

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

LIGHTRICKS LTD.,
Petitioner,

v.

PLOTAGRAPH, INC. and SASCHA CONNELLY,
Patent Owners

IPR2023-00153
U.S. Patent No. 10,346,017

**PATENT OWNERS PLOTAGRAPH, INC. AND SASCHA
CONNELLY'S RESPONSE UNDER 37 C.F.R. §42.107**

TABLE OF CONTENTS

Table of Contents

I.	Introduction.....	1
II.	Petitioner Has Not Shown Unpatentability of Any Claim of the ‘017 Patent	3
A.	Overview of the ‘017 Patent	3
1.	The Specification	3
2.	Exemplary Claim	8
B.	The Cited Art	11
1.	The AEM Reference	11
2.	The IMU Reference	16
3.	The Okabe Reference	17
4.	The Li Reference	19
C.	Claim Construction	20
1.	Construction of shifting a set of pixels “in a loop”	21
D.	Legal Standards	28
E.	Petitioner Fails to Invalidate Any Claim of the ‘017 Patent	29
1.	The prior art does not teach or suggest receiving starting or ending points through a user interface.....	30
2.	The prior art does not teach or suggest creating a link between the starting and ending points.....	33
3.	The AEM prior art does not teach or suggest identifying/selecting a set of pixels as the claims require	34
4.	The IMU-Okabe combination does not identify/select a “set of pixels” to be shifted.....	36
5.	The IMU-Okabe combinations do not teach or suggest shifting a set of pixels in a direction	38
6.	The prior art does not teach or suggest rendering indications on the user interface.....	40
7.	The prior art does not teach or suggest the step of shifting a set of pixels by rendering and re-rendering them “in a loop”	43
8.	The prior art is incapable of shifting a set of pixels “in a loop”	46
9.	The prior art does not show pixels are “prevented from shifting” by a mask	49
10.	The prior art does not disclose a non-linear pathway	51
III.	Conclusion	52

TABLE OF AUTHORITIES

Cases

CFMT, Inc. v. YieldUp Int’l Corp., 349 F.3d 1333 (Fed. Cir. 2003).....	29
Dynamic Drinkware, LLC v. Nat’l Graphics, Inc., 800 F.3d 1375 (Fed. Cir. 2015)	29
Harmonic Inc. v. Avid Tech, Inc., 815 F.3d 1356 (Fed. Cir. 2016)	29
In re CSB-Sys. Int’l, Inc., 832 F.3d 1341 (Fed. Cir. 2016)	20
In re Rijckert, 9 F.3d 1531 (Fed. Cir. 1993)	36
In re: Power Integrations, Inc., 884 F.3d 1370 (Fed. Cir. 2018).....	20, 28
Microsoft Corp. v. Biscotti, Inc., 878 F.3d 1052 (Fed. Cir. 2017)	29
Phillips v. AWH Corp., 415 F.3d 1303 (Fed. Cir. 2005)(en banc).....	21

CLAIM LISTING
For IPR2023-00153
U.S. Patent No. 10,346,017

Claim	Limitation No.	Limitation
1	1- preamble	A computer system for automating the shifting of pixels within a digital image, comprising: one or more processors; and one or more computer-readable media having stored thereon executable instructions that when executed by the one or more processors configure the computer system to perform at least the following:
	1a	access, from memory, a digital image file, wherein the digital image file comprises information that corresponds to individual pixels within the digital image;
	1b	receive a first starting point through a user interface, wherein the first starting point is received through a user selection of a first beginning portion of the digital image;
	1c	render a first visual indication of the first starting point on the user interface;
	1d	receive a first ending point through the user interface, wherein the first ending point is received through a user selection of a first ending portion of the digital image;
	1e	render a second visual indication of the first ending point on the user interface;
	1f	render, on the user interface, a first rendered link between the first starting point and the first ending point, wherein the first rendered link comprises: a first direction extending from the first starting point to the first ending point; and a first length between the first starting point and the first ending point;
	1g	identify a first set of pixels that are: parallel to the first rendered link,

		within a user-defined threshold distance from the first rendered link, and extending from the first starting point to the first ending point;
	1h	automatically shift the first set of pixels in the first direction wherein shifting the first set of pixels comprises rendering and re-rendering in a loop the first set of pixels being shifted;
	1i	receive a second starting point through the user interface, wherein the second starting point is received through a user selection of the second starting portion of the digital image, the second starting portion being different than both the first starting portion and the first ending portion;
	1j	render a third visual indication of the second starting point on the user interface;
	1k	receive, through the user interface, a second ending point associated with the second starting point, wherein the second ending point is received through a user selection of a second ending portion of the digital image, the second ending portion being different than both the first starting portion and the first ending portion;
	1l	render a fourth visual indication of the second end point on the user interface;
	1m	render, on the user interface, a second link between the second starting point and the second ending point;
	1n	identify a second set of pixels that are: parallel to the second rendered link, within the user-defined threshold distance from the first rendered link, and extending from the second starting point to the second ending point; and
	1o	automatically shift the second set of pixels between the second starting point and the second ending point, wherein shifting the second set of pixels comprises rendering in a loop the second set of pixels being shifted.
2	2a	The computer system of claim 1, wherein the executable instruction include instructions that are executable to configure the computer system to identify a particular portion of the digital image to mask,

	2b	wherein the mask prevents pixels covered by the mask from being shifted.
3		The computer system of claim 2, wherein receiving an indication of a particular portion of the digital image to mask comprises receiving through a user interface a selection of a particular pixel within the digital image.
4	4a	The computer system of claim 2, wherein the executable instructions include instructions that are executable to configure the computer system to generate the mask by:
	4b	identifying one or more edges that form a boundary around the particular pixel; and
	4c	generating the mask to cover area within the boundary.
5		The computer system of claim 1, wherein the first direction is different from the second direction.
6		The computer system of claim 1, wherein the magnitude of the shifting of the first set of pixels is proportionally related to the first length and the magnitude of the shifting of the second set of pixels is proportionally related to the second length.
7	7a	The computer system of claim 1, wherein: the first set of pixels comprise pixels that intersect the first rendered link; and
	7b	the first rendered link comprises a non-linear pathway extending between the first starting point and the first ending point.
8	8- Preamble	A method for automating the shifting of pixels within a digital image, comprising:
	8a	receiving a first indication of a first starting point through a user interface, wherein the first starting point is received through a user selection of a first starting portion of the digital image;
	8b	render a first visual indication of the first starting point on the user interface;
	8c	receiving, through the user interface, a first ending point associated with the first starting point, wherein the first ending point is received through a user selection of a first ending portion of the digital image;

	8d	render a second visual indication of the first ending point on the user interface;
	8e	render, on the user interface, a first rendered link between the first starting point and the first ending point;
	8f	selecting a first set of pixels that intersect with the first rendered link;
	8g	automatically shifting the first set of pixels along the first rendered link between the first starting point and the first ending point, wherein shifting the first set of pixels comprises rendering in a loop the first set of pixels being shifted;
	8h	receiving a second indication of a second starting point through the user interface, wherein the second starting point is received through a user selection of a second starting portion of the digital image, the second starting portion being different than both the first starting portion and the first ending portion;
	8i	render a third visual indication of the second starting point on the user interface;
	8j	receiving, through the user interface, a second ending point associated with the second starting point, wherein the second ending point is received through a user selection of a second ending portion of the digital image, the second ending portion being different than both the first starting portion and the first ending portion;
	8k	render a fourth visual indication of the second end point on the user interface;
	8l	render, on the user interface, a second rendered link between the second starting point and the second ending point;
	8m	selecting a second set of pixels that intersect with the second rendered link;
	8n	automatically shifting the second set of pixels along the second rendered link between the second starting point and the second ending point, wherein shifting the second set of pixels comprises rendering in a loop the second set of pixels being shifted.
9		The method as recited in claim 8, further comprising when a pixel selected from the first set of pixels reaches the ending point, rendering and re-rendering in the loop the pixel being shifted from the first starting point to the ending point.

10	10a	The method as recited in claim 8, further comprising receiving an indication to generate a mask over a portion of the digital image,
	10b	wherein pixels under the mask are prevented from shifting.
11		The method as recited in claim 10, further comprising receiving through a user interface a selection of a particular pixel within the digital image from which the mask should be generated.
12	12a	The method of claim 11, further comprising: identifying one or more edges that form a boundary around the particular pixel; and
	12b	generating the mask to cover area within the boundary.
13	13a	The method of claim 8, wherein: the first set of pixels further comprise: pixels that are parallel to the first rendered link, within a user-defined threshold distance from the first rendered link, and that extend from the first starting point to the first ending point; and
	13b	the first rendered link comprises a non-linear pathway extending between the first starting point and the first ending point.
14	14 preamble	A method for automating the shifting of pixels within a digital image, comprising:
	14a	accessing, from memory, a digital image file, wherein the digital image file comprises information that corresponds to individual pixels within the digital image;
	14b	receiving a first starting point through a user interface, wherein the first starting point is received through a user selection of a first beginning portion of the digital image;
	14c	rendering a first visual indication of the first starting point on the user interface;
	14d	receiving a first ending point through the user interface, wherein the first ending point is received through a user selection of a first ending portion of the digital image;

	14e	rendering a second visual indication of the first ending point on the user interface;
	14f	rendering, on the user interface, a first rendered link between the first starting point and the first ending point, wherein the first link comprises: a non-linear pathway extending between the first starting point and the first ending point; and a first length extending along the non-linear pathway;
	14g	identifying a first set of pixels that lie along the first rendered link;
	14h	automatically shifting the first set of pixels along the non-linear pathway, wherein shifting the first set of pixels comprises rendering and re-rendering in a loop the first set of pixels being shifted;
	14i	receiving a second starting point through the user interface, wherein the second starting point is received through a user selection of a second starting portion of the digital image, the second starting portion being different than both the first starting portion and the first ending portion;
	14j	rendering a third visual indication of the second starting point on the user interface;
	14k	receiving, through the user interface, a second ending point associated with the second starting point, wherein the second ending point is received through a user selection of second ending portion of the digital image, the second ending portion being different than both the first starting portion and the first ending portion;
	14l	rendering a fourth visual indication of the second end point on the user interface;
	14m	rendering, on the user interface, a second link between the second starting point and the second ending point;
	14n	selecting a second set of pixels along the second link; and
	14o	automatically shifting the second set of pixels between the second starting point and the second ending point, wherein shifting the second set of pixels comprises rendering in a loop the second set of pixels being shifted.
15	15a	The method of claim 14, further comprising identifying a particular portion of the digital image to mask,

	15b	wherein the mask prevents pixels covered by the mask from being shifted.
16		The method of claim 15, wherein receiving an indication of a particular portion of the digital image to mask comprises receiving through a user interface a selection of a particular pixel within the digital image.
17	17a	The method of claim 15, further comprising:
		generating the mask, wherein generating the mask comprises:
	17b	identifying one or more edges that form a boundary around the particular pixel; and
	17c	generating the mask to cover area within the boundary.
18		The method of claim 15, wherein: the first set of pixels further comprise: pixels that are parallel to the first rendered link, within a user-defined threshold distance from the first rendered link, and that extend from the first starting point to the first ending point.

Patent Owners Plotagraph, Inc. and Sascha Connelly (collectively “Patent Owner”) hereby submit this Response to the Petition of Lightricks Ltd. (“Petitioner”) for review of claims 1-18 of U.S. Patent No. 10,346,017 (the ‘017 Patent).

I. Introduction

The Institution Decision determined that the Petition showed a reasonable likelihood that Petitioner would prevail with respect to at least one of the claims challenged based on the IMU-Okabe and IMU-Okabe-Li grounds. Paper 34 at 31-32. As a result, the Board did not address Petitioner’s challenges based on AEM. However, Patent Owner will address each of the three grounds herein.

Reference to claim elements (i.e., 1a, 2b, etc.) herein will be as set forth in the preceding Claim Listing. Patent Owner submits that the cited prior art fails to teach or suggest several elements that are required by all of the claims of the ‘017 Patent. First, the AEM reference fails to teach or suggest selecting a set of pixels. To the extent that the asserted IMU-Okabe combination or the asserted IMU-Okabe-Li combination teach selecting a set of pixels they fail to teach or suggest shifting the selected set of pixels at all, not to mention between a starting point and an ending point. Also, none of the cited art

teaches or suggests shifting a set of pixels “in a loop” as all of the challenged claims require.

Dr. Greenspun’s supporting Declaration for Petitioner describes use of the Adobe After Effects CS6 and ImageMagick programs to perform processes which allegedly can move a pixel (Deform pin) from a starting point to an ending point and then return it to the starting point. This movement is then repeated. However, this fails to select a “set of pixels” as the claims require. There is no selection of any pixels which intersect or lie along or are parallel to and within a user-defined threshold distance of the link which extends between the starting and ending points. Perhaps more importantly, none of the cited references teach or suggest at least the element of shifting a set of pixels by rendering and re-rendering the pixels “in a loop” as recited in every independent claim of the ‘017 Patent. Dr. Greenspun’s movement of the Deform pin is from a starting point to an ending point and then repeating this movement. However, such movement of a pixel is not shifting “in a loop” as required by the claims and as described in the ‘017 Patent specification. Because the pixels in Petitioners’ cited references are not ever shifted “in a loop” the Petitioner’s challenge fails as to every single claim.

Not only is shifting “in a loop” not shown in Petitioner’s movement of a Deform pin, shifting “in a loop” is not possible at all using the Adobe After

Effects CS6 and ImageMagick programs because those programs shift pixels by merely stretching or deforming an image. The Okabe reference merely shows “infinite repetition” (i.e., repeated movement from start to end point) and therefore no movement “in a loop” as the claims require. No pixels within the image are ever looped around other pixels, and this is fatal to Petitioner’s challenge here.

II. Petitioner Has Not Shown Unpatentability of Any Claim of the ‘017 Patent

A. Overview of the ‘017 Patent

1. The Specification

The specification of the ‘017 Patent describes how a user can use the tools of paths (or digital links) with starting and ending points and masks to automate the shifting of pixels within a digital photo. In specification of the ‘017 Patent (entitled “Automated Pixel Shifting within a Digital Image”) the example of a photo of a house with grass in front is used to illustrate.

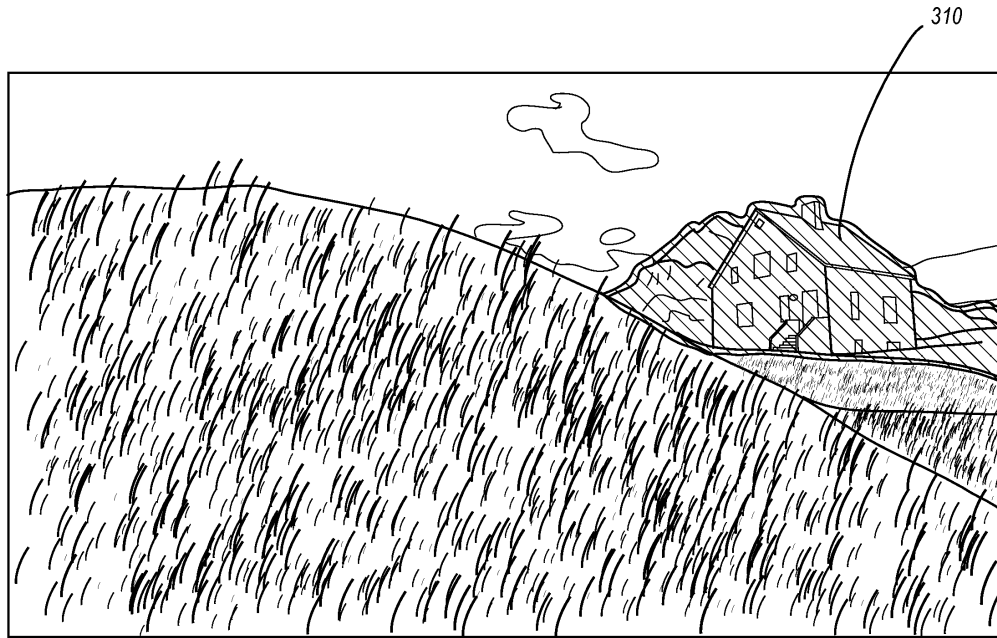
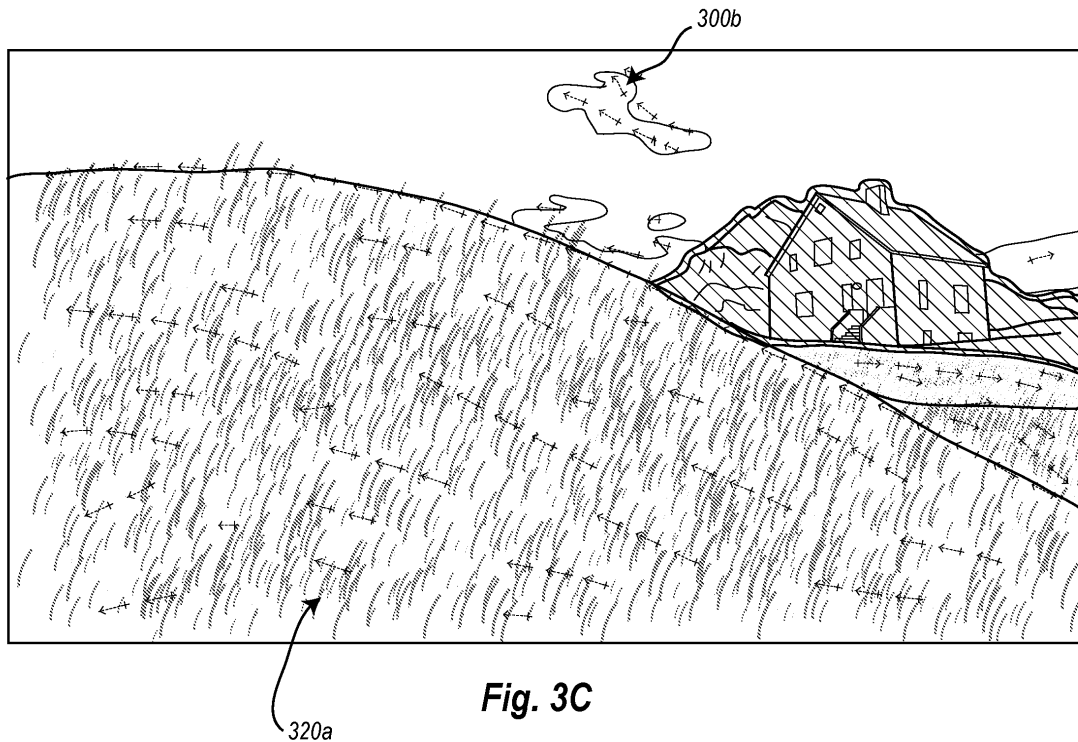


Fig. 3B

“FIG. 3B [of the ‘017 Patent] illustrates a mask 310 covering the house and the mountain behind the house, but excluding the sky and the grass in front of the house.” (‘017 Patent, col. 5, lines 17-19). “The mask prevents pixels covered by the mask from being shifted.” (*Id.*, col. 5, lines 21-22). The mask may be created using several techniques. In one, “the software architecture 200 performs an edge detection method to identify edges that form boundaries around the [user] selected area” (*Id.*, col. 5, lines 12-14). Alternatively, a user may use a manual mask painting option to create the mask. (*Id.*, col. 5, lines 25-28). In another embodiment, “stabilizer points are useable for creating a fence around an area that should not be shifted.” (*Id.*, col. 5, lines 31-33).

The patent specification then explains how digital links or paths having starting and ending points are used as tools which allow the computer to make the grass and sky in the photo appear to move.



First, a starting point for shifting of pixels is selected. A user can select starting points manually and/or “direct the software architecture 200 to automatically add a portion of the points (*Id.*, col. 5, lines 62-63). “For example, in FIG. 3C, various beginning points (also referred to herein as “starting points”) comprise the points nearest to the base of the grass and represented by crosses.” (*Id.*, col. 6, lines 5-8). Second, an ending point is selected.

The ending point is received through a user selection of an ending portion of the user interface. . . . For example, in FIG. 3C various ending points are selected on the outer periphery of the grass and are represented by arrows.

(*Id.*, col. 6, lines 13-19). Then, “. . . the software architecture creates a link between the starting point and first ending point.” (*Id.*, col. 6, lines 21-22). The links in FIG. 3C are the lines which extend between the crosses and arrows in the picture. “A link comprises a direction extending from the first starting point to the first ending point” (*Id.*, col. 6, lines 24-25).

The patent specification then describes selecting sets of pixels and illustrate the pixel shifting process which is carried out by the computer using the digital link(s).

FIG. 5A depicts the pixels 500, 510, 520 in their original state within the digital image. Once the software architecture 200 begins shifting the set of pixels in the direction of the link (extending from the starting point to the ending point) the pixels 500, 510, 520 are moved. . . . FIG. 5C depicts pixels 500, 510, 520 another step along the shift. In FIG. 5D the pixels 500, 510, 520 are shown at yet another step in the shift, but in this step the pixels 500, 510, 520 are beginning to loop around.

(*Id.*, col. 7, line 28-col. 8, line 1).

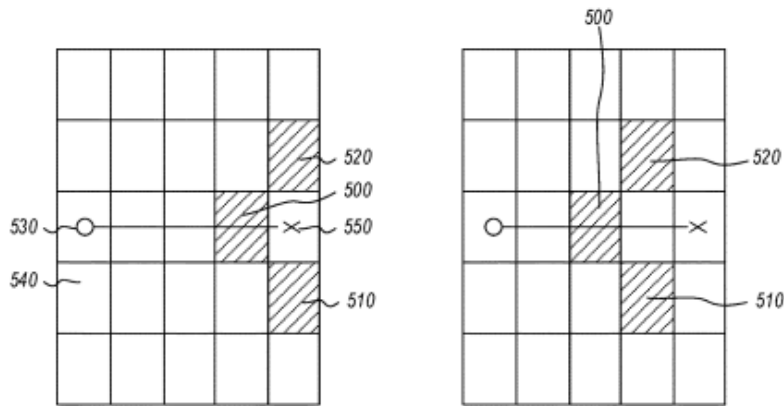


Fig. 5A

Fig. 5B

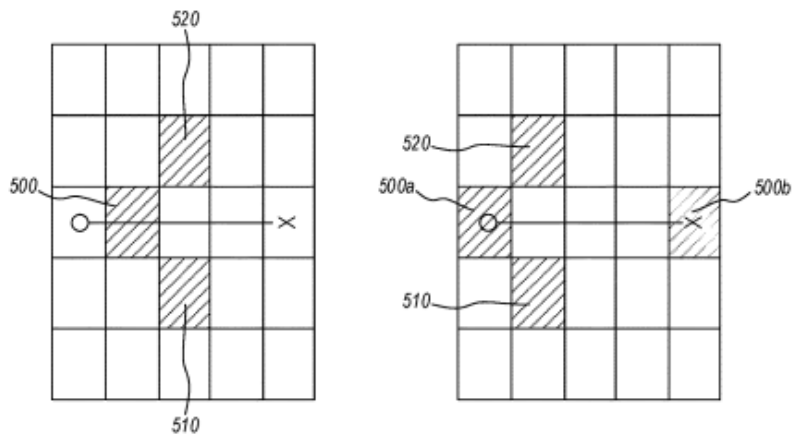


Fig. 5C

Fig. 5D

It can be seen from Figures 5A-5D that a set of three pixels, having been selected for shifting, is being shifted right to left. The specification is clear that the pixels wrap around and begin to appear at the starting point (550) as they are disappearing at the ending point (530).

For example, pixel 500a is at the left of the shift, while corresponding pixel 500b is reappearing at the right of the shift to begin the shifting loop. In various additional or alternative

embodiments, the set of pixels only begin to loop as pixels 500, 510, 520 leave the screen. For example, only pixel 500a reaches the end of the link (i.e., ending point 530), pixel 500a disappears on the next step. Upon disappearing, pixel 500b then reappears at the beginning of the link (i.e., starting point 550) to start the loop. Additionally, in at least one additional or alternative embodiment, the pixels 500, 510, 520 fade in and out at the beginning and ending of each loop. For example, upon reaching the end of the link, pixel 500a fades out of the image by slowly growing transparent, such that the disappearance of pixel 500a is not abrupt or jarring. Similarly, pixel 500b can fade in such that its appearance is not abrupt or jarring.

Id., col. 8, lines 1-18.

The specification also describes an alternative embodiment for shifting pixels in a loop.

For example, in at least one embodiment, the pixels immediately begin cycling as soon as the shifting begins. For example, instead of what is shown in FIGS. 5A-5D, as soon as pixel 500 is shifted to its new position depicted in FIG. 5B, pixel 500B would appear in the position depicted in FIG. 5D. As such, the pixels begin looping and blending into each other immediately.

Id., col. 8, lines 21-27.

2. Exemplary Claim

Claim 1 of the '017 Patent is reproduced here:

1. A computer system for automating the shifting of pixels within a digital image, comprising:

one or more processors; and

one or more computer-readable media having stored thereon executable instructions that when executed by the one or more processors configure the computer system to perform at least the following:

access, from memory, a digital image file, wherein the digital image file comprises information that corresponds to individual pixels within the digital image;

receive a first starting point through a user interface, wherein the first starting point is received through a user selection of a first beginning portion of the digital image;

render a first visual indication of the first starting point on the user interface;

receive a first ending point through the user interface, wherein the first ending point is received through a user selection of a first ending portion of the digital image;

render a second visual indication of the first ending point on the user interface;

render, on the user interface, a first rendered link between the first starting point and the first ending point, wherein the first link comprises:

a first direction extending from the first starting point to the first ending point; and

a first length between the first starting point and the first ending point;

identify a first set of pixels that are:

parallel to the first rendered link, and

within a user-defined threshold distance from the first rendered link, and

extending from the first starting point to the first ending point;

automatically shift the first set of pixels in the first direction, wherein shifting the first set of pixels comprises rendering and re-rendering in a loop the first set of pixels being shifted;

receive a second starting point through the user interface, wherein the second starting point is received through a user selection of a second ending portion of the digital image, the second ending portion being different than both the first starting portion and the first ending portion;

render a third visual indication of the second starting point on the user interface;

receive, through the user interface, a second ending point associated with the second starting point, wherein the second ending point is received through a user selection of a second ending portion of the digital image, the second ending portion being different than both the first starting portion and the first ending portion;

render a fourth visual indication of the second end point on the user interface;

render, on the user interface, a second link between the second starting point and the second ending point;

identify a second set of pixels that are:

parallel to the second rendered link,

within the user-defined threshold distance from the first rendered link, and

extending from the second starting point to the second ending point; and

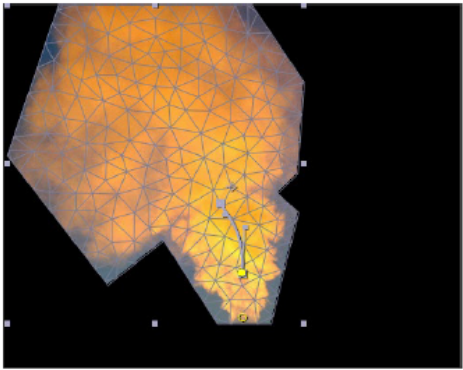
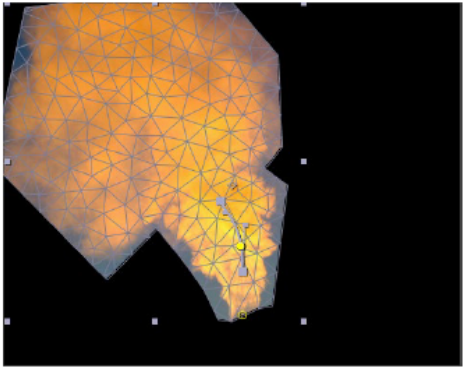
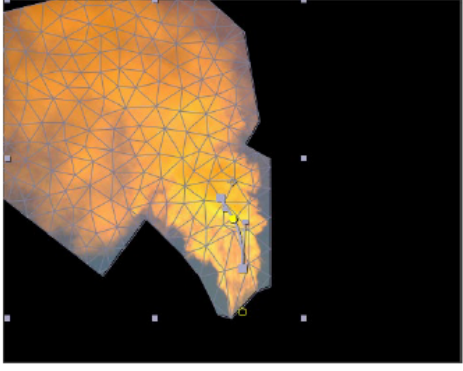
automatically shift the second set of pixels between the second starting point and the second ending point, wherein shifting the second set of pixels comprises rendering in a loop the second set of pixels being shifted.

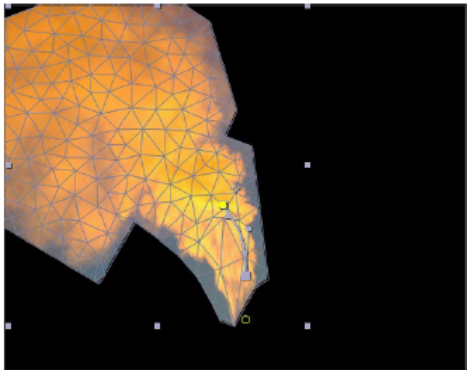
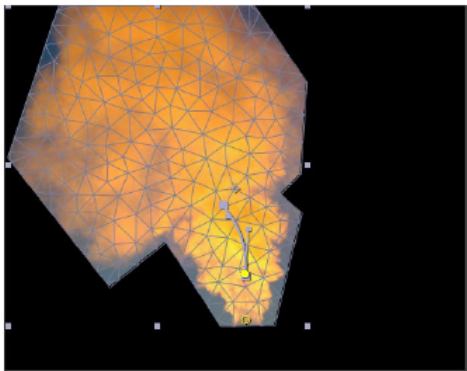
EX1001, claim 1.

B. The Cited Art

1. The AEM Reference

Ground 1 is the challenge to claims 1-18 over the “AEM” reference. “AEM” is how Petitioner refers to the User Manual for the Adobe After Effects CS6 software program (AECS6”). Petitioner relies upon the testimony of Dr. Greenspun to purportedly demonstrate the use of Adobe After Effects to shift a pixel “in a loop.” The Petition notes that: “[t]he following table includes a series of cropped screenshots of AECS6 showing the result of applying a “loopOut()” expression to the exemplary composition provided in Section VI.A.2.b.” Paper 1 at 31. Patent Owners reproduce the first five of these cropped screenshots below for illustration:

Animation Time Point	Composition Frame
(0:00:00:00)	
(0:00:01:00)	
(0:00:02:00)	

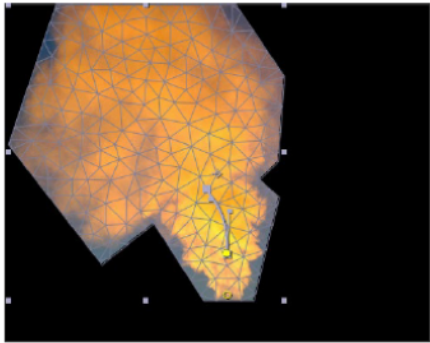
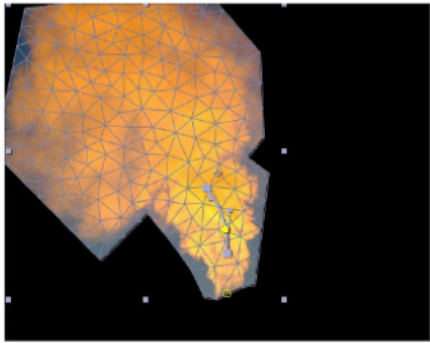
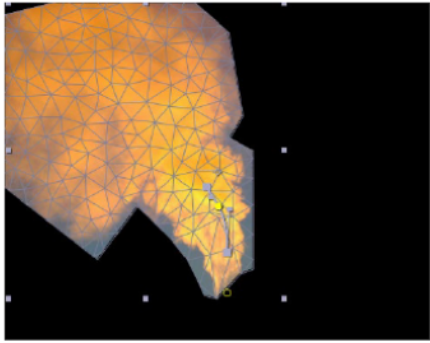
Animation Time Point	Composition Frame
(0:00:02:23)	
(0:00:03:01)	

Paper 1 at 32-33. What you are looking at is a small yellow dot (i.e., the “Deform pin”) which is at a starting point of a grey path at the beginning of a shift at Animation Time Point (00:00). The Deform pin is shifted to intermediate points along the path at Animation Time Points (01:00) and (02:00). At Animation Time Point (02:23) the Deform pin has reached the ending point of the grey path. At Animation Time Point (03:01) the Deform pin has been returned to its starting point as the keyframes are reset to the beginning (see Paper 1 at 33). Petitioner does not explain how repeating “a segment of time” is shifting a set of pixels in a loop.

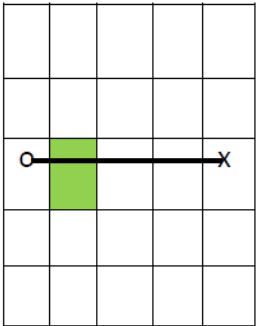
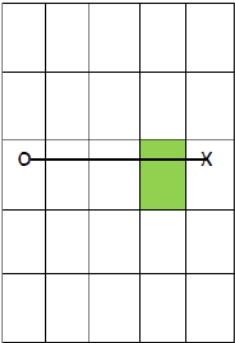
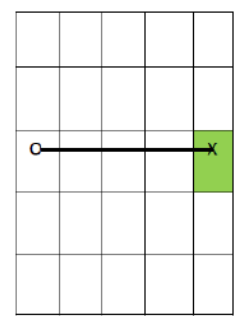
Petitioner does discuss the creating “different ending keyframes and motion paths for . . . additional Deform pins in the same way as for the first Deform pin” using the AEM reference (Paper 1 at 35), so the additional Deform pins are also merely being moved from a starting point to an ending point and then, as keyframes are reset, back to their starting points.

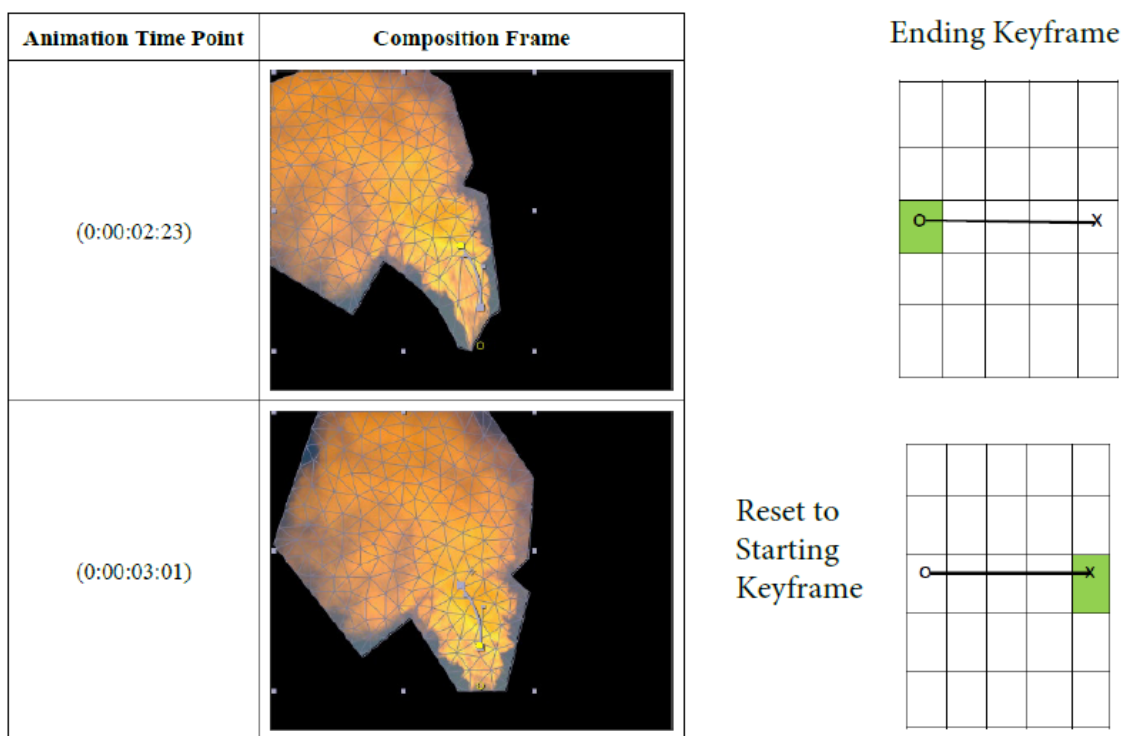
Patent Owners submit that movement of a Deform pin using the AEM reference could be modeled using the 5x5 grid featured in Figure 5 of the ‘342 Patent, and it would appear as follows with a Deform pin being represented as a single green pixel¹:

¹ Dr. Greenspun’s shifting of the Deform pin is indistinguishable from the shifting of a single pixel.

Animation Time Point	Composition Frame
(0:00:00:00)	
(0:00:01:00)	
(0:00:02:00)	

Starting Keyframe





Petitioner is moving a pixel from a starting point (represented by the “x”) to an ending point (represented by the “o”) and then returning it to the starting point.

2. The IMU Reference

Grounds 2 and 3 of the Petition allege that the instruction manual for ImageMagick Usage (Version 6)(i.e., “IMU”) “teaches how to incrementally apply IMV6’s effects to an image to produce a sequence of frames that, when played sequentially, result in an animation of the original image.” Petition at 57. Further, the Petition alleges that “. . . IMU teaches applying a ‘-loop’ operator to specify the ‘[n]umber of times the GIF animation is to cycle through the image sequence before stopping.’” Paper 1 at 59. IMU, then,

teaches precisely the same sequential process that AEM does. A number of frames are played sequentially which presumably show the movement of a pixel from a starting point to an ending point. Then the frames are replayed from the beginning. Petitioner provides no illustration for the movement of a pixel using IMU as it did with AEM. Therefore, Patent Owners submit that the 5x5 grid modeling of movement of a pixel using IMU would be the same as it would for AEM, as set forth above.

The Petition argues that IMU teaches performing “Shepard’s Distortion” as a warping function and thus pixel shifting. Paper 1 at 64-65. According to IMU, one “can think of [Shepard’s Distortion] as equivalent to a thick block of ‘taffy’ representing the source image, having pins driven into it and then the pins moved around.” EX1005 at 59. Shepard’s Distortion stretches portions of a still image.

3. The Okabe Reference

The Petition cites Okabe as teaching an algorithm that is “able to generate a flow animation in the in the target image corresponding to the flow direction and speed specified by the user’s strokes, and permits ‘infinite repetition’ of such an animation.” Paper 1 at 61. However, this

mischaracterizes what the “user’s strokes” actually do and ignores the fact that Okabe does not shift pixels in a still image.

In Okabe, a user “inputs a single target image and an alpha matte that extracts the fluid region of interest.” EX1009 at 3. A user can provide “a few optional suggestions about fluid motion, e.g., sketches of flow direction, shown as orange arrows.” *Id.* at 2. It is not apparent from Okabe that the orange arrows in Fig. 1 are “user-viewable” on a user interface as opposed to merely being illustrations added to the Okabe article by the author.

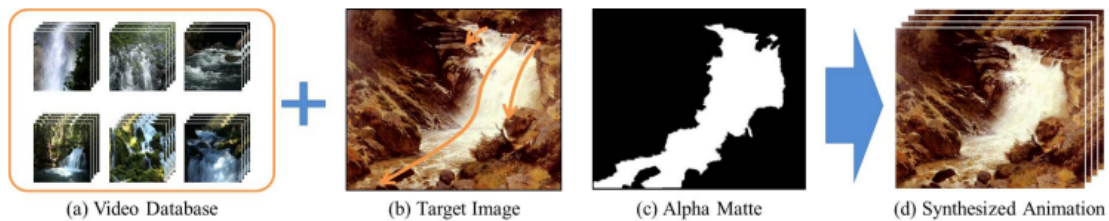


Figure 1: We employ a database of video examples of fluids (a). The user specifies a target image (b) with a few optional suggestions about fluid motion, e.g., sketches of flow direction, shown as orange arrows. The user also provides an alpha matte of the region of interest (c). The system synthesizes an animation (d).

Okabe then “employ[s] a video database that includes hundreds of video examples of fluids, which therefore helps the user to synthesize better quality animations” EX1009 at 1. Okabe “cut[s] each video example into smaller pieces (Fig. 2-c).” *Id.* at 3.

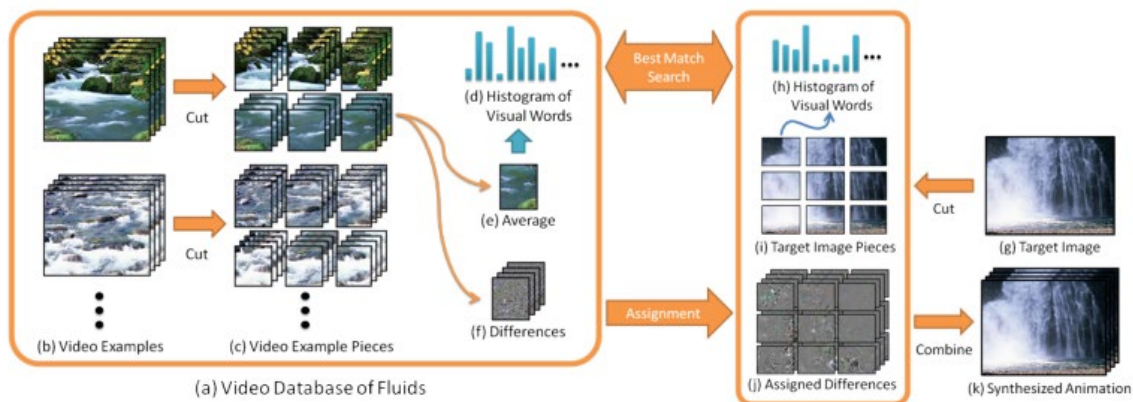


Figure 2: System overview.

These video examples are then patched on by “assigning them onto parts of the target image” *Id.* at 2. The video patches are selected by Okabe’s process for best match based upon appropriateness as determined by three criteria: appearance similarity, smoothness of motion fields, and appearance similarity between neighboring assigned pieces. See *Id.* at 5. When a user “provide[s] a desired motion field by sketching the flow direction and painting a speed map, [this] is used as a constraint in the assignment process.” *Id.* at 3.

4. The Li Reference

Ground 3 of the Petition contends that claims 2-4, 10-12, and 15-17 are obvious in view of IMU in combination with Okabe and Li. But Petitioner does not contend that Li teaches or suggests rendering and re-rendering pixels in a loop. The Petition cites Li as teaching “. . . that its ‘Lazy Snapping’ tool

allows a user to easily perform ‘image cutout’ – separating and removing an image’s background, leaving only the foreground visible – by using a ‘novel image segmentation algorithm’ to detect a boundary between the foreground and background based on ‘a quick object marking step’ and ‘a simple boundary editing step’ performed by the user.” Paper 1 at 88. Li has nothing to do with shifting pixels “in a loop.”

C. Claim Construction

To the extent that it is necessary to resolve any dispute in this matter, Patent Owner proposes the construction of certain claim language below pursuant to the broadest reasonable interpretation (BRI) for the sole purpose of this *inter partes* proceeding. The scope of patent claims is determined giving claims their broadest reasonable construction consistent with the specification; the words of a claim are given their plain meaning unless the plain meaning is inconsistent with the patent specification as understood by a person of skill in the art at the time of the invention. “Even under the broadest reasonable construction rubric, however, the board must always ‘consider the claims in light of the specification and teachings in the underlying patent.’” *In re: Power Integrations, Inc.*, 884 F.3d 1370, 1375 (Fed. Cir. 2018) (citing *In re CSB-Sys. Int’l, Inc.*, 832 F.3d 1335, 1341 (Fed. Cir. 2016)). The patent

specification “is the single best guide to the meaning of a disputed term” in a claim. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1315 (Fed. Cir. 2005)(en banc).

1. Construction of shifting a set of pixels “in a loop”

Patent Owner proposes that the claim language shifting of a set of pixels by rendering and re-rendering “in a loop” be construed as meaning that a leading pixel in the selected set of pixels be rendered at the starting point of the shift and behind one or more trailing pixels which it previously was leading. The specification of the ‘017 Patent supports Patent Owner’s construction of this claim language.

a. The Primary Embodiment Describes a Leading Pixel Looping Around Behind Trailing Pixels

The ‘017 Patent specification clearly explains that, in the primary illustrated embodiment, rendering and re-rendering pixels in a loop means that, during shifting, pixels loop around and begin to appear at the starting point (550) as they are disappearing at the ending point (530). In the illustrated embodiment:

For example, pixel 500a is at the left of the shift, while corresponding pixel 500b is reappearing at the right of the shift to begin the shifting loop. In various additional or alternative

embodiments, the set of pixels only begins to loop as pixels 500, 510, 520 leave the screen. For example, once pixel 500a reaches the end of the link (i.e., ending point 530), pixel 500a disappears on the next step. Upon disappearing, pixel 500b then reappears at the beginning of the link (i.e., starting point 550) to start the loop.

EX1001 at col. 8, lines 1-11. The leading pixel in the shift “loops” around so that it follows the trailing pixels. This is illustrated by Fig. 5D (annotated below) wherein the leading pixel (500) is appearing at 500b behind the trailing pixels (510, 520). Specifically, this happens:

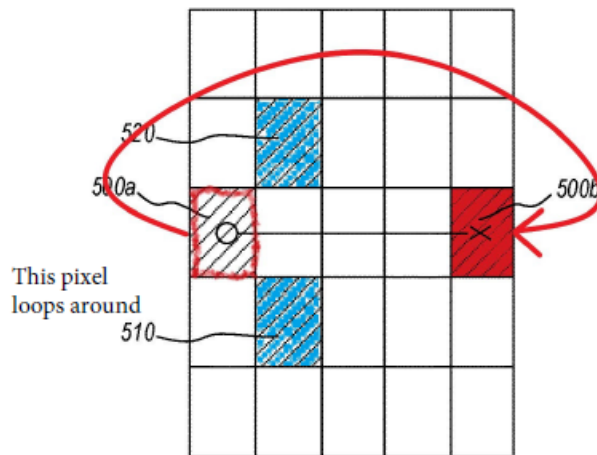


Fig. 5D

This is the “loop around” which characterizes shifting a set of pixels “in a loop.” The language of the description supports this interpretation. “In Fig. 5D the pixels 500, 510, 520 are shown at yet another step along the shift, but

in this step the pixels 500, 510, 520 are beginning to loop around.” EX1001, col. 7, line 66-col. 8, line 1 (emphasis added). In other words, leading pixel 500 does a “loop around” in the direction opposite of the direction of shift (shift is right-to-left) to end up behind (to the right of) the trailing pixels 510, 520.

b. The Alternate Embodiment Describes a Leading Pixel Looping Around Behind Trailing Pixels

In addition to what is shown in Figs. 5A-5D, the specification describes an alternative embodiment in which the leading pixel (500) also does a “loop around” to appear behind the trailing pixels (510, 520) as soon as the shift begins:

For example, in at least one embodiment, the pixels immediately begin cycling as soon as the shifting begins. For example, instead of what is shown in FIGS. 5A-5D, as soon as pixel 500 shifted to its new position depicted in FIG. 5B, pixel 500B would appear in the position depicted in FIG. 5D. As such, the pixels begin looping and blending into each other immediately.

EX1001, col. 8, lines 21-27. If illustrated, this embodiment of the loop around would appear as follows:

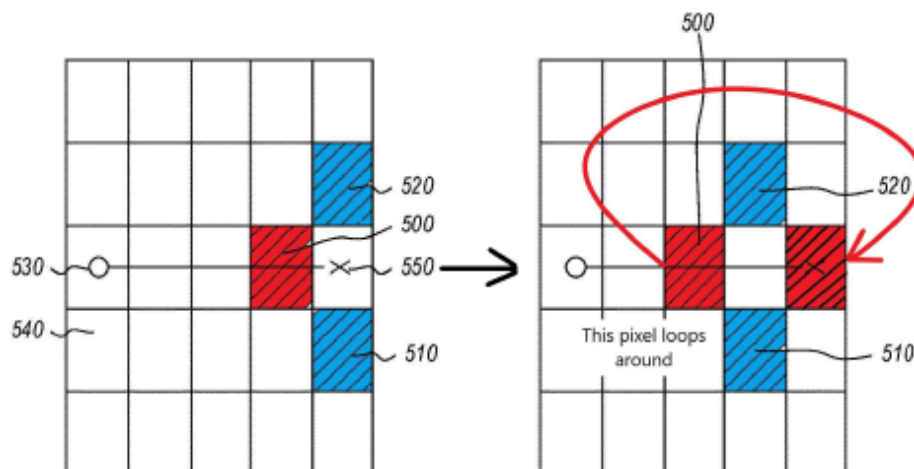


Fig. 5A

Fig. 5B

Patent Owner submits that because all the described embodiments in the specification have a leading pixel being looped around and rendered behind trailing pixels, this is the correct construction for a set of pixels being rendered and re-rendered “in a loop.” Conversely, if no pixel does the “loop around,” as taught by the specification, a set of pixels has not been shifted “in a loop.”

d. The Petitioner’s Implicit Proposed Construction for Shifting “in a Loop” is Clearly Incorrect

As mentioned previously, the Petition does not explicitly set forth any proposed construction for the shifting “in a loop” claim term. But Petitioner apparently is contending here that shifting a set of pixels in a loop means that a single pixel is simply moved from a starting point to an ending point and

then returned to the starting point. This is apparent from, for example, Petitioner's discussion of the AEM reference's teaching use of the Adobe After Effects program to shift a Deform pin:

Thus, after the Deform pin reaches the ending position/keyframe, its position will be reset to the starting position/keyframe, and keyframe interpolation will repeat."

Paper 1 at 16.

The problem with the Petitioner's apparent construction of the term "in a loop" is that it is not supported at all by the specification. If Petitioner's proposed construction for shifting pixels "in a loop" were reasonable, or even possible, Petitioner could point to the specification to show where it is described. But, in fact, the specification fails to show any pixel being moved from the starting point to the ending point and then returning to the starting point. In Figures 5A-5D none of the pixels 500, 510, 520 are moved from the starting point 550 to the ending point 530 and then returned to the starting point 550.

First, consider the case of the trailing pixels 510, 520 which begin in Fig. 5A at the starting point 550 and which are highlighted here in blue:

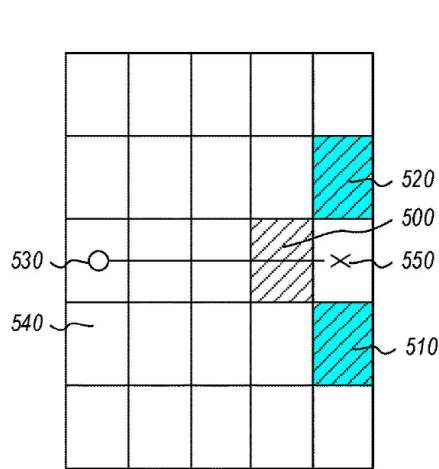


Fig. 5A

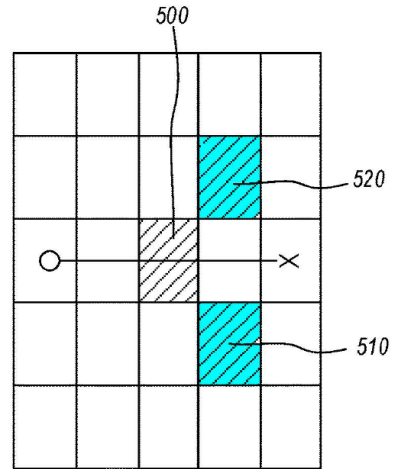


Fig. 5B

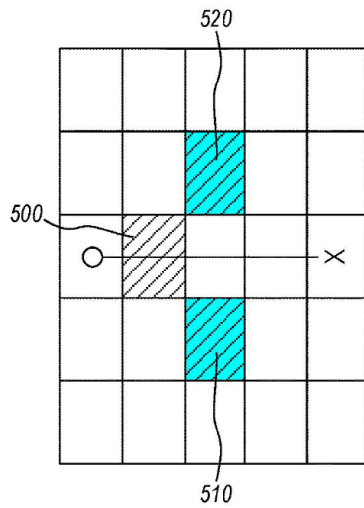


Fig. 5C

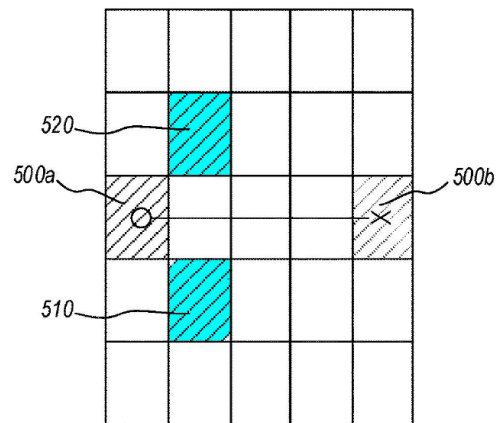


Fig. 5D

In the illustration of shifting in a loop, these pixels never return to the starting point 550 where they began. In fact, they never even reach the ending point 530. So the movement of these pixels does not support Petitioner's implicit interpretation of shifting "in a loop."

Next, consider the case of the leading pixel (500) which is highlighted in red below:

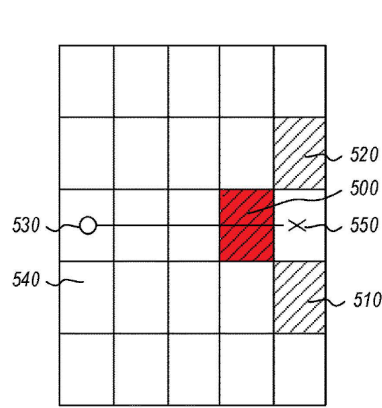


Fig. 5A

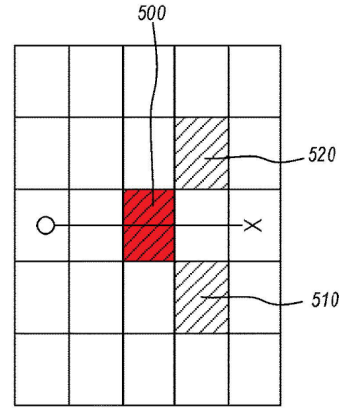


Fig. 5B

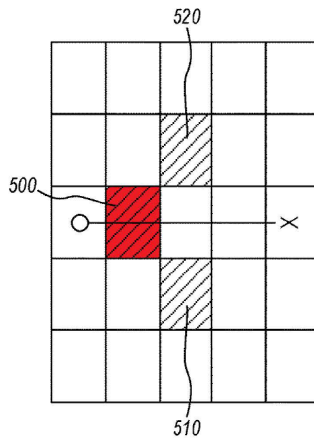


Fig. 5C

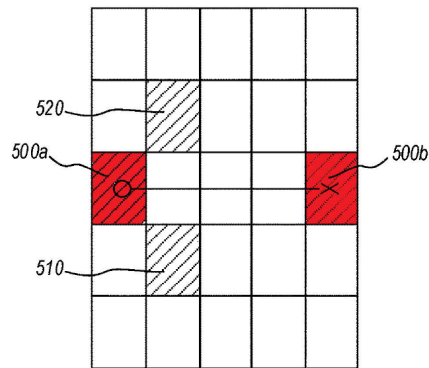


Fig. 5D

In Figure 5A (the start of the shift), pixel 500 is located one space ahead of (to the left of) the starting point 550. In Figure 5D (ending of the shift), pixel 500 has been rendered at the starting point 550, but that is one space behind (to the right of) where it started out. Pixel 500 does not return to the spot

where it started. Therefore, the movement of pixel 500 does not support Petitioner's implicit construction for shifting "in a loop" either. If Petitioner's implied construction of "in a loop" were followed, the pixel 500 would reappear where it started in Fig. 5A. But instead, it appears one space to the right of where it began, which is only possible because the pixel 500 is looped around.

Petitioner's implicit construction of the term "in a loop" is incorrect, unreasonable, and impossible because it is not supported by the specification. In *In re Power Integrations, Inc.* the Federal Circuit rejected the claim construction adopted by the board for a number of reasons, including that ". . . the board's unduly broad reading of the language of claim 1 is unsupported by the specification." 884 F.3d at 1376. Patent Owner's proposed claim construction is the only construction that is supported by the specification and the only construction that is reasonable. Further, the Board determined in a related case involving this same asserted art that ". . . '[m]erely returning the Deform pin back to its starting [point] is not shifting it "in a loop.'" IPR2023-00152, Paper 24 at 16.

D. Legal Standards

“[P]etitioner has the burden from the onset to show with particularity why the patent is challenges is unpatentable.” *Harmonic Inc. v. Avid Tech, Inc.*, 815 F.3d 1356, 1363 (Fed. Cir. 2016). This burden never shifts to the Patent Owner. *Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015).

Petitioner has the burden of proving that the prior art it relies upon shows every element of the challenged claims. “[A] prior art reference must provide every element of the claimed invention arranged as in the claims in order to anticipate.” *Microsoft Corp. v. Biscotti, Inc.*, 878 F.3d 1052, 1069 (Fed. Cir. 2017). Also, if a single element of a claim is absent from the prior art, the claim cannot be considered obvious. *See CFMT, Inc. v. YieldUp Int’l Corp.*, 349 F.3d 1333, 1342 (Fed. Cir. 2003) (“Obviousness requires a suggestion of all limitations in a claim.”).

E. Petitioner Fails to Invalidate Any Claim of the ‘017 Patent

The references asserted by Petitioner do not teach or suggest all the elements of the independent claims of the ‘017 Patent. AEM does not select or identify a set of pixels which are shifted along a pathway between a starting point and ending point. The IMU-Okabe and IMU-Okabe-Li combinations do not teach or suggest identifying/selecting a set of pixels which are then

shifted in a first direction. Indeed, Okabe does not shift any pixels at all but rather teaches assigning numerous video patches onto the target image meaning that any motion resulting in the final synthesis is provided by the video patches rather than shifting of pixels in the target image.

The alleged IMU-Okabe combination fails to teach or suggest the elements of rendering on the user interface visual indications of first or second starting points, first or second ending points, or rendered links between starting and ending points. There is no evidence that Okabe's orange arrows are user-viewable or appear on a user interface. Further, Okabe's user drawn orange arrows do not perform the function of specifying a direction of shift or of motion in the region of interest.

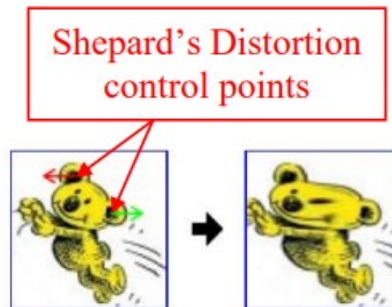
Also, each of the independent claims (1, 8, 14), and thus all the claims, recite rendering "in a loop" a selected set of pixels. At least this element is entirely missing from every cited reference. None of the prior art teaches or suggests it.

1. The prior art does not teach or suggest receiving starting or ending points through a user interface

Each of the independent claims (1, 8, 14) recite elements in which a first starting point (elements 1b, 8a, 14b) is received through a user interface. Claim 1 also recites a second starting point that is received through the user

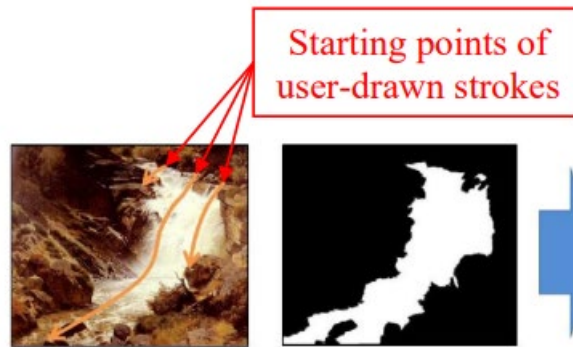
interface (element 1i). These elements are not taught or suggested by any of the cited art. For AEM, there is no starting point received through the user interface. What is shown at Animation Time Point (0:00:00:00) [Paper 1 at 38] is merely the initial position of a Deform pin before it is moved. It is not the starting point for the shifting of any set of pixels or any other pixels aside from the Deform pin.

In IMU, no starting point is received through the user interface. What is shown in the Petition at p. 68 is merely the initial position(s) of two control points before they are moved in opposite directions. These control points are not shown to be the starting points for the shifting of any set(s) of pixels or other pixels aside from the control points.



Paper 1 at 68.

Okabe also does not show receiving a starting point through a user interface at all. The Petition points to the beginning portions of Okabe's orange arrows as being the received starting points.

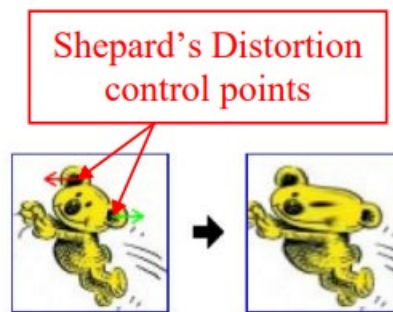


Paper 1 at 69. But these points have not been shown to be the starting points for anything. Okabe’s orange arrows are merely optional suggestions about fluid motion. They are not pathways along which pixels are ever shifted and, therefore, the beginnings of such arrows do not function as starting points for any pathways of shifting. *See* IPR2023-00152, Paper 7 at 22 (“ . . . Okabe’s strokes merely provide a general indication of the desired direction of movement – there is no indication that they define pathways that pixels are shifted along.”).

Independent claims 1, 8 and 14 also recite elements in which first ending points are received through a user interface (elements 1d, 8c, 14d). Claim 1 also recites a second ending point that is received through the user interface (element 1k). Patent Owner submits that these elements are not taught or suggested by the cited art for the same reasons that the starting points are not. For the AEM reference, there is no ending point received through the user interface because what is shown at Animation Time Point (0:00:02:23)

[Paper 1 at 39] is merely the ending position of the Deform pin and not the ending point for the shift of any set of pixels.

In IMU, no ending point is received through a user interface. What is shown in the Petition with the koala image (right side) is merely the final positions of two control points after they are moved.



Paper 1 at 74.

Okabe does not show receiving an ending point through a user interface either. The Petition points to the arrowheads of Okabe's orange arrows at being received ending points. Paper 1 at 69-70. But because the orange arrows do not function as pathways along which pixels are shifted, the arrowheads do not serve as ending points for any such shift.

2. The prior art does not teach or suggest creating a link between the starting and ending points

Independent claims 1, 8 and 14 require (elements 1f, 8e, 14f) "a first rendered link between the first starting point and the first ending point."

Claim 1 also requires “render[ing], on the user interface, a first link between the second starting point and the second ending point.” (element 1m). Claim 8 requires “a second rendered link between the second starting point and the second ending point” (element 8l).

The prior art does not teach or suggest these steps. Because starting points and ending points for shifting of pixels are not ever received through the user interface, as noted above, no link between such points is created.

3. The AEM prior art does not teach or suggest identifying/selecting a set of pixels as the claims require

Each of the independent claims (1, 8, 14) recite similar elements which use slightly different language. Independent claim 1 recites the step of: “identify a first set of pixels that are: parallel to the first rendered link, within a user-defined threshold distance from the first rendered link . . .” Independent claim 8 recites the step of “selecting a first set of pixels that intersect with the first rendered link . . .” Independent claim 14 recites the step of “identifying a first set of pixels that lie along the first rendered link . . .” The prior art fails to teach or suggest these steps.

The ‘017 Patent illustrates a set of pixels (500, 510, 520) in which two of the pixels (510, 520) are at the starting point (550) and pixel 500. This set of pixels intersects the link (as per claim 8) that extends between the starting

point 550 and the ending point 530. This set of pixels also lies along the link between the first starting point and first ending point, as claim 14 recites.

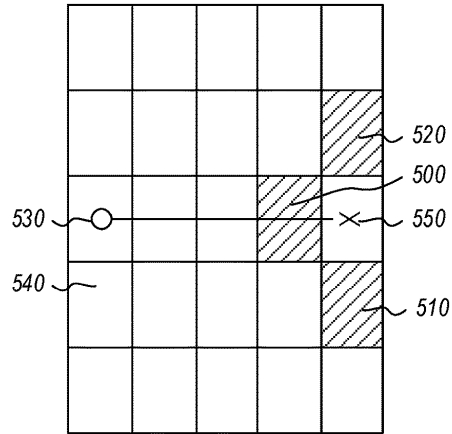
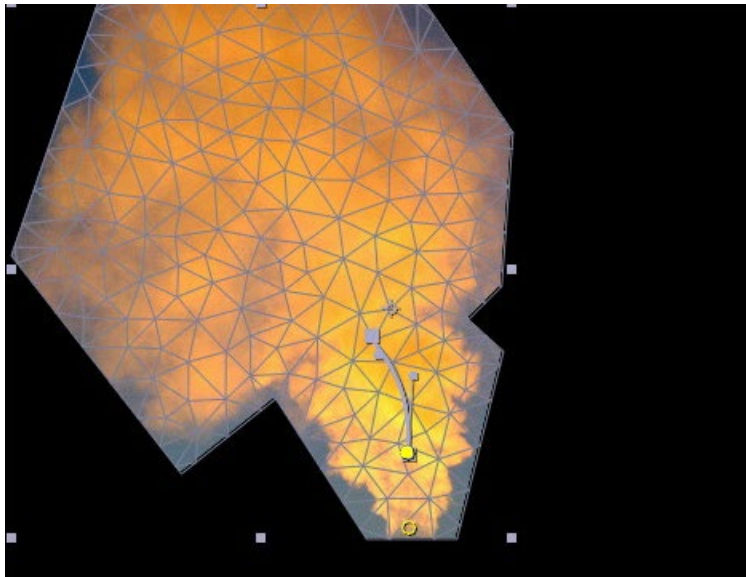


Fig. 5A

Petitioner does not select a set of pixels that meets any of these claim requirements with the AEM reference. Petitioner only selects a Deform pin with AEM.



Paper 1 at 32. This is essentially a single pixel which is located at the starting point of the shift in the same manner that pixels 510, 520 are in Figure 5. There are no other pixels selected which are parallel to and within a user-defined threshold distance from the link (claim 1). The Deform pin is merely the pixel which is at the first starting point. Even if the Deform pin were composed of more than just a single pixel, it is shifted in the same manner as a single pixel would be: from the starting point to the ending point and then reset to the starting point.

4. The IMU-Okabe combination does not identify/select a “set of pixels” to be shifted

The Petition contends that “[t]he IMU-Okabe Combination’s user . . . applies a matte to select pixels . . . to be in a region of interest and thus animated.” Paper 1 at 73. Patent Owner submits that this is nowhere taught or suggested by these references. No pixels within the region of interest selected with Okabe’s “matte” would be shifted given the teachings of the art. As noted previously, Okabe synthesizes fluid flow animation on a target image using a database of video examples of fluid flow. The database provides video examples of flow, and Okabe then cuts up selected video examples “into small pieces.” EX1009 at 3. Instead of shifting pixels within

the target image, sections of the selected pixels are replaced or patched over with the small pieces of video. See EX1009 at 3 (“Next we . . . assign video example pieces that have appearances similar to target image pieces.”). No pixels within the area selected by the matte are ever shifted with Okabe. Motion that occurs in the final synthesis is provided by the video patches rather than any pixels that are shifted.

Patent Owner submits that one of skill in the art reading Okabe would not be motivated to use IMU’s Shepard’s Distortion at all given the teachings of Okabe. Because the pixels in Okabe’s selected region are being patched over with video pieces, there is no utility in using Shepard’s Distortion to ‘animate’ them. Further, the video pieces provide the animation themselves, and using Shepard’s Distortion (to the extent it could be applied to a video piece) would defeat the purpose of the provided video example pieces/patches.

Additionally, Okabe teaches away from pixel-based shifting approach to teach a patch-based method. Okabe notes that “[o]ur method differs from Image Analogies in that it is not pixel-based but rather is a patch-based texture synthesis, similar to Image Quilting.” *Id.* at 3. One reading Okabe would not look to a pixel-based shifting mechanism such as Shepard’s Distortion in view of this distinction.

5. The IMU-Okabe combinations do not teach or suggest shifting a set of pixels in a direction

The Petition asserts that “. . . the IMU-Okabe Combination user automatically generates animation frames by shifting the selected pixels in the stroke’s direction using IMV6’s Shepard’s Distortion – “automatically shift the first set of pixels in the first direction.” Paper 1 at 73. In fact, the IMU-Okabe combination fails to do so as does the IMU-Okabe-Li combination. As noted previously, Okabe teaches synthesizing animation by patching over a target image with small video patches. Patent Owner notes that movement of pixels in these patches is confined to be within the discrete patches themselves. There is no mechanism, at least not that Petitioner has shown, for pixels to move between patches such that they could move from the starting point to the ending point. IMU also does not teach shifting of a set of pixels but rather a single control point. See Paper 1 at 72 (“. . . IMV6’s user shifts pixels as claimed by using Shepard’s Distortion to place and incrementally move a ‘control point’ on an image frame-by-frame in the user’s specified direction and speed.”). Any other pixels in an image are translated only as a result of the image being stretched. One of skill in the art would not look to Okabe for a solution to selecting a set of pixels because IMU is not concerned

with selecting a set of pixels. IMU is only concerned with selecting a single point: the control point.

Also, the orange arrows of Okabe do not provide a “first direction” or a “second direction” for the shifting of pixels. As noted above, Okabe assigns video patches to the target image, and it is the video patches assigned to the target image which provide the motion in the final synthesized result. No pixels in the original target image are ever shifted. When a user “provide[s] a desired motion field by sketching the flow direction and painting a speed map, [this] is used as a constraint in the assignment process.” EX1009 at 3. Therefore, providing a user-drawn suggestion would only function to remove unsuitable video patches from the pool of potential video patches which could be assigned to the target image. Patent Owner submits that this is not sufficient for a user-drawn suggestion to function to provide a direction. Petitioner also doesn’t show that removal of unsuitable video patches would not be done anyway by Okabe’s “best match” assignment process. At the very least, the removal of unsuitable potential video patches does not inherently function to provide a direction, and that is sufficient to show the absence of this element from the prior art. “The fact that a certain result or characteristic may occur or be present in the prior art is not sufficient to establish the inherency of the result or characteristic.” *In re Rijckert*, 9 F.3d 1531, 1534

(Fed. Cir. 1993). Okabe outright tells you that there are cases in which the “optional suggestions” of the user-drawn orange arrows are unnecessary in the assignment process. Okabe notes:

Waterfalls or regions with flames tend to have successful assignment [of video patches] even without any user input because they have strong image edges that correlate to the flow direction. For example, waterfalls and flames always move from top to bottom and from bottom to top, respectively.

EX1009 at 7. The user-drawn arrows don’t inherently function to provide a direction and therefore no pixels, to the extent any pixels are, are shifted in the ‘direction’ provided by the user. Patent Owner submits that none of the cited art teaches shifting a set of pixels in a first direction or in a second direction. That teaching is only provided by Patent Owner’s specification.

6. The prior art does not teach or suggest rendering indications on the user interface

With reference to the preceding Claim Listing, claim elements 1c, 8b and 14c recite the step of “render[ing] a first visual indication of the first starting point on the user interface.” Claim elements 1e, 8d, and 14e recite similar rendering of a first ending point on the user interface. Claim elements 1f, 8e, and 14f recite rendering, on the user interface, a first rendered link between the first starting point and the first ending point. The claims also

recite similar steps for rendering on the user interface a second starting point (1i, 8i, 14j) a second ending point (1l, 8k, 14l) and a second rendered link (1m, 8l, 14m). These elements are not taught or suggested by the cited art. For instance, with regard to the AEM reference, the Petition contends that the first visual indication [of the first starting point] occurs when AECS6 makes this starting keyframe ‘visible in the Composition panel and Layer panel . . .’” Paper 1 at 8. The Petition then presents an annotated screenshot from AECS6 despite admitting that it is not a patent or printed publication. *See* Paper 1 at 19-20, n. 6. In fact, no portion of AEM teaches or suggests rendering a visual indication of a first starting point on a user interface. The same would be true for a first ending point, a first rendered link, second starting and ending points and a second rendered link.

The asserted IMU-Okabe combination also fails to teach or suggest rendering these visual indications on a user interface. There is no indication in IMU that when “a user places a ‘control point’ on a pixel located at a user-specified coordinate of the image [Paper 1 at 67] any indication is rendered on a user-interface. The Petition refers several times to Okabe’s user drawn strokes as being “user-viewable.” *See* Petition at 68 (“ . . . Okabe’s user-drawn stroke includes a user-viewable indication of a starting point, direction, and speed across the animation”) citing to pages 3 and 7 of Okabe. However,

those pages and no portion of Okabe ever describe the user-drawn strokes as being “user-viewable.” There is no evidence to suggest that user-drawn strokes are rendered on a user interface at all. There is no evidence that the orange arrows in Okabe’s article are anything other than graphics added to the image by the author to illustrate optional suggestions about fluid motion.



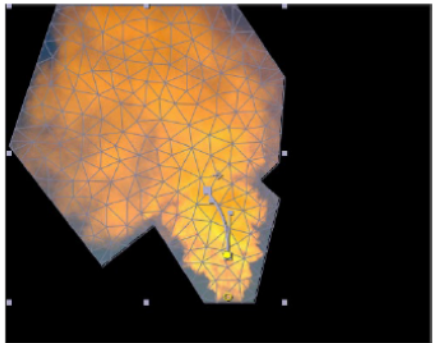
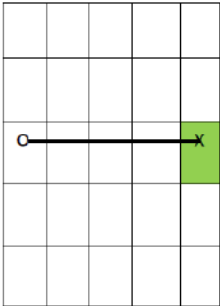
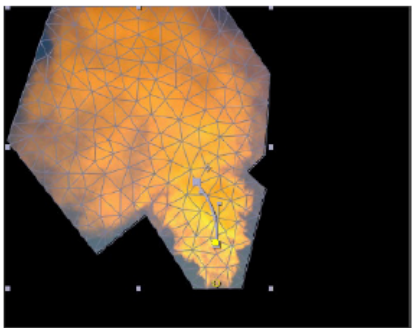
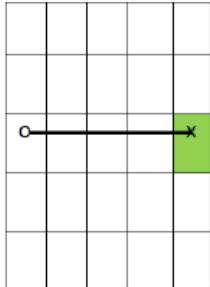
(b) Target Image

This is evident by the fact that the caption for Okabe’s Figure 1 notes that the optional suggestions about fluid motion are merely “shown as orange arrows” as opposed to, say, “which appear on a user interface as orange arrows.” EX1009 at 2. Petitioner appears to largely concede the point that there is no evidence that the arrows are “user-viewable” by noting that “. . . a POSITA would have nevertheless been motivated to make Okabe’s strokes user-viewable to aid users in visualizing the direction and speed of their animations” Paper 1 at 65, note 11. But there is no evidence provided to support this proposed motivation or to show that such a modification would have been known to a POSITA at the time.

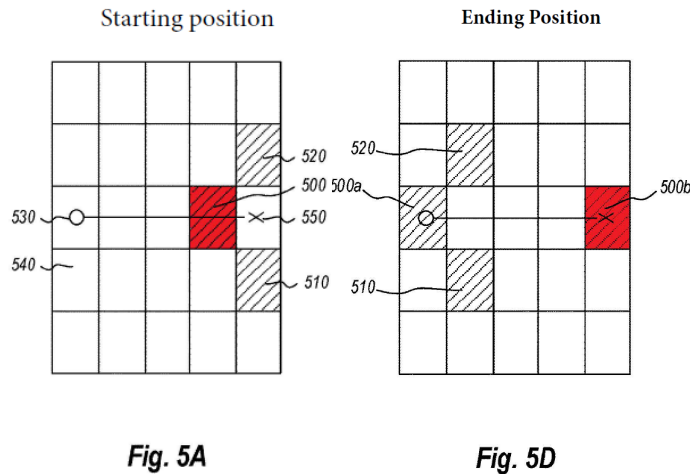
There is also no evidence that the end points of these orange arrows actually serve as the starting points or ending points for any shifted pixels. Because no pixels are shifted in Okabe, Patent Owner submits that the ends of these orange arrows do not function as starting and ending points.

7. The prior art does not teach or suggest the step of shifting a set of pixels by rendering and re-rendering them “in a loop”

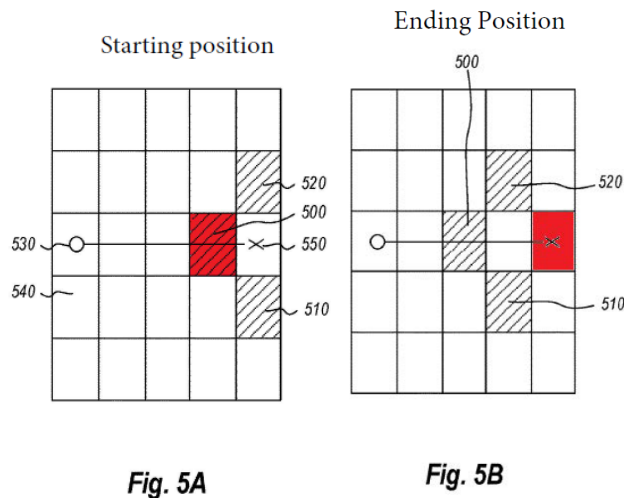
AEM does not teach or suggest shifting of any pixels by rendering and re-rendering in a loop because no pixel ever loops around any other pixels during shifting. Consider where the Deform pin starts and where it finishes in AEM:

Animation Time Point	Composition Frame	Starting Keyframe
(0:00:00:00)		
(0:00:03:01)		Reset to Starting Keyframe 

The Deform pin starts and ends up in the same position. But now consider where the leading pixel 500 starts (Fig. 5A) and where it ends up (Fig. 5D) after being shifted “in a loop” per the ‘017 Patent:



The pixel ends up one space behind (i.e., to the right of) where it started out. Also consider the starting and ending positions for the leading pixel 500 in the alternative embodiment described in the ‘017 Patent:



They are the same as for the first embodiment. The pixel being shifted in a loop ends up behind (to the right of) where it started out. Merely returning the Deform pin back to its starting point, as AEM does, is not shifting it “in a loop.” This conclusion has also been borne out in a related *inter partes* review decision involving related U.S. Patent No. 10,558,342. See IPR2023-00152, Paper No. 7 at 16 (“. . .we agree with Patent Owner that ‘[m]erely returning the Deform pin back to its starting [point] is not shifting it ‘in a loop.’”).

AEM also fails to teach shifting in a loop since it only describes looping a segment of time rather than a set of pixels, as the claims require. AEM discloses that the loopOut expression “[l]oops a segment of time that is measured from the last keyframe on the layer back toward the In point of the layer. The loop plays until the Out point of the layer.” EX1003, 562. It is not shown how looping a segment of time is the same thing as shifting a set of pixels in a loop. The Petition fails to explain the In point and Out point of the layer, and there is no disclosure in AEM that would suggest applying a loopOut expression to a Deform pin.

The IMU and Okabe references also fail to teach or suggest shifting “in a loop” but rather merely a return to a starting point and then repetition of this movement. The Petition maintains that “. . . IMV6’s user shifts pixels as claimed by using Shepard’s Distortion to place and incrementally move a

‘control point’ on an image frame-by-frame in the user’s specified direction and speed.” Paper No. 1 at 72. “The IMU-Okabe Combination thus likewise permits infinite repetition” Id. at 73. See also Okabe, EX1009 at 8 (“By applying one of these techniques to our database and defining the rule of infinite repetition for each video example, our synthesized animation can also permit infinite repetition.”).

8. The prior art is incapable of shifting a set of pixels “in a loop”

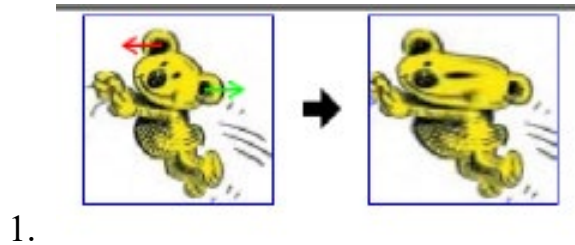
Both Adobe After Effects CS6 and ImageMagick Version 6 are unable to shift a set of pixels in a loop according to the Petitioner’s description of how they move pixels. In short, each of these programs moves pixels by stretching or deforming the image. As a result, no pixels are rearranged such as by looping one behind another. Every pixel which is leading a trailing pixel will remain ahead of the trailing pixel at every stage of the stretch/deform.

To illustrate this, consider what Petitioner uses to explain how ImageMagick can “apply effects and convert one still image into another.” Paper No. 1 at

56. Petitioner notes:

IMU provides an example of placing and moving Shepard’s Distortion control points on an image of a koala to “torture the ‘koala’ by pulling on his ears”

Id. As can be seen from the illustration provided, the image is stretched and distorted.



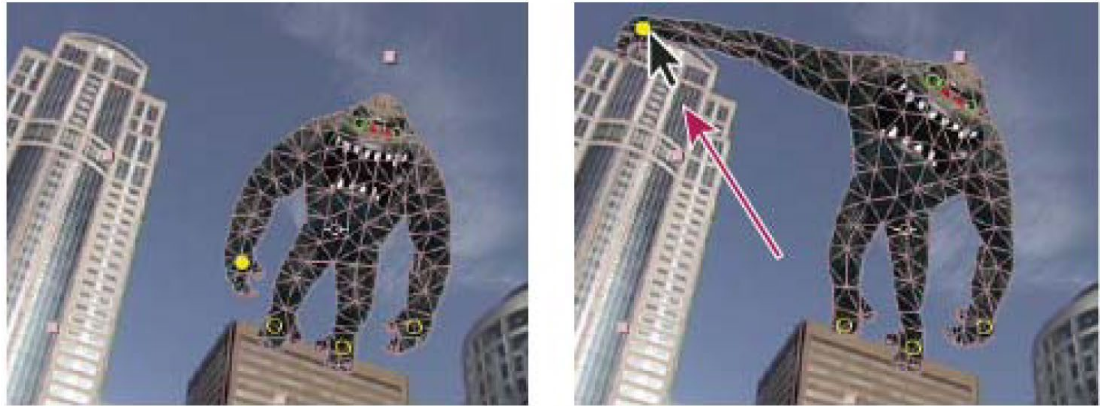
Id. However, no portion of the image loops around any other. The koala’s ear which has the red arrow on it, and the pixels making the ear up, will always remain on the left-hand side of the image. The ear which has the green arrow on it, and the pixels making it up, will always remain on the right-hand side of the image. The red arrow ear does not, for example, loop around to the right-hand side of the image and appear to the right of the green arrow ear. The IMU reference makes it clear that the Shepard’s Distortion used to deform the koala is a “taffy-like distort”. EX2005 at 59. “You can think of this [Shepard’s distortion] as equivalent to a thick block of ‘taffy’ representing the source image, having pins driven into it and then the pins moved around.” *Id.*

A similar effect is done with the Adobe After Effects program which relies on a Puppet Effect to shift a Deform pin. Petitioner explains:

This outline [of nontransparent pixels] is “automatically divided into a mesh of triangles,” where “[e]ach part of the mesh is also associated with the pixels of the image, so the pixels move with the mesh.” . . . Thus, when the user repositions the Deform pin,

“the mesh changes shape to accommodate this movement while keeping the overall mesh as rigid as possible, resulting in a corresponding movement of the layer’s nontransparent pixels.

Paper No. 1 at 12. The Petition then illustrates how movement of the pixels using an image of a gorilla atop buildings:



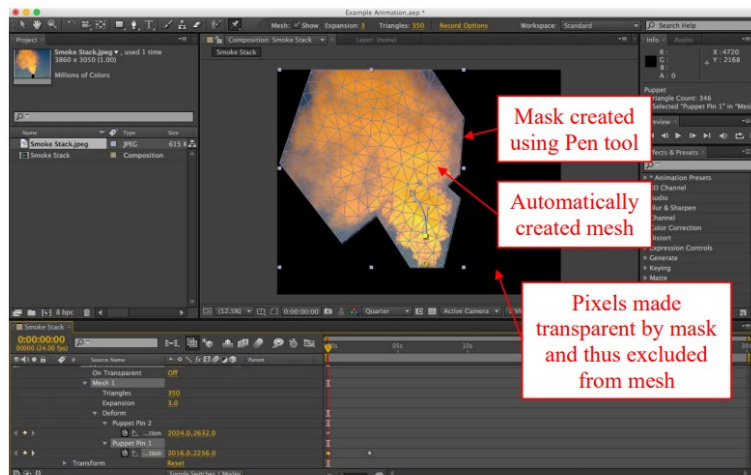
“[T]he Deform pin on the gorilla’s right arm is repositioned, causing that arm to move” *Id.* at 13. Portions of the image are merely stretched or deformed in order to move the Deform pin. No pixels are looped around behind other pixels.

Both Shepard’s Distortion (from IMU) and “puppet effect” (from AEM) are useless to shift a set of pixels in a loop, as all of the challenged claims require. This loop around is what you can never get with the stretching techniques of AEM and IMU. And this is precisely what you need to do to shift pixels “in a loop” as required by every claim of the ‘017 Patent. The prior art the Petitioner relies upon is not capable of shifting a set of pixels in

a loop, per the '017 Patent and, therefore, the challenge to the '017 Patent fails.

9. The prior art does not show pixels are “prevented from shifting” by a mask

Claims 2 and 15 recite identifying a particular portion of the digital image to mask, “wherein the mask prevents pixels covered by the mask from being shifted.” This element is not taught or suggested by the cited art. With regard to the AEM reference, the Petition points to a ‘mask’ created with an “Auto-trace” function to allegedly exclude pixels from a mesh.



Paper 1 at 41-43. However, the Petition doesn't say that these pixels are “prevented from shifting” or explain how they are prevented from shifting. Patent Owner submits that AEM does not teach or suggest any mechanism for preventing pixels from shifting.

The alleged IMU-Okabe combination also does not teach or suggest the “prevented from shifting” element. The Petition asserts that “[i]nherently, the matte [of Okabe] also specifies a region that includes pixels the user does not desire to animate.” Paper 1 at 76.



However, Okabe does not state anywhere that any pixels are prevented from shifting. As noted previously, Okabe does not shift any pixels in the target image but rather assigns video patches to the target image. Are areas that are simply not assigned video patches “prevented from shifting”? Okabe doesn’t say anywhere that any pixels are “prevented from shifting,” and Petitioner has not explained what mechanism would prevent shifting here.

The addition of the Li reference to the IMU-Okabe combination does not provide the missing element. No portion of Li states that any pixels, selected or not, are prevented from shifting. Li’s Lazy Snapping tool is used to cut out portions of images and not to mask any portions of images to prevent pixel shifting. See Paper 1 at 88. The “prevented from shifting” element is entirely missing from the cited art.

10. The prior art does not disclose a non-linear pathway

Claims 7 and 13 recite that “the first rendered link comprises a non-linear pathway extending between the first starting point and the first ending point.” Claim 14 requires rendering on the user interface “a non-linear pathway extending between the first starting point and the first ending point.” The cited art fails to teach or suggest these elements. AEM does not disclose a link that is a non-linear pathway because it does not disclose selecting a set of pixels that extend along a non-linear motion path. In AEM each pixel in a composite image is either transparent or nontransparent such that every pixel is “selected.” *See* EX1003 at 315. As found in a related case, the possibility that some of these pixels, in theory, could extend along a non-linear motion path of a Deform pin is mere happenstance. *See* IPR2023-00152, Paper 7 at 12.

The asserted IMU-Okabe combination also does not teach or suggest these elements. There is no indication that Okabe’s strokes define pathways that pixels are shifted along. *See Id.* at 22. With regard to claim 14, as discussed above, there is no clear teaching that Okabe’s user-drawn strokes are rendered on a user-interface at all.

III. Conclusion

Petitioner has not invalidated any claim of the '017 Patent. The prior art fails to select a set of pixels for shifting, as the claims require. The prior art Petitioner relies upon does not shift a set of pixels “in a loop,” as every challenged claim of the '017 Patent requires, and it is incapable of doing so. The cited art fails to teach masking of portions of an image so that pixels covered by the mask are “prevented from shifting.” Many other claim elements are not taught or suggested either.

Dated this 22nd day of November, 2023.

Respectfully submitted,

/s/ /Shawn Hunter/

Shawn Hunter

Reg. No. 36,168

David A. Walker

Reg. No. 52,334

SCHNEIDER WALLACE

COTTRELL KONECKY LLP

3700 Buffalo Speedway, Suite
960

Houston, Texas 77098

(713) 338-2560

shunter@schneiderwallace.com

[dwalker@schneiderwallace.co
m](mailto:dwalker@schneiderwallace.com)

Raymond R. Ferrera

Reg. No. 47,559

ADAMS AND REESE LLP
One Houston Center
1221 McKinney, Suite 4400
Houston, Texas 77010
(713) 308-0127
ray.ferrera@arlaw.com

Counsel for Patent Owners

Case no. IPR2-2023-00153
U.S. Patent No. 10,346.017

Lightricks, Ltd vs. Plotagraph, Inc. and Sascha Connelly

PATENT OWNERS' EXHIBIT LIST

Exhibit 2001 – Plaintiffs' Original Complaint and Demand for Jury Trial for Plotagraph v. Lightricks Litigation.

Exhibit 2002 – Declaration of Marcellous S. McZeal

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. §42.6(e) the undersigned certifies that on November 22, 2023, the foregoing PATENT OWNERS PLOTAGRAPH, INC. AND SASCHA CONNELLY'S RESPONSE UNDER 37 C.F.R. §42.107 was served via electronic mail upon the following counsel of record for the Petitioner:

Robert Greeson
robert.greeson@nortonrosefulbright.com
Reg. No. 52,966
R. Ross Viguet
ross.viguet@nortonrosefulbright.com
Reg. No. 42,203
Norton Rose Fulbright US LLP
2200 Ross Avenue, Suite 3600
Dallas, Texas 75201-7932
(214) 855-8185

Lu Ling
lu.ling@nortonrosefulbright.com
Reg. No. 78,121
Norton Rose Fulbright US LLP
1301 McKinney, Suite 5100
Houston, Texas 77010
(713) 651-5327
Counsel for Petitioner Lightricks, Ltd.

By: /s/Shawn Hunter
Shawn Hunter
SCHNEIDER WALLACE
COTTRELL KONECKY LLP
3700 Buffalo Speedway, Suite
960
Houston, Texas 77098
(713) 338-2560

shunter@schneiderwallace.com

CERTIFICATE OF WORD COUNT

The foregoing PATENT OWNERS PLOTAGRAPH, INC. AND SASCHA CONNELLY'S RESPONSE UNDER 37 C.F.R. §42.107 complies with the type-volume limitations of 37 C.F.R. §42.24(b)(2). Excluding the items listed in 37 C.F.R. 42.24(a)(1) as not included in the word count, and using the word count of the word-processing system used to prepare the paper (Microsoft Word), the Response contains 8918 words.

/s/ Shawn Hunter
Shawn Hunter