretation, it is rendered obvious by the proposed combination.

provided for claims 12 and 19. (Ex-1002, ¶194).

Moreover, even if the preamble is limiting, the proposed combination renders it obvious. Kostello’s effects teach a “lensing effect” (element [12a]), and Kostello teaches adding an effect that includes a **sound effect** because Kostello adds “audio” effects to individual segments in substantially the same way as other effects. (Ex-1006, 2:27-36, 7:65-8:10, 8:48-53, Fig. 7, Fig. 9 (step 930); Ex-1002, ¶196).

In addition, Hedinsson discloses or renders obvious that **the composite virtual clip** (the media stream and/or its playlist) **includes (N) saved virtual clips** (the number of video clips referenced by the playlist) for the reasons explained in element [12b]. (Ex-1004, 5:37-47, 9:37-53, Fig. 2; Ex-1002, ¶197).

IV. CONCLUSION

Petitioner respectfully requests that claims 1-24 of the ’103 patent be canceled.

Date: <u>August 12, 2025</u>	<u>/Matthew A. Smith/ (RN 49,003)</u> Matthew A. Smith SMITH BALUCH LLP 700 Pennsylvania Ave SE Ste. 2060 Washington, DC 20003 (202) 669-6207 smith@smithbaluch.com <i>Counsel for Petitioner Google LLC</i>
------------------------------	---

CLAIM LISTING APPENDIX

	Claim Language
1a	1. A method to make a composite virtual clip, comprising:
1b	selecting (N) saved virtual clips and an order of presentation for each virtual clip;
1c	initially setting (M)=1;
1d	configuring a (M)th pointer to indicate a storage location for a (M)th saved virtual clip, wherein a (M)th virtual clip comprises a (M)th content from one or more designated sources and (M) is greater than or equal to 1 and less than or equal to (N); saving said (M)th pointer;
1e	determining if (M) equals (N); when (M) is less than (N), setting (M) equal to (M)+1; and repeating said configuring, saving, and determining until (M) equals (N).
2a	2. The method of claim 1, wherein said composite virtual clip further comprises one or more first type virtual clips, wherein the one or more first type virtual clips comprise content from a Media File, playable or static, from a beginning to a first designated end point,
2b	wherein the Media File, playable or static, is selected from the group consisting of AVI file, MP3 file, MP4 file, WMA file, WAV file, FLASH, MPEG file, an image file, a PDF file, a text file, a VISIO file, a .ppt file, a .key file, a spreadsheet file, 3D media file which provides a Stereoscopic visual display that can be viewed through stereoscopic eyewear or played on 3D display technology, and a Virtual Reality/Augmented Reality file that can be viewed through Virtual Reality devices.
3a	3. The method of claim 2, further comprising:

	Claim Language
3b	displaying a first interactable graphical object, wherein the first interactable graphical object comprises a Cancel Clip link;
3c	simultaneously displaying a second interactable graphical object, wherein the second interactable graphical object comprises an End Clip link;
3d	when activating the first interactable graphical object, a begin time location, end time location, and first media duration of the first type virtual clip are not stored; and
3e	when activating the second interactable graphical object, the begin time location, end time location, and first media during of the first type virtual clip are stored.
4a	4. The method of claim 1, wherein said composite virtual clip further comprises one or more second type virtual clips, wherein the second type virtual clip comprising content from a Media File, playable or static, from a designated beginning to a second designated end point,
4b	wherein said Media File, playable or static, is selected from the group consisting of an AVI file, MP3 file, MP4 file, WMA file, WAV file, FLASH, MPEG file, an image file, a PDF file, a text file, a .ppt file, a .key file, a VISIO file, a spreadsheet file, a 3D media file which provides a Stereoscopic visual display that can be viewed through stereoscopic eyewear or played on 3D display technology, and a Virtual Reality/Augmented Reality file that can be viewed through Virtual Reality devices.
5a	5. The method of claim 4, further comprising:
5b	displaying a first interactable graphical object, wherein the first interactable graphical object comprises a Cancel Clip link;
5c	simultaneously displaying a second interactable graphical object, wherein the second interactable graphical object comprises an End Clip link;

	Claim Language
5d	when activating the first interactable graphical object, a begin time location, end time location, and first media duration of the second type virtual clip are not stored; and
5e	when activating the second interactable graphical object, the begin time location, end time location, and first media during of the second type virtual clip are stored.
6	6. The method of claim 1, wherein said composite virtual clip further comprises one or more first type virtual clips and one or more second type virtual clips.
7a	7. A method to display a composite virtual clip, comprising:
7b	providing a storage location for a composite virtual clip stored in a composite virtual clip file, wherein the composite virtual clip is configured to access (N) saved virtual clips;
7c	initially setting (P)=1;
7d	activating a (P)th link encoded in the composite virtual clip file to stream a (P)th saved virtual clip to a user's device;
7e	determining if (P) equals (N); when (P) is less than (N), setting (P) equals (P)+1; and repeating said activating and determining until (P) equals (N).
8a	8. The method of claim 7, wherein said composite virtual clip further comprises one or more first type virtual clips, wherein the first type virtual clip comprising content from a Media File, playable or static from a beginning to a first designated end point,.

	Claim Language
8b	wherein said Media File, static or playable, is selected from the group consisting of an AVI file, MP3 file, MP4 file, WMA file, WAV file, FLASH, MPEG file, an image file, a PDF file, a text file, a VISIO file, a .ppt file, a .key file, a spreadsheet file, a 3D media file which provides a Stereoscopic visual display that can be viewed through stereoscopic eyewear or played on 3D display technology, and a Virtual Reality/Augmented Reality file that can be viewed through Virtual Reality devices
9a	9. The method of claim 7, wherein said composite virtual clip further comprises one or more second type virtual clips, wherein the second type virtual clip comprising content from a Media File, playable or static from a designated start point to a second designated end point,
9b	wherein said Media File, static or playable, is selected from the group consisting of an AVI file, MP3 file, MP4 file, WMA file, WAV file, FLASH, MPEG file, an image file, a PDF file, a text file, a .ppt file, a .key file, a VISIO file, a spreadsheet file, a 3D media file which provides a Stereoscopic visual display that can be viewed through stereoscopic eyewear or played on 3D display technology, and a Virtual Reality/Augmented Reality file that can be viewed through Virtual Reality devices.
10	10. The method of claim 7, wherein said composite virtual clip further comprises one or more first type virtual clips and one or more second type virtual clips.
11a	11. The method of claim 7, further comprising adding a transition effect to the composite virtual clip by:
11b	selecting the composite virtual clip, wherein the composite virtual clip is defined by (N) saved virtual clips and an order of presentation of each virtual clip from 1 to N;

	Claim Language
11c	selecting a transition effect having a known storage location;
11d	initially setting (i)=1;
11e	configuring an (i)th transition effect link pointing to a known (i)th transition effect storage location; configuring the (i)th transition effect link to be activated after activating an (i)th virtual clip link and before activation of an (i+1)th virtual clip link, wherein (i) is between 1 and (N)-1; updating the composite virtual clip file to include the (i)th transition effect link;
11f	determining if (i) equals (N); when (i) is less than (N), setting (i) equals (i)+(1); and repeating selecting and configuring to add the (i)th transition effect to the composite virtual clip until (i) equals (N)-1.
12a	12. The method of claim 7, further comprising adding an effect to the composite virtual clip, wherein the effect includes at least one of a lensing effect and a second effect, by:
12b	selecting the composite virtual clip, wherein the composite virtual clip defined by (N) saved virtual clips and an order of presentation of each virtual clip from 1 to N;
12c	selecting an effect having a known storage location;
12d	setting (k) initially equals to 1;
12e	configuring an (k)th effect link pointing to a known (k)th storage location; configuring the (k)th effect link to be activated simultaneously with an (k)th virtual clip, wherein (k) is between 1 and (N); updating the composite virtual clip to include the (k)th effect link;

	Claim Language
12f	determining if (k) equals (N); when (k) is less than (N), setting (k) equals (k)+(1); and repeating selecting and configuring to add the (k)th effect to the composite virtual clip until (k) equals (N).
13a	13. A non-transitory computer usable medium encoded with a computer program product to make a composite virtual clip and usable with programmable computer processor disposed within a controller, comprising:
13b	computer readable program code which causes said programmable computer processor to select (N) saved virtual clips and an order of presentation for each virtual clip;
13c	computer readable program code which causes said programmable computer processor to initially set (M)=1;
13d	computer readable program code which causes said programmable computer processor to configure a (M)th pointer to indicate a storage location for a (M)th saved virtual clip, wherein a (M) th virtual clip comprises a (M)th content from one or more designated sources and (M) is greater than or equal to 1 and less than or equal to (N); computer readable program code which causes said programmable computer processor to save said (M)th pointer;
13e	computer readable program code which causes said programmable computer processor to determine if (M) equals (N); when (M) is less than (N), computer readable program code which causes said programmable computer processor to set (M) equal to (M)+1; and computer readable program code which causes said programmable computer processor to repeat said configuring, saving, and determining until (M) equals (N).

	Claim Language
14a	14. The computer program product of claim 13, wherein said composite virtual clip further comprises one or more first type virtual clips, wherein the first type virtual clip comprises content from a Media File, playable or static from a beginning to a first designated end point,
14b	wherein said Media File, static or playable, is selected from the group consisting of an AVI file, MP3 file, MP4 file, WMA file, WAV file, FLASH, MPEG file, an image file, a PDF file, a text file, a .ppt file, a .key file, a VISIO file, a spreadsheet file, a 3D media file which provides a Stereoscopic visual display that can be viewed through stereoscopic eyewear or played on 3D display technology, and a Virtual Reality/Augmented Reality file that can be viewed through Virtual Reality devices.
15a	15. The computer program product of claim 14, further comprising computer readable program code which causes said programmable computer processor to:
15b	display a first interactable graphical object, wherein the first interactable graphical object comprises a Cancel Clip link;
15c	simultaneously display a second interactable graphical object a, wherein the second interactable graphical object comprises an End Clip link;
15d	when activating the first interactable graphical object, not to store a begin time location, end time location, and first media duration of the first type virtual clip; and
15e	when activating the second interactable graphical object, store the begin time location, end time location, and first media during of the first type virtual clip.

	Claim Language
16a	16. The computer program product of claim 13, wherein said composite virtual clip further comprises one or more second type virtual clips, wherein the second type virtual clip comprising content from a Media File, playable or static from a designated beginning to a second designated end point,
16b	wherein said Media File, static or playable, is selected from the group consisting of an AVI file, MP3 file, MP4 file, WMA file, WAV file, FLASH, MPEG file, an image file, a PDF file, a .ppt file, a .key file, a text file, a VISIO file, a spreadsheet file, a 3D media file which provides a Stereoscopic visual display that can be viewed through stereoscopic eyewear or played on 3D display technology, and a Virtual Reality/Augmented Reality file that can be viewed through Virtual Reality devices.
17a	17. The computer program product of claim 16, further comprising computer readable program code which causes said programmable computer processor to:
17b	display a first interactable graphical object, wherein the first interactable graphical object comprises a Cancel Clip link;
17c	simultaneously display a second interactable graphical object, wherein the second interactable graphical object comprises an End Clip link;
17d	when activating the first interactable graphical object, not to store a begin time location, end time location, and first media duration of the second type virtual clip; and
17e	when activating the second interactable graphical object, store the begin time location, end time location, and first media during of the second type virtual clip.

	Claim Language
18	18. The computer program product of claim 13, wherein said composite virtual clip further comprises one or more first type virtual clips and one or more second type virtual clips.
19a	19. A non-transitory computer usable medium encoded with a computer program product to display a composite virtual clip and usable with programmable computer processor disposed within a controller, comprising:
19b	computer readable program code which causes said programmable computer processor to provide a storage location for a composite virtual clip stored in a composite virtual clip file, wherein the composite virtual clip is configured to access (N) saved virtual clips;
19c	computer readable program code which causes said programmable computer processor to initially set (P)=1;
19d	computer readable program code which causes said programmable computer processor to activate a (P)th link encoded in the composite virtual clip file to stream a (P)th saved virtual clip to a user's device;
19e	computer readable program code which causes said programmable computer processor to determine if (P) equals (N); when (P) is less than (N), computer readable program code which causes said programmable computer processor to set (P) equals (P)+1; and computer readable program code which causes said programmable computer processor to repeat said activating and determining until (P) equals (N).
20a	20. The computer program product of claim 19, wherein said composite virtual clip further comprises one or more first type virtual clips, wherein the first type virtual clip comprising content from a Media File, playable or static from a beginning to a first designated end point,

	Claim Language
20b	wherein said Media File, static or playable, is selected from the group consisting of an AVI file, MP3 file, MP4 file, WMA file, WAV file, FLASH, MPEG file, an image file, a PDF file, a text file, a .ppt file, a .key file, a VISIO file, a spreadsheet file, a 3D media file which provides a Stereoscopic visual display that can be viewed through stereoscopic eyewear or played on 3D display technology, and a Virtual Reality/Augmented Reality file that can be viewed through Virtual Reality devices.
21a	21. The computer program product of claim 19, wherein said composite virtual clip further comprises one or more second type virtual clips, wherein the second type virtual clip comprising content from a Media File, playable or static from a designated start point to a second designated end point,
21b	wherein said Media File, static or playable, is selected from the group consisting of an AVI file, MP3 file, MP4 file, WMA file, WAV file, FLASH, MPEG file, an image file, a PDF file, a text file, a .ppt file, a .key file, a VISIO file, a spreadsheet file, a 3D media file which provides a Stereoscopic visual display that can be viewed through stereoscopic eyewear or played on 3D display technology, and a Virtual Reality/Augmented Reality file that can be viewed through Virtual Reality devices.
22	22. The computer program product of claim 19, wherein said composite virtual clip further comprises one or more first type virtual clips and one or more second type virtual clips.
23a	23. The computer program product of claim 19, further comprising computer readable program code encoded with a computer program product to add a transition effect to the composite virtual clip and usable with programmable computer processor disposed within a controller, comprising:

	Claim Language
23b	computer readable program code which causes said programmable computer processor to select the composite virtual clip, wherein the composite virtual clip is defined by (N) saved virtual clips and an order of presentation of each virtual clip from 1 to N;
23c	computer readable program code which causes said programmable computer processor to select a transition effect having a known storage location;
23d	computer readable program code which causes said programmable computer processor to initially set (i)=1;
23e	computer readable program code which causes said programmable computer processor to configure an (i)th transition effect link pointing to a known (i)th transition effect storage location; computer readable program code which causes said programmable computer processor to configure the (i)th transition effect link to be activated after activating an (i)th virtual clip link and before activation of an (i+1)th virtual clip, link wherein (i) is between 1 and (N)-1; computer readable program code which causes said programmable computer processor to update the composite virtual clip file o include the (i)th transition effect link;
23f	computer readable program code which causes said programmable computer processor to determine if (i) equals (N); when (i) is less than (N), computer readable program code which causes said programmable computer processor to set (i) equals (i)+(1); and computer readable program code which causes said programmable computer processor to repeat selecting and configuring to add the (i)th transition effect to the composite virtual clip until (i) equals (N).

	Claim Language
24a	24. The computer program product of claim 19, further comprising computer readable program code encoded with a computer program product to add an effect including at least one of a lensing effect and a sound effect to the composite virtual clip and usable with programmable computer processor disposed within a controller, comprising:
24b	computer readable program code which causes said programmable computer processor to select the composite virtual clip, wherein the composite virtual clip includes (N) saved virtual clips and an order of presentation of each virtual clip from 1 to N;
24c	computer readable program code which causes said programmable computer processor to select an effect having a known storage location;
24d	computer readable program code which causes said programmable computer processor to set (k) initially equals to 1;
24e	computer readable program code which causes said programmable computer processor to configure an (k)th effect link pointing to the known (k)th storage location; computer readable program code which causes said programmable computer processor to configure the (k)th effect link to be activated simultaneously with an (k)th virtual clip, wherein (i) is between 1 and (N); computer readable program code which causes said programmable computer processor to update the composite virtual clip to include the (k)th effect link;
24f	computer readable program code which causes said programmable computer processor to determine if (k) equals (N); when (k) is less than (N), computer readable program code which causes said programmable computer processor to set (k) equals (k)+(1); and computer readable program code which causes said programmable computer processor to repeat selecting and configuring to add the (k)th effect to the composite virtual clip until (k) equals (N).

CERTIFICATE OF SERVICE

The undersigned hereby certifies that the foregoing petition for *inter partes* review, together with all exhibits and other documents filed therewith, was served by Priority Mail Express on August 12, 2025, on the Patent Owner's counsel of record at the United States Patent & Trademark Office having the following address:

Messner Reeves LLP

7250 North 16th Street

Suite 410

Phoenix, AZ 85020

UNITED STATES

Attn: Rowan P. Smith

Date: August 12, 2025

/Matthew A. Smith/ (RN 49,003)

CERTIFICATE OF WORD COUNT

The undersigned hereby certifies that the foregoing petition for *inter partes* review contains 13,961 words according to the word processing program used to prepare it.

Date: August 12, 2025

/Matthew A. Smith/ (RN 49,003)