

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of: Michael S. Smith
U.S. Patent No.: 10,936,114 Attorney Docket No. 50095-0171IP1
Issue Date: March 2, 2021
Appl. Serial No.: 16/558,026
Filing Date: August 30, 2019
Title: GESTURE-EQUIPPED TOUCH SCREEN SYSTEM, METHOD,
AND COMPUTER PROGRAM PRODUCT

Mail Stop Patent Board

Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

**PETITION FOR *INTER PARTES* REVIEW OF UNITED STATES PATENT
NO. 10,936,114 PURSUANT TO 35 U.S.C. §§ 311–319, 37 C.F.R. § 42**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	REQUIREMENTS FOR IPR UNDER 37 C.F.R. § 42.104.....	1
	A. Grounds for Standing Under 37 C.F.R. § 42.104(a).....	1
	B. Challenge Under 37 C.F.R. § 42.104(b) and Relief Requested	1
	C. Claim Construction	2
III.	SUMMARY OF THE '114 PATENT	3
	A. Brief Description.....	3
	B. Summary of the Prosecution History of the '114 Patent.....	3
IV.	THE CHALLENGED CLAIMS ARE UNPATENTABLE.....	4
	A. [GROUND 1A] – Kim anticipates or renders obvious claims 1, 27, and 32	4
	1. Kim	4
	2. Claim 1	12
	3. Claim 27	26
	4. Claim 27	32
	5. Claim 32	37
	B. [GROUND 1B] – The Kim-Lang Combination renders obvious claims 34, 59, and 64.....	41
	1. Lang.....	41
	2. The Kim-Lang Combination	41
	3. Claim 34	43
	4. Claim 59	43
	5. Claim 64	44
	C. [GROUND 1C] – The Kim-Burrough Combination renders obvious claim 30.....	44
	1. Burrough.....	44
	2. The Kim-Burrough Combination	45
	3. Claim 30	48
	D. [GROUND 1D] – The Kim-Lang-Burrough Combination renders obvious claim 62	51
	1. The Kim-Lang-Burrough Combination	51
	2. Claim 62	52
	E. [GROUND 2A] – The Kim-Bowens Combination renders obvious claims 1, 27, and 32	52

1.	Bowens	52
2.	The Kim-Bowens-Device (“KBD”).....	53
3.	Claim 1	62
4.	Claim 27	68
5.	Claim 32	74
F.	[GROUND 2B] – The Kim-Bowens-Lang Combination renders obvious claims 34, 59, and 64	79
1.	The Kim-Bowens-Lang Combination.....	79
2.	Claim 34	79
3.	Claim 59	80
4.	Claim 64	80
G.	[GROUND 2C] – The Kim-Bowens-Burrough Combination renders obvious claim 30	80
1.	The Kim-Bowens-Burrough Combination.....	80
2.	Claim 30	81
H.	[GROUND 2D] – The Kim-Bowens-Lang-Burrough Combination renders obvious claim 62	83
1.	The Kim-Bowens-Lang-Burrough Combination	83
2.	Claim 62	84
V.	ANALYSIS ON DISCRETION.....	84
A.	35 U.S.C. §325(d)—<i>Advanced Bionics</i>	84
B.	35 U.S.C. §314(a) - <i>Fintiv</i>	85
VI.	CONCLUSION AND FEES	87
VII.	MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(a)(1).....	87
A.	Real Party-In-Interest Under 37 C.F.R. § 42.8(b)(1).....	87
B.	Related Matters Under 37 C.F.R. § 42.8(b)(2)	87
C.	Lead And Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3).....	88
D.	Service Information	88

LIST OF EXHIBITS

APPLE-1001	U.S. Patent No. 10,936,114
APPLE-1002	U.S. Patent No. 10,936,114 File History
APPLE-1003	Declaration of Dr. Andrew Cockburn
APPLE-1004	U.S. Patent Publication No. 2010/0302281 to Kim (“Kim”)
APPLE-1005	U.S. Patent Publication No. 2009/0273571 to Bowens (“Bowens”)
APPLE-1006	U.S. Patent Publication No. 2006/0139320 to Lang (“Lang”)
APPLE-1007	U.S. Patent Publication No. 2010/0156818 to Burrough et al. (“Burrough”)
APPLE-1008	U.S. Patent Publication No. 2011/0310043 to Lacroix et al. (“Lacroix”)
APPLE-1009	Simon, Windows CE 2 for Dummies (1997)
APPLE-1010	U.S. Patent Publication No. 2011/0221666 to Newton et al. (“Newton”)
APPLE-1011	U.S. Patent Publication No. 2015/0324116 to Marsden et al. (“Marsden”)
APPLE-1012	U.S. Patent Publication No. 2018/0059928 to Westerman et al. (“Westerman”)
APPLE-1013	U.S. Patent Publication No. 2013/0212541 to Dolenc et al. (“Dolenc”)

APPLE-1014	U.S. Patent Publication No. 2005/0168488 to Montague (“Montague”)
APPLE-1015	U.S. Patent No. 8,954,887 to Tseng et al. (“Tseng”)
APPLE-1016	Smith’s Infringement Contention
APPLE-1017	U.S. Patent Publication No. 2007/0024593 to Schroeder (“Schroeder”)
APPLE-1018	U.S. Patent Publication No. 2009/0167508 to Fadell et al. (“Fadell”)
APPLE-1019	United States District Courts — National Judicial Caseload Profile (Dec. 31, 2023) United States Courts (uscourts.gov), https://www.uscourts.gov/sites/default/files/fcms_na_distprofile1231.2023_0.pdf (last visited June 25, 2024)
APPLE-1020	Memorandum, Interim Procedure for Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation (USPTO June 21, 2022) (“Guidance”)
APPLE-1021	Case Management Order, Smith Interface Technologies, LLC v. Apple Inc., 3-23-cv-01187, S.D. Cal. (June 20, 2024)

LISTING OF CLAIMS

Claim	Identifier	Claim Language
1	[1.pre]	An apparatus, comprising:
	[1.a]	at least one non-transitory memory storing instructions and a plurality of applications;
	[1.b]	a touch screen; and
	[1.c]	one or more processors in communication with the at least one non-transitory memory and the touch screen, wherein the one or more processors execute the instructions to:
	[1.d]	display, via the touch screen, an object and at least a portion of an interface;
	[1.e]	detect a gesture on the touch screen on the object;
	[1.f]	when a magnitude of a duration of the gesture being detected on the touch screen on the object is less than a first magnitude threshold, perform an operation;
	[1.g]	when the magnitude of the duration of the gesture being detected on the touch screen on the object is greater than the first magnitude threshold, perform a scaling in connection with at least the object,
	[1.h]	where the scaling includes a reduction in a size such that the size is inversely related to the magnitude of the duration of the gesture being detected on the touch screen on the object; and
	[1.i]	when the magnitude of the duration of the gesture being detected on the touch screen on the object is greater than a second magnitude threshold, perform another operation.
27	[27.pre]	An apparatus, comprising:
	[27.a]	at least one non-transitory memory storing instructions and a plurality of applications;
	[27.b]	a touch screen; and

Claim	Identifier	Claim Language
	[27.c]	one or more processors in communication with the at least one non-transitory memory and the touch screen, wherein the one or more processors execute the instructions to cause the apparatus to:
	[27.d]	display an object on at least a portion of an interface;
	[27.e]	detect a gesture via the touch screen on the object;
	[27.f]	perform a first function in a first touch state;
	[27.g]	perform a second function in a second touch state; and
	[27.h]	during detection of at least a portion of the gesture and based on a change in a duration of the gesture being detected via the touch screen on the object, perform a scale operation in connection with the at least portion of the interface, where the scale operation includes a reduction in a size that is inversely related to the duration of the gesture being detected via the touch screen on the object.
30	[30.pre]	An apparatus, comprising:
	[30.a]	at least one non-transitory memory storing instructions and a plurality of applications;
	[30.b]	a touch screen;
	[30.c]	a vibrator; and
	[30.d]	one or more processors in communication with the at least one non-transitory memory, the touch screen, and the vibrator, wherein the one or more processors execute the instructions to cause the apparatus to:
	[30.e]	display a first object and at least one other object;
	[30.f]	detect a gesture on the touch screen on the first object;
	[30.g]	during detection of at least a portion of the gesture before a completion thereof is detected, perform a scale operation, based on a change in a duration of the gesture being detected on the touch screen on the first object, on at least the at least one other object,

Claim	Identifier	Claim Language
	[30.h]	where the scale operation includes a reduction in a size that is inversely related to the duration of the gesture being detected on the touch screen on the first object; and
	[30.i]	after at least a portion of the scale operation, vibrate the apparatus and perform an action.
32	[32.pre]	An apparatus, comprising:
	[32.a]	at least one non-transitory memory storing instructions and a plurality of applications;
	[32.b]	a touch screen; and
	[32.c]	one or more processors in communication with the at least one non-transitory memory and the touch screen, wherein the one or more processors execute the instructions to:
	[32.d]	display, via the touch screen, a first object and at least one other object;
	[32.e]	detect at least one component of a gesture on the touch screen on the first object;
	[32.f]	when a first component of the gesture is detected where a magnitude of a duration of the gesture being detected on the touch screen on the first object is less than a first magnitude threshold, perform a function upon a cessation of the gesture being detected;
	[32.g]	when a second component of the gesture is detected where the magnitude of the duration of the gesture being detected on the touch screen on the first object is greater than the first magnitude threshold, perform a scale operation for the at least one other object,
	[32.h]	where the scale operation includes a reduction in a size so as to be inversely related to the magnitude of the duration of the gesture being detected on the touch screen on the first object; and
	[32.i]	when a third component of the gesture is detected where

Claim	Identifier	Claim Language
		the magnitude of the duration of the gesture being detected on the touch screen on the first object is greater than a second magnitude threshold, perform another function.
34	[34.pre]	A method, comprising:
	[34.a]	at an apparatus including a watch with
	[34.b]	at least one non-transitory memory storing a plurality of applications,
	[34.c]	a touch screen, and
	[34.d]	one or more processors in communication with the at least one non-transitory memory and the touch screen:
	[34.e]	displaying, via the touch screen, an object and at least a portion of an interface;
	[34.f]	detecting a gesture on the touch screen on the object;
	[34.g]	based on a magnitude of a duration of the gesture detected on the touch screen on the object being detected to be less than a first magnitude threshold, performing an operation;
	[34.h]	based on the magnitude of the duration of the gesture detected on the touch screen on the object being detected to be greater than the first magnitude threshold, performing a scaling in connection with at least the object,
	[34.i]	where the scaling includes a reduction in a size such that the size is inversely related to the magnitude of the duration of the gesture being detected on the touch screen on the object; and
	[34.j]	based on the magnitude of the duration of the gesture detected on the touch screen on the object being detected to be greater than a second magnitude threshold, performing another operation.
59	[59.pre]	A method, comprising:

Claim	Identifier	Claim Language
	[59.a]	at an apparatus including a watch with
	[59.b]	at least one non-transitory memory storing a plurality of applications,
	[59.c]	touch screen, and
	[59.d]	one or more processors in communication with the at least one non-transitory memory and the touch screen:
	[59.e]	displaying an object on at least a portion of an interface;
	[59.f]	detecting a gesture via the touch screen on the object;
	[59.g]	performing a first function in a first touch state;
	[59.h]	performing a second function in a second touch state; and
	[59.i]	based on a detection of at least a portion of the gesture and based on a change in a duration of the gesture being detected via the touch screen on the object, performing a scale operation in connection with the at least portion of the interface, where the scale operation includes a reduction in a size that is inversely related to the duration of the gesture being detected via the touch screen on the object.
62	[62.pre]	A method, comprising:
	[62.a]	at an apparatus including a watch
	[62.b]	having at least one non-transitory memory storing a plurality of applications,
	[62.c]	a touch screen,
	[62.d]	a vibrator, and
	[62.e]	one or more processors in communication with the at least one non-transitory memory, the touch screen, and the vibrator:
	[62.f]	displaying a first object and at least one other object;
	[62.g]	detecting a gesture on the touch screen on the first object;

Claim	Identifier	Claim Language
	[62.h]	in response to a detection of at least a portion of the gesture, performing a scale operation, based on a change in a duration of the gesture being detected on the touch screen on the first object, on at least the at least one other object,
	[62.i]	where the scale operation includes a reduction in a size that is inversely related to the duration of the gesture being detected on the touch screen on the first object; and
	[62.j]	subsequent to the performance of the scale operation, vibrating the apparatus and performing an action.
64	[64.pre]	A method, comprising:
	[64.a]	at an apparatus including a watch having
	[64.b]	at least one non-transitory memory storing a plurality of applications,
	[64.c]	a touch screen, and
	[64.d]	one or more processors in communication with the at least one non-transitory memory and the touch screen:
	[64.e]	displaying, via the touch screen, a first object and at least one other object;
	[64.f]	detecting at least one component of a gesture on the touch screen on the first object;
	[64.g]	in response to a first component of the gesture being detected where a magnitude of a duration of the gesture being detected on the touch screen on the first object is detected to be less than a first magnitude threshold, performing a function upon a cessation of the gesture being detected;
	[64.h]	in response to a second component of the gesture being detected where the magnitude of the duration of the gesture being detected on the touch screen on the first object is detected to be greater than the first magnitude threshold, performing a scale operation for the at least

Claim	Identifier	Claim Language
		one other object,
	[64.i]	where the scale operation includes a reduction in a size so as to be inversely related to the magnitude of the duration of the gesture being detected on the touch screen on the first object; and
	[64.j]	in response to a third component of the gesture being detected where the magnitude of the duration of the gesture being detected on the touch screen on the first object is detected to be greater than a second magnitude threshold, performing another function.

I. INTRODUCTION

Apple Inc. (“Apple” or “Petitioner”) petitions for IPR of claims 1, 27, 30, 32, 34, 59, 62, and 64 (“Challenged Claims”) of U.S. Patent No. 10,936,114 (“the ’114 patent”). As explained in this petition, there exists a reasonable likelihood that Petitioner will prevail with respect to at least one of the Challenged Claims.

The Challenged Claims are generally directed to reducing a size of a displayed object in response to a gesture detected on a touch screen, where the size is inversely related to the magnitude of the duration of the gesture.

The Challenged Claims are unpatentable based on teachings set forth in at least the references presented in this petition. Petitioner respectfully submits that an IPR should be instituted, and that the Challenged Claims should be canceled as unpatentable.

II. REQUIREMENTS FOR IPR UNDER 37 C.F.R. § 42.104

A. Grounds for Standing Under 37 C.F.R. § 42.104(a)

Petitioner certifies that the ’114 Patent is available for IPR. Petitioner is not barred or estopped from requesting this review. Petitioner was served with a complaint of infringement of the ’114 Patent less than one year ago.

B. Challenge Under 37 C.F.R. § 42.104(b) and Relief Requested

Petitioner requests an IPR of the Challenged Claims on the grounds set forth in the table shown below. Dr. Andrew Cockburn provides supporting testimony in

his Declaration. APPLE-1003, ¶¶25-258.

Ground	'114 Patent Claims	Basis for Rejection
Ground 1A	1, 27, 32	§§102/103: Kim
Ground 1B	34, 59, 64	§103: Kim, Lang
Ground 1C	30	§103: Kim, Burrough
Ground 1D	62	§103: Kim, Lang, Burrough
Ground 2A	1, 27, 32	§103: Kim, Bowens
Ground 2B	34, 59, 64	§103: Kim, Bowens, Lang
Ground 2C	30	§103: Kim, Bowens, Burrough
Ground 2D	62	§103: Kim, Bowens, Lang, Burrough

The '114 Patent claims priority to several provisional applications as early as August 5, 2011 (“Critical Date”). APPLE-1001, Cover. Petitioner does not concede that the claimed priority date is correct, but applies prior art that predates it. The references are prior art under §§102(a), 102(b), and/or 102(e).

Reference	Filing Date	Publication Date	Basis
Kim (APPLE-1004)	5/27/2010	12/2/2010	§§102(a),(e)
Bowens (APPLE-1005)	10/20/2008	11/5/2009	§§102(a),(b),(e)
Lang (APPLE-1006)	2/16/2996	6/29/2006	§§102(a),(b),(e)
Burrough (APPLE-1007)	4/6/2009	6/24/2010	§§102(a),(b),(e)

C. Claim Construction

All claim terms should be construed according to the *Phillips* standard.

Phillips v. AWH Corp., 415 F.3d 1303 (Fed. Cir. 2005); 37 C.F.R. §42.100. Apple submits that no claim terms need be construed to resolve issues of controversy in the present Petition. *Wellman, Inc. v. Eastman Chem. Co.*, 642 F.3d 1355, 1361 (Fed. Cir. 2011); APPLE-1003, ¶28.

III. SUMMARY OF THE '114 PATENT

A. Brief Description

The '114 Patent states that “[e]mbodiments of the present invention generally relate to touch screens. APPLE-1001, 1:37-38. Claim 27 of the '114 Patent, for example, recites “[a]n apparatus,” comprising “a touch screen” and “one or more processors [that] execute [] instructions to ... perform[s] a first function in a first touch state,” “perform[s] a second function in a second touch state,” and “perform[s] a scale operation in connection with the at least portion of the interface, where the scale operation includes a reduction in a size that is inversely related to the duration of the gesture being detected via the touch screen on the object.” APPLE-1003, ¶¶29-31.

B. Summary of the Prosecution History of the '114 Patent

During prosecution, the examiner issued no Office actions. *See, generally*, APPLE-1002; APPLE-1003, ¶¶32-34. Instead, Applicant offered a “Proposed Amendment,” which the parties discussed in an interview, and the examiner subsequently allowed the proposed amended claims without any prior art rejection.

Id., 305-358, 134, 74-133. Notably, despite the thousands of references provided to the Office, the examiner did not issue a single prior art rejection before allowing the claims.

IV. THE CHALLENGED CLAIMS ARE UNPATENTABLE

A. [GROUND 1A] – Kim anticipates or renders obvious claims 1, 27, and 32¹

1. Kim²

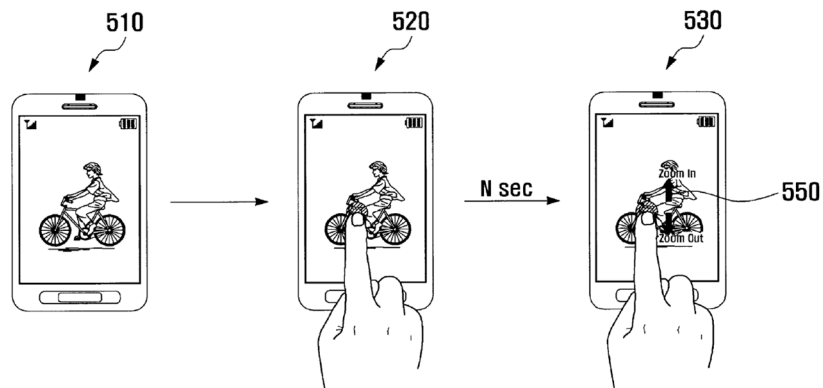
Kim describes a mobile device that “has a touch sensitive surface such as a touch screen” that provides “a touch-based zooming control method” for “portable and handheld devices.” APPLE-1004, Abstract, [0118]. At a high level, Kim explains that, “while performing a selected mode” (i.e., “a particular mode that allows displaying some kind of display target data outputted while a specific application selected at a user’s request runs in the mobile device”), “the mobile device detects the occurrence of an interrupt,” which is “a signal or instructions,

¹ Ground 1A comprises two separate Kim-based grounds based on §102 or §103 that are presented collectively for convenience.

² General descriptions provided for this and other references and combinations thereof are incorporated into each subsection and mapping of the claims that includes citations to these references.

inputted through a touch screen” and “determines whether an interrupt is a request for activating the zoom mode,” including, for example, determining “whether an interrupt is a long press input.” *Id.*, [0048]-[0049]. The interrupt may be, e.g., “a request for the zoom mode” or a request for “some action in connection with the interrupt,” such as “a selection input for a specific icon” or “a request for a specific menu.” *Id.*, [0049]-[0051]. Kim’s device “detect[s] a user gesture” by “receiving an input event from a user through the touch screen, and responsively the mobile device determines the type of an input event,” including, e.g., “whether an input event is one of a move event ... , a hold event ..., and a double tap event.” *Id.*, [0053]; APPLE-1003, ¶35.

In Kim, while “target data is displayed, the mobile device activates a zoom mode in response to a given gesture or input event,” and “executes a zooming function such as a zoom-in and a zoom-out in response to the received gesture input.” APPLE-1004, Abstract. Figure 5 “illustrates an example of a process of activating a zoom mode of a mobile device.” *Id.*, [0083]. Kim explains that the central point of the zoom can be determined by a tap. *Id.*, [0066]-[0067]. In Figure 5, “the mobile device displays target data of specific content on the screen”—here, a bike and bicyclist. *Id.*, [0084]; APPLE-1003, ¶36.



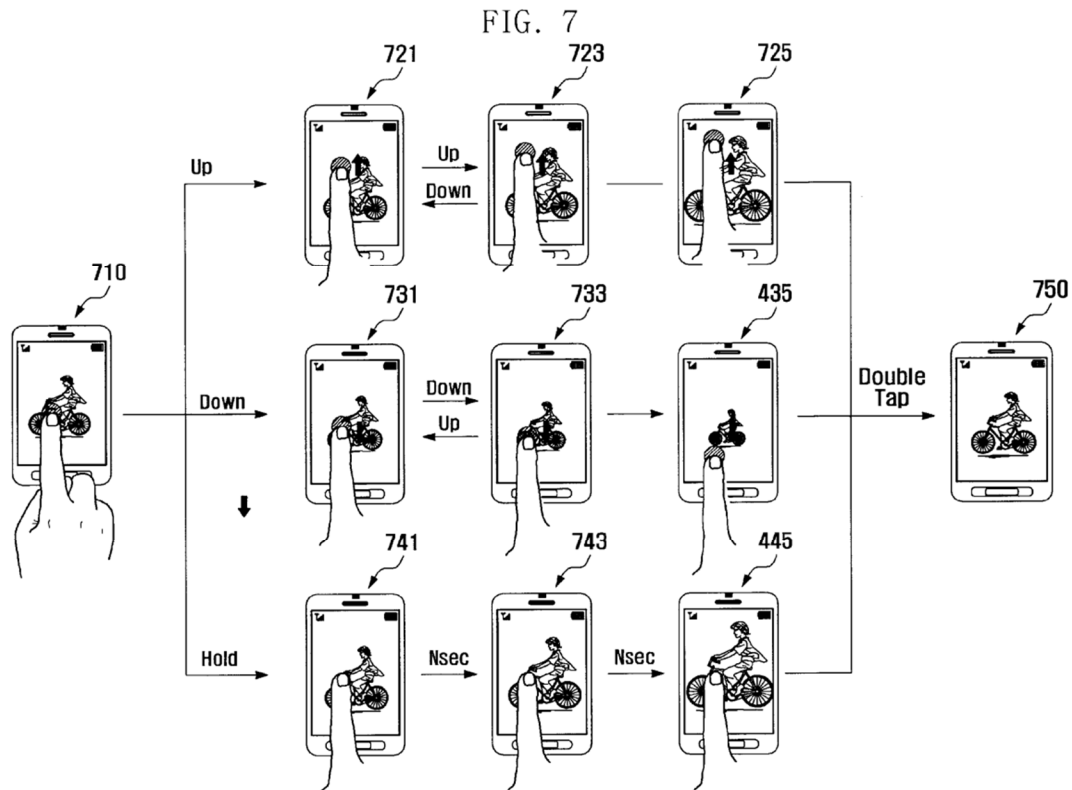
APPLE-1004, FIG. 5.

Kim explains that “a zoom mode can be activated when a user invokes a *long press* at a certain point (x, y) on a touch screen” and “can perform a zooming action in the following two ways.” APPLE-1004, [0139] (emphasis added). First, a user can move his finger up or down “from a pressed point (x, y) in order to perform a zoom-in or a zoom-out around that point (x, y).” *Id.*, [0140]. Or second, “a user can hold one of his or her finger ... without moving at a pressed point (x, y).” *Id.*; APPLE-1003, ¶37.

In more detail, Figure 7 “illustrates a process of executing the zooming function depending on a touch-based user gesture in a mobile device.” APPLE-1004, [0094]. In Figure 7, after “the zoom mode is activated, a user can manipulate the mobile device to perform a continuous zooming function (namely, a zoom-in or a zoom-out) through an appropriate gesture such as a move event or a hold event.” *Id.*, [0095]. Similarly, Figure 7 shows a mobile device displaying

“target data of specific content on the screen”—a bike and bicyclist. *Id.*, [0084],

FIG. 7; APPLE-1003, ¶38.



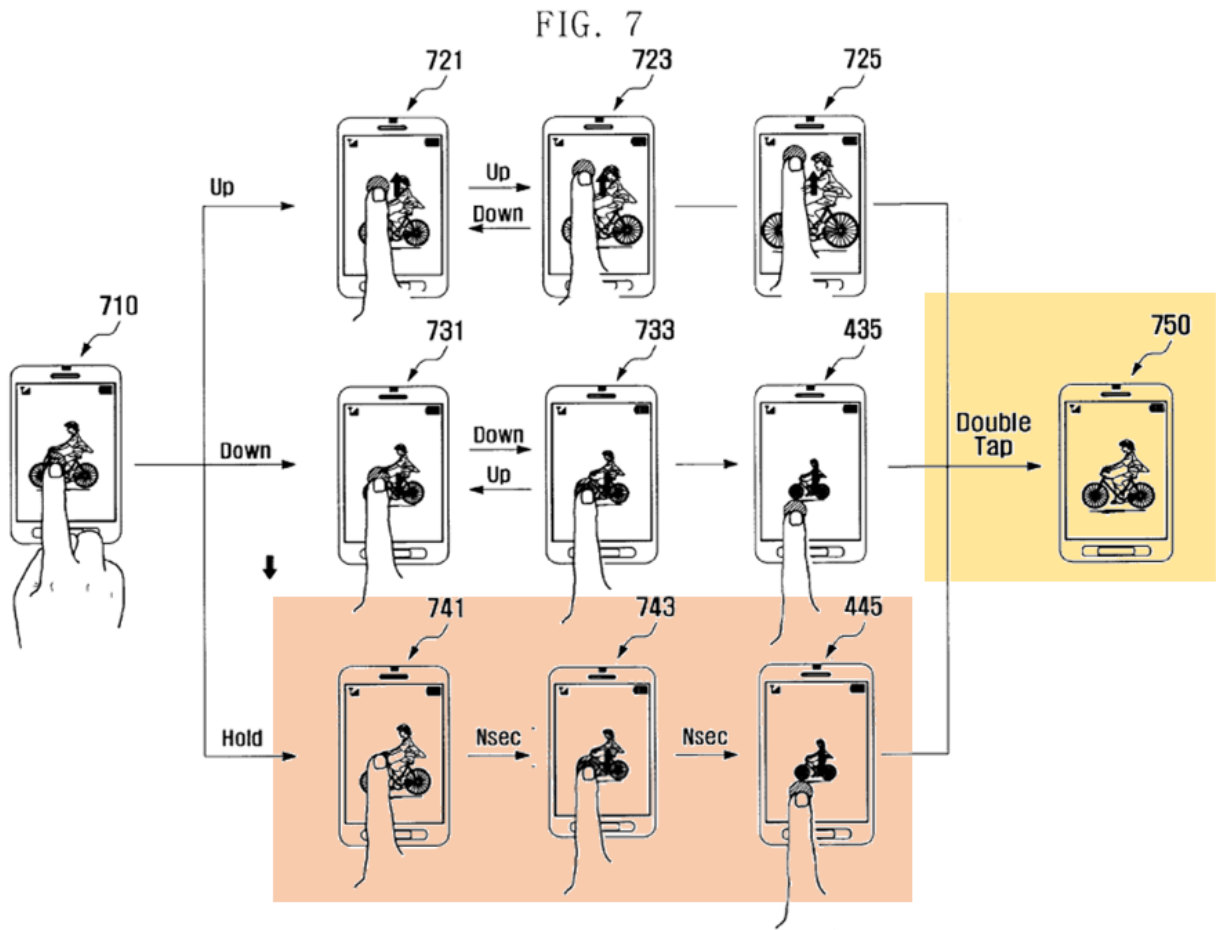
APPLE-1004, FIG. 7.

In the top two examples above, a user can “continuously invoke” either a move up or move down event at a selected point to cause a continuous zoom-in or zoom-out function, respectively. APPLE-1004, [0096]-[0103]. In the bottom path of Figure 7 “show[s] a continuous zoom-in action depending on a hold event.” *Id.*, [0104]. That is, the user touches the displayed bike (“selected point”), and the device executes a zoom-in function around that point on the bike, as evidenced by the bike (and its surrounding data) being displayed at a larger size. *Id.*; APPLE-

1003, ¶39. Therefore, Kim teaches or at least renders obvious that the user input is received on the object that is later scaled up (zoomed-in) or scaled down (zoomed-out). *Id.* Kim further explains that “the hold event is assigned to a continuous zoom-in function is an exemplary case only,” and “the hold event may be alternatively assigned to a continuous zoom-out function.” *Id.*, [0107]; APPLE-1003, ¶40.

Since “the hold event may be alternatively assigned to a continuous zoom-out function,” Kim teaches that bottom path of Figure 7 can be zoom-out instead of zoom-in, as illustrated in Figure 7 (modified), below (highlighted orange).

APPLE-1004, [0107]; APPLE-1003, ¶41.



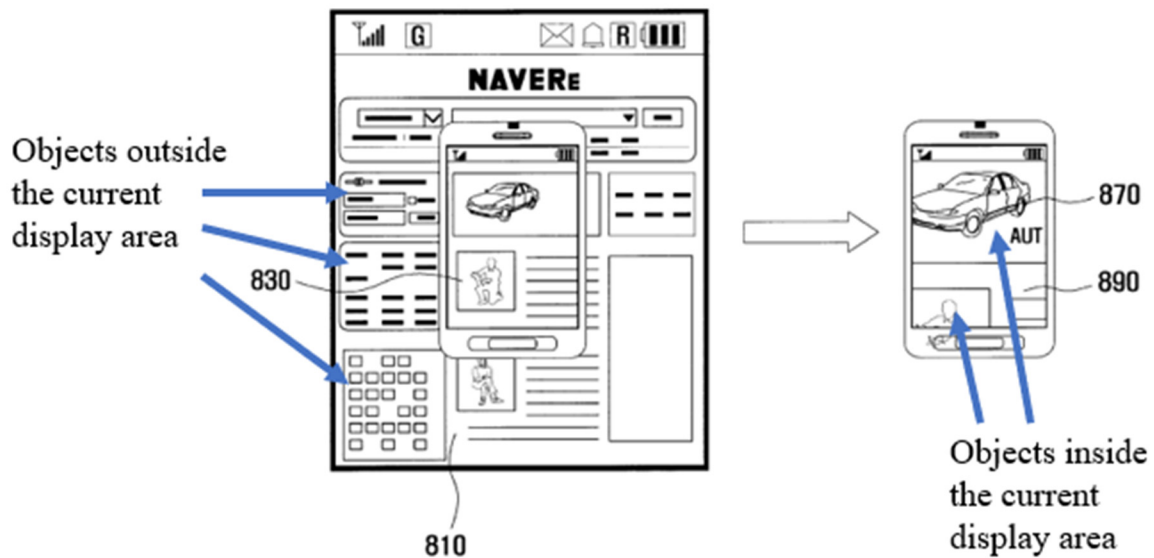
APPLE-1004, FIG. 7 (modified)

In addition to a hold event, Kim teaches that other gestures, such as press and move, can be used to control zooming. APPLE-1004, [0094]-[0103]. Kim also teaches that, after a zoom action, “a user can invoke a double tap event on the screen,” which causes “the mobile device [to] display[] target data with a given basic magnification value, as indicated by reference number 750” in Figure 7 (modified) above (highlighted yellow). APPLE-1004, [0099], [0103], [0106], FIG. 7; APPLE-1003, ¶42.

While Figure 7 shows a rider on a bicycle,³ Figure 8 illustrates that scaling can be performed on objects other than the object with respect to which the user gesture is directed or detected. APPLE-1004, FIG. 8; APPLE-1003, ¶43. Kim's device, "controls a zooming action ... in response to a user gesture" on a portion of the web page that fits on the device screen's interface. APPLE-1004, [0111]; APPLE-1003, ¶43. While Figure 8 does not explicitly indicate where the user gesture was received, a person of ordinary skill in the art ("POSITA")⁴ would have understood that the gesture was received on a single object (as opposed to on the entire touch screen), and likely, it was received on or near the car given the resulting enlarged display. APPLE-1003, ¶43. Moreover, in response to the user gesture, the zoom function affects both the car and surrounding text and image(s). APPLE-1004, FIG. 8.

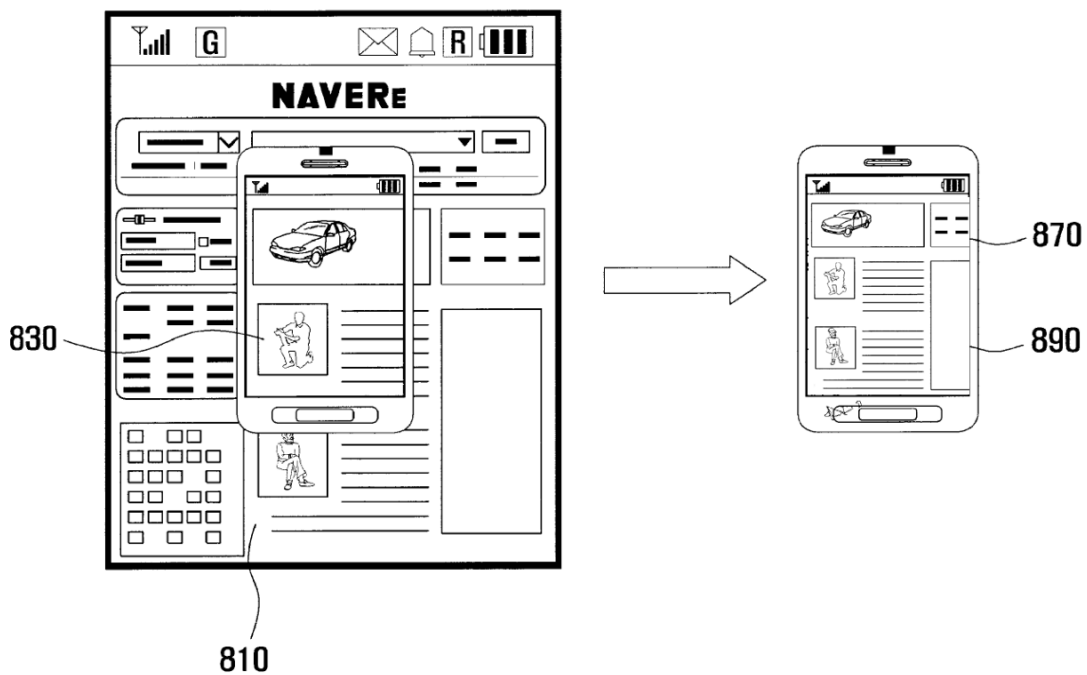
³ The bicycle and bicyclist shown in Figure 7 could be considered at least two separate objects, where the gesture is detected on the bicycle but not the bicyclist. *See* APPLE-1004, FIG. 7 (741).

⁴ APPLE-1003, ¶26.



APPLE-1004, FIG. 8 (annotated).

Per Kim's teachings, the zooming action shown in Figure 8 can also be a zoom-out action, as illustrated in modified FIG. 8. APPLE-1004, [0107], [0113]; APPLE-1003, ¶44.



APPLE-1004, FIG. 8 (modified)

In Kim, the “continuous zoom-in action depend[s] on a hold event” such that “the mobile device executes a continuous zoom-in function around the selected point at given periodic intervals while the hold event continues.” APPLE-1004, [0104]. In one example, “if a periodic interval is set for N seconds, the mobile device can perform a zooming control at intervals of N seconds while the hold event continues.” *Id.*, [0104]. In at least this way, Kim discloses performing another zoom operation or function based on duration of the detected gesture magnitude. *Id.* More, a POSITA would have understood from Kim’s disclosure that, after the periodic interval, the zooming would cease. APPLE-1003, ¶45.

2. Claim 1

[1.pre]

To the extent the preamble is limiting, Kim teaches or renders [1.pre] obvious. APPLE-1003, ¶46. Kim discloses a mobile device with “a touch sensitive surface such as a touch screen” that provides “a touch-based zooming control method.” APPLE-1004, Abstract. The mobile device “includes a plurality of electronic devices having a touch sensitive surface such as one of a touch screen and a touch pad” and can be any other kinds of portable and handheld devices.” *Id.*, [0118]. An exemplary mobile device is shown below in an excerpt from Kim’s Figure 5.

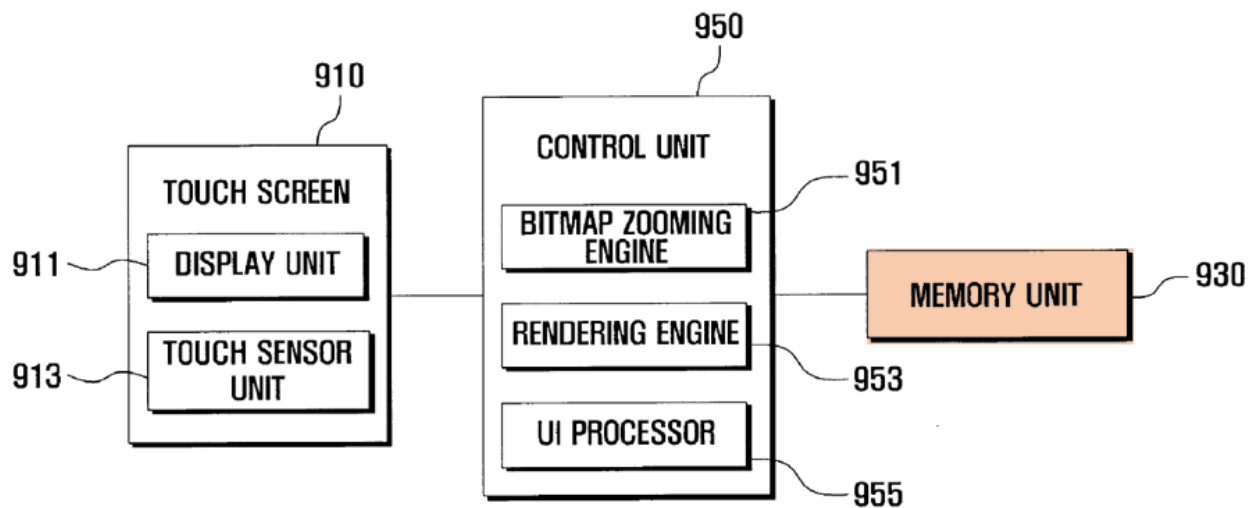


APPLE-1004, FIG. 5 (excerpt)

[1.a]

Kim teaches or renders [1.a] obvious. APPLE-1003, ¶47. With reference to Figure 9, Kim's device includes "a memory unit 930." APPLE-1004, [0122], [0127]-[0128]. The "memory unit 930 stores a plurality of applications required for controlling a general operation of the mobile device, and a plurality of applications required for controlling a zooming function based on a user gesture in the mobile device." *Id.*, [0128]. Kim further teaches that its methods are "realized in hardware or as software or computer code that can be stored in a recording medium." *Id.*, [0144]. From this and other Kim disclosures, a POSITA would have understood that the "applications," "hardware," "software," and/or "computer

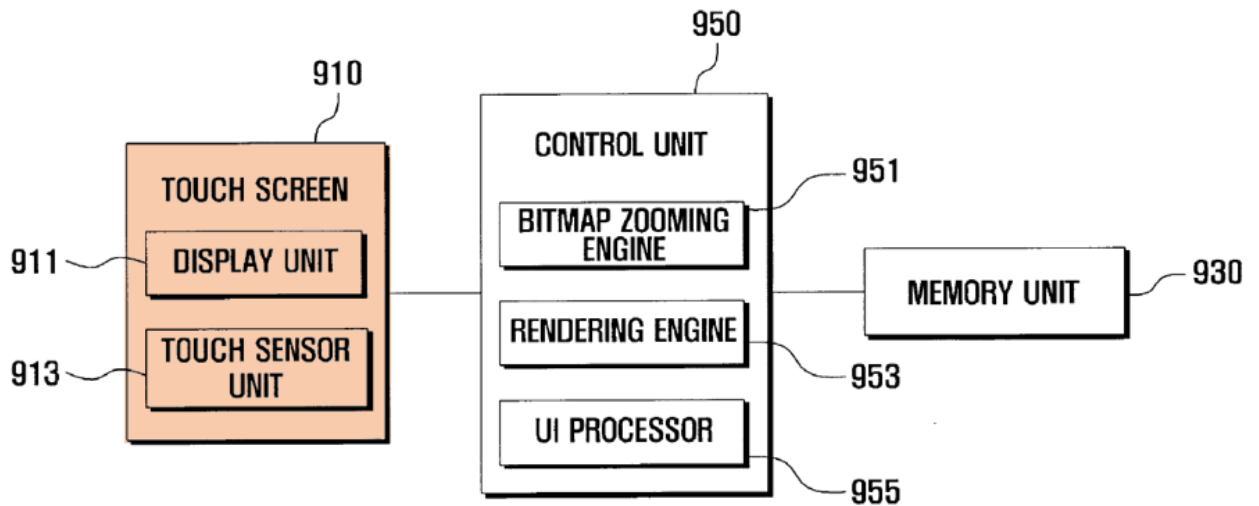
code” of Kim include a series of instructions for performing the methods described in Kim. *Id.*; APPLE-1003, ¶47.



APPLE-1004, FIG. 9 (highlighted)

[1.b]

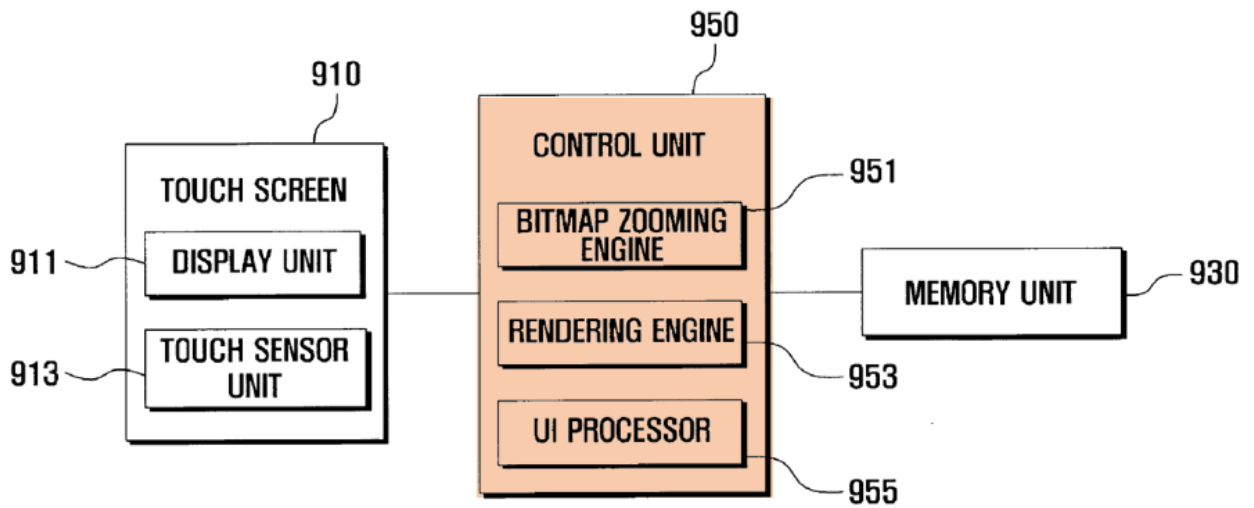
Kim teaches or renders [1.b] obvious. APPLE-1003, ¶48. Kim’s device includes “a touch sensitive surface such as a touch screen.” APPLE-1004, Abstract. For example, with reference to Figure 9, Kim shows in the inclusion of touch screen 910. *Id.*, [0122]-[0126], FIG. 9.



APPLE-1004, FIG. 9 (highlighted)

[1.c]

Kim teaches or renders [1.c] obvious. APPLE-1003, ¶49. With reference to Figure 9, Kim's device includes "a control unit 950," which is shown to be in communication with the touch screen 910 and the memory unit 930. APPLE-1004, [0122], [0129]-[0135], FIG. 9. "The control unit 950 performs an overall control function associated with the mobile device and controls the flow of signals between components in the mobile device," including "control[ling] final flows between the aforesaid components such as the touch screen 910 and the memory unit." *Id.*, [0129], [0135]. Thus, a POSITA would have understood from Kim's discussion of the control unit's functions that the control unit executes the instructions (e.g., computer code) stored in memory to perform the methods described in Kim. *Id.*, [0144]; APPLE-1003, ¶49.

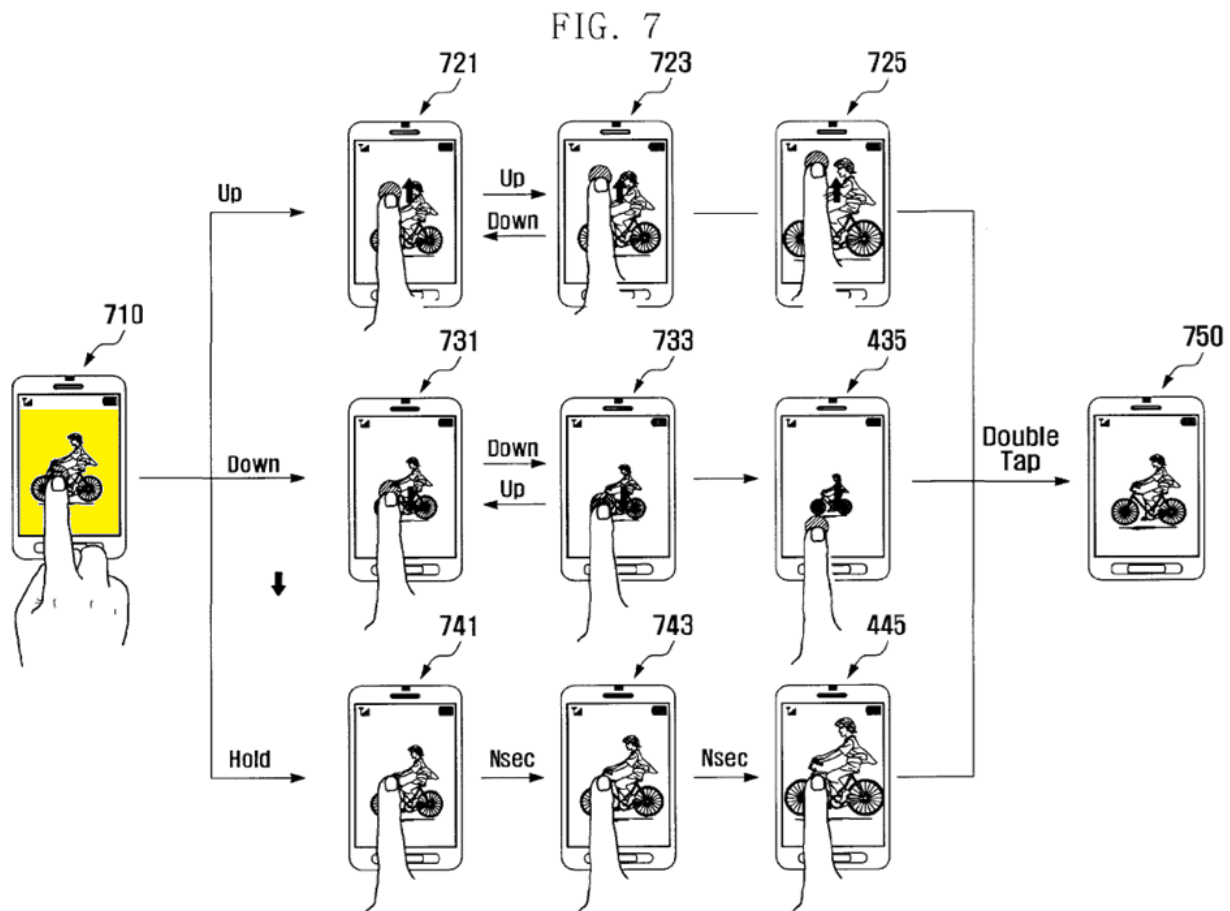


APPLE-1004, FIG. 9 (highlighted)

[1.d]

Kim teaches or renders [1.d] obvious. APPLE-1003, ¶50. Kim describes several examples of Kim's device "display[ing] target data of specific content on the screen." APPLE-1004, [0084]. For example, Kim describes "displaying some kind of display target data outputted while a specific application selected at a user's request runs in the mobile device including any of a web page display mode based on a web browser function or environment, a photo data display mode based on a camera function or environment, [and] a text display mode based on a document viewer function or environment." *Id.*, [0041], [0048]. In Figures 5-6, Kim's device displays "certain image data." *Id.*, [0083]-[0093], [0088], FIGS. 5-6. Similarly, in Figure 7, Kim's device "continuously offers a real-time, scaled-up bitmap image on the screen in response to the progress of the continuous move up

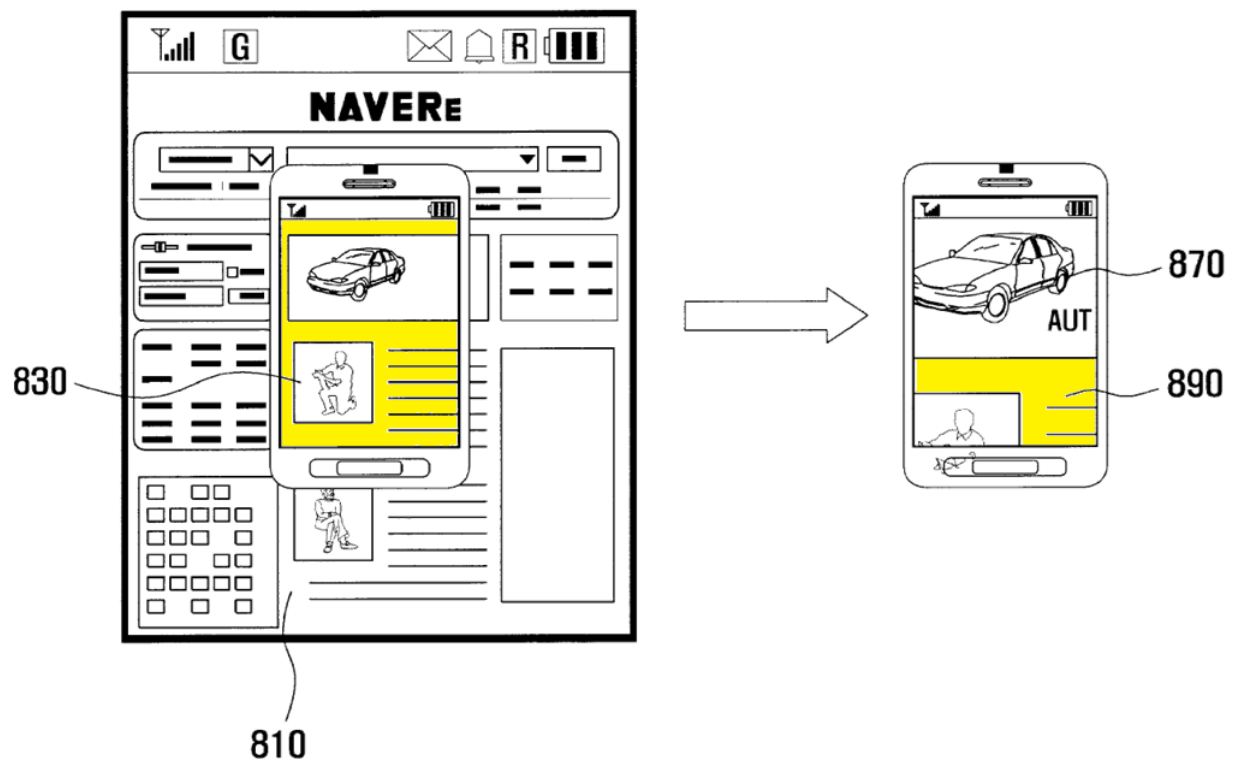
event.” *Id.*, [0094]-[0107], FIG. 7. Figure 8 shows the device displaying “target data” in a bitmap image. *Id.*, [0109]-[0114], FIG. 8. In each example, Kim’s device displays an object (e.g., bicyclist on bicycle) surrounded by at least a portion of an interface (highlighted yellow in Figure 7). APPLE-1003, ¶50.



APPLE-1004, FIG. 7 (highlighted)

In another example, Kim’s Figure 8 (below) displays an object (e.g., car, text hyperlink) and a surrounding interface (highlighted yellow). APPLE-1004, [0109] (“target data”), FIG. 8; APPLE-1003, ¶51.

FIG. 8

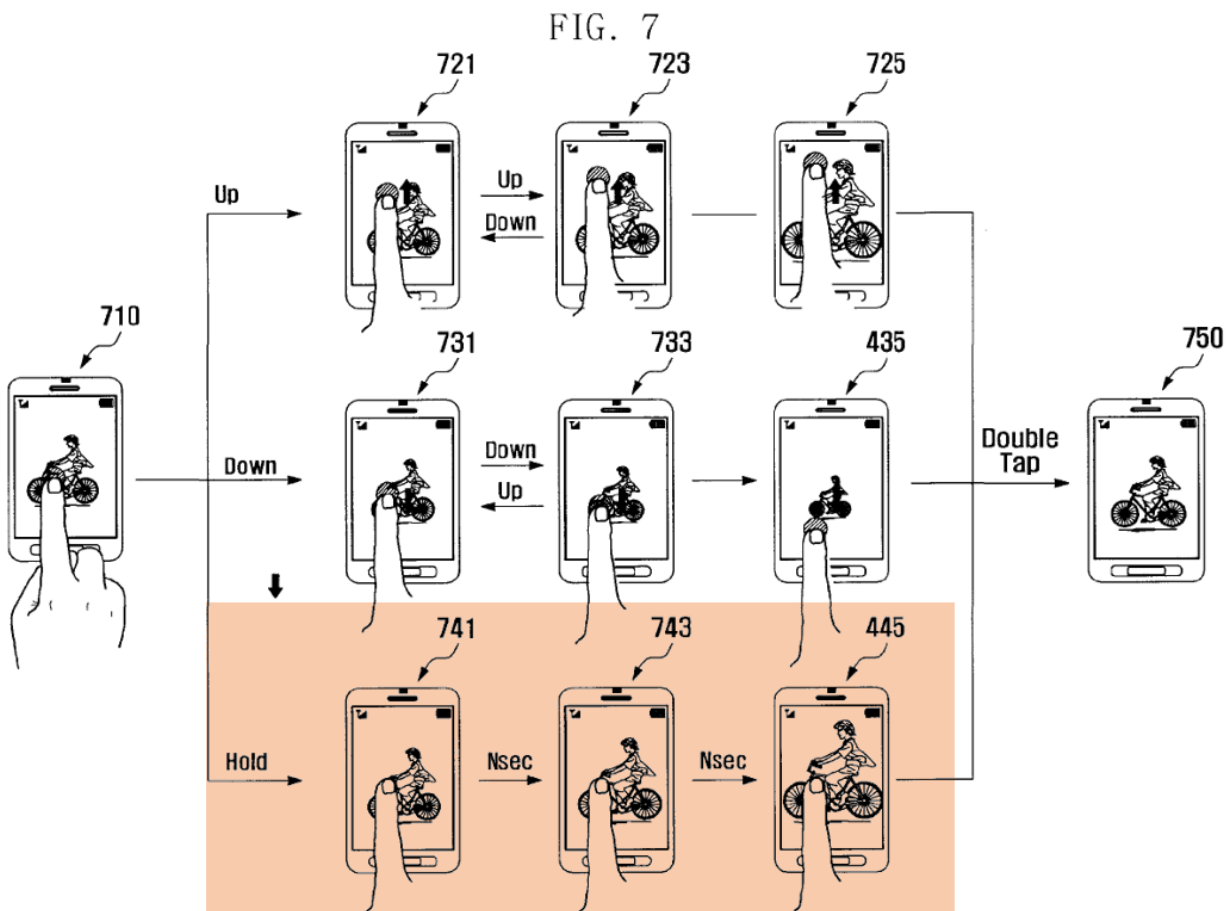


APPLE-1004, FIG. 8 (highlighted)

[1.e]

Kim teaches or renders [1.e] obvious. APPLE-1003, ¶52. Kim describes various scenarios where a gesture is received on its touch screen. APPLE-1004, [0041]-[0045], [0048]-[0049], [0053] (“If detecting a user gesture, namely, if receiving an input event from a user through the touch screen”), [0064], [0066], [0068], [0072], [0076], [0099], [0103]-[0104], [0106], [0130], FIGS. 7-8. In the highlighted example of Figure 7 (below), Kim describes “a continuous zoom-in action depending on a hold event” detected on an object displayed on the touch

screen. APPLE-1004, [0104]. Moreover, as shown in Figure 7, the detected gesture is shown to be on the touch screen and *on the object* (bike and bicyclist), or a POSITA would have at least found it obvious that the gesture was detected on the object. APPLE-1004, FIG. 7; APPLE-1003, ¶52; §IV.A.1; *see also* APPLE-1004, FIGS. 5-6.



APPLE-1004, FIG. 7 (highlighted)

Kim provides various examples of gestures detected on the touch screen. In each of the examples discussed with respect to Kim's Figures 5-7, Kim's device detects a gesture on the touch screen on the object, at least as illustrated by the

finger shown in each of those figures. APPLE-1004, FIGS. 5-7; APPLE-1003, ¶53; *see also* APPLE-1004, [0108]-[0115], FIG. 8.

[1.f]

Kim teaches or renders [1.f] obvious. APPLE-1003, ¶54. Kim describes “a double tap event for returning to an initial state with a basic display size after a zoom-in or zoom-out action.” APPLE-1004, [0031], [0064], [0099], [0103], [0106]. In describing the touch as a tap, Kim discloses that the duration of each of the taps of the double tap gesture must be less than a threshold value associated with a longer-lasting (i.e., hold) event (the “first magnitude threshold”). APPLE-1003, ¶54. In particular, from this disclosure and based on common knowledge of a tap, a POSITA would have understood that a double tap—a well known user input by the Critical Date—comprises two short duration taps in quick succession, which individually and collectively have a duration shorter than a threshold duration associated with a more persistent gesture (e.g., a hold). APPLE-1005, [0057]-[0058], [0069]; APPLE-1003, ¶54 (citing APPLE-1009; APPLE-1012). The duration of each of the taps of the double tap gesture is “a duration of the gesture.” APPLE-1003, ¶54 (citing APPLE-1010; APPLE-1011). In Kim’s device, when a double tap event is generated, the “device displays target data with a given basic magnification value,” thus performing another operation. APPLE-1004, [0106]; APPLE-1003, ¶55.

[1.g]

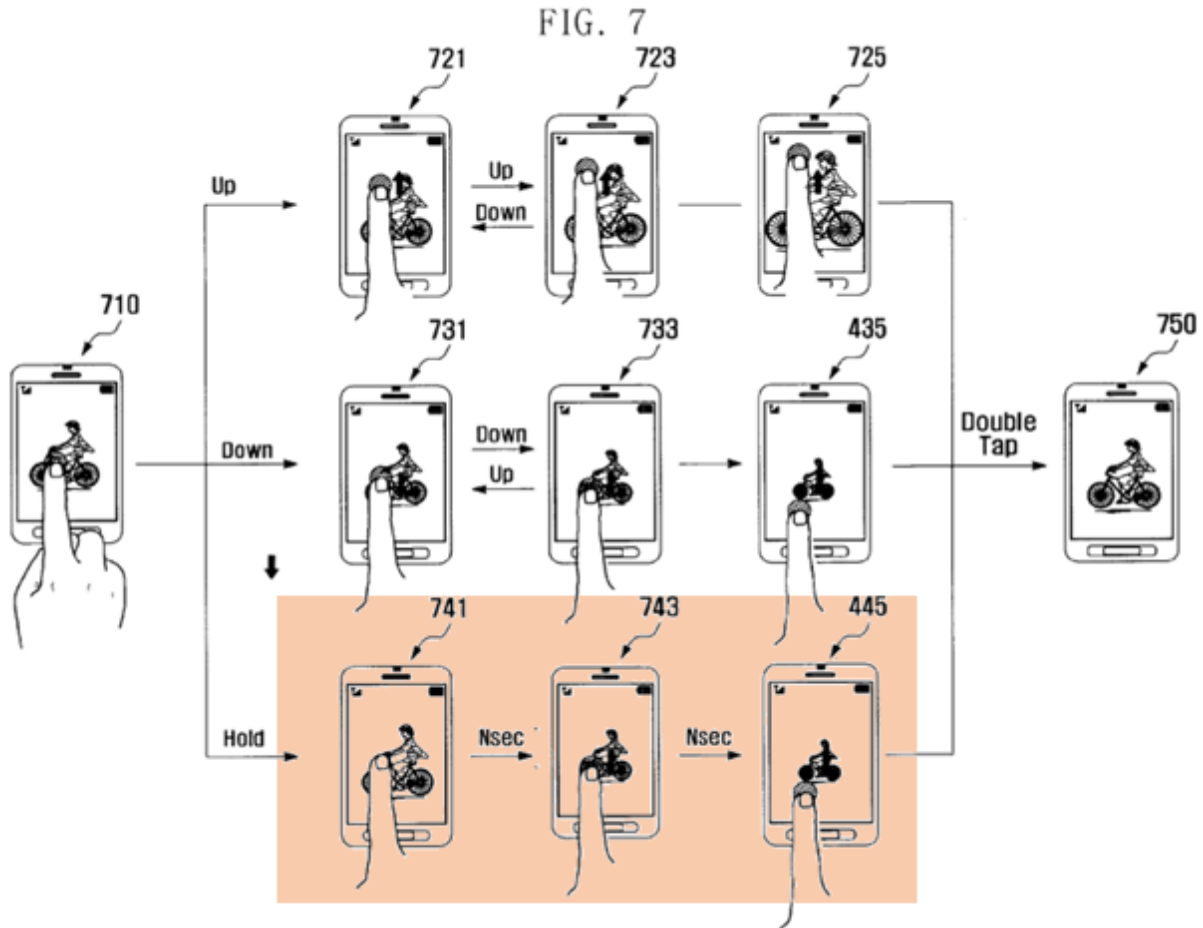
Kim teaches or renders [1.g] obvious. APPLE-1003, ¶56. “[R]eference numbers 741, 743 and 745 show a continuous zoom-in action depending on a hold event,” shown in the bottom path of Figure 7. APPLE-1004, [0104]. For such a hold event, a POSITA would have understood that the magnitude of the duration of the gesture being detected on the touch screen on the object is greater than the first magnitude threshold (threshold value associated with a hold event). APPLE-1003, ¶56 (citing APPLE-1013; APPLE-1014; APPLE-1015). Kim explains that, in “a zoom-in process, the mobile device continuously offers a real-time, scaled-up bitmap image on the screen in response to the duration of the hold event,” and when “the hold event is released from the screen, the mobile device displays an enlarge display target data around the selected point on the touch screen, depending on a zoom-in ratio of a bitmap image.” APPLE-1004, [0105]. Kim further explains that “the hold event is assigned to a continuous zoom-in function is an exemplary case only,” and “the hold event may be alternatively assigned to a continuous zoom-out function.” *Id.*, [0107]; APPLE-1003, ¶57.

Since “the hold event may be alternatively assigned to a continuous zoom-out function,” Kim teaches that bottom path of Figure 7 can be zoom-out instead of zoom-in, as illustrated in Figure 7 (modified), below (highlighted orange).

APPLE-1004, [0107]; APPLE-1003, ¶58. Figure 7 (modified) below shows a user

gesture (hold event) detected on an object (person on bicycle) and the corresponding scaling operation performed on the object. APPLE-1003, ¶58; APPLE-1004, FIG. 7.

Kim also refers to the request for the zoom mode being activated in response to a “long press input” and a “long press event.” APPLE-1004, [0031], [0034], [0042], [0049], [0085], [0139], Claim 2. A POSITA would have understood that Kim’s long press would be discriminated from a normal press based on the contact being maintained (without moving) for longer than a threshold duration. APPLE-1004, [0140]; APPLE-1003, ¶59.



APPLE-1004, FIG. 7 (“hold” scenario modified); APPLE-1003, ¶59

[1.h]⁵

Kim teaches or renders [1.h] obvious. APPLE-1003, ¶60. A POSITA would have understood that a “continuous zoom-out function” on an object, such

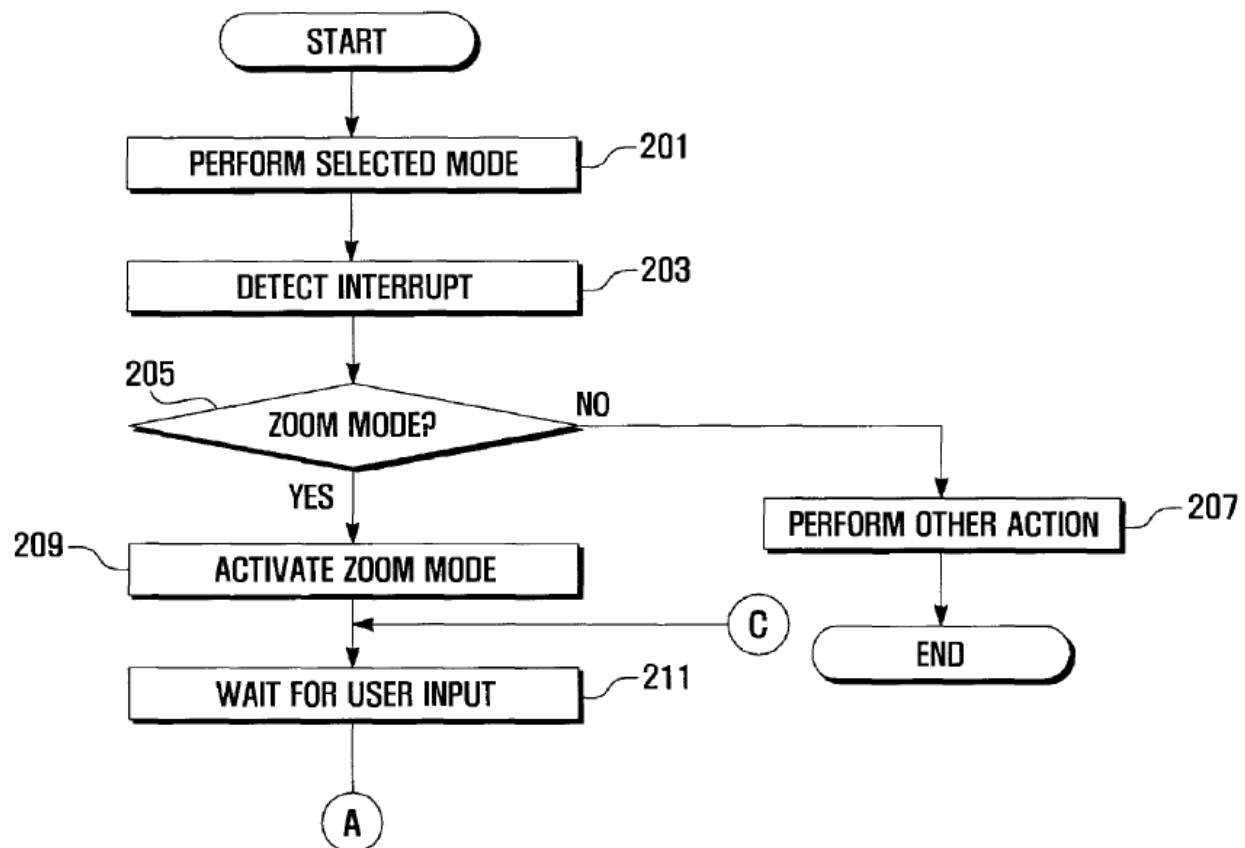
⁵ The discussion of [27.h] in §IV.A.3 is incorporated herein by reference to the extent “such that the size is inversely related to the magnitude of the duration...” is interpreted differently.

as the bike and bicyclist of Kim's Figure 7 or the hyperlink of Figure 8, is the equivalent of scale operation that includes a reduction in size of the object. *Id.* Additionally, Kim teaches that the zoom-out is "continuous": in a zoom-out process, "the mobile device continuously offers a real-time, scaled-[down] bitmap image on the screen in response to the duration of the hold event." *Id.*; APPLE-1004, [0105]. Thus, the longer the duration of the gesture (press and hold), the more the object is zoomed-out, or the more the size of the object is reduced. *Id.* In this way, the reduction in a size is inversely related to the magnitude of the duration of the gesture (the larger the duration of the gesture, the smaller the size). *Id.*; APPLE-1003, ¶¶60-61.

[1.i]

Kim teaches or renders [1.i] obvious. APPLE-1003, ¶62. Kim further describes "another operation" that is performed when the magnitude of the duration of the gesture is greater than a second magnitude threshold—specifically, a minimum threshold to be deemed an intentional touch. Specifically, Kim explains that, "[i]f detecting a user gesture, namely, if receiving an input event from a user through the touch screen, the mobile device determines the type of an input event." APPLE-1004, [0053]. Kim states that, "[i]f an interrupt is not a request for the activation of the zoom mode as the result of determination in step 205, the mobile device can perform some action in connection with an interrupt

(step 207),” shown in Figure 2a below. APPLE-1004, [0050], FIG. 2a; APPLE-1003, ¶63.



APPLE-1004, FIG. 2a

Kim additionally explains that if the mobile device determines that a detected user gesture does not correspond to any particular input event, “the mobile device disregards the received input event.” APPLE-1004, [0053]. Based on Kim’s disclosure of “disregarding” a received input, a POSITA would have understood that, for the detected interrupt (step 203) to be sufficient to cause the

mobile device to “perform other action” (step 207), a magnitude greater than a minimum threshold value (e.g., the minimum threshold of an intentional user input) is relied upon, or rendered obvious, by Kim. APPLE-1004, FIG. 2a; APPLE-1003, ¶64. Thus, when the magnitude of the gesture is greater than a second magnitude threshold, the device “performs other action” (e.g., display requested menu). APPLE-1004, [0050], FIG. 2a; APPLE-1003, ¶64. A POSITA would have understood that such an interrupt would be caused by a gesture that is longer in duration than the second threshold (as necessary to avoid the gesture from being disregarded), shorter than the duration necessary for a long press threshold, and a different gesture than a double tap gesture. APPLE-1003, ¶64.

3. Claim 27

[27.pre]-[27.e]

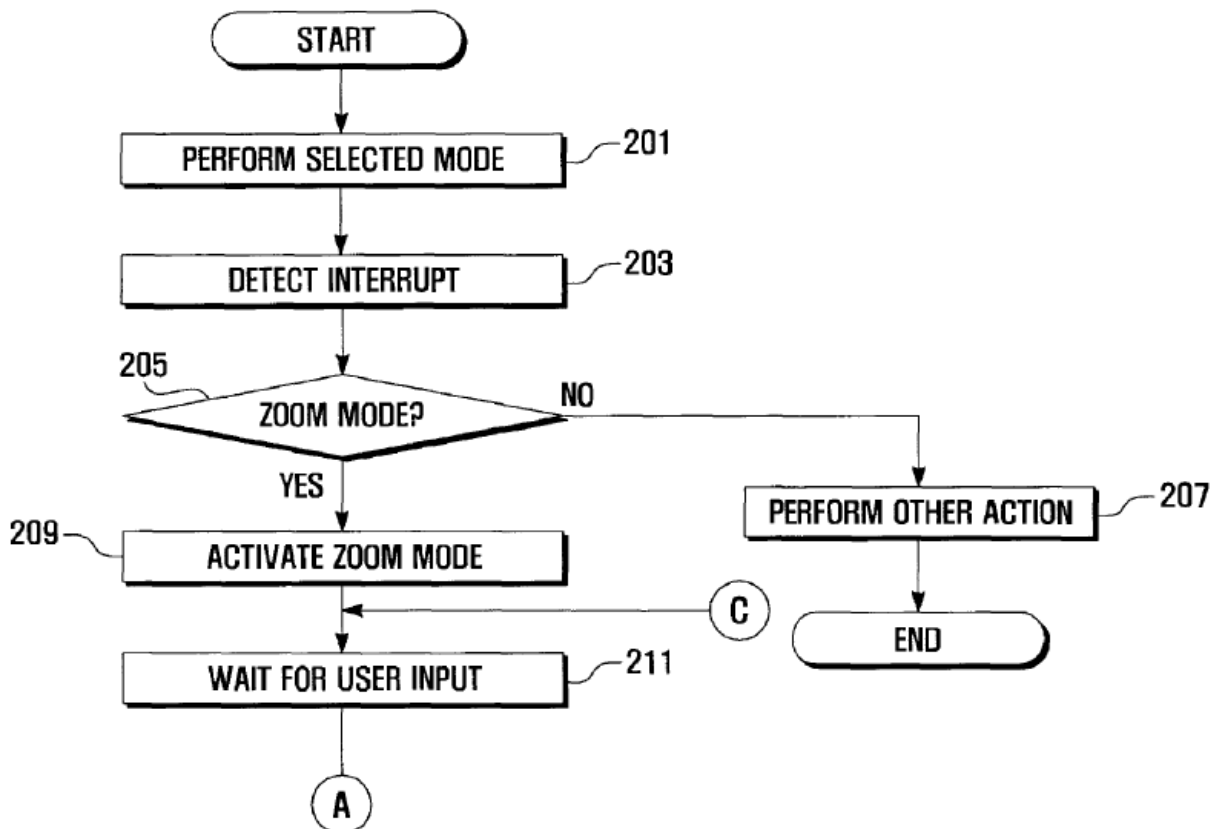
See [1.pre]-[1.e] (Ground 1A).

[27.f]

Kim teaches or renders [27.f] obvious. APPLE-1003, ¶71. Kim describes different modes, or touch states, with respect to its device (e.g., zoom mode, toggle mode) and corresponding functions performed responsive to detected gestures in the respective mode. APPLE-1004, [0034]-[0035]. In Figure 2A, Kim teaches receiving an interrupt (e.g., gesture) at step 203 and “determin[ing] whether [the] interrupt is a request for activating the zoom mode. *Id.*, [0048]-[0049], FIG. 2A.

If not (e.g., if not in a zoom mode), “the mobile device can perform some action in connection with an interrupt (step 207).” *Id.*, [0050] (e.g., display a requested menu, launch an application). “[I]f the interrupt is a request for the toggle mode, the mobile device performs appropriate actions in the toggle mode.” *Id.* In this way, Kim teaches performing a first function (“perform other action 207”) in a first touch state (i.e., when not in the “zoom mode” state, or in “toggle mode”). *Id.*; APPLE-1003, ¶71.

FIG. 2a

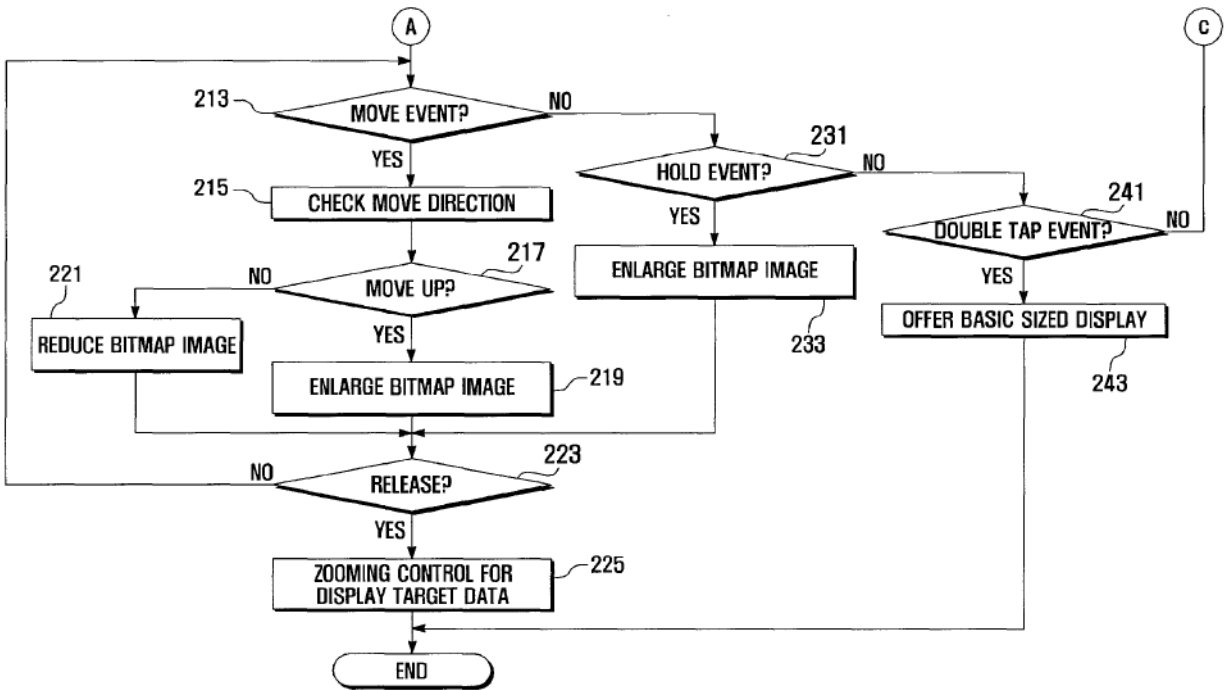


APPLE-1004, FIG. 2A

[27.g]

Kim teaches or renders [27.g] obvious. APPLE-1003, ¶72. Kim teaches performing different functions in response to detected gestures when in a different mode (e.g., zoom mode), or touch state. APPLE-1004, [0034]. “If an interrupt is a request for the activation of the zoom mode as a result of the determination in the step 205, the mobile device activates the zoom mode (step 209).” *Id.*, [0051]. In the zoom mode, the device can perform various actions. *Id.*, [0051]-[0064], FIG. 2B. For instance, with reference to Figure 2B, if the mobile device detects an “input event [that] is a double tap event,” “the mobile device offers display target data with a predefined basic size.” *Id.*, [0064], FIG. 2B. In this way, Kim teaches performing a second function (“offer basic sized display 243”) in a second touch state (i.e., when in the “zoom mode” state). *Id.*; APPLE-1003, ¶72.

FIG. 2b



APPLE-1004, FIG. 2B

[27.h]

Kim teaches or renders [27.h] obvious. APPLE-1003, ¶73. As discussed for [1.g]-[1.h]⁶, in response to a press and hold gesture, the device performs a continuous zoom out (or, alternatively, a zoom in) function. APPLE-1004, [0104]-[0105], [0107], FIG. 7.

⁶ Where a section of the Petition references descriptions from other addressed claim elements, as here, those descriptions are incorporated by reference in the particular section.

A POSITA would have understood that a “continuous zoom-out function” on an object, such as the bike and bicyclist of Kim’s Figure 7, is the equivalent of scale operation that includes a reduction in size of the object. APPLE-1003, ¶74. Additionally, Kim teaches that the zoom-out is “continuous”: in a zoom-out process, “the mobile device continuously offers a real-time, scaled-[down] bitmap image on the screen in response to the duration of the hold event.” *Id.*; APPLE-1004, [0105]; APPLE-1003, ¶75. Thus, the longer the duration of the gesture (press and hold) (i.e., it has a continuous change in duration because the gesture persists), the greater the zoom-out, or the more the size is reduced. *Id.* Accordingly, to the extent this claim is interpreted to require a reduction in size, wherein the size is inversely related to the duration,⁷ Kim discloses the limitation. *Id.*

In the alternative, to the extent the claim is interpreted to require that the *reduction* in size and the *duration* are inversely related (as opposed to the *size* and the *duration* being inversely related), then such an element would have at least been obvious. APPLE-1003, ¶76. For example, Kim teaches that a double tap event (i.e., short duration) causes “enlarged display target data” to be “given basic

⁷ This interpretation is consistent with [1.h] (“a reduction in a size such that the size is inversely related to the magnitude of the duration”). §IV.A.2.

magnification value” (i.e., reduction in size), while a hold event (i.e., long duration) causes a zoom-out function (i.e., reduction in size). APPLE-1004, [0105]-[0107]. That is, in the case of a double tap, if the “enlarged display target data” was very large, the double tap would cause a large reduction in size, whereas, a hold that invokes a sustained and thus longer press would cause a zoom and correspondingly a decrease in the double-tap caused reduction in size. *Id.* In this way, the Kim scale operation responds to a change in a duration of the gesture (e.g., from a short duration to a long one) with a scale reduction that is inversely related to the length of touch duration. *Id.* Indeed, as this example demonstrates, Kim discloses or renders obvious at least one instance or use case in which a gesture with a longer duration (sustained hold or long press) would cause comparatively smaller reduction in size. *Id.*; APPLE-1003, ¶76. The relationship between the duration of a touch and its responsive zoom function (e.g., zoom in/out, rate of zoom, etc.) was a known design choice, and was known to be settable by the system in accordance with a designer’s programming or even by a user. APPLE-1003, ¶76. In this way, the *reduction* in a size is inversely related to the *duration* of the gesture (the larger the duration of the gesture, the less the reduction (i.e., the greater in size)). *Id.*

In another alternative, if the phrase “where the scale operation includes a reduction in a size that is inversely related to the duration” is interpreted to mean

that either the *reduction in size* or the *rate of reduction in size* are inversely related to *duration*, the element would still have at least been obvious. APPLE-1003, ¶77. In particular, a POSITA would have understood that as an image is reduced in size, fewer pixels are changed per step in the zoom-out animation. *Id.* In this way, the *reduction* in a size is inversely related to the *duration* of the gesture. *Id.*

Finally, while the example in Figure 7 shows an object (bike and bicyclist) with a surrounding interface that zooms in or zooms out in response to the press and hold, Figure 8 shows yet another example of the zoom-in or zoom-out effect. APPLE-1004, [0111]-[0115], FIG. 8; APPLE-1003, ¶78. In particular, Figure 8 shows an example where, in response to a user gesture, a zooming action is performed on both an object (e.g., a car) and its surrounding interface (e.g., the remainder of the web page surrounding the object). APPLE-1004, [0111]-[0115], FIG. 8. Thus, Kim's scale operation is performed in connection with the at least portion of the interface. *Id.*; APPLE-1003, 78.

4. Claim 27

This analysis of claim 27 presents "touch states" in a manner consistent with Smith's infringement contentions. In the previous analysis of claim 27 in §IV.A.3, the claimed "touch states" are mapped to Kim's "modes," which cause different functions to be performed, whereas, here, each claimed "touch state" is mapped to a particular user input and its corresponding function.

[27.pre]-[27.e]

See [1.pre]-[1.e] (Ground 1A).

[27.f]

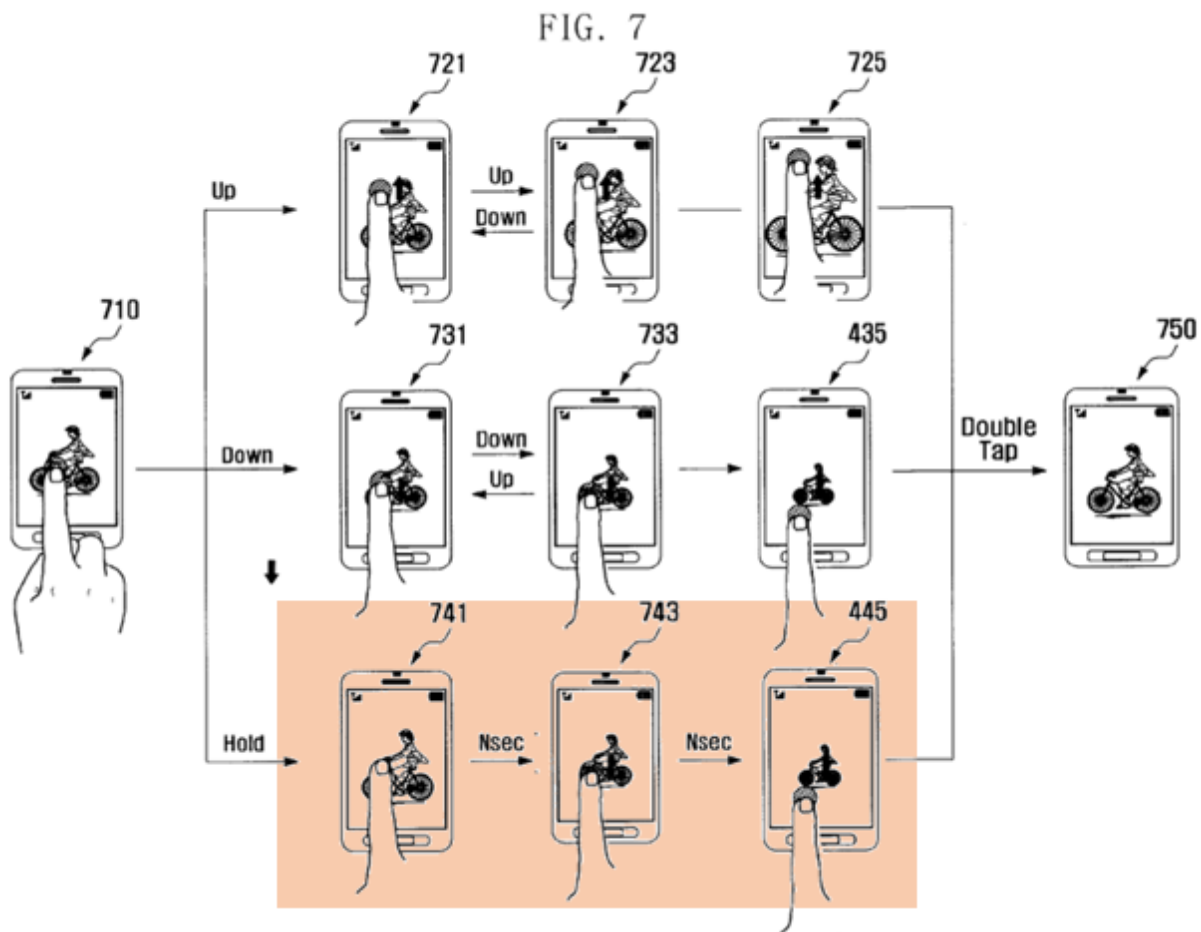
Kim teaches or renders [27.f] obvious. APPLE-1003, ¶86. Smith alleges turning on Bluetooth functionality in response to a tap input meets the requirements of [27.f], thus equating a first touch state with a particular user input. APPLE-1016, 13-14.

Kim similarly teaches a function performed in response to a user input. The top path of Figure 7 “show[s] a continuous zoom-in action depending on a move event.” APPLE-1004, [0104]. Kim explains that “a user can continuously invoke a move up event at a selected point,” and “the mobile device executes a continuous zoom-in function around the selected point in response to the progress of a continuous move up event.” *Id.*, [0096], FIG. 7. In this way, Kim teaches performing a first function (continuous zoom-in function) in a first touch state (continuous move up event). *Id.*; APPLE-1003, ¶¶86-87.

[27.g]

Kim teaches or renders [27.g] obvious. APPLE-1003, ¶88. Smith alleges causing a menu of icons to pop up in response to a long press input meets the requirements of [27.g], thus equating a second touch state with a particular user input. APPLE-1016, 14-15.

Kim similarly teaches a function performed in response to a user input. Specifically, the bottom path of Figure 7 “show[s] a continuous zoom-in action depending on a hold event.” APPLE-1004, [0104]. Kim explains that, in “a zoom-[out] process, the mobile device continuously offers a real-time, scaled-[down] bitmap image on the screen in response to the duration of the hold event.” *Id.*, [0105], [0107] (explaining that the process may be zoom-in or zoom-out); APPLE-1003, ¶¶88-90. In this way, Kim teaches performing a second function (zooming out, or, alternatively, zooming in) in a second touch state (press and hold). *Id.*



APPLE-1004, FIG. 7 (“hold” scenario modified)

[27.h]

Kim teaches or renders [27.h] obvious. APPLE-1003, ¶91. As discussed for [27.g], in response to a press and hold gesture, the device performs a continuous zoom out (or, alternatively, a zoom in) function. APPLE-1004, [0104]-[0105], [0107], FIG. 7.

A POSITA would have understood that a “continuous zoom-out function” on an object, such as the bike and bicyclist of Kim’s Figure 7, is the equivalent of scale operation that includes a reduction in size of the object. APPLE-1003, ¶¶92-93. Additionally, Kim teaches that the zoom-out is “continuous”: in a zoom-out process, “the mobile device continuously offers a real-time, scaled-[down] bitmap image on the screen in response to the duration of the hold event.” *Id.*; APPLE-1004, [0105]. Thus, the longer the duration of the gesture (press and hold), the greater the zoom-out, or the more the size is reduced. *Id.* Accordingly, to the extent this claim is interpreted to require a reduction in size, wherein the size is inversely related to the duration, Kim discloses the limitation. *Id.*

In the alternative, to the extent the claim is interpreted to require that the *reduction* in size and the *duration* are inversely related (as opposed to the *size* and the *duration* being inversely related), then such an element would have at least been obvious. APPLE-1003, ¶94. For example, Kim teaches that a double tap

event (i.e., short duration) causes “enlarged display target data” to be “given basic magnification value” (i.e., reduction in size), while a hold event (i.e., long duration) causes a zoom-out function (i.e., reduction in size). APPLE-1004, [0105]-[0107]. That is, in the case of a double tap, if the “enlarged display target data” was very large, the double tap would cause a large reduction in size, where, a long press that was relatively short would cause a smaller reduction in size. *Id.* Thus, the scale operation is responsive to a change in a duration of the gesture (e.g., from a short duration to a long one). *Id.* Thus, the gesture with the longer duration of the gesture (long press), would cause a comparatively smaller reduction in size. *Id.*; APPLE-1003, ¶94. In this way, the *reduction* in a size is inversely related to the *duration* of the gesture (i.e., a gesture with a larger duration causes a smaller reduction in size than a gesture with a shorter duration). *Id.*

In another alternative, if the phrase “where the scale operation includes a reduction in a size that is inversely related to the duration” is interpreted to mean that either the *reduction in size* or the *rate of reduction in size* are inversely related to *duration*, then such an element would have at least been obvious. APPLE-1003, ¶95. In particular, a POSITA would have understood that as an image is reduced in size (as zoomed out over time), fewer pixels are changed per step in the zoom-out animation (e.g., a step change from 100% to 99% zoom would change a larger number of pixels than a step change from 2% to 1%). *Id.* In this way, the

reduction in a size is inversely related to the *duration* of the gesture. *Id.*

Finally, while the example in Figure 7 shows an object (rider on a bicycle) with a surrounding interface that zooms in or zooms out in response to the press and hold, Figure 8 shows yet another example of the zoom-in or zoom-out effect. APPLE-1004, [0111]-[0115], FIG. 8. In particular, Figure 8 shows an example where, in response to a user gesture, both an object (e.g., a car) and its surrounding interface (e.g., the remainder of the web page surrounding the object), a zooming action is performed on both the object and its surrounding interface. *Id.* Thus, Kim's scale operation is performed in connection with the at least portion of the interface. *Id.*; APPLE-1003, ¶96.

5. Claim 32

[32.pre]-[32.c]

See [27.pre]-[27.c] (Ground 1A).

[32.d]

Kim teaches or renders [32.d] obvious. APPLE-1003, ¶101. As discussed for [1.d], Kim displays a first object. *Id.* To the extent the device shown in Kim's Figure 7 is interpreted as a single object, a POSITA would have found the display of at least one other object obvious and understood Figure 7 to simply be an example. *Id.* Indeed, Figure 8 of Kim, for example, shows more than one object (e.g., a car, text, a person, link assigned to text "AUTO"). APPLE-1004, FIG. 8.

Thus, a POSITA would have found the display of any number of objects at least to be obvious. APPLE-1003, ¶101.

[32.e]

Kim teaches or renders [32.e] obvious. APPLE-1003, ¶102. *See* [1.e] (Ground 1A). A POSITA would have understood or found obvious that the gesture detected on Kim's device's touch screen includes a plurality of time-related components, including, for example, how long the gesture has been maintained at any point in time during or at cessation of the gesture. *Id.* The implication of these time-related components in Kim would have been readily understood by a POSITA to reveal or render obvious various types of gestures based on the length of the gesture. *See* APPLE-1004, [0034] ("*long* press"), [0061] ("duration of the hold event"), [0105] ("*duration* of the hold event"), [0139] ("*long* press"); APPLE-1003, ¶102. Additionally, a POSITA would have understood from Kim's discussion of interacting with content on the touch screen that the gesture can be received (and detected) and any number of objects displayed on the screen. APPLE-1003, ¶103. For instance, regarding Figure 8, Kim describes, among the displayed objects, a hyperlink with the text "AUTO" ("first object"); a POSITA would have found it obvious for a user to input a gesture on the hyperlink, while not inputting the gesture on other objects (e.g., car). APPLE-1004, [0113]-[0114], FIG. 8.

[32.f]

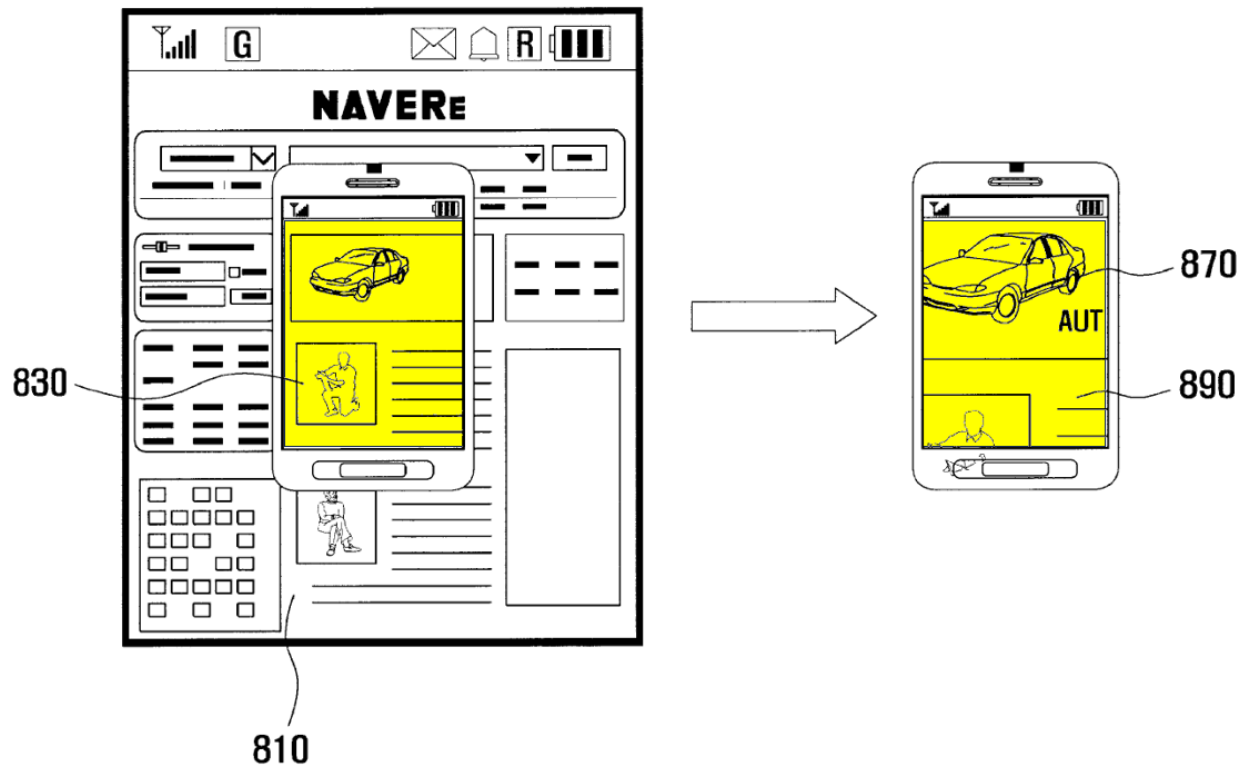
Kim teaches or renders [32.f] obvious for at least the same reasons as discussed for [1.f]. APPLE-1003, ¶104. Specifically, when a first component of the gesture (component when a double tap event is generated) is detected where the duration detected on the touch screen is less than a first magnitude (e.g., threshold associated with a hold event), Kim's device performs a function (displays target data with a given basic magnification value). *Id.*; *see* [1.f]. Moreover, a POSITA would have understood that, where the function occurs after timeout, shortly after cessation of the gesture (when the user releases his finger), the function is performed upon cessation of the gesture being detected. APPLE-1003, ¶104; APPLE-1004, [0103] ("Thereafter, a user can invoke a double tap event on the screen. Then, *in response to a double tap event*, the mobile device displays target data with a given basic magnification value...").

[32.g]

Kim teaches or renders [32.g] obvious for at least the same reasons as discussed for [1.g]. APPLE-1003, ¶105. In particular, when a second component of the gesture (component when a hold event is detected) is detected where the duration detected on the touch screen is greater than the first magnitude (threshold associated with a hold event), Kim's device performs a function (a zoom or scale operation). *Id.*; *see* [1.g] (Ground 1A). Moreover, a POSITA would have

understood that the scale operation is performed on content displayed on the screen, including both the first object (hyperlink) and the at least one other object (e.g., car, person). APPLE-1003, ¶105. Figure 8 of Kim (below) shows an example showing how all displayed content (highlighted) is scaled in response to the gesture. *Id.*; APPLE-1004, FIG. 8.

FIG. 8



APPLE-1004, FIG. 8 (highlighted)

[32.h]

See [1.h].

[32.i]

Kim teaches or renders [32.i] obvious for at least the same reasons as discussed for [1.i]. APPLE-1003, ¶107. Specifically, when a third component of the gesture (component when the interrupt is detected) is detected where the duration detected on the touch screen is greater than a second magnitude (the minimum threshold of an intentional user input), Kim’s device performs another function (“perform other action”). *Id.*; APPLE-1004, FIG. 2a (207); *see* [1.i].

B. [GROUND 1B] – The Kim-Lang Combination renders obvious claims 34, 59, and 64

1. Lang

Lang is directed to a “user interface for small computing devices,” including “smart watches.” APPLE-1006, Abstract, [0015]. “An exemplary smart watch 102 embodying the present invention” includes input elements, such as “touch sensitive elements.” *Id.*, [0016]-[0017]. The smart watch includes a “display screen 118” that “may be touch-sensitive, thereby providing activation signals to the smart watch 102 when the display screen 118 is contacted by the user.” *Id.*, [0020]; APPLE-1003, ¶108.

2. The Kim-Lang Combination

Kim describes a mobile device with “a touch sensitive surface such as a touch screen” that can “include, but is not limited to, a device selected from the group comprising *any other kinds of portable and handheld devices.*” APPLE-

1004, [0118]. To further understand what other types of devices Kim’s mobile device could be implemented as, a POSITA would have naturally looked at other similar discussions of portable devices. APPLE-1003, ¶109.

For example, Lang describes a “user interface for small computing devices,” explaining that “[s]uch small computing devices include ... *smart watches*.” APPLE-1006, Abstract, [0015] (emphasis added). Like Kim, Lang describes a smart watch having “touch sensitive elements,” such as a “display screen 118” that is “touch-sensitive, thereby providing activation signals to the smart watch 102 when the display screen 118 is contacted by the user.” APPLE-1006, [0017], [0020]. Thus, a POSITA would have understood that “any other kinds of portable and handheld devices” include smart watches, as Lang teaches. APPLE-1003, ¶110; APPLE-1006, [0015].

Given that Lang’s smart watch includes a touch-sensitive display screen, a POSITA would have understood that Kim’s device could be implemented as such a smart watch, e.g., allowing a user to invoke zooming functions, discussed above in §IV.A. APPLE-1003, ¶111. Moreover, because Kim contemplates implementing its device in “any other kinds of portable and handheld devices” a POSITA would have reasonably expected success in implementing Kim’s device as a smart watch. *Id.* Combining Kim with Lang (the Kim-Lang Device, or “KLD”) would have been obvious at least because such a combination would have

merely involved (1) combining prior art elements according to known methods to yield predictable results and (2) the use of known technique to improve similar devices (methods, or products) in the same way. *See KSR*, 550 U.S. at 415-421 (2007); MPEP §2143.

3. Claim 34

[34.pre]

To the extent the preamble is limiting, the KLD renders [34.pre] obvious. APPLE-1003, ¶112. Specifically, as discussed for claim 1, Kim’s device performs various functions. §IV.A. For at least the same reasons, the KLD performs a similar functions including functions recited in [34.e]-[34.j], which were discussed in other referenced/incorporated sections of this Petition. APPLE-1003, ¶112; *see* [1.pre] (Ground 1A).

[34.a]

The KLD renders [34.a] obvious. APPLE-1003, ¶113. Kim explains that its mobile device can be “any other kinds of portable and handheld devices.” APPLE-1004, [0118]. As discussed in §IV.B.2, a POSITA would have found it obvious to implement the KLD as a smart watch, per Lang. APPLE-1003, ¶113.

[34.b]-[34.j]

See [1.a]-[1.i] (Ground 1A).

4. Claim 59

[59.pre]-[59.d]

See [34.pre]-[34.d] (Ground 1B).

[59.e]-[59.i]

See [27.d]-[27.h] (Ground 1A).

5. Claim 64

[64.pre]-[64.d]

See [34.pre]-[34.d] (Ground 1B).

[64.e]-[64.j]

See [32.d]-[32.i] (Ground 1A).

C. [GROUND 1C] – The Kim-Burrough Combination renders obvious claim 30

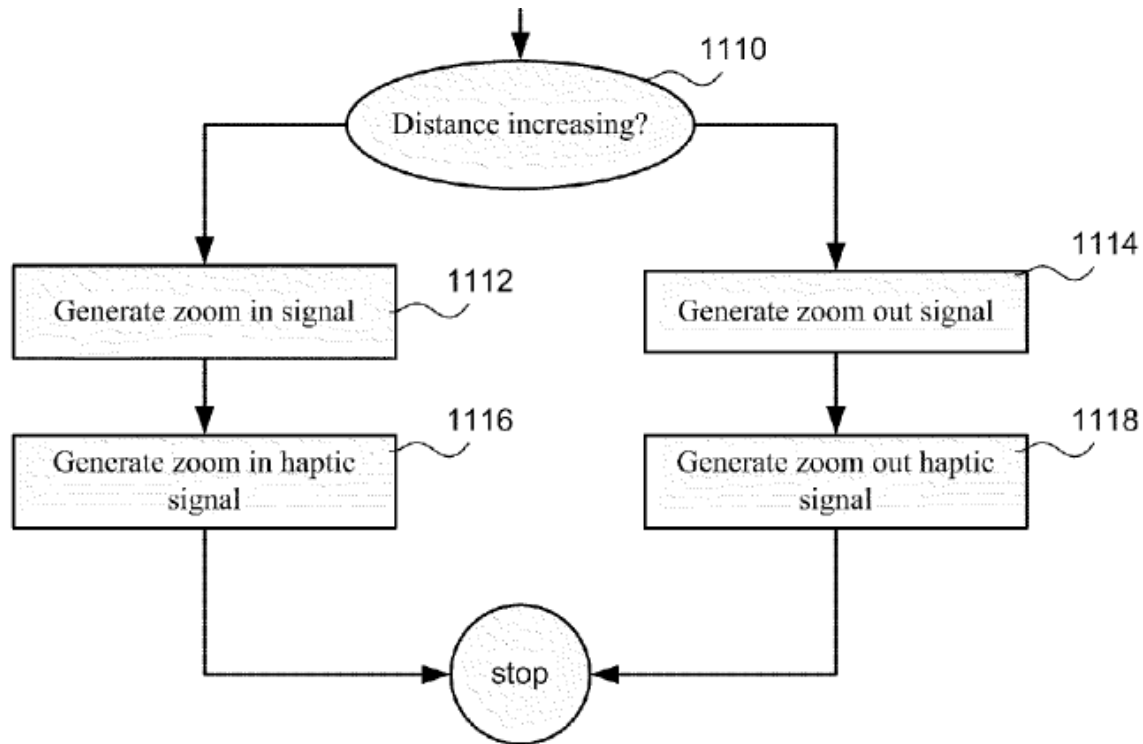
1. Burrough

Burrough describes “processing touch inputs.” APPLE-1007, Abstract.

Burrough explains that “providing [a] multi-touch surface with the ability to provide physical (haptic) feedback makes the multi-touch experience even more efficient and realistic to the user.” *Id.*, [0008]. Burrough describes a portable device that includes a “haptic actuator” that “creates haptic response” (“vibro-tactile response”). *Id.*, [0037], [0047], FIGS. 1A-1E; *see also id.*, [0055]-[0056], [0013], [0033]; APPLE-1003, ¶133.

With reference to Figure 11, Burrough describes “a zoom gesture method” “performed on multi-touch multi-haptic surface.” APPLE-1007, [0079], FIG. 11.

As shown below, following any zoom signal (1112, 1114), a haptic signal is generated (1116, 1118). *Id.*, FIG. 11; APPLE-1003, ¶134.



APPLE-1007, FIG. 11 (excerpted)

2. The Kim-Burrough Combination

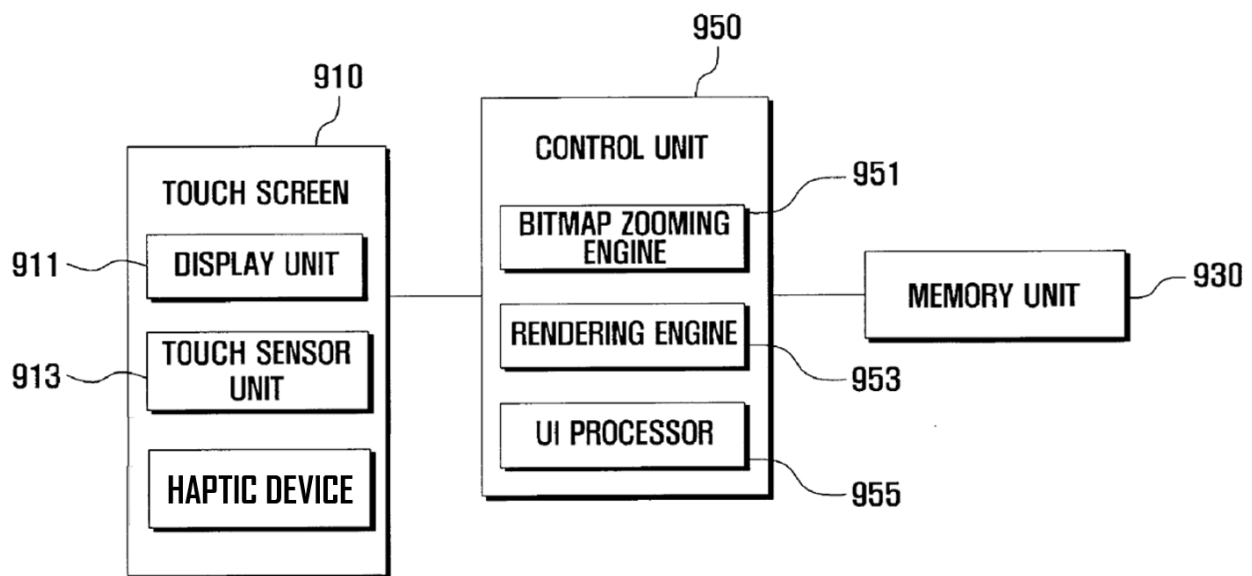
Kim describes a mobile device that “has a touch sensitive surface such as a touch screen” that provides “a touch-based zooming control method.” APPLE-1004, Abstract. Kim does not explain whether tactile feedback is provided to a user, instead leaving this detail to POSITAs well aware of haptic feedback by the Critical Date. APPLE-1003, ¶135 (citing APPLE-1008). The lack of a physical keyboard or tactile means was known to create challenges for users (e.g., visually distracted or visually impaired, poor lighting, small screen). *Id.* (citing APPLE-

1017; APPLE-1018). Thus, a POSITA would have been motivated to look to references, such as Burrough, which teaches using haptic feedback in conjunction with touch screens. *Id.*; APPLE-1007, Abstract.

Indeed, Burrough recognizes that “physical sensation [cannot] be provided by a conventional multi-touch system thereby forcing the user to visually locate the home key thereby making keyboard use less efficient and fatiguing.” APPLE-1007, [0008]. However, “providing the multi-touch surface with the ability to provide physical (haptic) feedback makes the multi-touch experience even more efficient and realistic to the user.” *Id.* Thus, Burrough teaches providing haptic feedback in response to user input, including zooming operations. *Id.*, [0079]-[0081], FIG. 11. The haptic feedback is provided using a haptic device (e.g., piezoelectric actuator assembly) in the portable device. *Id.*, [0013], [0033], [0037], [0047], [0055]-[0056], FIGS. 1A-1E; APPLE-1003, ¶136.

A POSITA would therefore have been motivated to include a haptic device and use of haptic feedback, per Burrough, in the device to provide haptic feedback to improve the user experience with the touch screen. APPLE-1003, ¶137. Specifically, a POSITA would have been motivated to provide haptic feedback to a user invoking a zoom operation, per Burrough, e.g., to provide an indication of whether a zoom gesture has been invoked. *Id.*; APPLE-1007, [0080]. Burrough’s haptic actuator is in communication with a micro-controller and processor.

APPLE-1007, [0046]-[0047], FIG. 1B; APPLE-1003, ¶137. In the Kim-Burrough Device (“Kim-Burrough-Device”), the haptic device would be in communication with the processor (“control unit”), such that the control unit controls the haptic device and generating haptic responses, as illustrated below in modified Figure 9 from Kim. APPLE-1003, ¶137.



APPLE-1004, FIG. 9 (modified); APPLE-1003, ¶137

Given that Burrough teaches including a haptic device in a portable device, and Kim provides such a portable device, a POSITA would have had a reasonable expectation of success in implementing the combination. APPLE-1003, ¶138.

Combining Kim with Burrough would have been obvious at least because such a combination would have merely involved (1) combining prior art elements according to known methods to yield predictable results and (2) the use of known

technique to improve similar devices (methods, products) in the same way. *See KSR*, 550 U.S. at 415-421 (2007); MPEP §2143.

3. Claim 30

[30.pre]-[30.b]

See [1.pre]-[1.b] (Ground 1A).

[30.c]

The Kim-Burrough-Device renders [1.c] obvious. APPLE-1003, ¶142. As discussed in §IV.C.2, the Kim-Burrough-Device further includes a haptic device (e.g., piezoelectric actuator assembly), which is a vibrator that provides a “vibro-tactile response.” APPLE-1007, [0013], [0033], [0037], [0047], [0055]-[0056], FIGS. 1A-1E; APPLE-1003, ¶142; §IV.C.2.

[30.d]

The Kim-Burrough-Device renders [30.d] obvious for at least the same reasons as discussed for [1.c]. APPLE-1003, ¶143. As discussed in §IV.C.2, the Kim-Burrough-Device further includes a vibrator in communication with the control unit of Kim. APPLE-1003, 143; §IV.C.2.

As discussed in §IV.C.2, the Kim-Burrough-Device further includes a vibrator in communication with the one or more processors. APPLE-1003, ¶144; §IV.C.2.

[30.e]

See [32.d] (Ground 1A).

[30.f]

See [1.e] (Ground 1A).

[30.g]

The Kim-Burrough-Device renders [30.g] obvious for at least the same reasons as discussed for [32.g]-[32.h]. §IV.A.3; APPLE-1003, ¶147. As discussed for [32.g]-[32.h], the Kim-Burrough-Device performs a scale operation on a displayed object other than the first object subject to the gesture based on a change in a duration of a detected gesture on the first object (e.g., the gesture continuing and thus increasing in duration). APPLE-1003, ¶147. As discussed for [30.e], a POSITA would have found the display of more than one object to at least be obvious. *Id.* Additionally, a POSITA would have found it obvious that the scale operation would be performed on the first object and at least one other object, as clearly shown in Figure 8 of Kim. *Id.*; APPLE-1004, FIG. 8.

Moreover, Kim teaches that the scaling operation (e.g., continuous zoom) occurs during the tap and hold event: “the mobile device executes a continuous zoom-in [or zoom-out] function around the selected point at given periodic intervals while the hold event continues.” APPLE-1004, [0104], [0107]. Thus, a POSITA would have understood that the scale operation is performed during

detection of at least a portion of the gesture before a completion thereof is detected.

APPLE-1003, ¶148.

[30.h]

See [32.h] (Ground 1A).

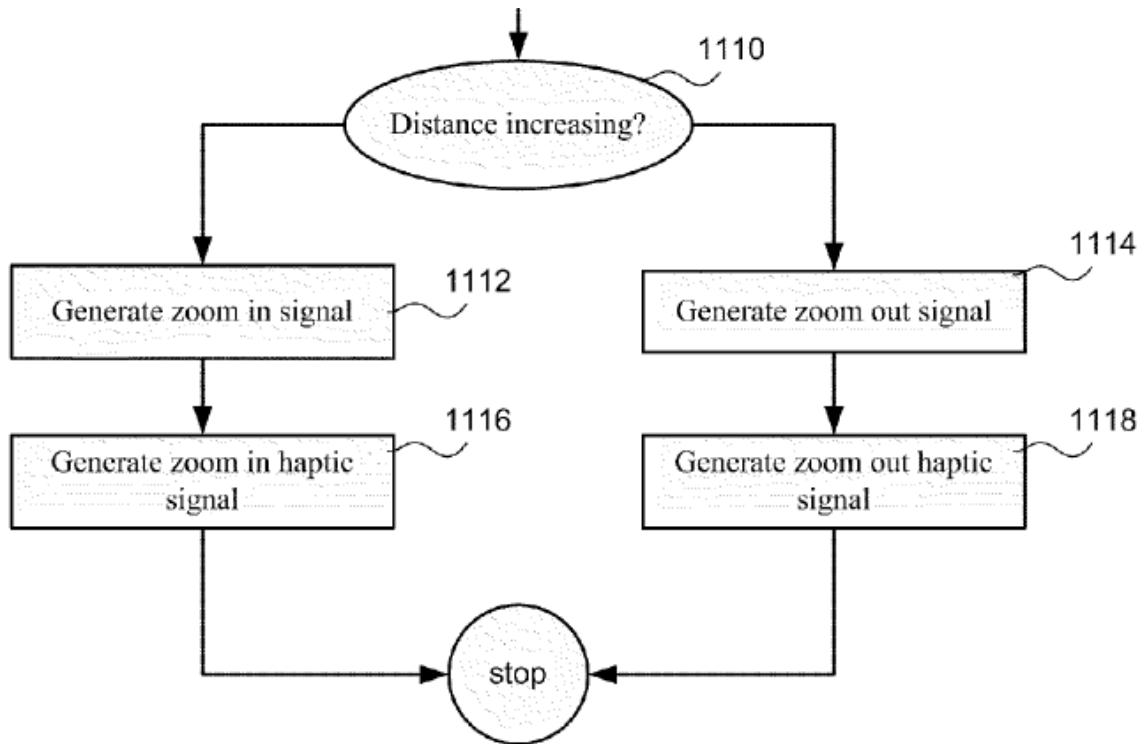
[30.i]

The Kim-Burrough-Device renders [30.i] obvious. APPLE-1003, ¶150.

Kim explains that after a hold event that causes a zoom-in (or zoom-out) process, “[t]hereafter, a user can invoke a double tap event on the screen,” causing the device to display “target data with a given basic magnification value.” APPLE-1004, [0106], FIG. 7; *see id.*, FIG. 2b. Thus, after at least a portion of the scale operation, the device performs an action (displays data with a given basic magnification). *Id.*; APPLE-1003, ¶150.

As discussed in §IV.C.2, the Kim-Burrough-Device includes a haptic device that provides a “vibro-tactile response.” APPLE-1007, [0013], [0033], [0037], [0047], [0055]-[0056], FIGS. 1A-1E, FIG. 11; APPLE-1003, ¶151; §IV.C.2. As shown in Figure 11 (excerpted below), Burrough teaches “generate a zoom out haptic signal” or “generate a zoom in haptic signal” after generating a zoom out or zoom in signal. APPLE-1004, [0080], FIG. 11. A POSITA would have understood that a haptic signal (1116/1118 of Figure 11) follows any zooming signal in the Kim-Burrough-Device, and therefore, the zooming action that occurs

to “offer basic sized display” (243 in FIG. 2b of Kim) in the Kim-Burrough-Device would be accompanied by a haptic output, as Burrough teaches. APPLE-1003, ¶151.



APPLE-1007, FIG. 11 (excerpted)

D. [GROUND 1D] – The Kim-Lang-Burrough Combination renders obvious claim 62

1. The Kim-Lang-Burrough Combination

The Kim-Lang Device (KLD) is discussed above in §IV.B.2. For at least the same reasons discussed in §IV.C.2, it would have further been obvious to a POSITA to modify the KLD to include a haptic device and use of haptic feedback, as Burrough teaches and suggests. APPLE-1003, ¶152.

2. Claim 62

[62.pre]-[62.c]

See [34.pre]-[34.c] (Ground 1B).

[62.d]-[62.j]

See [30.c]-[30.i] (Ground 1C).

E. [GROUND 2A] – The Kim-Bowens Combination renders obvious claims 1, 27, and 32

1. Bowens

Bowens is directed to a “state machine gesture recognition algorithm for interpreting streams of coordinates received from a touch sensor.” APPLE-1005, Abstract. “A touch” is determined to be a touch “if certain criteria are satisfied,” and “[i]f a touch of a duration longer than a threshold duration is sensed in the touch state, then control flow passes to a press state module, wherein the press state is for handling longer touches.” *Id.*, [0023]; APPLE-1003, ¶164.

Bowens describes various types of user actions and how each is identified. APPLE-1005, [0055]-[0077], FIGS. 1-3. For example, a “tap event is generated when the user makes a tap gesture” (i.e., “when the user quickly touches and releases the touch surface”) and could be used to activate a hyperlink. *Id.*, [0056], [0065]. “A double tap happens when the user quickly touches and releases the touch surface twice in quick succession,” and can be “used, for example, to select a word in a displayed document.” *Id.*, [0057]; APPLE-1003, ¶165.

“A press event is generated when the user touches the touch surface and leaves their finger stationary.” *Id.*, [0067]. “A short press event is generated after a press event, when the user has continued to leave their finger stationary for a period”; a “long press event is generated after a short press event, when the user has continued to leave their finger stationary for a period.” *Id.*, [0068]-[0069]; APPLE-1003, ¶166.

2. The Kim-Bowens-Device (“KBD”)

As discussed in §IV.A.1, Kim describes a mobile device that performs various functions (e.g., zoom operation) responsive to a particular input gesture. APPLE-1003, ¶¶167-170.

While Kim provides these teachings relating to a zoom function, Kim leaves to well-informed POSITAs various specific implementation details pertaining to how the system identifies and differentiates the various touches and gestures. APPLE-1003, ¶171; *see* APPLE-1004, [0053]. Thus, a POSITA would have naturally looked to and reasonably expected to succeed in integrating the disclosed detail found in prior art related to touch gestures, such as Bowens, which includes, e.g., implementation details and suggestions on how to distinguish between different types of touch gestures and their corresponding operations, as was well known by the Critical Date. APPLE-1003, ¶171.

As discussed in §IV.E.1, Bowens is directed to a “state machine gesture

recognition algorithm for interpreting streams of coordinates received from a touch sensor.” APPLE-1005, Abstract, FIGS. 1-3. Like Kim, Bowens describes “events” that are generated in response to detected gestures. APPLE-1003, ¶172; *see, e.g.*, APPLE-1004, [0035]; APPLE-1005, [0065], [0066]-[0069].

Significantly, Bowens describes various types of user gestures are identified and distinguished from one another. APPLE-1005, [0055]-[0077]. For example, a “tap event is generated when the user makes a tap gesture” and could be used, e.g., to active a hyperlink on a web page. *Id.*, [0056], [0065]. Bowens teaches that the “touched state occurs when a user has touched the touch surface,” and “[w]hen the Touched state is entered a timeout is started, i.e. a timer is set running which expires after a preset time.” *Id.*, [0080]; APPLE-1003, ¶173.

In this Touched state, if the user releases the touch, “the state machine enters the Second Tap Pending state,” which causes a timeout to start that “is used to decide if the user has made a tap gesture.” *Id.* If the timeout expires, “a tap event is generated.” *Id.*, [0081]. However, if “the user presses the touch surface” during the Second Tap Pending state and at a location within threshold distance of the location of the first touch, “the state machine enters the Double Tap pending state,” which is later confirmed as a double tap if the user releases his touch within a timeout parameter. *Id.*, [0082]; APPLE-1003, ¶174.

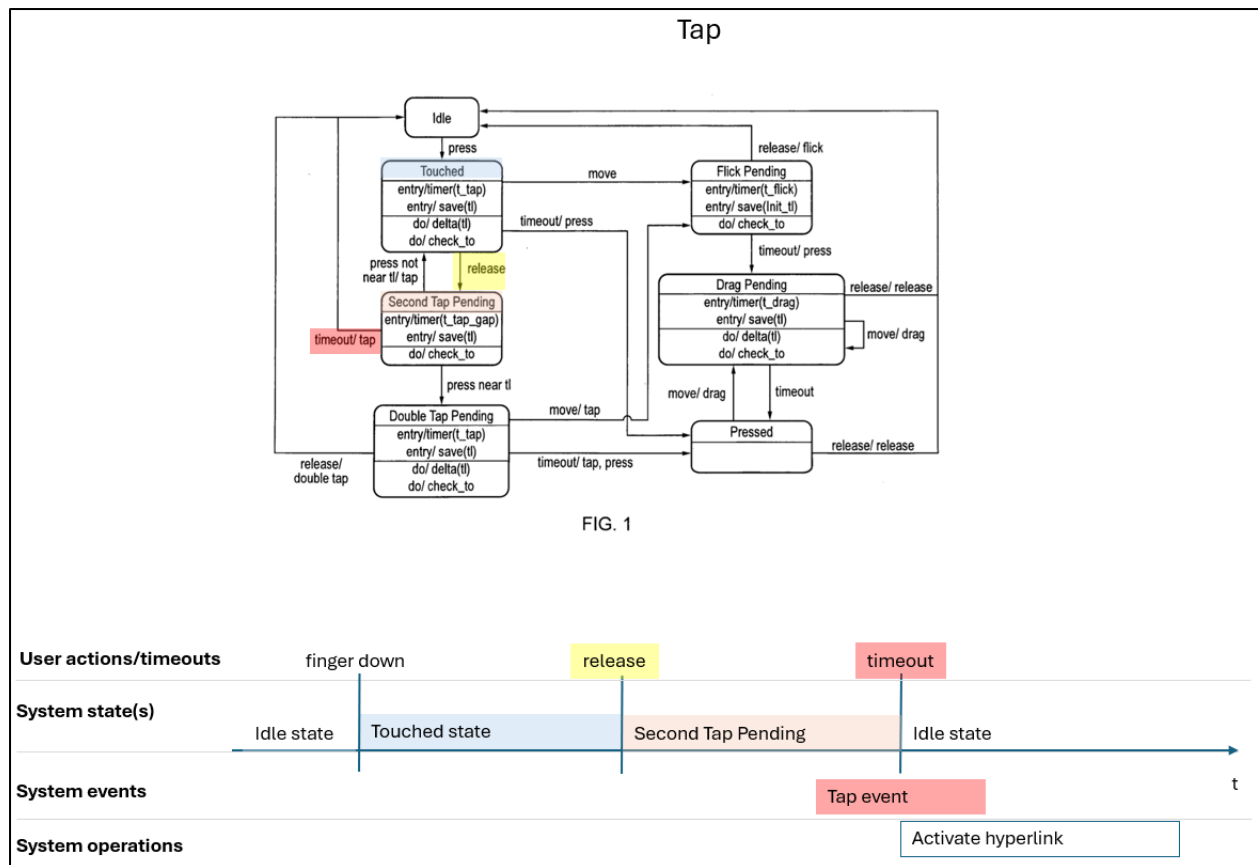
Bowens further teaches that, “[i]f the [Touched state] timeout expires, the

state machine undergoes a transition by generating a press event and then enters the Pressed state.” APPLE-1005, [0080], [0023]; *see also* [0029], [0080].

Notably, like Kim, Bowens contemplates a “long press”: A “Long Pending state is for when a user has pressed the touch surface and has been stationary for some time,” at which point “a timeout is started,” which is “used to determine when to generate a long press event.” APPLE-1003, ¶175; APPLE-1005, [0085]-[0098], FIG. 3. When “the timeout expires, a long press event is generated.” *Id.*

In view of Bowens’s teachings with respect to detecting and distinguishing between different types of user inputs, a POSITA would have found it obvious to implement Kim’s device with methods for recognizing gestures (e.g., distinguishing between a tap, double tap, and long press) and employing well-known tap-related functions, per Bowens. APPLE-1003, ¶176. Specifically, a POSITA would have implemented the Kim-Bowens device (“KBD”) such that the device uses a gesture recognition algorithm, per Bowens, for addressing received user inputs and for well-known tap-related functionality. *Id.* For example, after the KBD receives a touch input on the touch screen, the KBD first enters a Touched state, storing the location of the touch and starting a timer. *Id.*; APPLE-1005, [0080], [0023], FIG. 2. If the user releases his/her touch during the Touched state, the device enters a “Second Tap Pending state,” and another timer starts. APPLE-1003, ¶176; APPLE-1005, [0081], FIG. 2. As in Bowens, in the KBD, the

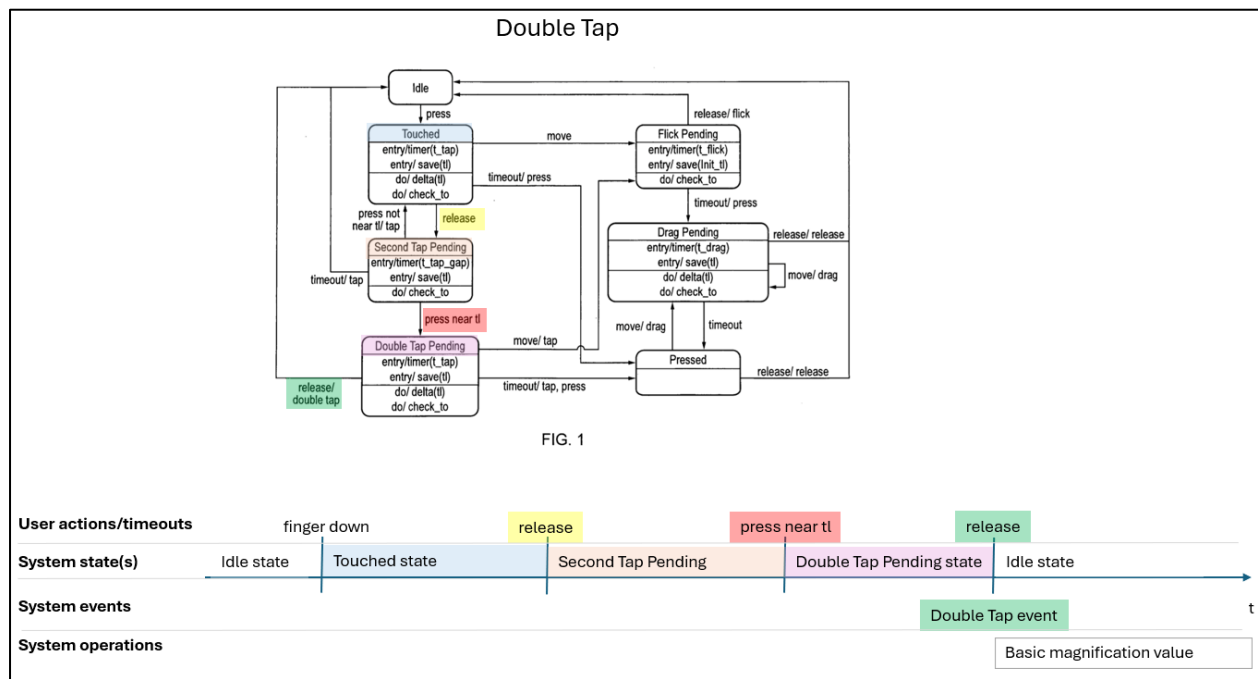
“Second Tap Pending state occurs after a user has tapped the touch surface” and causes a timeout to start, which “is used to decide if the user has made a tap gesture.” *Id.* If that timeout expires, “a tap event is generated” and the gesture is identified as a tap, which, as Bowens teaches, “activates a hyperlink on a displayed web page.” APPLE-1003, ¶176; APPLE-1005, [0056], [0081]. An illustration of a “tap event” in the KBD is provided below using Bowens’s Figure 1 flow chart:



APPLE-1003, ¶176; APPLE-1005, FIG. 1 (highlighted)

The KBD would further use a method for identifying a double tap, per Bowens. APPLE-1003, ¶177. Specifically, during the Second Tap Pending state,

if another touch is detected at a near to the first touch prior to the expiry of the timeout, then the state is changed to “Double Tap Pending.” APPLE-1005, [0081]-[0082]; APPLE-1003, ¶177. In the KBD, if a double tap is detected (“the state machine generates a double tap event and enters the Idle state,” per Bowens’s method), the “device displays target data with a given basic magnification value.” APPLE-1004, [0106]; APPLE-1005, [0082]; APPLE-1003, ¶177. An illustration of a “double tap event” in the KBD is provided below using Bowens’s Figure 1 flow chart:

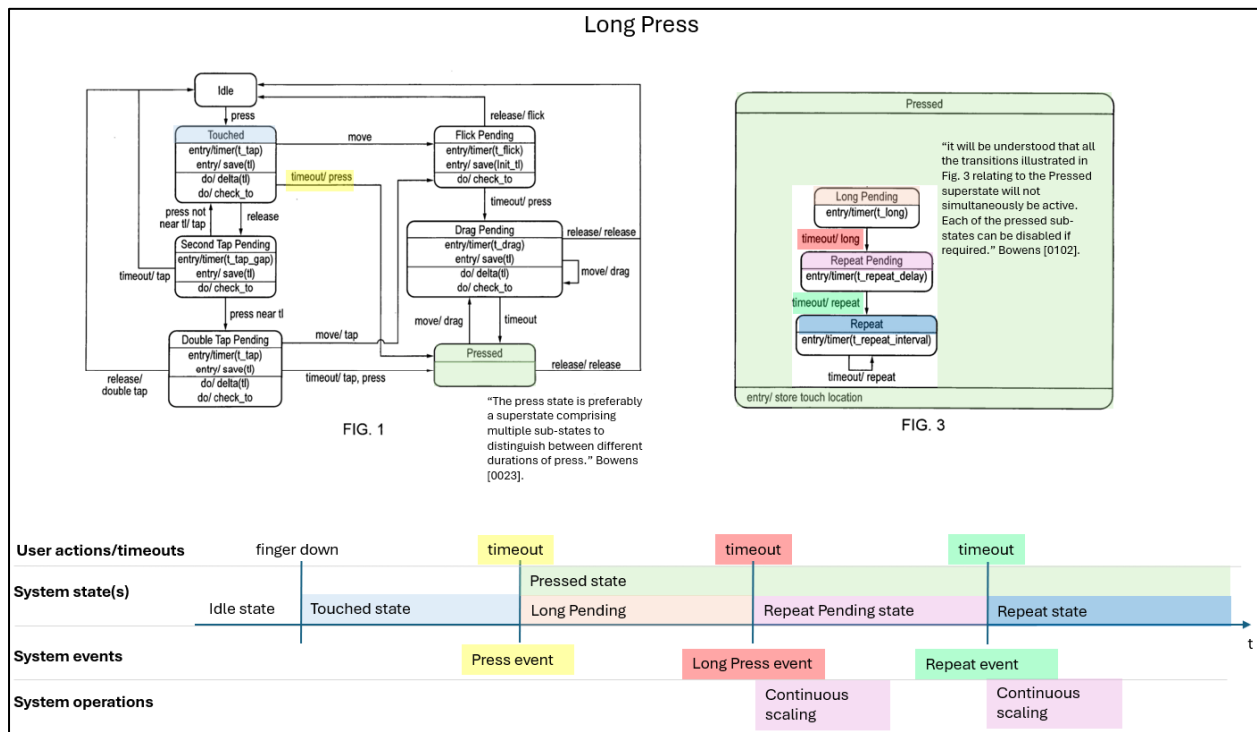


APPLE-1003, ¶177; APPLE-1005, FIG. 1 (highlighted)

If the touch persists for a longer period of time (from the Touched or Double Tap Pending states), “then control flow passes to a press state module, wherein the

press state is for handling longer touches.” APPLE-1003, ¶178; APPLE-1005, [0023], [0080] (“If the timeout expires, the state machine undergoes a transition by generating a press event and then enters the Pressed state.”), FIGS. 2-3. When a “user has pressed the” touch screen of the device “and been stationary for some time,” the device is in the “Long Pending state,” which starts a timer (t_long) on entry. APPLE-1003, ¶178; APPLE-1005, [0098], FIG. 3. “If the timeout expires, a long press event is generated and the state machine enters the Repeat Pending or Press Complete states,” as Bowens describes, which a POSITA would have understood as corresponding to Kim’s “tap and hold” or “long press,” which causes the KBD to *invoke a continuous zoom-in or zoom-out function*. APPLE-1003, ¶178; APPLE-1004, [0104]-[0107], [0139]-[0140], FIG. 7; APPLE-1005, [0098].

An illustration of a “long press event” in the KBD is provided below using Bowens’s Figure 1 and Figure 3 flow charts:



APPLE-1003, ¶179; APPLE-1005, FIGS. 1 and 3, [0023], [0102] (highlighted)

A POSITA would have understood that implementing a device (per Kim) with tap-related operation (activate a hyperlink on a displayed web page, per Bowens) would offer distinct functions associated with a tap, double tap, and a long press, respectively. APPLE-1003, ¶180; APPLE-1005, [0056]. Indeed, Kim offers an exemplary scenario where “a zooming action associated with web display data in a web browser” is provided. APPLE-1004, [0113]. It would have been obvious to a POSITA that the device would respond with different operations depending on the type of input received. APPLE-1003, ¶180.

A POSITA would have further understood that implementing the device (per Kim) with a “gesture recognition algorithm for interpreting streams of coordinates

received from a touch sensor” (per Bowens) would provide a clear method for distinguishing between the different types of touch gestures Kim describes.

APPLE-1003, ¶181; APPLE-1005, Abstract. Specifically, a POSITA would have understood that implementing the device with the various “states” (e.g., Touched state, Second Tap Pending state, Idle state, Pressed state), and setting timers when entering each state, would provide the implementation details for Kim’s teachings of distinguishing a tap from a double tap or long press. APPLE-1003, ¶181; APPLE-1004, [0139], APPLE-1005, [0080]. A POSITA would have understood that applying Bowens’s teachings and using timers/timeouts (and threshold durations) for distinguishing between taps, double taps, and long presses in the device per Kim would have led to predictable results without significantly altering or hindering the functions performed by device, particularly because Kim already contemplates receiving different types of inputs (tap, long press, double tap).

APPLE-1003, 181; APPLE-1004, [0042], [0053], [0103]-[0104], [0106]. Kim further contemplates determining the particular type of input a gesture is. APPLE-1004, [0042] (“the mobile device can determine whether an interrupt is a long press input or a double tap input...”), [0049], [0053], [0072]. A POSITA would have been motivated to implement the device per Kim with Bowens’s teachings in the KBD to achieve the predictable results offered by Bowens’s description of algorithms to detect gestures, related gesture events, and how such gestures and

events are determined. APPLE-1003, ¶181; APPLE-1005, [0055]-[0077].

Because recognizing predetermined touch gestures was known, and Kim and Bowens both describe the use of touchscreen devices and predetermined criteria for registering touches on the devices, a POSITA would have reasonably understood the combination of Kim and Bowens to be successful with no unexpected results. APPLE-1003, ¶181; APPLE-1004, [0003], [0011], [0030], [0039], [0041], [0094]-[0107], [0122]-[0137], FIGS. 5-9; APPLE-1005, Abstract, [0018], [0039], [0055]-[0078], [0080]-[0101], [0141].

Combining Kim and Bowens would have been obvious at least because the combination would have merely involved (1) combining prior art elements according to known methods to yield predictable results and (2) the use of known technique to improve similar devices (methods, or products) in the same way. *See KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 415-421 (2007); MPEP §2143.

Furthermore, the elements of the resulting combination would each perform functions they had previously been known to perform—the device would perform the same functions to zoom-out responsive to a long press, and receive different types of input gestures, per Kim and Bowens, and would leverage Bowens' teachings regarding the implementation details of how to distinguish between the different gestures to ultimately perform the respective operation. APPLE-1003, ¶182. Accordingly, a POSITA would have naturally expected success when

incorporating Bowens's teachings into Kim. *Id.*

3. Claim 1

[1.pre]-[1.d]

See [1.pre]-[1.d] (Ground 1A).

[1.e]

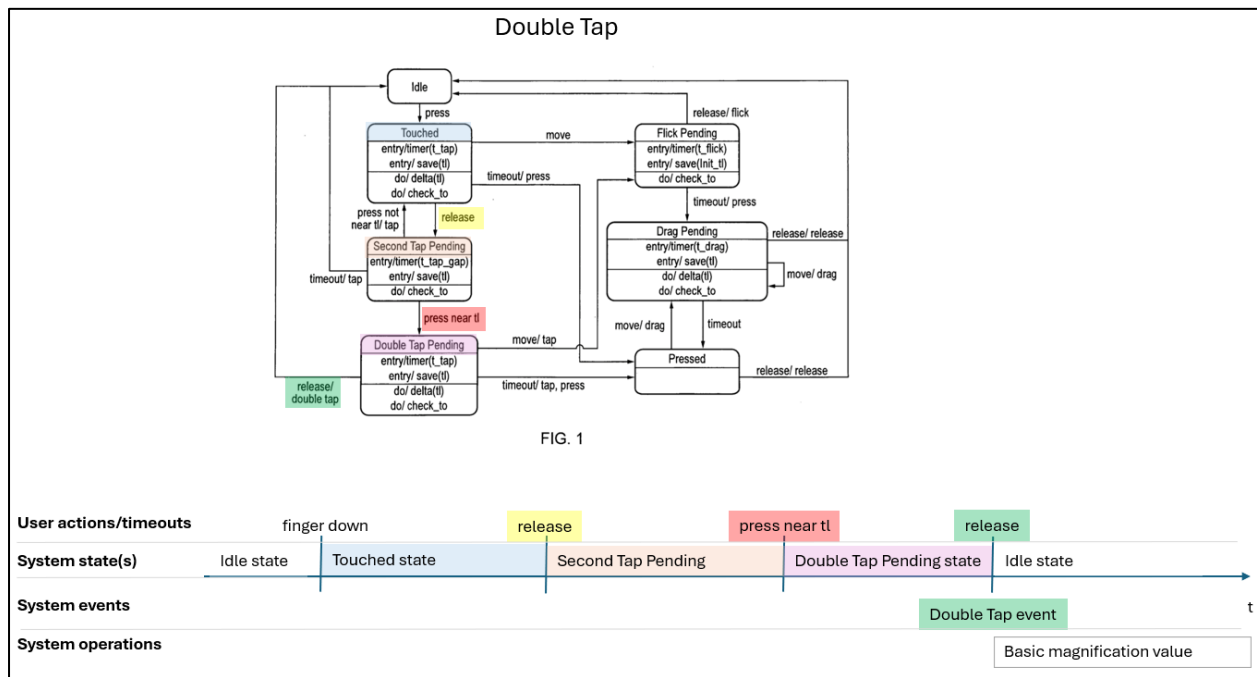
See [1.e] (Ground 1A). APPLE-1003, ¶188.

Additionally, like Kim, Bowens teaches a device that recognizes and generates events for different gesture inputs, such as a tap, a press, and a long press. APPLE-1005, [0055]-[0077]; APPLE-1003, ¶189.

[1.f]

The KBD renders [1.f] obvious. APPLE-1003, ¶190. As discussed above, the KBD includes various system states and corresponding operations. §IV.E.2; APPLE-1003, ¶190. In one example, the KBD can generate a double tap event. APPLE-1003, ¶190; APPLE-1005, [0081]-[0082]. As illustrated below, a double tap event is detected based on a duration of the touch, which is less than a first magnitude threshold (e.g., the timeout for the transition from Touched to Pressed). *Id.* In particular, a POSITA would have understood that the duration of each of the taps of the double tap gesture must be less than a timeout because, if either timeout expires, the gesture cannot be a double tap. APPLE-1003, ¶190. For the first tap, the release must occur before the “timeout/ press” transition from the Touched

state to the Pressed state. *Id.*; APPLE-1005, FIG.1. For the second tap, the release must occur before the “timeout/ tap, press” transition from the Double Tap Pending state to the Pressed state. *Id.* The duration of each of the taps of the double tap gesture is “a duration of the gesture.” A POSITA would have understood that gesture recognition encompasses identification of multiple components of touch data. APPLE-1003, ¶190. Consistent with this understanding, the specification of the ’114 patent states that “gesture input refers to the collection of inputs, flags, and signals by which a gesture is identified and parameterized,” that “[p]ossible gesture inputs may include, but are not limited to, touch event attributes (both initial and over time),” and that “[s]ome gestures may be able to be identified solely from initial touch event attributes. Other gestures may only be identified after the gesture has been performed for some period of time.” APPLE-1001, 58:13-15, 58:19-20, 58:27-30. In the KBD, when a double tap event is generated, the “device displays target data with a given basic magnification value,” thus performing another operation. APPLE-1003, ¶190; APPLE-1004, [0106]; APPLE-1005, [0082].

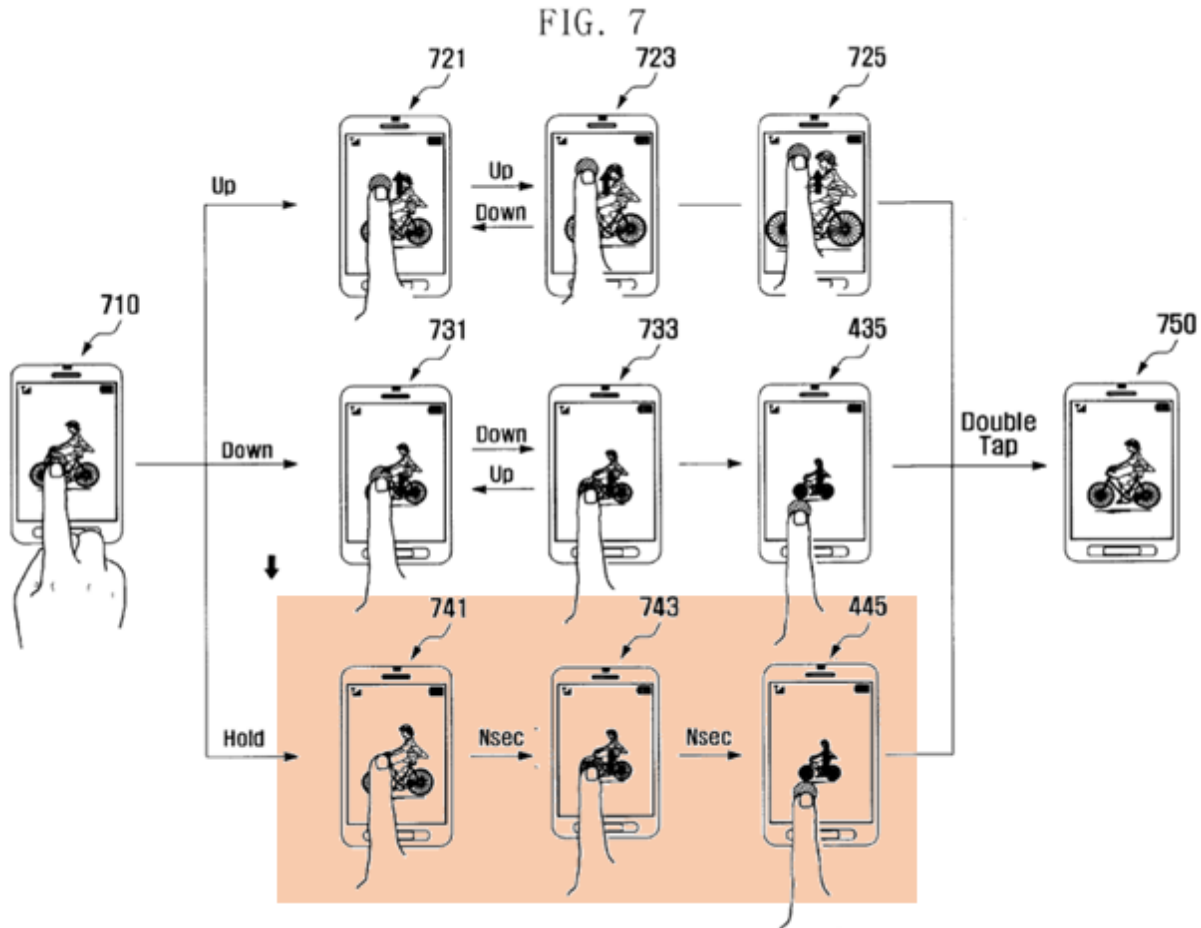


APPLE-1003, ¶190; APPLE-1005, FIG. 1 (highlighted)

[1.g]

The KBD renders [1.g] obvious. APPLE-1003, ¶191. As discussed above, if contact with the touch screen persists beyond the Touched state to a Pressed state and then to a Long Pending state, a timer (t_{long}) starts. §IV.E.2; APPLE-1003, ¶191; APPLE-1005, [0098], FIG. 3. “If the timeout expires, a long press event is generated and the state machine enters the Repeat Pending or Press Complete states,” as Bowens describes, which a POSITA would have understood as corresponding to Kim’s “tap and hold” or “long press,” which causes the KBD to *invoke a continuous zoom-in or zoom-out function*. APPLE-1003, ¶191; APPLE-1004, [0104]-[0107], [0139]-[0140], FIG. 7; APPLE-1005, [0098]. Bowens

teaches that the “long press event” is generated when the duration of the touch is longer, for example, than a tap (or double tap) (e.g., greater than the first magnitude threshold). APPLE-1005, [0023] (“If a touch of a duration longer than a threshold duration is sensed in the touch state, then control flow passes to a press state module, wherein the press state is for handling longer touches.”), FIG. 3. While Figure 7 shows the object (person on bicycle) at reference numbers 741, 743, and 745 increasing in size (e.g., zoom in), Kim explains that “the hold event may be alternatively assigned to a *continuous zoom-out function*” such that the tap and hold input causes the object (person on bicycle) to decrease in size (e.g., zoom out). *Id.*, [0107] (emphasis added), FIG. 7. Modified Figure 7 (below) illustrates Kim’s alternative assignment. A POSITA would have understood that this same functionality can also be applied to other content, including, for example, the content discussed in Figure 8 that includes a displayed hyperlink. APPLE-1003, ¶191; APPLE-1004, [0114]-[0115], [0145], FIG. 8.



APPLE-1004, FIG. 7 (“hold” scenario modified)

Thus, when the magnitude of the duration of the gesture on the touch screen on the object is greater than the first magnitude threshold (e.g., the timeout for the transition from Touched to Pressed), the KBD performs a scaling of the object (e.g., hyperlink, image, etc.). APPLE-1003, ¶192.

[1.h]

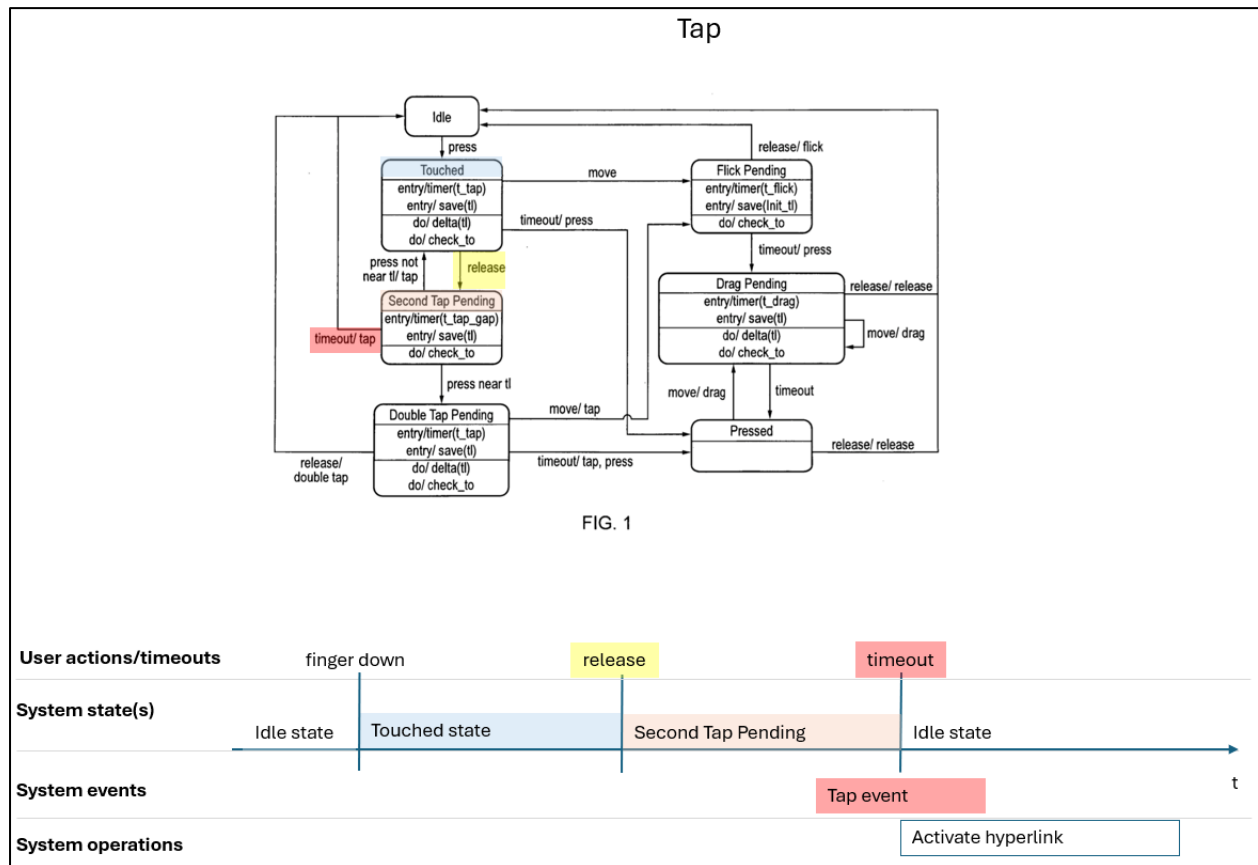
The KBD renders [1.h] obvious. APPLE-1003, ¶193. A POSITA would have understood that a “continuous zoom-out function” on an object, such as the

bike and bicyclist of Kim's Figure 7 or the hyperlink of Figure 8, is the equivalent of scale operation that includes a reduction in size of the object. *Id.* Additionally, Kim teaches that the zoom-out is "continuous": in a zoom-out process, "the mobile device continuously offers a real-time, scaled-[down] bitmap image on the screen in response to the duration of the hold event." *Id.*, ¶194; APPLE-1004, [0105]. Thus, the longer the duration of the gesture (press and hold), the more the object is zoomed-out, or the more the size of the object is reduced. *Id.* In this way, the reduction in a size is inversely related to the magnitude of the duration of the gesture (the larger the duration of the gesture, the smaller the size). *Id.*

[1.i]

The KBD renders [1.i] obvious. APPLE-1003, ¶195. As discussed for [1.e], the detected gesture may be received on the touchscreen on an object (e.g., a hyperlink). §IV.E.2. As discussed above, "after a user has tapped the touch surface" and "the user releases their touch" (highlighted yellow below), a timeout starts in the KBD device, which "is used to decide if the user has made a tap gesture." §IV.E.2; APPLE-1005, [0065], [0080]-[0081]. If the timeout expires, "a tap event is generated." *Id.* If that timeout expires, "a tap event is generated" and the gesture is then identified as a tap, which, as Bowens teaches, "activates a hyperlink on a displayed web page." APPLE-1003, ¶195; APPLE-1005, [0056], [0081]. A POSITA would have understood that if the gesture is determined to be a

tap gesture (which ceased at the time labeled “release,” highlighted yellow below), then the magnitude of the duration of the gesture on the touch screen is greater than a second magnitude (the “timeout/tap” transition from SecondTapPending to Idle). APPLE-1003, ¶195. In this scenario, the KBD performs another operation (activating the hyperlink). *Id.*; APPLE-1005, [0056], [0081].



APPLE-1003, ¶195; APPLE-1005, FIG. 1 (highlighted)

4. Claim 27

[27.pre]-[27.e]

See [1.pre]-[1.e] (Ground 1A).

[27.f]

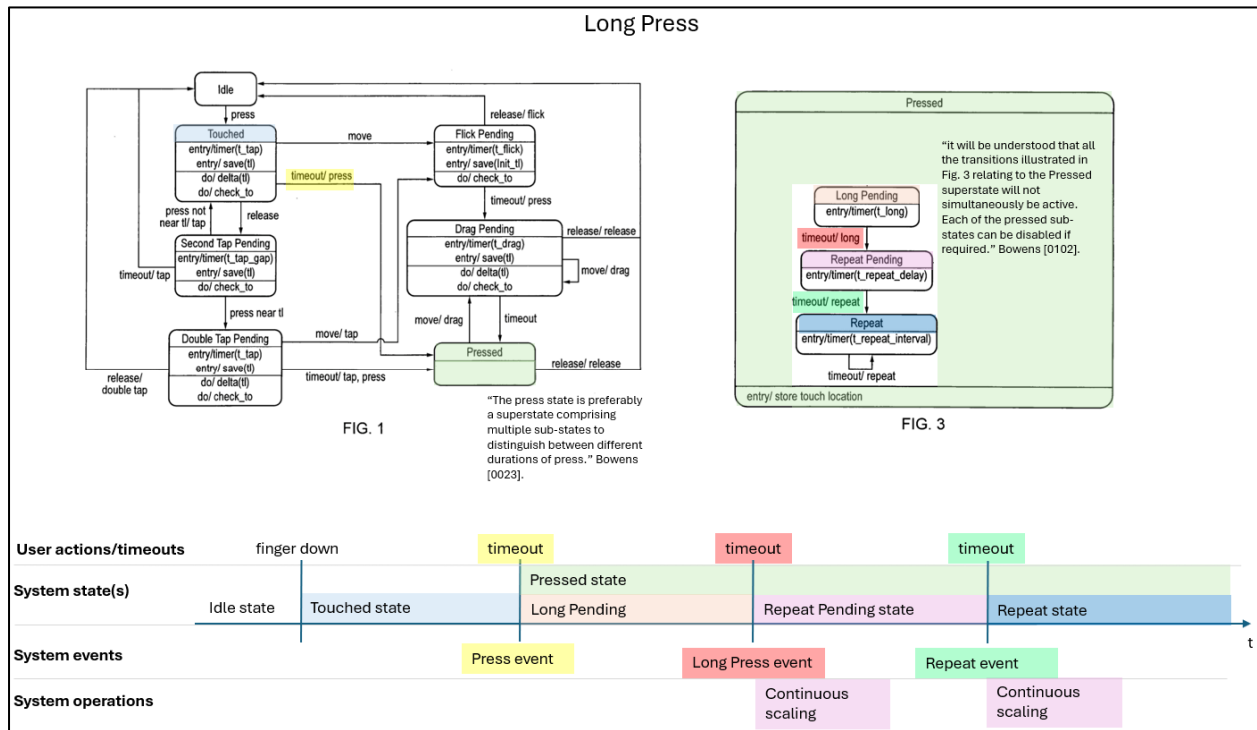
The KBD renders [27.f] obvious. APPLE-1003, ¶202. As discussed above, the KBD is in a first touch state (“Second Tap Pending”) “after a user has tapped the touch surface,” which causes the device to generate a “tap event.” §IV.E.2; APPLE-1005, [0065], [0081]. In the KBD, when a tap event occurs in a “a web browser environment,” a “hyperlink on a displayed web page” is activated. APPLE-1003, ¶202; APPLE-1004, [0113], FIG. 8; APPLE-1005, [0056]. Thus, in the Second Tap Pending state, the function of activating a hyperlink is performed. *Id.*

[27.g]

The KBD renders [27.g] obvious. APPLE-1003, ¶203. As discussed above, in the KBD, a “Double Tap Pending state occurs when a user has tapped the touch surface and pressed it again.” §IV.E.2; APPLE-1005, [0082]. When the KBD is in a “Double Tap Pending state” and “the user releases their touch [from the second touch], the state machine generates a double tap event and enters the Idle state.” *Id.* In this second touch state (“Idle state”) following the double tap detection, the KBD performs a second function (e.g., “displays target data with a given basic magnification value”). APPLE-1003, ¶203; APPLE-1004, [0106], FIG. 7; APPLE- 1005, [0057], [0082].

Alternatively, when the KBD is in a “Repeat Pending state” (second touch

state) following a “Long Press event,” the device performs continuous scaling (second function), as discussed in more detail with respect to [27.h]. *See* §IV.E.2, [27.h]; APPLE-1003, ¶204.

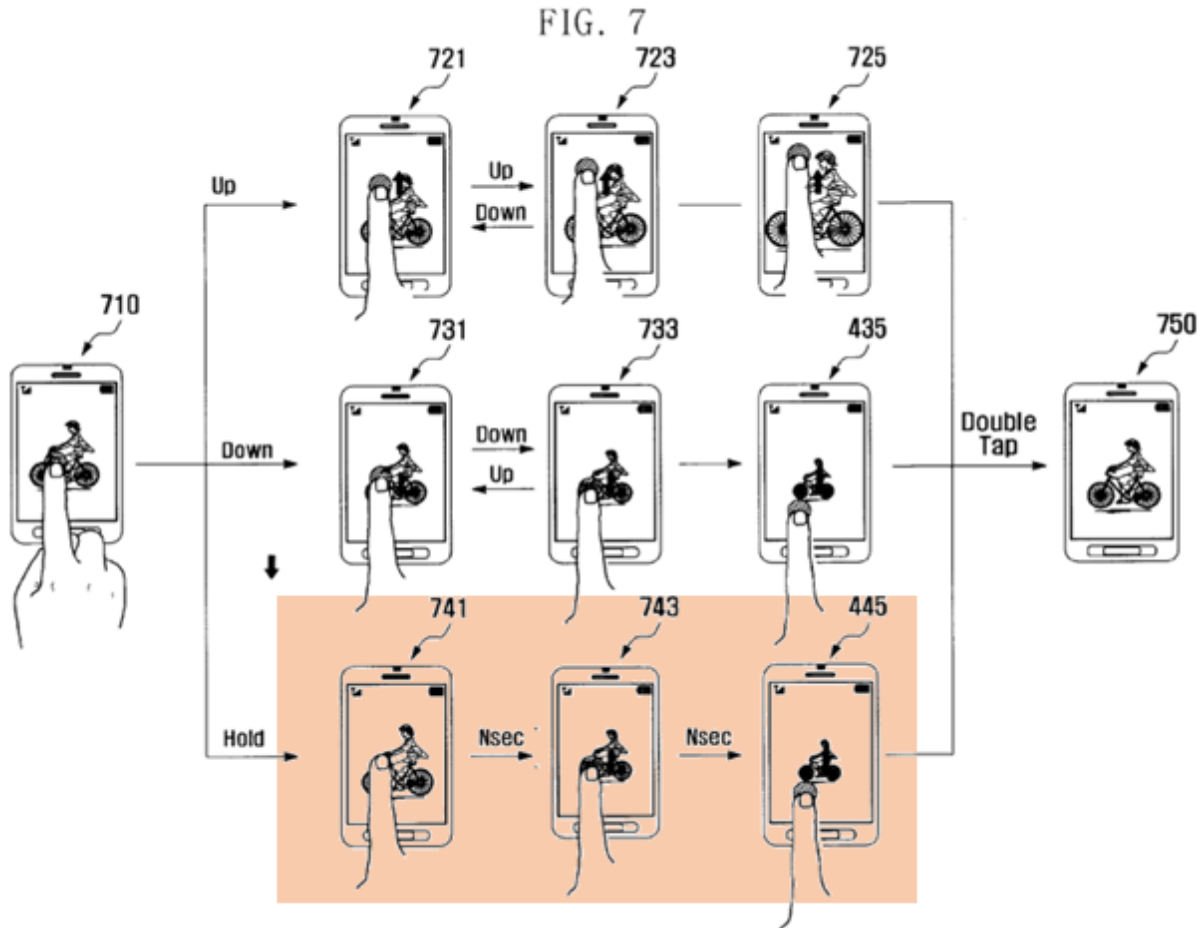


APPLE-1003, ¶204; APPLE-1005, FIGS. 1 and 3; [0023], [0102] (highlighted)

[27.h]

The KBD renders [27.h] obvious. APPLE-1003, ¶205. As discussed above, if contact with the touch screen persists beyond the Touched state to a Pressed state and then to a Long Pending state, a timer (t_{long}) starts. §IV.E.2; APPLE-1003, ¶205; APPLE-1005, [0098], FIG. 3. “If the timeout expires, a long press event is generated and the state machine enters the Repeat Pending or Press Complete states,” as Bowens describes, which a POSITA would have understood as

corresponding to Kim's "tap and hold" or "long press," which causes the KBD to *invoke a continuous zoom-in or zoom-out function*. APPLE-1003, ¶205; APPLE-1004, [0104]-[0107], [0139]-[0140], FIG. 7; APPLE-1005, [0098]. While Figure 7 shows the object (person on bicycle) at reference numbers 741, 743, and 745 increasing in size (e.g., zoom in), Kim explains that "the hold event may be alternatively assigned to a *continuous zoom-out function*" such that the tap and hold input causes the object (person on bicycle) to decrease in size (e.g., zoom out). *Id.*, [0107] (emphasis added), FIG. 7. Modified Figure 7 (below) illustrates Kim's alternative assignment.



APPLE-1004, FIG. 7 (“hold” scenario modified); APPLE-1003, ¶205

A POSITA would have understood that a “continuous zoom-out function” on an object, such as the bike and bicyclist of Kim’s Figure 7, responsive to a persistent gesture (i.e., one with a constantly changing duration), is the equivalent of scale operation that includes a reduction in size. APPLE-1003, ¶206.

Additionally, Kim teaches that the zoom-out is “continuous”: in a zoom-out process, “the mobile device continuously offers a real-time, scaled-[down] bitmap image on the screen in response to the duration of the hold event.” *Id.*, ¶207;

APPLE-1004, [0105]. Thus, the longer the duration of the gesture (press and hold), the greater the zoom-out, or the more the size is reduced. *Id.* Accordingly, to the extent this claim is interpreted to require a reduction in size, wherein the size is inversely related to the duration, Kim discloses the limitation. *Id.*

In the alternative, to the extent the claim is interpreted to require that the *reduction* in size and the *duration* are inversely related (as opposed to the *size* and the *duration* being inversely related), then such an element would have at least been obvious. APPLE-1003, ¶208. For example, Kim teaches that a double tap event (i.e., short duration) causes “enlarged display target data” to be “given basic magnification value” (i.e., reduction in size), which a hold event (i.e., long duration) causes a zoom-out function (i.e., reduction in size). APPLE-1004, [0105]-[0107]. That is, in the case of a double tap, if the “enlarged display target data” was very large, the double tap would cause a large reduction in size, whereas, a long press that was relatively short would cause a smaller reduction in size. *Id.* Thus, the scale operation is responsive to a change in a duration of the gesture (e.g., from a short duration to a long one). *Id.* Thus, the gesture with the longer duration of the gesture (long press) would cause a comparatively smaller reduction in size. *Id.*; APPLE-1003, ¶208. In this way, the *reduction* in a size is inversely related to the *duration* of the gesture (i.e., a gesture with a larger duration causes a smaller reduction in size than a gesture with a shorter duration). *Id.*

In another alternative, if the phrase “where the scale operation includes a reduction in a size that is inversely related to the duration” is interpreted to mean that either the *reduction in size* or the *rate of reduction in size* are inversely related *duration*, then such an element would have at least been obvious. APPLE-1003, ¶209. In particular, a POSITA would have understood that as an image is reduced in size (as zoomed out over time), fewer pixels are changed per step in the zoom-out animation (e.g., a step change from 100% to 99% zoom would change a larger number of pixels than a step change from 2% to 1%). *Id.* In this way, the *reduction in a size* is inversely related to the *duration* of the gesture. *Id.*

Finally, while the example in Figure 7 shows an object (rider on a bicycle) with a surrounding interface that zooms in or zooms out in response to the press and hold, Figure 8 shows yet another example of the zoom-in or zoom-out effect. APPLE-1004, [0111]-[0115], FIG. 8; APPLE-1003, ¶210. In particular, Figure 8 shows an example where, in response to a user gesture, both an object (e.g., a car) and its surrounding interface (e.g., the remainder of the web page surrounding the object), a zooming action is performed on both the object and its surrounding interface. *Id.* Thus, Kim’s scale operation is performed in connection with the at least portion of the interface. *Id.*; APPLE-1003, ¶210.

5. Claim 32

[32.pre]-[32.c]

See [32.pre]-[32.c] (Ground 1A).

[32.d]

The KBD renders [32.d] obvious. APPLE-1003, ¶215.

As discussed for [27.d], the KBD displays a first object. APPLE-1004, FIG. 7; APPLE-1003, ¶215. To the extent the device shown in Kim's Figure 7 is interpreted as a single object, a POSITA would have found the display of at least one other object obvious and understood Figure 7 to simply be an example. APPLE-1003, ¶215. Figure 8 of Kim, for example, shows more than one object (e.g., a car, text, a person). APPLE-1004, FIG. 8. Thus, a POSITA would have found the display of any number of objects to be obvious. APPLE-1003, ¶215.

[32.e]

The KBD renders [32.e] obvious. APPLE-1003, ¶216.

See [32.e] (Ground 1A). A POSITA would have understood or found obvious that the gesture detected on the KBD's touch screen includes a plurality of time-related components, including, for example, how long the gesture has been maintained at any point in time during or at cessation of the gesture. *Id.* The significance of these time-related components is clear in both Kim's and Bowens's teachings, which each define various types of gestures based on the length of the gesture. *See* APPLE-1004, [0034] ("*long press*"), [0061] ("duration of the hold

event”), [0105] (“*duration* of the hold event”), [0139] (“*long* press”); APPLE-1005, [0023] (“touch of a *duration longer* than a threshold *duration*”) (“a very *long* press”), [0029] (“if a touch of a *duration longer* than a threshold *duration* is sensed”) (“distinguish between different *durations* of press”), [0056]-[0059] (“*short* touch *duration*”), [0080] (“*short duration* gestures such as flicks and taps, and *long duration* gestures such as a press”), [0096] (“multiple interpretations of a single touch depending on its *duration*”).

[32.f]

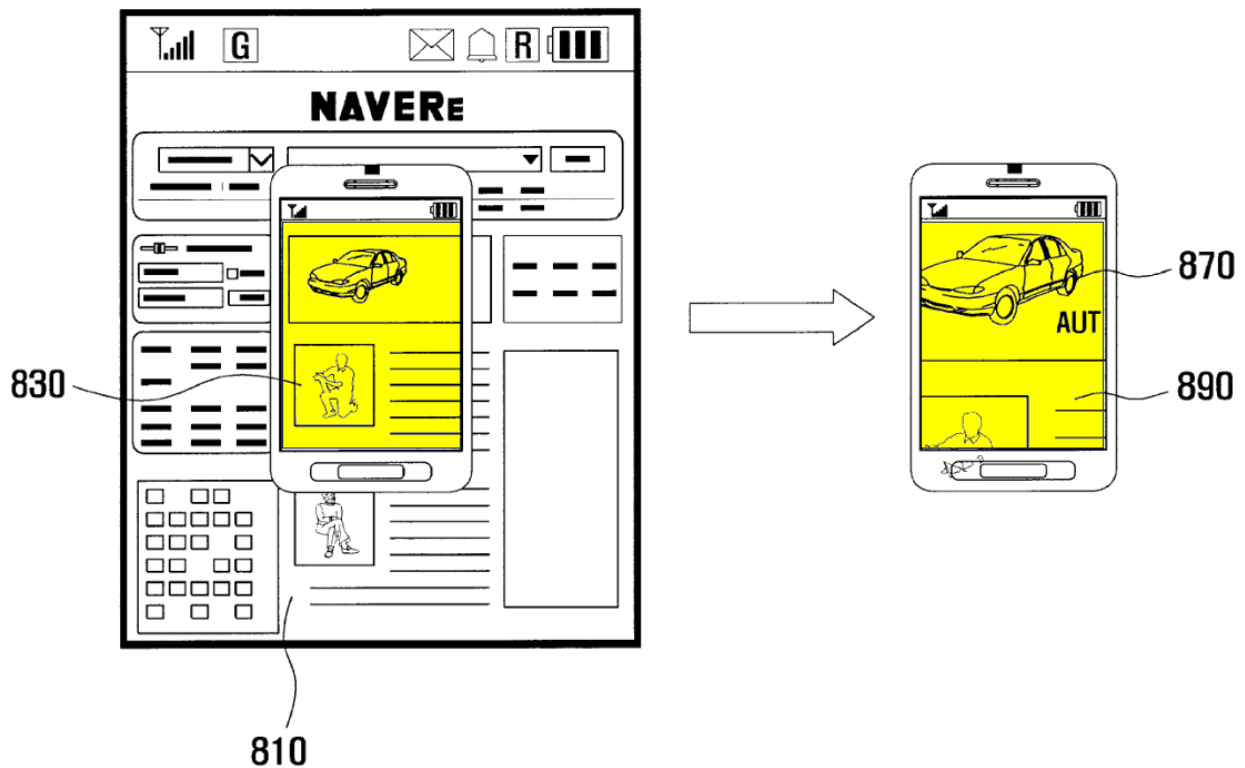
The KBD renders [32.f] obvious for at least the same reasons as discussed for [1.f] (Ground 2A). APPLE-1003, ¶317. Specifically, when a first component of the gesture (time-related component when a double tap event is generated) is detected where the duration detected on the touch screen is less than a first magnitude (e.g., the timeout for the transition from Touched to Pressed), the KBD performs a function (displays target data with a given basic magnification value). APPLE-1003, ¶317; *see* [1.f] (Ground 2A). Moreover, a POSITA would have understood that, where the function occurs after a timeout, shortly after cessation of the gesture (when the user releases his finger), the function is performed upon cessation of the gesture being detected. APPLE-1003, ¶317; APPLE-1004 (“Thereafter, a user can invoke a double tap event on the screen. Then, *in response to a double tap event*, the mobile device displays target data with a given basic

magnification value...”).

[32.g]

The KBD renders [32.g] obvious for at least the same reasons as discussed for [1.g] (Ground 2A). APPLE-1003, ¶218. In particular, when a second component of the gesture (time-related component when a long press event is generated) is detected where the duration detected on the touch screen is greater than the first magnitude (the timeout for the transition from Touched to Pressed), the KBD performs a function (a zoom or scale operation). *Id.*; *see* [1.g] (Ground 2A). Moreover, a POSITA would have understood that the scale operation is performed on content displayed on the screen, including both the first object (hyperlink) and the at least one other object (e.g., car, person). APPLE-1003, ¶218. Figure 8 of Kim (below) shows an example showing how all of the displayed content (highlighted) is scaled in response to the gesture. *Id.*; APPLE-1004, FIG. 8.

FIG. 8



APPLE-1004, FIG. 8 (highlighted)

[32.h]

See [1.h] (Ground 2A).

[32.i]

The KBD renders [32.i] obvious for at least the same reasons as discussed for [1.i] (Ground 2A). APPLE-1003, ¶220. In particular, when a third component of the gesture (e.g., time-related component when a tap event is generated) is detected where the duration detected on the touch screen is greater than a second magnitude (the “timeout/tap” transition from SecondTapPending to Idle), the KBD

performs another function (activating the hyperlink). APPLE-1003, ¶220; *see* [1.i] (Ground 2A).

F. [GROUND 2B] – The Kim-Bowens-Lang Combination renders obvious claims 34, 59, and 64

1. The Kim-Bowens-Lang Combination

The Kim-Bowens Device is discussed above in §IV.E.2. APPLE-1003, ¶221. For at least the same reasons discussed in §IV.B.2, it would have further been obvious to a POSITA to implement the KBD in a smart watch, per Lang, forming the Kim-Bowens-Lang Device (“KBLD”). APPLE-1003, ¶221.

2. Claim 34

[34.pre]

To the extent the preamble is limiting, the KBLD renders [34.pre] obvious. APPLE-1003, ¶222. Specifically, as discussed with respect to claim 1, the KBD (apparatus) performs a method. §IV.E.2. For at least the same reasons, the KBLD performs a method, as discussed in more detail with respect to [34.e]-[34.j]. APPLE-1003, ¶222; *see* [1.pre] (Ground 2A).

[34.a]

The KBLD renders [34.a] obvious. APPLE-1003, ¶223. Kim explains that its mobile device can be “any other kinds of portable and handheld devices.” APPLE-1004, [0118]. As discussed in §IV.B.2, a POSITA would have found it obvious to implement the KBLD as a smart watch, per Lang. APPLE-1003, ¶223;

APPLE-1005, [0015].

[34.b]-[34.j]

See [1.a]-[1.i] (Ground 2A).

3. Claim 59

[59.pre]-[59.d]

See [34.pre]-[34.d] (Ground 2B).

[59.e]-[59.i]

See [27.d]-[27.h] (Ground 2A).

4. Claim 64

[64.pre]-[64.d]

See [34.pre]-[34.d] (Ground 2B).

[64.e]-[64.j]

See [32.d]-[32.i] (Ground 2A).

G. [GROUND 2C] – The Kim-Bowens-Burrough Combination renders obvious claim 30

1. The Kim-Bowens-Burrough Combination

The Kim-Bowens Device is discussed above in §IV.E.2. For at least the same reasons discussed in §IV.C.2, it would have further been obvious to a POSITA to modify the KBD to include a haptic device and use of haptic feedback, as Burrough teaches and suggests, forming the Kim-Bowens-Burrough Device (“KBBD”). APPLE-1003, ¶¶254-257.

2. Claim 30

[30.pre]-[30.b]

See [1.pre]-[1.b] (Ground 1A).

[30.c]

The KBBD renders [30.c] obvious. APPLE-1003, ¶261. As discussed in §IV.G.1 and §IV.C.2, the KBBD further includes a haptic device (e.g., piezoelectric actuator assembly), which provides a “vibro-tactile response,” and thus is a vibrator. APPLE-1007, [0013], [0033], [0037], [0047], [0055]-[0056], FIGS. 1A-1E; APPLE-1003, ¶261; §§IV.C.2, IV.G.1.

[30.d]

The KBBD renders [30.d] obvious with respect to the one or more processors in communication with memory and the touch screen for at least the same reasons as discussed for [1.c]. APPLE-1003, ¶262. As discussed in §IV.G.1 and §IV.C.2, the KBBD further includes a vibrator in communication with the control unit of Kim. APPLE-1003, ¶262; §§IV.C.2, IV.G.1.

As discussed in §IV.G.1 and §IV.C.2, the KBBD further includes a vibrator in communication with the one or more processors. APPLE-1003, ¶263; §§IV.C.2 and IV.G.1.

[30.e]

See [32.d] (Ground 1A).

[30.f]

See [1.e] (Ground 2A).

[30.g]

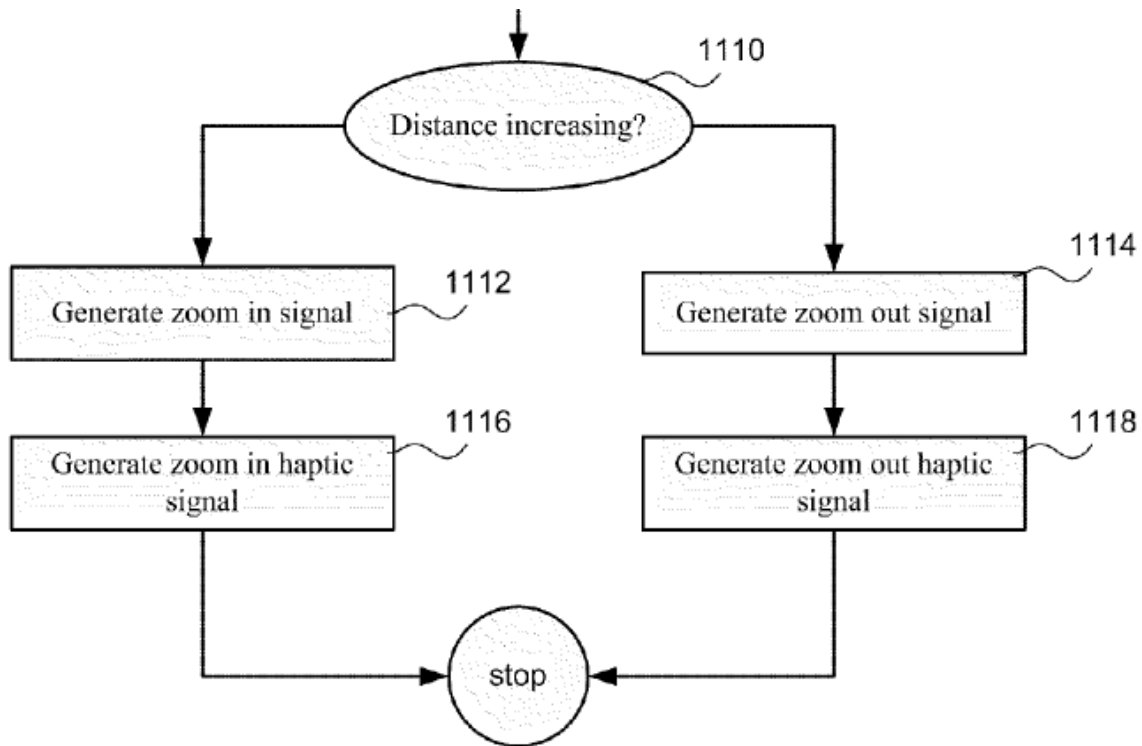
See [30.g] (Ground 1C).

[30.h]

See [32.h] (Ground 2A).

[30.i]

The KBBD renders [30.i] obvious. APPLE-1003, ¶268. As discussed in §IV.G.1 and §IV.C.2, the KBBD includes a haptic device (e.g., piezoelectric actuator assembly) that provides a “vibro-tactile response.” APPLE-1007, [0013], [0033], [0037], [0047], [0055]-[0056], FIGS. 1A-1E, FIG. 11; APPLE-1003, ¶268; §§IV.C.2, IV.G.1. As shown in Figure 11 (excerpted below), Burrough teaches “generate a zoom out haptic signal” or “generate a zoom in haptic signal” after generating a zoom out or zoom in signal. APPLE-1004, [0080], FIG. 11.



APPLE-1007, FIG. 11 (excerpted)

Additionally, as explained in §IV.E.2, in the KBD (which Burrough modifies), as the gesture persists and the timeout expires, “the state machine enters the Repeat Pending or Press Complete states” (perform an action), as Bowens describes. APPLE-1003, ¶269; APPLE-1005, [0100].

H. [GROUND 2D] – The Kim-Bowens-Lang-Burrough Combination renders obvious claim 62

1. The Kim-Bowens-Lang-Burrough Combination

The Kim-Bowens-Lang Device (KBLD) is discussed above in §IV.F.1. For at least the same reasons discussed in §IV.C.2, it would have further been obvious to a POSITA to modify the KBLD to include a haptic device and use of haptic

feedback, as Burrough teaches and suggests. APPLE-1003, ¶270.

2. Claim 62

[62.pre]-[62.n]

See [34.pre]-[34.b] (Ground 2B).

[62.c]-[62.j]

See [30.b]-[30.i] (Ground 2C).

V. ANALYSIS ON DISCRETION

A. 35 U.S.C. §325(d)—*Advanced Bionics*

The *Advanced Bionics* framework strongly favors institution. None of the prior art relied upon for Grounds 1A-1D and 2A-2D were cited among the thousands of references disclosed by Applicant and/or cited by the Examiner. Nor did either the Applicant or Examiner explicitly discuss or consider the combinations presented herein. Thus, neither condition of Part One in the *Advanced Bionics* two-part framework is met.

Further, while there is no need to reach Part Two of the *Advanced Bionics* framework, the record shows that the Challenged Claims were allowed based on a material error by the Examiner at least as illustrated by the Grounds presented herein. Had the Examiner properly assessed the references relied upon for Grounds 1A-1D and 2A-2D and appreciated their teachings in view of the prior art as a whole, the Challenged Claims would not have issued. *See* Section IV, *supra*.

B. 35 U.S.C. §314(a) - *Fintiv*

Institution should not be discretionarily denied under §314(a) because no *Fintiv* factors favor denial. Rather, five of the six factors enumerated in *Apple Inc. v. Fintiv, Inc.* weigh against discretionary denial, and the other factor is neutral on the issue. IPR2020-00019, Paper 11 at 2-3 (PTAB “precedential” Mar. 20, 2020) (“*Fintiv*”).

Factor 1 weighs against a discretionary denial. Apple intends to file a motion to stay the parallel litigation. Thus, factor 1 does not favor discretionary denial and, at worst, is neutral.

Factor 2 weighs against a discretionary denial. Director Vidal’s Guidance instructed the Board that “when considering the proximity of the district court’s trial date to the date when the PTAB final written decision will be due, the PTAB will consider the median time from filing to disposition of the civil trial for the district in which the parallel litigation resides.” APPLE-1020, 3, 8. According to the latest published district court statistics, the median time to trial in the Southern District of California (SDCal) is 40.2 months, predicting a trial in November 2027—significantly later than a Final Written Decision would be issued in this case. APPLE-1019, 69; APPLE-1020, 8 (“A court’s scheduled trial date, therefore, is not by itself a good indicator of whether the district court trial will occur before the statutory deadline for a final written decision.”). For this reason

alone, Factor 2 weighs against discretionary denial. Though a pre-trial conference of October 23, 2025 has been scheduled in co-pending litigation, no trial date has been set, and it is uncertain whether the schedule will stick, particularly if Apple's motion to stay is granted. APPLE-1021, 6.

Factor 3 also weighs against discretionary denial because Petitioner has diligently filed this Petition while the co-pending litigation is still in its early stages. Petitioner filed as soon as practicable following Smith's election of 200 asserted claims of the over 1000 claims in the asserted patents just weeks before filing. Beyond the service of preliminary infringement contentions, the parties and the District Court have yet to expend significant resources. No invalidity contentions have been served, and discovery is in its infancy.

Factor 4 weighs against discretionary denial. As noted above, no trial date has been scheduled. And even if a pre-trial conference date of October 2025 is used as a reference point, if Apple's to-be-filed motion to stay is granted, the case schedule is going to change and the trial date will be further delayed. Accordingly, it is likely that a final written decision will be issued before trial. Indeed, no stipulation has been offered given the lack of a trial date, and it otherwise remains too early to assess the suitability of a stipulation with invalidity contentions still not exchanged.

Factor 5 The parties in the parallel co-pending litigation are the same.

Factor 6 favors institution because the merits of this Petition are compelling, as described in this Petition. Indeed, the grounds presented herein apply, in a straightforward manner, to demonstrate the obviousness of the claims. *See* §IV, *supra*.

VI. CONCLUSION AND FEES

The Challenged Claims are unpatentable. Please charge fees to Deposit Account 06-1050.

VII. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(a)(1)

A. Real Party-In-Interest Under 37 C.F.R. § 42.8(b)(1)

Apple Inc. is the petitioner and the real party-in-interest.

B. Related Matters Under 37 C.F.R. § 42.8(b)(2)

Petitioner is not aware of any disclaimers, reexamination certificates or petitions for inter partes review for the '114 Patent. The '114 Patent is the subject of civil action: *Smith Interface Technologies, LLC v. Apple Inc.*, Case No. 3-23-cv-01187 (SDCA), filed June 27, 2023.

C. Lead And Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3)

Petitioner provides the following designation of counsel.

Lead Counsel	Backup counsel
W. Karl Renner, Reg. No. 41,265 Fish & Richardson P.C. 60 South Sixth Street Minneapolis, MN 55402 Tel: 202-783-5070 Fax: 877-769-7945 Email: IPR50095-0171IP1@fr.com	Jennifer J. Huang, Reg. No. 64,297 Fish & Richardson P.C. 60 South Sixth Street Minneapolis, MN 55402 Tel: 202-783-5070 Fax: 877-769-7945 Email: IPR50095-0171IP1@fr.com

D. Service Information

Please address all correspondence and service to the address listed above.

Petitioner consents to electronic service by email at IPR50095-0171IP1@fr.com.

Respectfully submitted,

Dated: June 26, 2024

/W. Karl Renner/
W. Karl Renner, Reg. No. 41,265
Jennifer J. Huang, Reg. No. 64,297
Fish & Richardson P.C.
60 South Sixth Street
Minneapolis, MN 55402
T: 202-783-5070
F: 877-769-7945

(Control No. IPR2024-01091)

Attorneys for Petitioner

CERTIFICATION UNDER 37 CFR § 42.24

Under the provisions of 37 CFR § 42.24(d), the undersigned hereby certifies that the word count for the foregoing Petition for *Inter Partes* Review totals 13,998 words, which is less than the 14,000 allowed under 37 CFR § 42.24.

Dated: June 26, 2024

/W. Karl Renner/

W. Karl Renner, Reg. No. 41,265
Jennifer J. Huang, Reg. No. 64,297
Fish & Richardson P.C.
60 South Sixth Street
Minneapolis, MN 55402
T: 202-783-5070
F: 877-769-7945

(Control No. IPR2024-01091)

Attorneys for Petitioner

CERTIFICATE OF SERVICE

Pursuant to 37 CFR §§ 42.6(e)(4)(i) *et seq.* and 42.105(b), the undersigned certifies that on June 26, 2024, a complete and entire copy of this Petition for *Inter Partes* Review and all supporting exhibits were provided via USPS Priority Express Mail, to the Patent Owner by serving the correspondence address of record as follows:

THE CALDWELL FIRM, LLC
PO Box 59655
Dept. SVIPGP
Dallas, TX 75229
UNITED STATES
(214) 734-2313

/Crena Pacheco/

Crena Pacheco
Fish & Richardson P.C.
60 South Sixth Street
Minneapolis, MN 55402
pacheco@fr.com