

EXHIBIT A



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

90/012,332

05/30/2012

7844915

P4895USREX1/120730-003US

5963

20872 7590 07/26/2013

MORRISON & FOERSTER LLP
425 MARKET STREET
SAN FRANCISCO, CA 94105-2482

EXAMINER

YIGDALL, MICHAEL J

ART UNIT

PAPER NUMBER

3992

MAIL DATE

DELIVERY MODE

07/26/2013

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

DO NOT USE IN PALM PRINTER

(THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS)

BRYAN CAVE LLP
1290 AVENUE OF THE AMERICAS
NEW YORK, NY 10104

***EX PARTE* REEXAMINATION COMMUNICATION TRANSMITTAL FORM**

REEXAMINATION CONTROL NO. 90/012,332.

PATENT NO. 7,844,915.

ART UNIT 3992.

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above identified *ex parte* reexamination proceeding (37 CFR 1.550(f)).

Where this copy is supplied after the reply by requester, 37 CFR 1.535, or the time for filing a reply has passed, no submission on behalf of the *ex parte* reexamination requester will be acknowledged or considered (37 CFR 1.550(g)).

Office Action in Ex Parte Reexamination	Control No. 90/012,332	Patent Under Reexamination 7844915
	Examiner Michael J. Yigdall	Art Unit 3992

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

- a ☒ Responsive to the communication(s) filed on 19 March 2013. b ☒ This action is made FINAL.
c ☐ A statement under 37 CFR 1.530 has not been received from the patent owner.

A shortened statutory period for response to this action is set to expire 2 month(s) from the mailing date of this letter. Failure to respond within the period for response will result in termination of the proceeding and issuance of an *ex parte* reexamination certificate in accordance with this action. 37 CFR 1.550(d). **EXTENSIONS OF TIME ARE GOVERNED BY 37 CFR 1.550(c).** If the period for response specified above is less than thirty (30) days, a response within the statutory minimum of thirty (30) days will be considered timely.

Part I THE FOLLOWING ATTACHMENT(S) ARE PART OF THIS ACTION:

1. ☐ Notice of References Cited by Examiner, PTO-892. 3. ☐ Interview Summary, PTO-474.
2. ☐ Information Disclosure Statement, PTO/SB/08. 4. ☐ ____.

Part II SUMMARY OF ACTION

- 1a. ☒ Claims 1-21 are subject to reexamination.
1b. ☐ Claims ____ are not subject to reexamination.
2. ☐ Claims ____ have been canceled in the present reexamination proceeding.
3. ☐ Claims ____ are patentable and/or confirmed.
4. ☒ Claims 1-21 are rejected.
5. ☐ Claims ____ are objected to.
6. ☐ The drawings, filed on ____ are acceptable.
7. ☐ The proposed drawing correction, filed on ____ has been (7a) ☐ approved (7b) ☐ disapproved.
8. ☐ Acknowledgment is made of the priority claim under 35 U.S.C. § 119(a)-(d) or (f).
a) ☐ All b) ☐ Some* c) ☐ None of the certified copies have
1 ☐ been received.
2 ☐ not been received.
3 ☐ been filed in Application No. ____ .
4 ☐ been filed in reexamination Control No. ____.
5 ☐ been received by the International Bureau in PCT application No. ____.
* See the attached detailed Office action for a list of the certified copies not received.
9. ☐ Since the proceeding appears to be in condition for issuance of an *ex parte* reexamination certificate except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte* Quayle, 1935 C.D. 11, 453 O.G. 213.
10. ☐ Other: ____

cc: Requester (if third party requester)

Application/Control Number: 90/012,332
Art Unit: 3992

Page 2

DETAILED ACTION

1. Claims 1-21 of U.S. Patent No. 7,844,915 (“the ‘915 patent”) are under reexamination.

Procedural Posture

2. A request for *ex parte* reexamination of claims 1-21 of the ‘915 patent was filed on May 30, 2012, and an order granting the request was mailed on August 17, 2012. A first Office action was mailed on December 19, 2012.

A patent owner’s response to the Office action mailed on December 19, 2012 was filed on March 19, 2013. The response includes a statement of the substance of the interview held on March 14, 2013, at pages 2-4, and a declaration from Dr. Jason Nieh signed on March 18, 2013 (“the Nieh declaration”). No claims are amended or canceled.

Prior Art Cited in the Order

3. The following patents and printed publications were cited in the order granting the request for *ex parte* reexamination:

U.S. Patent No. 7,724,242 to Hillis et al. (“Hillis”).

U.S. Pub. No. 2005/0057524 to Hill et al. (“Hill”).

Dean Harris Rubine, “The Automatic Recognition of Gestures,” CMU-CS-91-202, December 1991 (“Rubine”).

Japanese Pub. No. 2000-163031A to Nomura et al. (English translation) (“Nomura”).

International Pub. No. WO 03/081458 to Lira (“Lira”).

U.S. Patent No. 6,677,965 to Ullmann et al. (“Ullmann”).

U.S. Patent No. 6,757,673 to Makus et al. (“Makus”).

Application/Control Number: 90/012,332

Page 3

Art Unit: 3992

Response to Arguments

4. Claim construction: In reexamination proceedings, claims are to be given the broadest reasonable interpretation consistent with the specification. See MPEP § 2258. As set forth in MPEP § 2111, the Federal Circuit’s *en banc* decision in *Phillips v. AWH Corp.*, 415 F.3d 1303, 75 USPQ2d 1321 (Fed. Cir. 2005) expressly recognized that the Office employs the “broadest reasonable interpretation” standard:

The Patent and Trademark Office (“PTO”) determines the scope of claims in patent applications not solely on the basis of the claim language, but upon giving claims their broadest reasonable construction “in light of the specification as it would be interpreted by one of ordinary skill in the art.” *In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1364[, 70 USPQ2d 1827] (Fed. Cir. 2004). Indeed, the rules of the PTO require that application claims must “conform to the invention as set forth in the remainder of the specification and the terms and phrases used in the claims must find clear support or antecedent basis in the description so that the meaning of the terms in the claims may be ascertainable by reference to the description.” 37 CFR 1.75(d)(1).

415 F.3d at 1316, 75 USPQ2d at 1329.

The broadest reasonable interpretation standard “is also [the] correct standard in reexamination proceedings.” *In re Yamamoto*, 740 F.2d 1569, 222 USPQ 934 (Fed. Cir. 1984).

Each of independent claims 1, 8 and 15 of the ‘915 patent recites, in pertinent part:

... distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation; ...

The patent owner’s arguments with respect to the independent claims are based on a particular interpretation of the language cited above. Specifically, the patent owner distills the “distinguishing” element into “distinguishing between a single input point ... and two or more input points,” and argues that the claims recite “a test that distinguishes (i.e., recognizes a

Application/Control Number: 90/012,332

Page 4

Art Unit: 3992

difference) between (a) a single input point and (b) two or more input points” (remarks, page 2).

The patent owner contends that the claims “require particularly identifying a single input point” (remarks, page 6), further arguing that “if there is one input point, the input is interpreted as a scroll operation” and “if there is not one input point ... the input is interpreted as a gesture operation” (remarks, page 8; emphasis added). **In other words, the patent owner interprets the claimed “distinguishing” element as a “one/not one” test.**

However, the examiner submits that the patent owner’s interpretation is not the broadest reasonable interpretation consistent with the specification. The independent claims of the ‘915 patent recite distinguishing between a single input point that is interpreted as a scroll operation and two or more input points that are interpreted as a gesture operation. Thus, giving these claims the broadest reasonable interpretation (as is the standard), any algorithm in the prior art that distinguishes between (a) a single input point that is interpreted as a scroll operation and (b) two input points or more than two input points that are interpreted as a gesture operation would meet the language of the claims. Furthermore, consistent with the broadest reasonable interpretation an algorithm in the prior art that distinguishes between (a) a single input point that is interpreted as a scroll operation and (b) two inputs points that are interpreted as a gesture operation would meet the language of the claims.

Indeed, as set forth in the Office action, Nomura teaches distinguishing between a single input point that is interpreted as a scroll operation (e.g., “moving one finger”) and two input points that are interpreted as a gesture operation (e.g., “two fingers moving apart” and “two fingers moving toward each other”) (see Nomura at paragraph [0053]). Nomura clearly recites a single input to be a scroll and two inputs to be a gesture, thus meeting the claimed limitation (see

Application/Control Number: 90/012,332

Page 5

Art Unit: 3992

Nomura at paragraph [0053]). Likewise, Hillis teaches distinguishing between a single input point that is interpreted as a scroll operation (e.g., “drawing a finger across the display surface”) and two or more input points that are interpreted as a gesture operation (e.g., “placing his fingertips on the display surface and moving them in an outwardly separating manner”) (see Hillis at column 8, lines 44-48 and column 3, lines 42-46).

Moreover, the examiner notes that there is no description in the ‘915 patent of a specific algorithm (or a specific interpretation) for distinguishing between a single input point and two or more input points, much less a description of a “one/not one” test such as in the patent owner’s arguments. That is to say, the specification of the ‘915 patent **does not describe any particular way or any particular algorithm for performing the claimed “distinguishing” element.** The term “distinguishing” is even not found within the specification of ‘915 patent, nor was the term recited in the claims as originally filed. Instead, the ‘915 patent describes that a single input point “may be” interpreted as a scroll operation and that two or more input points “may be” interpreted as a gesture operation:

FIG. 1 is flow chart of a method for responding to a user input of a device. The method 100 includes receiving a user input at block 102. The user input may be in the form of an input key, button, wheel, touch, or other means for interacting with the device. The method 100 further includes creating an event object in response to the user input at block 104. The method 100 further includes determining whether the event object invokes a scroll or gesture operation at block 106. For example, a single touch that drags a distance across a display of the device may be interpreted as a scroll operation. In one embodiment, a two or more finger touch of the display may be interpreted as a gesture operation. In certain embodiments, determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period.

(Column 6, lines 32-46; emphasis added.)

Application/Control Number: 90/012,332

Page 6

Art Unit: 3992

Thus, even when read in light of the specification, the Hillis and Nomura references teach the claimed “distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation” (emphasis added).

5. The patent owner contends that Nomura “does not and cannot distinguish between a single input point and two or more input points,” arguing that the disclosure of scroll and gesture operations in the Nomura reference “confuses the result (i.e., scroll and gesture operations) with the mechanism for achieving that result (i.e., distinguishing one input point from multiple input points)” (remarks, pages 7-8). The patent owner argues that in Nomura, “the processing of a single input point is handled no differently than the processing of three, or more input points, assuming other conditions (e.g., movement and pressure) are the same,” and contends that a person of ordinary skill in the art “would not have been motivated to modify [Nomura] to perform the distinguishing claimed in the ‘915 patent” (remarks, page 12).

The examiner generally agrees with the patent owner’s analysis of the algorithm shown in Figures 33, 34 and 37 of Nomura. See the patent owner’s remarks at pages 8-12 and the Nieh declaration at paragraphs 16-17. However, the premise of the patent owner’s argument is that “the claimed subject matter involves performing a test that distinguishes one input point versus two or more input points” (remarks, page 8) i.e. the Patent Owner’s claim interpretation is specific to the mechanism of achieving the result. However, the claim language does not support the specific mechanism and furthermore, the specification also does not support the specific mechanism or interpretation of the claim language (also see above response) i.e. the specification and the claim language does not support interpreting the claim language as a

Application/Control Number: 90/012,332

Page 7

Art Unit: 3992

“one/not one” test. Instead, the claims recite distinguishing between an (single) input that is interpreted as a scroll operation and an input (two or more (this includes also only two)) that is interpreted as a gesture operation.

Nomura teaches a finger movement detector 10 “for detecting the movement history of fingers on the display area ... for input of operations by the user such as zoom-in, zoom-out, rotate, scroll and the like” (see, e.g., paragraph [0049]). Zooming and rotating are examples of gesture operations; scrolling is a scroll operation. Nomura further teaches an operations details determination part 30 for determining whether the input is a scroll or gesture operation:

The operations details determination part 30 judges the operation details input by the user based on the finger movement history detected by the finger movement detector 10. Specifically, the operations details determination part 30 judges finger movement history detected by the finger movement detector 10 of two fingers moving apart as input of a map image zoom-in operation. Furthermore, for example, the operations details determination part 30 judges finger movement history detected by the finger movement detector 10 of two fingers moving toward each other as input of a map image zoom-out operation. Furthermore, the operations details determination part 30 judges finger movement history detected by the finger movement detector 10 of one finger rotating with another finger as an axis as input of a map image rotate operation. Furthermore, for example, the operations details determination part 30 judges finger movement history detected by the finger movement detector 10 of action of moving one finger as input of a map scroll operation.

(Nomura at paragraph [0053]; emphasis added.)

Thus, Nomura teaches distinguishing between an input consisting of a single input point (i.e., one finger) that is interpreted as a scroll operation and an input consisting of two or more input points (i.e., two fingers) that is interpreted as a gesture operation.

The patent owner’s argument that the processing of three or more input points in Nomura is apparently the same as the processing of a single input point is well taken. The examiner agrees that as shown in Figure 34 of Nomura, at step S110 (“Is there contact with two items?”),

Application/Control Number: 90/012,332

Page 8

Art Unit: 3992

the algorithm would follow the “N” branch to step S140 (“Is the contact point moving?”) in all cases where the input does not consist of two input points. However, none of the claims of the ‘915 patent recites an operation involving three or more input points. As noted above, Nomura teaches the claimed “distinguishing” element, distinguishing between an input consisting of a single input point that is interpreted as a scroll operation, such as “moving one finger,” and an input consisting of two or more (this includes also only two) input points that are interpreted as a gesture operation, such as “two fingers moving apart” or “two fingers moving toward each other.”

6. The patent owner acknowledges that the Office action does not rely on the Nomura reference for a teaching of “creating an event object,” but contends that the Office action “does not fully note the extent of Nomura’s lack of disclosure,” arguing that Nomura “fails to disclose the event object, scroll calls, gesture calls, and the view associated with an event object recited in the ‘915 claims” (remarks, pages 12-14).

However, the examiner disagrees. With respect to the claimed event object, the Office action reasoned that “creating an event object in response to the user input” would have been obvious in view of the Rubine reference. With respect to the claimed scroll and gesture calls, the Office action reasoned that Nomura teaches “issuing at least one scroll or gesture call” in terms of calling at least one of a zoom-in processor 42, a zoom-out processor 44, a rotation processor 46 and a scroll processor 48 based on the user input (see, e.g., paragraph [0054]). With respect to the claimed “view associated with the event object,” the Office action noted that the scrolling, zooming or rotating in Nomura is performed on the view associated with the user input. Here, the patent owner argues that Nomura “must determine each time a user input is received which

Application/Control Number: 90/012,332

Page 9

Art Unit: 3992

application should be activated depending on the area of the input,” rather than “associating a view with an event object and a software application” (remarks, page 13). The examiner notes, however, that the claims do not recite a step of “associating” the view with the event object and do not preclude some determination based on “the area of the input.”

7. The patent owner contends that Rubine “does not disclose using object-oriented programming for gestures,” arguing that the Office action “improperly conflates objects and gestures” and that Rubine “provides no enabling disclosure of creating an event object in response to multiple inputs” (remarks, pages 14-15).

However, Rubine clearly teaches an object-oriented programming system for gestures:

A single idea motivated the author to use object-oriented toolkits to construct gesture-based systems: gestures should be associated with objects on the screen. Just as an object's class determines the messages it understands, the author believed the class could and should be used to determine which gestures an object understands. The ideas of inheritance and overriding then naturally apply to gestures. ... Thus, the author created GRANDMA.

(Rubine at page 95.)

Moreover, the Office action does not conflate objects and gestures. In the GRANDMA system, gestures are represented as objects. Rubine describes, “In GRANDMA, gestural input is handled by objects of class GestureEventHandler” (see, e.g., page 128). Rubine further describes, “When input occurs, it is represented as an event” (see, e.g., page 105). These input events (i.e., the gestures) are represented as objects. Rubine states, “Input events are full-blown objects” (see, e.g., page 120), and describes how such “GestureEvent” objects are handled in the GRANDMA system (see, e.g., page 133). Thus, Rubine clearly teaches creating an event object in response to user input such as recited in the claims.

Application/Control Number: 90/012,332

Page 10

Art Unit: 3992

With respect to the user input consisting of “multiple input points,” the examiner refers instead to the Nomura reference. As set forth in the Office action, Nomura teaches the claimed user input, wherein “the user input is one or more input points applied to the touch-sensitive display,” and Rubine suggests creating an event object in response to that user input.

8. The patent owner contends that a person of ordinary skill in the art “would not have been motivated to combine Nomura and Rubine,” arguing that the flexibility of the GRANDMA system in Rubine “comes with a cost, including a performance cost” (remarks, page 16). The patent owner further argues that “Nomura explicitly teaches away from” support for different input devices simultaneously such as described in Rubine (remarks, page 17). The patent owner contends that Rubine “does not employ a touch-sensitive display,” but instead “uses a ‘Sensor Frame’ ... mounted on the CRT display of a Silicon Graphics workstation” (remarks, page 17). The patent owner further argues that because Rubine describes that GRANDMA “is purely a research system,” and describes a “simpler” non-object oriented alternative to GRANDMA, that a person of ordinary skill in the art “would not have been motivated to combine the event object creation in [Rubine] with Nomura” (remarks, pages 17-18).

However, the examiner disagrees. Combining the teachings of references does not involve an ability to combine the specific structures of the references. See *In re Nievelt*, 482 F.2d 965, 179 USPQ 224, 226 (CCPA 1973). While Rubine does state that GRANDMA itself involves “a great deal of mechanism” (see, e.g., page 181), the Office action does not propose incorporating the entirety of the GRANDMA system into the device of Nomura. Rather, the Office action reasoned that “creating an event object in response to the user input” in Nomura would have been obvious in view of the teachings of Rubine.

Application/Control Number: 90/012,332

Page 11

Art Unit: 3992

For example, a person of ordinary skill in the art could program the device of Nomura to represent the user input as an “event object” with a reasonable expectation of success. The flexibility of an object-oriented implementation is not limited to supporting different input devices simultaneously; Rubine further suggests that an object-oriented implementation would allow a user interface view to have any number of event handlers, where user input events are “automatically routed to the appropriate handler,” and would allow different views to share the same event handler (see, e.g., page 121). At the very least, creating an event object in response to the user input in Nomura would simply “impose structure” (e.g., a data structure) on the input event (see, e.g., Rubine at page 120). Thus, a person of ordinary skill in the art would have been prompted to create an event object in Nomura in order to store and manage, in a structured way, the “movement history, contact pressure, and contact area of the finger” defining the user input event (see, e.g., Nomura at paragraph [0139]).

With respect to the claimed “touch-sensitive display,” the Office action does not rely on the Rubine reference. Instead, Nomura teaches the touch-sensitive display in the form of “touch panel 1060, consisting of a transparent touch sensor ... overlaid on the display 1070” (see, e.g., paragraph [0130]). Nomura describes that the display includes a finger movement detector 1110 “for detecting the movement history, contact pressure, and contact area of the finger performed on the display area by the user” (see, e.g., paragraph [0139]). The teachings of Rubine therefore are analogous to those of Nomura because the “Sensor Frame” mounted on the display “detects the XY positions of up to three fingertips in a plane approximately one half inch in front of the display” (see, e.g., Rubine at page 79).

Application/Control Number: 90/012,332

Page 12

Art Unit: 3992

9. The patent owner further notes, generally, “the substantial secondary considerations supporting the non-obviousness of the claims of the ‘915 patent ... [including] the failure of others, evidence of copying demonstrated in litigation, and the unprecedented commercial success of the iPhone” (remarks, page 18).

However, evidence of secondary considerations must be factually supported with actual proof and must establish a nexus with the claimed invention. Consequently, the patent owner’s assertion of non-obviousness is of little probative value. See MPEP §§ 716.01(c) and 716.01(b). See also MPEP §§ 716.04, 716.06 and 716.03 for a discussion of evidence relating to the failure of others, copying and commercial success, respectively.

10. The patent owner contends that nothing in Nomura teaches or suggests that “determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period” such as recited in claims 5, 12 and 19 of the ‘915 patent (remarks, page 19). The patent owner contends that Nomura “does not use time or duration in its determination,” arguing that the “passage of time element” described in Nomura “does not include the recited limitation of determining whether the drag duration is above or below any particular threshold” (remarks, pages 19-20).

However, the examiner notes that “determining whether the drag duration is above or below any particular threshold” is not a limitation of the claims. Nomura describes:

Using the present invention, the user can input at least one operation selected from rotate, zoom-in, zoom-out, and scroll of a map image displayed in the display area through the movement history of his fingers.

The movement history of fingers contacting the display area is a concept including a passage of time element and is distinguished from an operation of

Application/Control Number: 90/012,332

Page 13

Art Unit: 3992

simply touching an input mark or the like with a finger that does not include a passage of time element.

(Nomura at paragraphs [0009]-[0010]; emphasis added.)

In Nomura, mere contact of one or more input points (i.e., without movement) is not enough to signal a scroll or gesture operation. Instead, identifying a scroll or gesture operation is based on the movement history of the one or more input points moving or “dragging” for some amount of time (i.e., the “passage of time element”). The tests for movement or dragging in Nomura are illustrated in Figure 34 at step S140 (“Is the contact point moving?”) and in Figure 37 at steps S320 (“Did the distance between the two [contact] points become larger?”) and S340 (“Did the distance between the two [contact] points get reduced?”), for example. Thus, Nomura teaches that “determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period” such as recited in the claims.

Moreover, while the patent owner argues that “it is irrelevant” (remarks, page 20), the examiner notes that Nomura further contemplates testing for “contact by fingers within a specified amount of time” (see, e.g., paragraph [0193]).

11. The patent owner contends that the Hillis reference “does not involve distinguishing between one input point and two or more input points,” arguing that Hillis “is silent on this test” and that nothing in Hillis “discloses that the number of inputs is used to distinguish between scroll and gesture operations” (remarks, pages 20-21).

However, Hillis “distinguishes” the exact number of input points:

In step 201, the user initiates (and the display/computer detects) the user’s physical contact with the display surface 124. Without any intended limitation, the illustrated embodiment of the sequence 200 performs one instance of the (repeating) steps 202-204 for each such contact initiated. The

Application/Control Number: 90/012,332

Page 14

Art Unit: 3992

contact of step 201 is referred to as the “current” contact. In one gesture recognition scheme, the computer 126 tracks a predetermined number of distinct contact locations (such as two). If the computer identifies another contact location (such as a third), the computer 126 ignores it until the user releases a sufficient number of the existing contact locations.

(Hillis at column 7, lines 4-14; emphasis added.)

Hillis goes on to describe determining whether the detected number of input points matches a predetermined pattern:

In step 208, the computer 126 determines whether activity of the current contact matches a predetermined pattern, and therefore constitutes a “gesture.” Step 208 repeats continually, utilizing some or all of the position, position history (movement), velocity, and force information from steps 202, 204, 206. More particularly, in step 208 the computer 126 compares the history of contact position, size, movement, velocity, and/or force to the dictionary 126a of predetermined gestures to determine if the user has performed any of these gestures.

(Hillis at column 7, lines 46-55.)

The patterns and operations described in Hillis include “panning, zooming, rotating, and the like” (see, e.g., column 8, lines 4-8). Zooming and rotating are examples of gesture operations and panning is an example of a scroll operation.

As set forth in the Office action, Hillis describes a single input point that is interpreted as a pan or scroll operation (i.e., “drawing a finger across the display surface”) (see, e.g., column 8, lines 44-48), and describes two or more input points that are interpreted as a gesture operation (i.e., “placing his fingertips on the display surface and moving them in an outwardly separating manner”) (see, e.g., column 3, lines 42-46). Thus, Hillis teaches distinguishing between an input consisting of a single input point that is interpreted as a scroll operation and an input consisting of two or more input points that is interpreted as a gesture operation.

Application/Control Number: 90/012,332

Page 15

Art Unit: 3992

12. The patent owner contends that Hillis does not teach the claimed “event object,” arguing that the Office action “recognizes that there cannot be the recited ‘event object’ in the absence of an object-oriented environment in Nomura,” and that the “machine readable output” described in Hillis “is a non-specific term that does not require any higher-level programming abstractions and certainly not event objects in particular” (remarks, pages 21-22).

However, in the case of Nomura and Rubine, the Office action cited the object-oriented programming system of Rubine because Nomura was silent with respect to “creating an event object in response to the user input.” The Hillis reference is not silent. Rather, Hillis teaches creating “a machine readable output” that “is representative of the position, size, shape, and timing of each contact region,” and further teaches creating and storing “a position history for each contact region” that “provides a record of how each contact region moves and/or changes shape over time” (see, e.g., column 7, lines 15-25). Thus, Hillis teaches “creating an event object in response to the user input” such as recited in the claims.

The examiner notes that the claims do not specify or limit what constitutes an “event object,” nor does the specification of the ‘915 patent define the term “event object.” At best, the ‘915 patent describes, “A multi-touch driver of the device receives the user input and packages the event into an event object” (column 12, lines 30-32). Thus, a reasonable interpretation is that the claimed event object is an entity or “package” of data representing the user input event. The “machine readable output” and “position history” representing the position, size, shape and timing of each input point in Hillis is such an entity.

The patent owner further contends that Hillis does not teach the claimed “view associated with the event object,” arguing that the description of “[panning] the imagery” in Hillis “does not

Application/Control Number: 90/012,332

Page 16

Art Unit: 3992

disclose a view associated with an event object” (remarks, page 22). However, the examiner submits that the imagery or part of the imagery that is panned in response to the user input in Hillis is in fact the “view associated with the event object.”

13. The patent owner contends that Hillis does not teach the claimed “touch-sensitive display,” arguing that the display surface 124 in Hillis “is not a touch sensitive display” and that nothing “indicates that the display itself responds to touch” (remarks, pages 22-23).

However, Hillis clearly teaches “[an] interactive display system, including a touch sensitive display” (see, e.g., abstract; emphasis added). Hillis describes, for example, that “[the] table 122 detects touch input from human users as applied to the display surface 124” (see, e.g., column 3, lines 4-7), and further describes “various approaches to detect when and where a user touches the display surface” (see, e.g., column 4, lines 17-19). Thus, Hillis teaches a “touch-sensitive display” such as recited in the claims.

14. The patent owner further contends that Hillis does not teach a touch-sensitive display “that is integrated with” the device, arguing that the term “integrated” means that the device “is in a single physical housing” (remarks, page 23). The patent owner argues, “Large physical size was important to Hillis,” and while “handheld displays were known at the time ... Hillis fails to discuss them in the context of its disclosure” (remarks, page 24).

However, the examiner notes that “large physical size” and “integrated” are not mutually exclusive. Furthermore, the physical size and “physical housing” of the components are not limitations of the claims. Hillis teaches a touch-sensitive display “that is integrated with” the device in the sense that the display surface 124 (i.e., the “touch-sensitive display”) is integrated

Application/Control Number: 90/012,332

Page 17

Art Unit: 3992

with the interactive display system 120 (see, e.g., FIG. 1A and column 2, lines 60-65). The display surface 124 and the computer 126 are necessary components of the interactive display system 120 and therefore are “integrated with” the system.

15. The patent owner contends that Hillis does not teach that “determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period” such as recited in claims 5, 12 and 19 of the ‘915 patent, arguing that Hillis does not describe that “it is the duration of a drag above a certain time period that distinguishes a scroll from a gesture” (remarks, pages 24-25).

However, none of the claims recites that “it is the duration of a drag above a certain time period that distinguishes a scroll from a gesture.” In Hillis, identifying a scroll or gesture operation is based on “the timing of each contact region” and “a record of how each contact region moves and/or changes shape over time” (see, e.g., column 7, lines 15-25). A contact with the display surface (i.e., an input point) that moves over a period of time constitutes “receiving a drag user input for a certain time period.” Hillis further describes that identifying the scroll or gesture operation is based on the movement, velocity and force of the (drag) user input:

In step 208, the computer 126 determines whether activity of the current contact matches a predetermined pattern, and therefore constitutes a “gesture.” Step 208 repeats continually, utilizing some or all of the position, position history (movement), velocity, and force information from steps 202, 204, 206. More particularly, in step 208 the computer 126 compares the history of contact position, size, movement, velocity, and/or force to the dictionary 126a of predetermined gestures to determine if the user has performed any of these gestures.

(Hillis at column 7, lines 46-55; emphasis added.)

Application/Control Number: 90/012,332

Page 18

Art Unit: 3992

The velocity of the user input, for example, represents the change in the position of the contact over time. Thus, Hillis teaches the limitation that “determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period” such as recited in the claims.

16. The patent owner contends that Lira does not teach “rubberbanding a scrolling region displayed within the window by a predetermined maximum displacement when the scrolling region exceeds a window edge based on the scroll” such as recited in claims 2, 9 and 16 of the ‘915 patent (remarks, page 25). First, the patent owner argues that Lira does not perform the rubberbanding “when the scrolling region exceeds a window edge based on the scroll” because Lira “does not disclose what will or should happen if and when the user tries to scroll past the edge of the page,” and instead “discloses that this re-centering functionality occurs whether or not the edge of the content (i.e., webpage) is reached” (remarks, pages 26-27). Second, the patent owner refers to the description in the specification of the ‘915 patent that “[at] the end of the scroll, the content slides back making the region outside of the content no longer visible on the display” (column 7, lines 65-67), and argues that Lira “does not disclose anything that occurs in the region outside the content” (remarks, page 28).

However, “the region outside the content” is not a limitation of the claims. As set forth in the Office action, Lira teaches snapping or “rubberbanding” a column of the page based on whether the horizontal scrolling of the column exceeds a threshold:

Referring to Fig. 14B, in another implementation, the vertical alignment control is enabled when the user lifts the pen 1200 from the display 1205. This causes the logical column 1220 to snap into alignment with the display window 1205 as the user stops scrolling. The user can adjust the snap sensitivity by, for example, setting the alignment control to snap to the nearest

Application/Control Number: 90/012,332

Page 19

Art Unit: 3992

logical column based on a user-defined snap threshold. If the user's scrolling does not exceed the threshold, which indicates an intention to continue to view the text column 1220, the display 1205 centers the logical column 1210 as the pen 1200 is lifted from the screen. If the user's scrolling exceeds the threshold, which indicates an intention to move beyond the boundary of the logical column 1220, the display is snapped to the adjacent or repositioned column. In other implementations, no snapping occurs when the user's scrolling exceeds the threshold. The snap-on-column feature can also be animated to provide an appearance of movement as the display scrolls to the correct column-viewing position.

(Lira at page 15, lines 18-31.)

When the scrolling does not exceed the threshold, the column is centered and “[snapped] into alignment with the display window 1205.” When the scrolling exceeds the threshold and “[moves] beyond the boundary” of the column, the display window 1205 snaps to the next nearest column. Thus, Lira teaches “rubberbanding” a column of the page.

Reaching the edge of the page and “[trying] to scroll past the edge of the page,” such as in the patent owner's argument, is immaterial because the threshold in Lira relates to the edge of each column, rather than the edge of the page. Lira describes that the columns are sized such that “[the] width of each logical column is less than or equal to the display window width” (see, e.g., page 11, lines 10-17). Therefore, in the case where the width of the column is the same as the width of the display window 1205, the snapping or “rubberbanding” is performed when the horizontal scrolling of the column “exceeds a window edge based on the scroll” such as recited in the claims. Thus, as set forth in the Office action, the teachings of Lira would have suggested “rubberbanding a scrolling region displayed within the window by a predetermined maximum displacement when the scrolling region exceeds a window edge based on the scroll” to those of ordinary skill in the art.

Application/Control Number: 90/012,332

Page 20

Art Unit: 3992

17. The patent owner further contends that a person of ordinary skill in the art “would not have been motivated to combine Lira with Hillis or Nomura” (remarks, page 28). The patent owner argues that “the recentering approach of Lira is incompatible with the scrolling or panning disclosed in Nomura and Hillis” because “new content is revealed in opposing ways in Lira versus either Hillis or Nomura” (remarks, page 28). The patent owner further argues that “there would not have been a reason to implement a recentering approach such as that disclosed in Lira with the large scale maps shown in Nomura” because the maps “would keep scrolling virtually endlessly in each direction” (remarks, page 29).

However, the examiner does not agree with the patent owner’s conclusions. As Lira suggests, the snapping or “rubberbanding” discussed above would “provide an appearance of movement as the display scrolls to the correct [viewing] position” (see, e.g., Lira at page 15, lines 29-31). A person of ordinary skill in the art would have been prompted to implement such a feature in Hillis and Nomura. Namely, as set forth in the Office action, a person of ordinary skill in the art would have been prompted to implement the teachings of Hillis or Nomura such that the scrolled area is automatically and visually snapped back into alignment with the window based on whether the scrolling exceeds a threshold.

The direction of the scrolling in Lira as compared to the direction of the scrolling in Hillis or Nomura would not have prevented a person of ordinary skill in the art from combining the teachings of the references. In fact, choosing the direction of the scrolling (and/or the direction from which “new content is revealed”) would have been within the level of ordinary skill. A person of ordinary skill is also a person of ordinary creativity, not an automaton. See *KSR Int’l Co. v. Teleflex Inc.*, 82 USPQ2d 1385, 1397 (U.S. 2007). Likewise, even if the maps displayed

Application/Control Number: 90/012,332

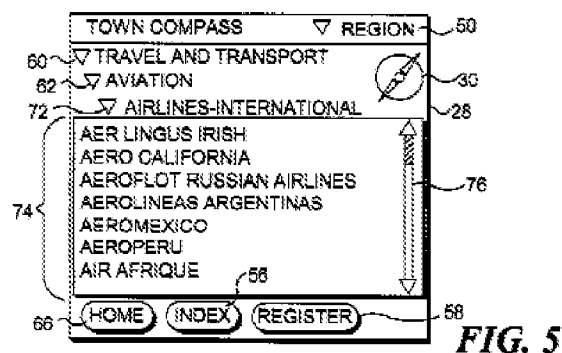
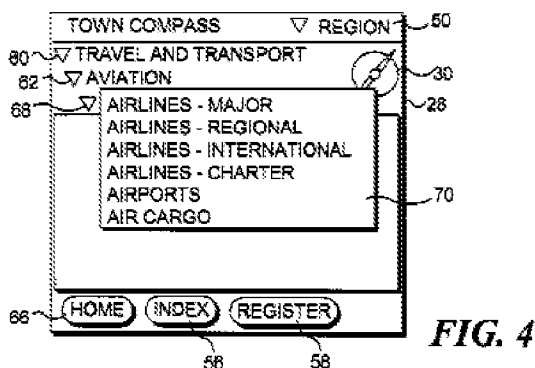
Page 21

Art Unit: 3992

in Nomura were to “keep scrolling virtually endlessly in each direction,” as the patent owner argues, a person of ordinary skill could implement the snapping or “rubberbanding” for a logical boundary within the map (e.g., at the boundary of a geographical area), analogous to the logical columns of the pages described in Lira (see, e.g., Lira at page 15, lines 18-31).

18. The patent owner contends that Makus does not teach or suggest “attaching scroll indicators” such as recited in claims 3, 4, 10, 11, 17 and 18 of the ‘915 patent, arguing that Makus “appears to disclose scroll indicators that are either ... present or not present from the outset” and that Makus does not “attach” them (remarks, pages 29-30).

However, the claims do not specify when the “scroll indicators” are “attached,” nor do the claims preclude attaching them “from the outset.” Nonetheless, in the Makus reference, the scroll bar 76 (i.e., the “scroll indicator”) is attached when the user touches the display screen 28, “causing a list 74 of [items] to be displayed” (see, e.g., column 8, lines 59-64). The scroll bar 76 is displayed (i.e., “attached” to the display) simultaneously because more items are included in the list 74 “than can be displayed in the available space on display screen 28 at one time” (see, e.g., column 8, line 64 to column 9, line 1). FIGS. 4 and 5 of Makus illustrate the “before” and “after,” respectively, of attaching the scroll bar 76 to the display:



Application/Control Number: 90/012,332

Page 22

Art Unit: 3992

Thus, the teachings of Makus would have suggested “attaching scroll indicators” to those of ordinary skill in the art.

19. The patent owner further contends that a person of ordinary skill in the art “would not have been motivated to combine Makus with Nomura ... because scrolled maps do not need scroll indicators,” arguing that “commercially available map systems generally do not use scroll indicators even today” and that “maps generally have no starting position from which relative position can be measured” (remarks, pages 30-31).

However, the examiner disagrees. The test for obviousness is what the combined teachings of the references would have suggested to those of ordinary skill in the art. See *In re Keller*, 642 F.2d 413, 425, 208 USPQ 871, 881 (CCPA 1981). Whether or not “commercially available map systems ... even today” attach “scroll indicators” to the display is immaterial. In Nomura, maps such as the “detailed city maps” cited in the patent owner’s argument (remarks, page 30) would have a length and a width. The part of the map viewable to the user changes as the map is scrolled; a “scroll indicator” such as described in Makus would illustrate, for example, the relative position of that view with respect to the size of the full map. Thus, as set forth in the Office action, “attaching scroll indicators” to the maps of Nomura would have been obvious to those of ordinary skill in the art.

20. The patent owner further contends that a person of ordinary skill in the art “would not have been motivated to combine Makus with Hillis,” arguing that the description of “slider mode” in Hillis “has no relationship to [its] disclosure of panning,” and that Hillis would not “have any need of scroll indicators” to access more content than what fits on the screen because

Application/Control Number: 90/012,332

Page 23

Art Unit: 3992

Hillis “already discloses” both panning and the slider mode (remarks, pages 31-32). The patent owner further argues “there would have been no reason to combine attaching scroll indicators ... with the types of documents disclosed in Hillis,” referring to documents such as “maps, aerial photographs, or integrated circuits” (remarks, page 32).

However, the examiner disagrees. The slider mode described in Hillis involves attaching “a slider tool” to the display such as “a linearly moveable slider bar 560” that “observes an appropriately convenient scale” (see, e.g., FIG. 5B and column 16, lines 29-39). The slider bar 560 is a form of a “scroll indicator.” Thus, Hillis teaches “attaching a scroll indicator” to the display. To the extent that the slider mode of Hillis is separate from the description of panning or scrolling the maps and other documents, the Office action cites the Makus reference. As noted above, Makus teaches a scroll bar 76 (see, e.g., FIG. 5) that would illustrate, for example, the relative position of the part of the map viewable to the user with respect to the size of the full map. Thus, as set forth in the Office action, “attaching scroll indicators” to the maps and other documents of Hillis would have been obvious to those of ordinary skill in the art.

Summary of Rejections

21. The following rejections of the claims are set forth below in this Office action:

Ground 1: Claims 1, 5-8, 12-15 and 19-21 are rejected under 35 U.S.C. § 102(e) as anticipated by Hillis.

Ground 2: Claims 2, 9 and 16 are rejected under 35 U.S.C. § 103(a) as unpatentable over Hillis in view of Lira.

Ground 3: Claims 3, 4, 10, 11, 17 and 18 are rejected under 35 U.S.C. § 103(a) as unpatentable over Hillis in view of Makus.

Application/Control Number: 90/012,332
Art Unit: 3992

Page 24

Ground 4: Claims 1, 5-8, 12-15 and 19-21 are rejected under 35 U.S.C. § 103(a) as unpatentable over Nomura in view of Rubine.

Ground 5: Claims 2, 9 and 16 are rejected under 35 U.S.C. § 103(a) as unpatentable over Nomura in view of Rubine and further in view of Lira.

Ground 6: Claims 3, 4, 10, 11, 17 and 18 are rejected under 35 U.S.C. § 103(a) as unpatentable over Nomura in view of Rubine and further in view of Makus.

Claim Rejections under 35 U.S.C. § 102

22. The following is a quotation of the appropriate paragraphs of 35 U.S.C. § 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

23. Ground 1: Claims 1, 5-8, 12-15 and 19-21 are rejected under 35 U.S.C. § 102(e) as anticipated by Hillis.

Claim 1

A machine implemented method for scrolling on a touch-sensitive display of a device comprising:

Hillis teaches a machine-implemented method for panning (i.e., scrolling) on a touch-sensitive display of a device (see, e.g., column 1, lines 29-36 and column 3, lines 21-24).

Application/Control Number: 90/012,332

Page 25

Art Unit: 3992

receiving a user input, the user input is one or more input points applied to the touch-sensitive display that is integrated with the device;

Hillis teaches receiving user input comprising one or more contact points applied to the touch-sensitive display (see, e.g., column 6, lines 59-63, "Broadly, the steps 202, 204, 206 run continuously to process user contact with the display surface 124 as it occurs. Steps 202, 204, 206 therefore serve to analyze contact occurring when the user contacts the surface 124 at one or more contact regions utilizing one or more fingers, hands, arms, etc."). The display is integrated with the device (see, e.g., column 2, lines 60-65, "One aspect of the present disclosure concerns an interactive touch detecting display system, which may be embodied by various hardware components and interconnections, with one example being described in FIG. 1A. The system 120 includes a table 122 with a display surface 124, computer 126, and projector 128.").

creating an event object in response to the user input;

Hillis teaches creating an event object in response to the user input (see, e.g., column 7, lines 15-25, "In step 202, the table 122 detects and monitors the position, size, shape, and timing of the current contact region. Namely, the table 122 provides a machine readable output to the computer 126, which is representative of the position, size, shape, and timing of each contact region, or contains information from which this information can be calculated or derived. The timing output may be satisfied, for example, by the table 122 providing its output in real time. Also in step 202, the computer 126 stores a position history for each contact region. The position history provides a record of how each contact region moves or and/or changes shape over time.").

determining whether the event object invokes a scroll or gesture operation by distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation;

Hillis teaches distinguishing the number of contact points and determining whether the event object matches a gesture pattern (see, e.g., column 7, lines 4-12, "In step 201, the user initiates (and the display/computer detects) the user's physical contact with the display surface 124. Without any intended limitation, the illustrated embodiment of the sequence 200 performs one instance of the (repeating) steps 202-204 for each such contact

Application/Control Number: 90/012,332

Page 26

Art Unit: 3992

initiated. The contact of step 201 is referred to as the 'current' contact. In one gesture recognition scheme, the computer 126 tracks a predetermined number of distinct contact locations (such as two)," and see, e.g., column 7, lines 46-65, "In step 208, the computer 126 determines whether activity of the current contact matches a predetermined pattern, and therefore constitutes a 'gesture.' Step 208 repeats continually, utilizing some or all of the position, position history (movement), velocity, and force information from steps 202, 204, 206. ... if step 208 detects that the user has initiated a gesture (208c), the computer in step 214 utilizes the mapping 126c to identify the action 126b associated with the gesture that was identified in step 208.").

Hillis further teaches that the gestures include a panning (i.e., scrolling) operation and other gesture operations such as zooming (see, e.g., column 8, lines 5-7, "As described in greater detail below, some examples of actions 126b include panning, zooming, rotating, and the like.")). Hillis distinguishes between a single contact point that is interpreted as the scrolling operation (see, e.g., column 8, lines 44-48) and two or more contact points that are interpreted as the zooming operation (see, e.g., FIG. 1B and column 3, lines 42-46).

issuing at least one scroll or gesture call based on invoking the scroll or gesture operation;

Hillis teaches issuing a call to perform the scrolling or other gesture operation (see, e.g., column 7, line 65 to column 8, line 3, "As mentioned above, the predefined actions include various machine implemented operations for updating the presentation of imagery by the display. In one embodiment, gestures are both identified (208) and associated (214) with display control commands via a single procedure," and see, e.g., column 8, lines 4-8, "After step 214, the computer 126 initiates performance of the identified action (step 216). As described in greater detail below, some examples of actions 126b include panning, zooming, rotating, and the like. Thus, step 216 starts the requested pan, zoom, rotate, or other operation.").

Application/Control Number: 90/012,332

Page 27

Art Unit: 3992

responding to at least one scroll call, if issued, by scrolling a window having a view associated with the event object based on an amount of a scroll with the scroll stopped at a predetermined position in relation to the user input; and

Hillis teaches responding to the call by panning (i.e., scrolling) the view associated with the event object based on the amount of scroll (see, e.g., column 8, lines 44-52, “Another example is where the computer 126 detects (FIG. 2, step 208) that the user has initiated a pan gesture by drawing a finger across the display surface at a particular velocity, and lifted his/her finger from the surface while still moving (FIG. 2, step 218b). With the optional inertia feature enabled, the computer 126 continues (FIG. 2, step 222) to pan the imagery in the initiated direction at the velocity implied by the gesture at the time the finger was lifted until a stopping or slowing naturally occurs (step 224).”). Hillis further teaches that the scrolling is stopped at a predetermined position based on the user input (see, e.g., column 8, line 63 to column 9, line 6).

responding to at least one gesture call, if issued, by scaling the view associated with the event object based on receiving the two or more input points in the form of the user input.

Hillis teaches responding to the call by zooming (i.e., scaling) the view associated with the event object based on the two or more contact points (see, e.g., column 3, lines 42-49, “In the example of FIG. 1B, a user 16 has gestured by placing his fingertips on the display surface and moving them in an outwardly separating manner. As discussed in greater detail below, this particular gesture 17 is associated with a zoom-in command. When the computer 126 performs a zoom-in command, it directs the projector to provide 128 a closer, more detailed view of the displayed imagery.”).

Claim 8

A machine readable storage medium storing executable program instructions which when executed cause a data processing system to perform a method comprising:

Hillis teaches a machine-readable storage medium storing instructions for causing a data processing system to perform a method (see, e.g., column 1, lines 29-36 and column 3, lines 21-24, and see, e.g., column 5, lines 48-59).

receiving a user input, the user input is one or more input points applied to a touch-sensitive display that is integrated with the data processing system;

Hillis teaches receiving user input comprising one or more contact points applied to a touch-sensitive display (see, e.g., column 6, lines 59-63, “Broadly, the steps 202, 204, 206 run continuously to process

Application/Control Number: 90/012,332

Page 28

Art Unit: 3992

user contact with the display surface 124 as it occurs. Steps 202, 204, 206 therefore serve to analyze contact occurring when the user contacts the surface 124 at one or more contact regions utilizing one or more fingers, hands, arms, etc.”). The display is integrated with the data processing system (see, e.g., column 2, lines 60-65, “One aspect of the present disclosure concerns an interactive touch detecting display system, which may be embodied by various hardware components and interconnections, with one example being described in FIG. 1A. The system 120 includes a table 122 with a display surface 124, computer 126, and projector 128.”).

creating an event object in response to the user input;

Hillis teaches creating an event object in response to the user input (see, e.g., column 7, lines 15-25, “In step 202, the table 122 detects and monitors the position, size, shape, and timing of the current contact region. Namely, the table 122 provides a machine readable output to the computer 126, which is representative of the position, size, shape, and timing of each contact region, or contains information from which this information can be calculated or derived. The timing output may be satisfied, for example, by the table 122 providing its output in real time. Also in step 202, the computer 126 stores a position history for each contact region. The position history provides a record of how each contact region moves or and/or changes shape over time.”).

determining whether the event object invokes a scroll or gesture operation by distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation;

Hillis teaches distinguishing the number of contact points and determining whether the event object matches a gesture pattern (see, e.g., column 7, lines 4-12, “In step 201, the user initiates (and the display/computer detects) the user’s physical contact with the display surface 124. Without any intended limitation, the illustrated embodiment of the sequence 200 performs one instance of the (repeating) steps 202-204 for each such contact initiated. The contact of step 201 is referred to as the ‘current’ contact. In one gesture recognition scheme, the computer 126 tracks a predetermined

Application/Control Number: 90/012,332

Page 29

Art Unit: 3992

number of distinct contact locations (such as two),” and see, e.g., column 7, lines 46-65, “In step 208, the computer 126 determines whether activity of the current contact matches a predetermined pattern, and therefore constitutes a ‘gesture.’ Step 208 repeats continually, utilizing some or all of the position, position history (movement), velocity, and force information from steps 202, 204, 206. ... if step 208 detects that the user has initiated a gesture (208c), the computer in step 214 utilizes the mapping 126c to identify the action 126b associated with the gesture that was identified in step 208.”).

Hillis further teaches that the gestures include a panning (i.e., scrolling) operation and other gesture operations such as zooming (see, e.g., column 8, lines 5-7, “As described in greater detail below, some examples of actions 126b include panning, zooming, rotating, and the like.”). Hillis distinguishes between a single contact point that is interpreted as the scrolling operation (see, e.g., column 8, lines 44-48) and two or more contact points that are interpreted as the zooming operation (see, e.g., FIG. 1B and column 3, lines 42-46).

issuing at least one scroll or gesture call based on invoking the scroll or gesture operation;

Hillis teaches issuing a call to perform the scrolling or other gesture operation (see, e.g., column 7, line 65 to column 8, line 3, “As mentioned above, the predefined actions include various machine implemented operations for updating the presentation of imagery by the display. In one embodiment, gestures are both identified (208) and associated (214) with display control commands via a single procedure,” and see, e.g., column 8, lines 4-8, “After step 214, the computer 126 initiates performance of the identified action (step 216). As described in greater detail below, some examples of actions 126b include panning, zooming, rotating, and the like. Thus, step 216 starts the requested pan, zoom, rotate, or other operation.”).

responding to at least one scroll call, if issued, by scrolling a window having a view associated with the event object;

Hillis teaches responding to the call by panning (i.e., scrolling) the view associated with the event object (see, e.g., column 8, lines 44-52, “Another example

Application/Control Number: 90/012,332

Page 30

Art Unit: 3992

and

is where the computer 126 detects (FIG. 2, step 208) that the user has initiated a pan gesture by drawing a finger across the display surface at a particular velocity, and lifted his/her finger from the surface while still moving (FIG. 2, step 218b). With the optional inertia feature enabled, the computer 126 continues (FIG. 2, step 222) to pan the imagery in the initiated direction at the velocity implied by the gesture at the time the finger was lifted until a stopping or slowing naturally occurs (step 224).”).

responding to at least one gesture call, if issued, by scaling the view associated with the event object based on receiving the two or more input points in the form of the user input.

Hillis teaches responding to the call by zooming (i.e., scaling) the view associated with the event object based on the two or more contact points (see, e.g., column 3, lines 42-49, “In the example of FIG. 1B, a user 16 has gestured by placing his fingertips on the display surface and moving them in an outwardly separating manner. As discussed in greater detail below, this particular gesture 17 is associated with a zoom-in command. When the computer 126 performs a zoom-in command, it directs the projector to provide 128 a closer, more detailed view of the displayed imagery.”).

Claim 15

An apparatus, comprising:

Hillis teaches an apparatus (see, e.g., column 1, lines 29-36 and column 3, lines 21-24, and see, e.g., column 5, lines 17-21).

means for receiving *[under § 112, ¶ 6, the “multi-touch driver” described at column 12, lines 30-32]*, through a hardware device, a user input on a touch-sensitive display of the apparatus, the user input is one or more input points applied to the touch-sensitive display that is integrated with the apparatus;

Hillis teaches equivalent software (see, e.g., column 5, lines 52-55) for receiving user input comprising one or more contact points applied to a touch-sensitive display (see, e.g., column 6, lines 59-63, “Broadly, the steps 202, 204, 206 run continuously to process user contact with the display surface 124 as it occurs. Steps 202, 204, 206 therefore serve to analyze contact occurring when the user contacts the surface 124 at one or more contact regions utilizing one or more fingers, hands, arms, etc.”). The display is integrated with the apparatus (see, e.g., column 2, lines 60-65, “One aspect of the present disclosure concerns an interactive touch detecting display

Application/Control Number: 90/012,332

Page 31

Art Unit: 3992

system, which may be embodied by various hardware components and interconnections, with one example being described in FIG. 1A. The system 120 includes a table 122 with a display surface 124, computer 126, and projector 128.”).

means for creating [*under § 112, ¶ 6, the “multi-touch driver” described at column 12, lines 30-32*] an event object in response to the user input;

Hillis teaches equivalent software (see, e.g., column 5, lines 52-55) for creating an event object in response to the user input (see, e.g., column 7, lines 15-25, “In step 202, the table 122 detects and monitors the position, size, shape, and timing of the current contact region. Namely, the table 122 provides a machine readable output to the computer 126, which is representative of the position, size, shape, and timing of each contact region, or contains information from which this information can be calculated or derived. The timing output may be satisfied, for example, by the table 122 providing its output in real time. Also in step 202, the computer 126 stores a position history for each contact region. The position history provides a record of how each contact region moves or and/or changes shape over time.”).

means for determining [*under § 112, ¶ 6, the “window server” described at column 12, lines 32-34*] whether the event object invokes a scroll or gesture operation by distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation;

Hillis teaches equivalent software (see, e.g., column 5, lines 52-55) for distinguishing the number of contact points and determining whether the event object matches a gesture pattern (see, e.g., column 7, lines 4-12, “In step 201, the user initiates (and the display/computer detects) the user’s physical contact with the display surface 124. Without any intended limitation, the illustrated embodiment of the sequence 200 performs one instance of the (repeating) steps 202-204 for each such contact initiated. The contact of step 201 is referred to as the ‘current’ contact. In one gesture recognition scheme, the computer 126 tracks a predetermined number of distinct contact locations (such as two),” and see, e.g., column 7, lines 46-65, “In step 208, the computer 126 determines whether activity of the current contact matches a predetermined pattern, and therefore constitutes a ‘gesture.’ Step 208 repeats continually, utilizing some or all of the position, position history (movement), velocity, and force

Application/Control Number: 90/012,332

Page 32

Art Unit: 3992

information from steps 202, 204, 206. ... if step 208 detects that the user has initiated a gesture (208c), the computer in step 214 utilizes the mapping 126c to identify the action 126b associated with the gesture that was identified in step 208.”).

Hillis further teaches that the gestures include a panning (i.e., scrolling) operation and other gesture operations such as zooming (see, e.g., column 8, lines 5-7, “As described in greater detail below, some examples of actions 126b include panning, zooming, rotating, and the like.”). Hillis distinguishes between a single contact point that is interpreted as the scrolling operation (see, e.g., column 8, lines 44-48) and two or more contact points that are interpreted as the zooming operation (see, e.g., FIG. 1B and column 3, lines 42-46).

means for issuing *[under § 112, ¶ 6, the “user interface software” described at column 12, lines 34-37]* at least one scroll or gesture call based on invoking the scroll or gesture operation;

Hillis teaches equivalent software (see, e.g., column 5, lines 52-55) for issuing a call to perform the scrolling or other gesture operation (see, e.g., column 7, line 65 to column 8, line 3, “As mentioned above, the predefined actions include various machine implemented operations for updating the presentation of imagery by the display. In one embodiment, gestures are both identified (208) and associated (214) with display control commands via a single procedure,” and see, e.g., column 8, lines 4-8, “After step 214, the computer 126 initiates performance of the identified action (step 216). As described in greater detail below, some examples of actions 126b include panning, zooming, rotating, and the like. Thus, step 216 starts the requested pan, zoom, rotate, or other operation.”).

means for responding *[under § 112, ¶ 6, the “window or view” described at column 12, lines 44-46]* to at least one scroll call, if issued, by scrolling a window having a view associated with the event object; and

Hillis teaches equivalent software (see, e.g., column 5, lines 52-55) for responding to the call by panning (i.e., scrolling) the view associated with the event object (see, e.g., column 8, lines 44-52, “Another example is where the computer 126 detects (FIG. 2, step 208) that the user has initiated a pan gesture by drawing a finger across the display surface at a particular velocity, and lifted his/her finger from the

Application/Control Number: 90/012,332

Page 33

Art Unit: 3992

surface while still moving (FIG. 2, step 218b). With the optional inertia feature enabled, the computer 126 continues (FIG. 2, step 222) to pan the imagery in the initiated direction at the velocity implied by the gesture at the time the finger was lifted until a stopping or slowing naturally occurs (step 224).”).

means for responding *[under § 112, ¶ 6, the “window or view” described at column 12, lines 44-46]* to at least one gesture call, if issued, by scaling the view associated with the event object based on receiving the two or more input points in the form of the user input.

Hillis teaches equivalent software (see, e.g., column 5, lines 52-55) for responding to the call by zooming (i.e., scaling) the view associated with the event object based on the two or more contact points (see, e.g., column 3, lines 42-49, “In the example of FIG. 1B, a user 16 has gestured by placing his fingertips on the display surface and moving them in an outwardly separating manner. As discussed in greater detail below, this particular gesture 17 is associated with a zoom-in command. When the computer 126 performs a zoom-in command, it directs the projector to provide 128 a closer, more detailed view of the displayed imagery.”).

Claim 5

The method as in claim 1, wherein determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period.

Claims 5, 12 and 19

Hillis further teaches that determining whether the event object matches a gesture pattern is based on receiving a drag input for a period of time (see, e.g., column 7, lines 15-25, “In step 202, the table 122 detects and monitors the position, size, shape, and timing of the current contact region. Namely, the table 122 provides a machine readable output to the computer 126, which is representative of the position, size, shape, and timing of each contact region, or contains information from which this information can be calculated or derived. The timing output may be satisfied, for example, by the table 122 providing its output in real time. Also in step 202, the computer 126 stores a position history for each contact region. The position history provides a record of how each contact region moves or and/or changes shape over time,” and see, e.g., column 7, lines 46-55, “In step 208, the computer 126 determines whether activity of the current contact matches a predetermined pattern, and

Claim 12

The medium as in claim 8, wherein determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period.

Claim 19

The apparatus as in claim 15, wherein determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a

Application/Control Number: 90/012,332

Page 34

Art Unit: 3992

certain time period.

therefore constitutes a 'gesture.' Step 208 repeats continually, utilizing some or all of the position, position history (movement), velocity, and force information from steps 202, 204, 206. More particularly, in step 208 the computer 126 compares the history of contact position, size, movement, velocity, and/or force to the dictionary 126a of predetermined gestures to determine if the user has performed any of these gestures.”).

Claim 6

The method as in claim 1, further comprising:

responding to at least one gesture call, if issued, by rotating a view associated with the event object based on receiving a plurality of input points in the form of the user input.

Claim 13

The medium as in claim 8, further comprising:

responding to at least one gesture call, if issued, by rotating a view associated with the event object based on receiving a plurality of input points in the form of the user input.

Claim 20

The apparatus as in claim 15, further comprising:

means for responding [*under § 112, ¶ 6, the “window or view” described at column 12, lines 44-46*] to at least one gesture call, if issued, by rotating a view associated with the event object based on

Claims 6, 13 and 20

Hillis describes that the gesture operations include rotating, and further teaches responding to a call to perform the rotating operation by rotating the view associated with the event object based on the user input (see, e.g., column 7, line 65 to column 8, line 3, “As mentioned above, the predefined actions include various machine implemented operations for updating the presentation of imagery by the display. In one embodiment, gestures are both identified (208) and associated (214) with display control commands via a single procedure,” and see, e.g., column 8, lines 4-8 and 36-39, “After step 214, the computer 126 initiates performance of the identified action (step 216). As described in greater detail below, some examples of actions 126b include panning, zooming, rotating, and the like. Thus, step 216 starts the requested pan, zoom, rotate, or other operation. ... For example, if the action identified in step 214 was ‘rotate,’ then the computer 126 in step 222 directs the projector 128 to additionally continue the requested rotation after the gesture terminates.”). Hillis further teaches software equivalent to the claimed means (see, e.g., column 5, lines 52-55).

Application/Control Number: 90/012,332
 Art Unit: 3992

Page 35

receiving a plurality of input points in the form of the user input.

Claim 7

The method as in claim 1, wherein the device is one of: a data processing device, a portable device, a portable data processing device, a multi touch device, a multi touch portable device, a wireless device, and a cell phone.

Claim 14

The medium as in claim 8, wherein the data processing system is one of: a data processing device, a portable device, a portable data processing device, a multi touch device, a multi touch portable device, a wireless device, and a cell phone.

Claim 21

The apparatus as in claim 15, wherein the apparatus is one of: a data processing device, a portable device, a portable data processing device, a multi touch device, a multi touch portable device, a wireless device, and a cell phone.

Claims 7, 14 and 21

Hillis further teaches that the device, data processing system or apparatus is a data processing device and a multi-touch device (see, e.g., column 5, lines 17-21, “Data processing entities such as the computer 126 may be implemented in various forms. One example is a digital data processing apparatus, as exemplified by the hardware components and interconnections of the digital data processing apparatus 100 of FIG. 1D,” and see, e.g., column 6, lines 59-63, “Broadly, the steps 202, 204, 206 run continuously to process user contact with the display surface 124 as it occurs. Steps 202, 204, 206 therefore serve to analyze contact occurring when the user contacts the surface 124 at one or more contact regions utilizing one or more fingers, hands, arms, etc.”).

Claim Rejections under 35 U.S.C. § 103

24. The following is a quotation of 35 U.S.C. § 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said

Application/Control Number: 90/012,332

Page 36

Art Unit: 3992

subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

25. Ground 2: Claims 2, 9 and 16 are rejected under 35 U.S.C. § 103(a) as unpatentable over Hillis in view of Lira.

Claim 2

The method as in claim 1, further comprising:

rubberbanding a scrolling region displayed within the window by a predetermined maximum displacement when the scrolling region exceeds a window edge based on the scroll.

Claim 9

The medium as in claim 8, further comprising:

rubberbanding a scrolling region displayed within the window by a predetermined maximum displacement when the scrolled region exceeds a window edge based on the scroll.

Claim 16

The apparatus as in claim 15, further comprising:

means for rubberbanding [under § 112, ¶ 6, the “application programming interface” described at column 7, lines 48-51] a scrolling region displayed within the window by a predetermined maximum displacement when the scrolling region exceeds a window edge based on the scroll.

Claims 2, 9 and 16

Hillis does not explicitly describe rubberbanding a scrolling region displayed within the window by a predetermined maximum displacement when the scrolling region exceeds a window edge based on the scroll.

However, in an analogous art, Lira teaches receiving user input on a touch-sensitive display (see, e.g., page 16, lines 4-11). The display includes a window displaying a page or document that is separated into columns (see, e.g., page 11, lines 1-17). Lira further describes scrolling a column within the window based on the user input, and snapping (i.e., rubberbanding) the column according to a predetermined maximum threshold based on whether the scrolling exceeds the threshold (see, e.g., FIG. 14B and page 15, lines 18-31, “Referring to Fig. 14B, in another implementation, the vertical alignment control is enabled when the user lifts the pen 1200 from the display 1205. This causes the logical column 1220 to snap into alignment with the display window 1205 as the user stops scrolling. The user can adjust the snap sensitivity by, for example, setting the alignment control to snap to the nearest logical column based on a user-defined snap threshold. If the user’s scrolling does not exceed the threshold, which indicates an intention to continue to view the text column 1220, the display 1205 centers the logical column 1210 as the pen 1200 is lifted from the screen. If the user’s scrolling exceeds the threshold, which indicates an intention to move beyond the boundary of the logical column 1220, the display is snapped to the adjacent or repositioned

Application/Control Number: 90/012,332

Page 37

Art Unit: 3992

column. In other implementations, no snapping occurs when the user's scrolling exceeds the threshold. The snap-on-column feature can also be animated to provide an appearance of movement as the display scrolls to the correct column-viewing position."). Lira further teaches software equivalent to the claimed means (see, e.g., page 7, lines 9-10).

A person of ordinary skill in the art could have combined the teachings of Hillis and Lira with predictable results, and would have been prompted to implement the teachings of Hillis such that a scrolling region is automatically and visually "snapped" back into alignment within the window based on whether the scrolling exceeds a threshold, as Lira suggests (see, e.g., page 15, "The snap-on-column feature can also be animated to provide an appearance of movement as the display scrolls to the correct ... position."). Therefore, it would have been obvious to those of ordinary skill in the art at the time of the invention to rubberband a scrolling region displayed within the window by a predetermined maximum displacement when the scrolling region exceeds a window edge based on the scroll.

26. Ground 3: Claims 3, 4, 10, 11, 17 and 18 are rejected under 35 U.S.C. § 103(a) as unpatentable over Hillis in view of Makus.

Claim 3

The method as in claim 1, further comprising:

attaching scroll indicators to a content edge of the window.

Claim 10

The medium as in claim 8, further comprising:

Claims 3, 10 and 17

To the extent that Hillis does not explicitly describe attaching scroll indicators to a content edge of the window or view, Hillis does teach attaching a slider tool such as a slider bar (i.e., a scroll bar or scroll indicator) to the display (see, e.g., FIG. 5B and column 16, lines 29-39, "When slider mode is activated (step 537), the display 124 presents a slider tool in step 538. Broadly, the slider tool includes a bar, knob, button, dial, or other suitable GUI component. The presently described embodiment

Application/Control Number: 90/012,332

Page 38

Art Unit: 3992

attaching scroll indicators to a content edge of the view.

Claim 17

The apparatus as in claim 15, further comprising:

means for attaching [*under § 112, ¶ 6, the “application programming interface” described at column 11, lines 16-20*] scroll indicators to a content edge of the window.

utilizes a linearly movable slider bar 560 illustrated in FIG. 5B. In this example, each designated linear position of the slider bar corresponds to a different image layer of step 532. In other words, the slider bar is set up so that different positions of the slider bar correspond to different positions in the prescribed sequence of images. The slider bar observes an appropriately convenient scale.”).

In an analogous art, Makus teaches attaching a scroll bar 76 (i.e., a scroll indicator) to a content edge of a window or view (see, e.g., FIG. 5 and column 8, line 59 to column 9, line 1, “Any of the third-level categories can be selected by touching the subcategory on display 28 with a finger or stylus. For example, as shown in FIG. 5, the user has selected a third-level subcategory 72 entitled ‘Airlines – International,’ causing a list 74 of international airlines to be displayed. Since more international airlines are included within list 74 than can be displayed in the available space on display screen 28 at one time, a scroll bar 76 is included for selectively accessing other international airlines in list 74 that are not currently shown.”). Makus further teaches software equivalent to the claimed means (see, e.g., column 6, lines 24-25).

A person of ordinary skill in the art would have been prompted to attach scroll indicators to a content edge of the window or view in Hillis in order to illustrate, for example, the relative position of the view using “an appropriately convenient scale,” as Hillis suggests, or to provide access to more content than what fits on the screen at one time, as Makus suggests. Therefore, it would have been obvious to those of ordinary skill in the art at the time of the invention to attach scroll indicators to a content edge of the window or view.

Claim 4

The method as in claim 1, further comprising:

Claims 4, 11 and 18

To the extent that Hillis does not explicitly describe attaching scroll indicators to the window edge or a

Application/Control Number: 90/012,332

Page 39

Art Unit: 3992

attaching scroll indicators to the window edge.

Claim 11

The medium as in claim 8, further comprising:

attaching scroll indicators to a window edge of the view.

Claim 18

The apparatus as in claim 15, further comprising:

means for attaching *[under § 112, ¶ 6, the “application programming interface” described at column 11, lines 16-20]* scroll indicators to the window edge.

window edge of the view, Hillis does teach attaching a slider tool such as a slider bar (i.e., a scroll bar or scroll indicator) to the display (see, e.g., FIG. 5B and column 16, lines 29-39, “When slider mode is activated (step 537), the display 124 presents a slider tool in step 538. Broadly, the slider tool includes a bar, knob, button, dial, or other suitable GUI component. The presently described embodiment utilizes a linearly movable slider bar 560 illustrated in FIG. 5B. In this example, each designated linear position of the slider bar corresponds to a different image layer of step 532. In other words, the slider bar is set up so that different positions of the slider bar correspond to different positions in the prescribed sequence of images. The slider bar observes an appropriately convenient scale.”).

In an analogous art, Makus teaches attaching a scroll bar 76 (i.e., a scroll indicator) to a window edge (see, e.g., FIG. 5 and column 8, line 59 to column 9, line 1, “Any of the third-level categories can be selected by touching the subcategory on display 28 with a finger or stylus. For example, as shown in FIG. 5, the user has selected a third-level subcategory 72 entitled ‘Airlines – International,’ causing a list 74 of international airlines to be displayed. Since more international airlines are included within list 74 than can be displayed in the available space on display screen 28 at one time, a scroll bar 76 is included for selectively accessing other international airlines in list 74 that are not currently shown.”). Makus further teaches software equivalent to the claimed means (see, e.g., column 6, lines 24-25).

A person of ordinary skill in the art would have been prompted to attach scroll indicators to the window edge or a window edge of the view in Hillis in order to illustrate, for example, the relative position of the view using “an appropriately convenient scale,” as Hillis suggests, or to provide access to more content than what fits on the screen at one time, as Makus suggests. Therefore, it would have been obvious to those of ordinary skill in the art at the time of the

Application/Control Number: 90/012,332

Page 40

Art Unit: 3992

invention to attach scroll indicators to a content edge of the window or view.

27. Ground 4: Claims 1, 5-8, 12-15 and 19-21 are rejected under 35 U.S.C. § 103(a) as unpatentable over Nomura in view of Rubine.

Claim 1

A machine implemented method for scrolling on a touch-sensitive display of a device comprising:

receiving a user input, the user input is one or more input points applied to the touch-sensitive display that is integrated with the device;

creating an event object in response to the user input;

Nomura teaches a machine-implemented method for scrolling on a touch-sensitive display of a device (see, e.g., paragraphs [0008] and [0049]).

Nomura teaches receiving user input comprising one or more contact points applied to the touch-sensitive display integrated with the device (see, e.g., paragraph [0130], “When the display screen is contacted by the tip of a user’s finger or pen 1120 or the like, the touch panel 1060 detects contact position information, contact pressure, and contact area of the finger ...,” and see, e.g., paragraph [0139], “The finger movement detector 1110 is for detecting the movement history, contact pressure, and contact area of the finger performed on the display area by the user.”).

Nomura teaches providing data based on the user input (see, e.g., paragraph [0139], “Detection data obtained by the finger movement detector 1110 is input to the processor 1100.”), but does not explicitly describe creating an event object in response to the user input.

However, in an analogous art, Rubine teaches creating an event object in response to the user input (see, e.g., page 105, “When input occurs, it is represented as an event which is raised. Raising an event results in a search for an active event handler that will handle the event,” and pages 120-121, “Input events are full-blown objects. The Event hierarchy imposes structure on events without imposing device dependencies. ... Much flexibility

Application/Control Number: 90/012,332

Page 41

Art Unit: 3992

is possible; for example, a Sensor Frame device might raise a single SensorFrameEvent describing the current set of fingers in the plane of the frame, or separate DragEvents for each finger ...,” and see, e.g., page 133, “Raising the GestureEvent initiates the search for the possible gesture classes given the initial event. ... The GestureEvent, handled by the same passive event handler mechanism, will thus be propagated to other GestureEventHandlers in the correct order. Each passive gesture handler that would have handled the initial event sends a message to the gesture handler which raised the GestureEvent indicating the set of gesture classes it recognizes and the view with which it is associated.”).

Rubine describes, “A single idea motivated the author to use object-oriented toolkits to construct gesture-based systems: gestures should be associated with objects on the screen. Just as an object’s class determines the messages it understands, the author believed the class could and should be used to determine which gestures an object understands. The ideas of inheritance and overriding then naturally apply to gestures” (see, e.g., page 95).

Thus, a person of ordinary skill in the art could have combined the teachings of Nomura and Rubine with predictable results, and would have been prompted to represent the user input of Nomura in the form of an event object. As Rubine suggests, such an implementation would impose structure on the input event while allowing for flexibility (see, e.g., pages 120-121, “Input events are full-blown objects. The Event hierarchy imposes structure on events without imposing device dependencies. ... Much flexibility is possible”). Therefore, it would have been obvious to those of ordinary skill in the art at the time of the invention to implement the teachings of Nomura so as to create an event object in response to the user input.

determining whether the event object invokes a scroll or gesture operation by

Nomura teaches distinguishing between and determining whether the user input consists of a

Application/Control Number: 90/012,332

Page 42

Art Unit: 3992

distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation;

single contact point that is interpreted as a scroll operation or two or more contact points that are interpreted as a gesture operation such as zoom in, zoom out or rotate (see, e.g., paragraph [0053], “The operations details determination part 30 judges the operation details input by the user based on the finger movement history detected by the finger movement detector 10. Specifically, the operations details determination part 30 judges ... two fingers moving apart as input of a map image zoom-in operation ... two fingers moving toward each other as input of a map image zoom-out operation ... one finger rotating with another finger as an axis as input of a map image rotate operation ... [and] action of moving one finger as input of a map scroll operation.”).

issuing at least one scroll or gesture call based on invoking the scroll or gesture operation;

Nomura teaches issuing a call to perform the scroll, zoom in, zoom out or rotate operation (see, e.g., paragraph [0054], “The map operation processor 40 performs processing to generate a map image with the operation judged by the operation details determination part 30 implemented and includes a zoom-in processor 42, a zoom-out processor 44, a rotation processor 46, and a scroll processor 48,” and see, e.g., paragraph [0137], “When the user performs various types of finger operation[s] on the display ... the CPU 1010 executes the operation indicated by the user based on the input details.”).

responding to at least one scroll call, if issued, by scrolling a window having a view associated with the event object based on an amount of a scroll with the scroll stopped at a predetermined position in relation to the user input; and

Nomura teaches responding to the call by scrolling the view associated with the user input based on the amount of scroll (see, e.g., FIG. 8 and paragraph [0067], “As shown in Fig. 8, when a finger is placed on the screen and moved in the desired direction while pressing a finger on the screen (map scroll gesture), a map moved in the direction of movement of the finger and the same distance as the moved finger is displayed.”). Nomura further teaches that the scrolling is stopped at a predetermined position based on the user input (see, e.g., paragraph [0016], “With the present invention, the operation amount can be inputted for each type of operation based on the movement history of fingers; therefore, a user

Application/Control Number: 90/012,332

Page 43

Art Unit: 3992

responding to at least one gesture call, if issued, by scaling the view associated with the event object based on receiving the two or more input points in the form of the user input.

can input the desired quantity of operation by adjusting finger motion according to his needs.”).

Nomura teaches responding to the call by scaling the view associated with the user input based on the two or more contact points (see, e.g., FIG. 5 and paragraph [0063], “In the case that the obtaining of information near ‘Bombay’ is desired, as shown in Fig. 6 (B), the placement of a thumb and forefinger near ‘Bombay’ on the screen with the thumb and forefinger close together and the action of moving the thumb and forefinger away from each other (map zoom-in gesture) are performed (see Fig. 5). When performed in this manner, a zoomed-in map image corresponding to the movement history of the thumb and forefinger is displayed. In other words, an increased level of separation of the thumb and forefinger leads to a smaller scale and a more detailed map,” and see, e.g., FIG. 9 and paragraph [0068], “When desiring a zoomed-out map for a case such as to see a map image with larger scale, or to display information for a wide range or the like, first place a thumb and forefinger apart on the map displayed on the screen as shown in Fig. 6 (A), then move the thumb and forefinger toward each other as shown in Fig. 9 (map zoom-out gesture). When performed in this manner, a zoomed-out map image corresponding to the movement history of the thumb and forefinger is displayed. In other words, an increased level of separation of the thumb and forefinger leads to a smaller scale and a more detailed map.”).

Claim 8

A machine readable storage medium storing executable program instructions which when executed cause a data processing system to perform a method comprising:

receiving a user input, the user input is one or more input points applied to a

Nomura teaches a machine-readable storage medium storing instructions for causing a data processing system to perform a method (see, e.g., paragraphs [0008] and [0049], and see, e.g., paragraphs [0132] and [0134]).

Nomura teaches receiving user input comprising one or more contact points applied to a touch-sensitive

Application/Control Number: 90/012,332

Page 44

Art Unit: 3992

touch-sensitive display that is integrated with the data processing system;

display integrated with the data processing system (see, e.g., paragraph [0130], “When the display screen is contacted by the tip of a user’s finger or pen 1120 or the like, the touch panel 1060 detects contact position information, contact pressure, and contact area of the finger ...,” and see, e.g., paragraph [0139], “The finger movement detector 1110 is for detecting the movement history, contact pressure, and contact area of the finger performed on the display area by the user.”).

creating an event object in response to the user input;

Nomura teaches providing data based on the user input (see, e.g., paragraph [0139], “Detection data obtained by the finger movement detector 1110 is input to the processor 1100.”), but does not explicitly describe creating an event object in response to the user input.

However, in an analogous art, Rubine teaches creating an event object in response to user input (see, e.g., page 105, “When input occurs, it is represented as an event which is raised. Raising an event results in a search for an active event handler that will handle the event,” and pages 120-121, “Input events are full-blown objects. The Event hierarchy imposes structure on events without imposing device dependencies. ... Much flexibility is possible; for example, a Sensor Frame device might raise a single SensorFrameEvent describing the current set of fingers in the plane of the frame, or separate DragEvents for each finger ...,” and see, e.g., page 133, “Raising the GestureEvent initiates the search for the possible gesture classes given the initial event. ... The GestureEvent, handled by the same passive event handler mechanism, will thus be propagated to other GestureEventHandlers in the correct order. Each passive gesture handler that would have handled the initial event sends a message to the gesture handler which raised the GestureEvent indicating the set of gesture classes it recognizes and the view with which it is associated.”).

Rubine describes, “A single idea motivated the

Application/Control Number: 90/012,332

Page 45

Art Unit: 3992

author to use object-oriented toolkits to construct gesture-based systems: gestures should be associated with objects on the screen. Just as an object's class determines the messages it understands, the author believed the class could and should be used to determine which gestures an object understands. The ideas of inheritance and overriding then naturally apply to gestures" (see, e.g., page 95).

Thus, a person of ordinary skill in the art could have combined the teachings of Nomura and Rubine with predictable results, and would have been prompted to represent the user input of Nomura in the form of an event object. As Rubine suggests, such an implementation would impose structure on the input event while allowing for flexibility (see, e.g., pages 120-121, "Input events are full-blown objects. The Event hierarchy imposes structure on events without imposing device dependencies. ... Much flexibility is possible"). Therefore, it would have been obvious to those of ordinary skill in the art at the time of the invention to implement the teachings of Nomura so as to create an event object in response to the user input.

determining whether the event object invokes a scroll or gesture operation by distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation;

Nomura teaches distinguishing between and determining whether the user input consists of a single contact point that is interpreted as a scroll operation or two or more contact points that are interpreted as a gesture operation such as zoom in, zoom out or rotate (see, e.g., paragraph [0053], "The operations details determination part 30 judges the operation details input by the user based on the finger movement history detected by the finger movement detector 10. Specifically, the operations details determination part 30 judges ... two fingers moving apart as input of a map image zoom-in operation ... two fingers moving toward each other as input of a map image zoom-out operation ... one finger rotating with another finger as an axis as input of a map image rotate operation ... [and] action of moving one finger as input of a map scroll operation.").

Application/Control Number: 90/012,332

Page 46

Art Unit: 3992

issuing at least one scroll or gesture call based on invoking the scroll or gesture operation;

Nomura teaches issuing a call to perform the scroll, zoom in, zoom out or rotate operation (see, e.g., paragraph [0054], “The map operation processor 40 performs processing to generate a map image with the operation judged by the operation details determination part 30 implemented and includes a zoom-in processor 42, a zoom-out processor 44, a rotation processor 46, and a scroll processor 48,” and see, e.g., paragraph [0137], “When the user performs various types of finger operation[s] on the display ... the CPU 1010 executes the operation indicated by the user based on the input details.”).

responding to at least one scroll call, if issued, by scrolling a window having a view associated with the event object; and

Nomura teaches responding to the call by scrolling the view associated with the user input (see, e.g., FIG. 8 and paragraph [0067], “As shown in Fig. 8, when a finger is placed on the screen and moved in the desired direction while pressing a finger on the screen (map scroll gesture), a map moved in the direction of movement of the finger and the same distance as the moved finger is displayed.”).

responding to at least one gesture call, if issued, by scaling the view associated with the event object based on receiving the two or more input points in the form of the user input.

Nomura teaches responding to the call by scaling the view associated with the user input based on the two or more contact points (see, e.g., FIG. 5 and paragraph [0063], “In the case that the obtaining of information near ‘Bombay’ is desired, as shown in Fig. 6 (B), the placement of a thumb and forefinger near ‘Bombay’ on the screen with the thumb and forefinger close together and the action of moving the thumb and forefinger away from each other (map zoom-in gesture) are performed (see Fig. 5). When performed in this manner, a zoomed-in map image corresponding to the movement history of the thumb and forefinger is displayed. In other words, an increased level of separation of the thumb and forefinger leads to a smaller scale and a more detailed map,” and see, e.g., FIG. 9 and paragraph [0068], “When desiring a zoomed-out map for a case such as to see a map image with larger scale, or to display information for a wide range or the like, first place a thumb and forefinger apart on the map displayed on the screen as shown in Fig. 6 (A), then move the thumb and forefinger toward each other as

Application/Control Number: 90/012,332

Page 47

Art Unit: 3992

shown in Fig. 9 (map zoom-out gesture). When performed in this manner, a zoomed-out map image corresponding to the movement history of the thumb and forefinger is displayed. In other words, an increased level of separation of the thumb and forefinger leads to a smaller scale and a more detailed map.”).

Claim 15

An apparatus, comprising:

Nomura teaches an apparatus (see, e.g., paragraphs [0008] and [0049], and see, e.g., paragraphs [0132] and [0134]).

means for receiving [*under § 112, ¶ 6, the “multi-touch driver” described at column 12, lines 30-32*], through a hardware device, a user input on a touch-sensitive display of the apparatus, the user input is one or more input points applied to the touch-sensitive display that is integrated with the apparatus;

Nomura teaches equivalent software (see, e.g., paragraph [0134]) for receiving user input comprising one or more contact points applied to a touch-sensitive display integrated with the apparatus (see, e.g., paragraph [0130], “When the display screen is contacted by the tip of a user’s finger or pen 1120 or the like, the touch panel 1060 detects contact position information, contact pressure, and contact area of the finger ...,” and see, e.g., paragraph [0139], “The finger movement detector 1110 is for detecting the movement history, contact pressure, and contact area of the finger performed on the display area by the user.”).

means for creating [*under § 112, ¶ 6, the “multi-touch driver” described at column 12, lines 30-32*] an event object in response to the user input;

Nomura teaches equivalent software (see, e.g., paragraph [0134]) for providing data based on the user input (see, e.g., paragraph [0139], “Detection data obtained by the finger movement detector 1110 is input to the processor 1100.”), but does not explicitly describe creating an event object in response to the user input.

However, in an analogous art, Rubine teaches creating an event object in response to user input (see, e.g., page 105, “When input occurs, it is represented as an event which is raised. Raising an event results in a search for an active event handler that will handle the event,” and pages 120-121, “Input events are full-blown objects. The Event

Application/Control Number: 90/012,332

Page 48

Art Unit: 3992

hierarchy imposes structure on events without imposing device dependencies. ... Much flexibility is possible; for example, a SensorFrame device might raise a single SensorFrameEvent describing the current set of fingers in the plane of the frame, or separate DragEvents for each finger ...,” and see, e.g., page 133, “Raising the GestureEvent initiates the search for the possible gesture classes given the initial event. ... The GestureEvent, handled by the same passive event handler mechanism, will thus be propagated to other GestureEventHandlers in the correct order. Each passive gesture handler that would have handled the initial event sends a message to the gesture handler which raised the GestureEvent indicating the set of gesture classes it recognizes and the view with which it is associated.”).

Rubine describes, “A single idea motivated the author to use object-oriented toolkits to construct gesture-based systems: gestures should be associated with objects on the screen. Just as an object’s class determines the messages it understands, the author believed the class could and should be used to determine which gestures an object understands. The ideas of inheritance and overriding then naturally apply to gestures” (see, e.g., page 95).

Thus, a person of ordinary skill in the art could have combined the teachings of Nomura and Rubine with predictable results, and would have been prompted to represent the user input of Nomura in the form of an event object. As Rubine suggests, such an implementation would impose structure on the input event while allowing for flexibility (see, e.g., pages 120-121, “Input events are full-blown objects. The Event hierarchy imposes structure on events without imposing device dependencies. ... Much flexibility is possible”). Therefore, it would have been obvious to those of ordinary skill in the art at the time of the invention to implement the teachings of Nomura so as to create an event object in response to the user input.

Application/Control Number: 90/012,332

Page 49

Art Unit: 3992

means for determining *[under § 112, ¶ 6, the “window server” described at column 12, lines 32-34]* whether the event object invokes a scroll or gesture operation by distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation;

Nomura teaches equivalent software (see, e.g., paragraph [0134]) for distinguishing between and determining whether the user input consists of a single contact point that is interpreted as a scroll operation or two or more contact points that are interpreted as a gesture operation such as zoom in, zoom out or rotate (see, e.g., paragraph [0053], “The operations details determination part 30 judges the operation details input by the user based on the finger movement history detected by the finger movement detector 10. Specifically, the operations details determination part 30 judges ... two fingers moving apart as input of a map image zoom-in operation ... two fingers moving toward each other as input of a map image zoom-out operation ... one finger rotating with another finger as an axis as input of a map image rotate operation ... [and] action of moving one finger as input of a map scroll operation.”).

means for issuing *[under § 112, ¶ 6, the “user interface software” described at column 12, lines 34-37]* at least one scroll or gesture call based on invoking the scroll or gesture operation;

Nomura teaches equivalent software (see, e.g., paragraph [0134]) for issuing a call to perform the scroll, zoom in, zoom out or rotate operation (see, e.g., paragraph [0054], “The map operation processor 40 performs processing to generate a map image with the operation judged by the operation details determination part 30 implemented and includes a zoom-in processor 42, a zoom-out processor 44, a rotation processor 46, and a scroll processor 48,” and see, e.g., paragraph [0137], “When the user performs various types of finger operation[s] on the display ... the CPU 1010 executes the operation indicated by the user based on the input details.”).

means for responding *[under § 112, ¶ 6, the “window or view” described at column 12, lines 44-46]* to at least one scroll call, if issued, by scrolling a window having a view associated with the event object; and

Nomura teaches equivalent software (see, e.g., paragraph [0134]) for responding to the call by scrolling the view associated with the user input (see, e.g., FIG. 8 and paragraph [0067], “As shown in Fig. 8, when a finger is placed on the screen and moved in the desired direction while pressing a finger on the screen (map scroll gesture), a map moved in the direction of movement of the finger and the same distance as the moved finger is

Application/Control Number: 90/012,332
 Art Unit: 3992

Page 50

means for responding *[under § 112, ¶ 6, the “window or view” described at column 12, lines 44-46]* to at least one gesture call, if issued, by scaling the view associated with the event object based on receiving the two or more input points in the form of the user input.

displayed.”).

Nomura teaches equivalent software (see, e.g., paragraph [0134]) for responding to the call by scaling the view associated with the user input based on the two or more contact points (see, e.g., FIG. 5 and paragraph [0063], “In the case that the obtaining of information near ‘Bombay’ is desired, as shown in Fig. 6 (B), the placement of a thumb and forefinger near ‘Bombay’ on the screen with the thumb and forefinger close together and the action of moving the thumb and forefinger away from each other (map zoom-in gesture) are performed (see Fig. 5). When performed in this manner, a zoomed-in map image corresponding to the movement history of the thumb and forefinger is displayed. In other words, an increased level of separation of the thumb and forefinger leads to a smaller scale and a more detailed map,” and see, e.g., FIG. 9 and paragraph [0068], “When desiring a zoomed-out map for a case such as to see a map image with larger scale, or to display information for a wide range or the like, first place a thumb and forefinger apart on the map displayed on the screen as shown in Fig. 6 (A), then move the thumb and forefinger toward each other as shown in Fig. 9 (map zoom-out gesture). When performed in this manner, a zoomed-out map image corresponding to the movement history of the thumb and forefinger is displayed. In other words, an increased level of separation of the thumb and forefinger leads to a smaller scale and a more detailed map.”).

Claim 5

The method as in claim 1, wherein determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period.

Claim 12

Claims 5, 12 and 19

Nomura further teaches that the determining is based on receiving a drag input for a period of time (see, e.g., paragraph [0009], “Using the present invention, the user can input at least one operation selected from rotate, zoom-in, zoom-out, and scroll of a map image displayed in the display area through the movement history of his fingers,” and see, e.g., paragraph [0010], “The movement history of fingers

Application/Control Number: 90/012,332

Page 51

Art Unit: 3992

The medium as in claim 8, wherein determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period.

contacting the display area is a concept including a passage of time element and is distinguished from an operation of simply touching an input mark or the like with a finger that does not include a passage of time element.”).

Claim 19

The apparatus as in claim 15, wherein determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period.

Claim 6

The method as in claim 1, further comprising:

responding to at least one gesture call, if issued, by rotating a view associated with the event object based on receiving a plurality of input points in the form of the user input.

Claim 13

The medium as in claim 8, further comprising:

responding to at least one gesture call, if issued, by rotating a view associated with the event object based on receiving a plurality of input points in the form of the user input.

Claim 20

The apparatus as in claim 15, further comprising:

means for responding *[under § 112, ¶ 6, the “window or view” described at*

Claims 6, 13 and 20

Nomura further teaches responding to the call by rotating the view associated with the user input based on the two or more contact points (see, e.g., FIG. 10 and paragraph [0072], “If change of orientation of a map without changing scale is desired, first place a thumb and forefinger apart on the map displayed on the screen as shown in Fig. 6 (A), then hold either the thumb or forefinger in one place and rotate the other finger with the other finger as an axis (map rotation gesture) (see Fig. 10). When performed in this manner, a map image with rotation corresponding to the movement history of the thumb and forefinger is displayed.”). Nomura further teaches software equivalent to the claimed means (see, e.g., paragraph [0134]).

Application/Control Number: 90/012,332

Page 52

Art Unit: 3992

column 12, lines 44-46] to at least one gesture call, if issued, by rotating a view associated with the event object based on receiving a plurality of input points in the form of the user input.

Claim 7

The method as in claim 1, wherein the device is one of: a data processing device, a portable device, a portable data processing device, a multi touch device, a multi touch portable device, a wireless device, and a cell phone.

Claim 14

The medium as in claim 8, wherein the data processing system is one of: a data processing device, a portable device, a portable data processing device, a multi touch device, a multi touch portable device, a wireless device, and a cell phone.

Claim 21

The apparatus as in claim 15, wherein the apparatus is one of: a data processing device, a portable device, a portable data processing device, a multi touch device, a multi touch portable device, a wireless device, and a cell phone.

Claims 7, 14 and 21

Nomura further teaches that the device, data processing system or apparatus is a portable data processing device and a multi-touch device (see, e.g., paragraph [0047], “A characteristic of the present invention is performing rotate, zoom-in, zoom-out, or scrolling of a map image through finger movement on a map image displayed on a mobile information device or electronic book that can display a map image.”).

28. Ground 5: Claims 2, 9 and 16 are rejected under 35 U.S.C. § 103(a) as unpatentable over

Nomura in view of Rubine and further in view of Lira.

Claim 2

The method as in claim 1, further

Claims 2, 9 and 16

Nomura does not explicitly describe rubberbanding a

Application/Control Number: 90/012,332

Page 53

Art Unit: 3992

comprising:

rubberbanding a scrolling region displayed within the window by a predetermined maximum displacement when the scrolling region exceeds a window edge based on the scroll.

Claim 9

The medium as in claim 8, further comprising:

rubberbanding a scrolling region displayed within the window by a predetermined maximum displacement when the scrolled region exceeds a window edge based on the scroll.

Claim 16

The apparatus as in claim 15, further comprising:

means for rubberbanding [under § 112, ¶ 6, the “application programming interface” described at column 7, lines 48-51] a scrolling region displayed within the window by a predetermined maximum displacement when the scrolling region exceeds a window edge based on the scroll.

scrolling region displayed within the window by a predetermined maximum displacement when the scrolling region exceeds a window edge based on the scroll.

However, in an analogous art, Lira teaches receiving user input on a touch-sensitive display (see, e.g., page 16, lines 4-11). The display includes a window displaying a page or document that is separated into columns (see, e.g., page 11, lines 1-17). Lira further describes scrolling a column within the window based on the user input, and snapping (i.e., rubberbanding) the column according to a predetermined maximum threshold based on whether the scrolling exceeds the threshold (see, e.g., FIG. 14B and page 15, lines 18-31, “Referring to Fig. 14B, in another implementation, the vertical alignment control is enabled when the user lifts the pen 1200 from the display 1205. This causes the logical column 1220 to snap into alignment with the display window 1205 as the user stops scrolling. The user can adjust the snap sensitivity by, for example, setting the alignment control to snap to the nearest logical column based on a user-defined snap threshold. If the user’s scrolling does not exceed the threshold, which indicates an intention to continue to view the text column 1220, the display 1205 centers the logical column 1210 as the pen 1200 is lifted from the screen. If the user’s scrolling exceeds the threshold, which indicates an intention to move beyond the boundary of the logical column 1220, the display is snapped to the adjacent or repositioned column. In other implementations, no snapping occurs when the user’s scrolling exceeds the threshold. The snap-on-column feature can also be animated to provide an appearance of movement as the display scrolls to the correct column-viewing position.”). Lira further teaches software equivalent to the claimed means (see, e.g., page 7, lines 9-10).

A person of ordinary skill in the art could have combined the teachings of Nomura and Lira with predictable results, and would have been prompted to implement the teachings of Nomura such that a

Application/Control Number: 90/012,332
 Art Unit: 3992

Page 54

scrolling region is automatically and visually “snapped” back into alignment within the window based on whether the scrolling exceeds a threshold, as Lira suggests (see, e.g., page 15, “The snap-on-column feature can also be animated to provide an appearance of movement as the display scrolls to the correct ... position.”). Therefore, it would have been obvious to those of ordinary skill in the art at the time of the invention to rubberband a scrolling region displayed within the window by a predetermined maximum displacement when the scrolling region exceeds a window edge based on the scroll.

29. Ground 6: Claims 3, 4, 10, 11, 17 and 18 are rejected under 35 U.S.C. § 103(a) as unpatentable over Nomura in view of Rubine and further in view of Makus.

Claim 3

The method as in claim 1, further comprising:

attaching scroll indicators to a content edge of the window.

Claim 10

The medium as in claim 8, further comprising:

attaching scroll indicators to a content edge of the view.

Claim 17

The apparatus as in claim 15, further comprising:

means for attaching [*under § 112, ¶ 6, the “application programming interface” described at column 11, lines 16-20*]

Claims 3, 10 and 17

To the extent that Nomura does not explicitly describe attaching scroll indicators to a content edge of the window, Nomura does teach attaching controls to edge of the display for scrolling or “turning” pages of the content (see, e.g., Figure 3 and paragraph [0062], “The center of the screen is primarily used as an information display area and the edges of the screen are primarily used for search tag and toolbar display areas. ... Furthermore, images 240 and 242 that simulate the ‘thickness of a book’ are displayed on the right and left edges of the display area,” and see, e.g., paragraph [0089], “Here, the page-turning speed can be adjusted by adjustment of the user’s touch pressure. ... When the user finds the page he wants to read and removes his hand from the aforementioned ‘thickness of book,’ page-turning is stopped and the display screen displays this screen.”).

In an analogous art, Makus teaches attaching a scroll bar 76 (i.e., a scroll indicator) to a content edge of a window or view (see, e.g., FIG. 5 and column 8, line

Application/Control Number: 90/012,332

Page 55

Art Unit: 3992

scroll indicators to a content edge of the window.

59 to column 9, line 1, “Any of the third-level categories can be selected by touching the subcategory on display 28 with a finger or stylus. For example, as shown in FIG. 5, the user has selected a third-level subcategory 72 entitled ‘Airlines – International,’ causing a list 74 of international airlines to be displayed. Since more international airlines are included within list 74 than can be displayed in the available space on display screen 28 at one time, a scroll bar 76 is included for selectively accessing other international airlines in list 74 that are not currently shown.”). Makus further teaches software equivalent to the claimed means (see, e.g., column 6, lines 24-25).

A person of ordinary skill in the art would have been prompted to attach scroll indicators to a content edge of the window or view in Nomura in order to illustrate, for example, the relative position of the view or to provide access to more content than what fits on the screen at one time, as Makus suggests. Therefore, it would have been obvious to those of ordinary skill in the art at the time of the invention to attach scroll indicators to a content edge of the window or view.

Claim 4

The method as in claim 1, further comprising:

attaching scroll indicators to the window edge.

Claim 11

The medium as in claim 8, further comprising:

attaching scroll indicators to a window edge of the view.

Claim 18

Claims 4, 11 and 18

To the extent that Nomura does not explicitly describe attaching scroll indicators to the window edge or a window edge of the view, Nomura does teach attaching controls to edge of the display for scrolling or “turning” pages of the content (see, e.g., Figure 3 and paragraph [0062], “The center of the screen is primarily used as an information display area and the edges of the screen are primarily used for search tag and toolbar display areas. ... Furthermore, images 240 and 242 that simulate the ‘thickness of a book’ are displayed on the right and left edges of the display area,” and see, e.g., paragraph [0089], “Here, the page-turning speed can be adjusted by adjustment of the user’s touch pressure. ... When the user finds the page he wants

Application/Control Number: 90/012,332

Page 56

Art Unit: 3992

The apparatus as in claim 15, further comprising:

means for attaching *[under § 112, ¶ 6, the “application programming interface” described at column 11, lines 16-20]* scroll indicators to the window edge.

to read and removes his hand from the aforementioned ‘thickness of book,’ page-turning is stopped and the display screen displays this screen.”).

In an analogous art, Makus teaches attaching a scroll bar 76 (i.e., a scroll indicator) to a window edge (see, e.g., FIG. 5 and column 8, line 59 to column 9, line 1, “Any of the third-level categories can be selected by touching the subcategory on display 28 with a finger or stylus. For example, as shown in FIG. 5, the user has selected a third-level subcategory 72 entitled ‘Airlines – International,’ causing a list 74 of international airlines to be displayed. Since more international airlines are included within list 74 than can be displayed in the available space on display screen 28 at one time, a scroll bar 76 is included for selectively accessing other international airlines in list 74 that are not currently shown.”). Makus further teaches software equivalent to the claimed means (see, e.g., column 6, lines 24-25).

A person of ordinary skill in the art would have been prompted to attach scroll indicators to the window edge or a window edge of the view in Nomura in order to illustrate, for example, the relative position of the view or to provide access to more content than what fits on the screen at one time, as Makus suggests. Therefore, it would have been obvious to those of ordinary skill in the art at the time of the invention to attach scroll indicators to a content edge of the window or view.

Conclusion

30. THIS ACTION IS MADE FINAL. A shortened statutory period for response to this action is set to expire TWO (2) MONTHS from the mailing date of this action.

Extensions of time under 37 CFR 1.136(a) do not apply in reexamination proceedings.

The provisions of 37 CFR 1.136 apply only to “an applicant” and not to any party in a

Application/Control Number: 90/012,332

Page 57

Art Unit: 3992

reexamination proceeding. Further, 35 U.S.C. § 305 and 37 CFR 1.550(a) require that these proceedings “will be conducted with special dispatch within the Office.”

Extensions of time in reexamination proceedings are provided for in 37 CFR 1.550(c). A request for extension of time must be filed on or before the day on which a response to this action is due, and it must be accompanied by the petition fee set forth in 37 CFR 1.17(g). The mere filing of a request will not effect any extension of time. An extension of time will be granted only for sufficient cause, and for a reasonable time specified.

The filing of a timely first response to this final action will be construed as including a request to extend the shortened statutory period for an additional month, which will be granted even if previous extensions have been granted. In no event, however, will the statutory period for response expire later than SIX (6) MONTHS from the mailing date of the final action. See MPEP § 2265.

31. All correspondence relating to this *ex parte* reexamination proceeding should be directed:

By mail to: Mail Stop *Ex Parte* Reexam
Attn: Central Reexamination Unit
Commissioner for Patents
United States Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

By fax to: (571) 273-9900
Central Reexamination Unit

By hand: Customer Service Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

By EFS: Registered users may submit correspondence via the EFS-Web electronic filing system at <https://efs.uspto.gov/efile/myportal/efs-registered>.

Application/Control Number: 90/012,332
Art Unit: 3992

Page 58


Any inquiry concerning this communication should be directed to the Central
Reexamination Unit at telephone number (571) 272-7705.

/Michael J. Yigdall/
Primary Examiner, Art Unit 3992

Conferees:

/Stephen J Ralis/
Primary Examiner, Art Unit 3992

/Sudhanshu C Pathak/
Supervisory Patent Examiner, Art Unit 3992

Reexamination 	Application/Control No. 90012332	Applicant(s)/Patent Under Reexamination 7844915
	Certificate Date	Certificate Number

Requester Correspondence Address: <input type="checkbox"/> Patent Owner <input checked="" type="checkbox"/> Third Party
BRYAN CAVE LLP 1290 AVENUE OF THE AMERICAS NEW YORK, NY 10104

LITIGATION REVIEW <input checked="" type="checkbox"/>	MY (examiner initials)	06/04/2012 (date)
Case Name	Director Initials	
1:11cv611 (Open)		
5:11cv1846 (Open)		

COPENDING OFFICE PROCEEDINGS	
TYPE OF PROCEEDING	NUMBER
1. (No copending Office proceedings)	

--	--