

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner

v.

SOFTVIEW LLC,
Patent Owner

Patent No. 10,083,154

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 10,083,154**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	MANDATORY NOTICES	1
III.	PAYMENT OF FEES UNDER 37 C.F.R. §§ 42.15 AND 42.103	5
IV.	GROUND FOR STANDING.....	6
V.	PRECISE RELIEF REQUESTED AND GROUNDS RAISED.....	6
VI.	LEVEL OF ORDINARY SKILL	7
VII.	OVERVIEW OF THE '154 PATENT	8
VIII.	RELATED PATENTS	9
	A. The '926 patent.....	9
	B. The '353 patent.....	9
IX.	THE CHALLENGED CLAIMS SHOULD BE CANCELED BECAUSE THEY ARE PATENTABLY INDISTINCT FROM UNPATENTABLE CLAIMS.....	10
X.	CLAIM CONSTRUCTION	17
XI.	DETAILED EXPLANATION OF GROUNDS.....	18
	A. Ground 1: Zaurus, Pad, and SVG Render Claims 1-35 Obvious.....	18
	1. Zaurus (EX1004).....	18
	2. Pad (EX1005 and EX1007)	19
	3. SVG (EX1008).....	19
	4. Motivation to Combine	19
	5. Reasonable Expectation of Success	20
	6. Claim 1	21
	7. Claim 2: “The mobile hand-held device of claim 1, wherein the mobile hand-held device is a mobile phone.”	55
	8. Claim 3: “The mobile hand-held device of claim 1, wherein the mobile hand-held device is a hand-held computer.”	56

9.	Claim 4: “The mobile hand-held device of claim 1, wherein execution of the instructions further enables the user to zoom in on a user-selectable portion of a view of the HTML document in response to a user interface input made via the touch-sensitive display.”	56
10.	Claim 5: “The mobile hand-held device of claim 1, wherein the views of the HTML document are rendered in real-time to effect zooming operations.”	57
11.	Claim 6: “The mobile hand-held device of claim 5, wherein the zooming operations are effected in a fraction of a second.”	57
12.	Claim 7	58
13.	Claim 8: “The mobile hand-held device of claim 1, wherein the HTML document is an HTML parent document for a Web page.”	59
14.	Claim 9: “The mobile hand-held device of claim 1, wherein the HTML elements includes an HTML table having a plurality of cells, each cell having at least one of associated text content and associated image content, and the CSS code includes CSS code defining how the table and its associated content is to be rendered.”	59
15.	Claim 10	60
16.	Claim 11	63
17.	Claim 12: “The mobile hand-held device of claim 1, wherein the primary datum is an origin in a two-dimensional XY coordinate system and the vectors from the primary datum to the object datums for the HTML object bounding boxes comprise XY coordinates in the coordinate system.”	64
18.	Claims 13-35	66
B.	Ground 2: Zaurus, Pad, SVG, and SVF Render Claims 1-35	
	Obvious	68
1.	SVF	68
2.	Motivation to Combine	69
3.	Reasonable Expectation of Success	70

4. Claims 1-35	71
XII. DISCRETIONARY DENIAL IS NOT APPROPRIATE	73
XIII. CONCLUSION.....	74

LIST OF EXHIBITS

EX1001	U.S. Patent No. 10,083,154 to Rohrabough et al. (“the ’154 patent”)
EX1002	Declaration of Benjamin Goldberg, Ph.D
EX1003	Prosecution History of U.S. Patent App. No. 15/356,456
EX1004	Power Zaurus Specifications: User Manual - Mobile Business Tool - Model MI-106 / MI-106M / MI-110M (Nov. 1997) (“Zaurus”)
EX1005	Pad++ Reference Manual Version 0.2.7 (Oct. 2, 1996) (“Pad_1”)
EX1006	Curriculum Vitae of Benjamin Goldberg, Ph.D
EX1007	Pad++ Tour (Apr. 1997) (“Pad_2”)
EX1008	Ferraiolo, Jon, <i>Scalable Vector Graphics Requirements: W3C Working Draft</i> (Oct. 29, 1998) (“SVG”)
EX1009	Ex. A to Pl.’s P.R. 4-2 Disclosures, <i>SoftView LLC v. Samsung Elecs. Co.</i> , No. 2:25-cv-00246-JRG (E.D. Tex. Feb. 11, 2026).
EX1010	Pl. SoftView LLC’s P.R. 3-1 & 3-2 Disclosure of Asserted Claims & Infringement Contentions, <i>SoftView, LLC v. Samsung Elecs. Co.</i> , No. 2:25-cv-00246 (E.D. Tex. June 11, 2025).
EX1011	U.S. Patent No. 7,831,926 to Rohrabough et al. (“the ’926 patent”)
EX1012	RESERVED
EX1013	Final Written Decision, <i>Kyocera Corporation v. SoftView LLC</i> , IPR2013-00004, Paper 53 (P.T.A.B. March 27, 2014), <i>aff’d</i> , 592 F. App’x 949 (Fed. Cir. 2015)
EX1014	Decision <i>Sua Sponte</i> Terminating Reexamination Proceeding, <i>Motorola Mobility LLC. v. SoftView LLC</i> , Reexamination Control No. 95/002,126 (Feb. 10, 2025).
EX1015	U.S. Patent No. 7,461,353 to Rohrabough et al. (“the ’353 patent”)
EX1016	RESERVED

Petition for *Inter Partes* Review
Patent No. 10,083,154

EX1017	Final Written Decision, <i>Kyocera Corporation v. SoftView LLC</i> , IPR2013-00007, Paper 51 (P.T.A.B. March 27, 2014), <i>aff'd</i> , 592 F. App'x 947 (Fed. Cir. 2015)
EX1018	RESERVED
EX1019	<i>SoftView LLC v. Apple Inc.</i> , 108 F.4th 1366 (Fed. Cir. 2024)
EX1020	Board Decision, <i>Apple Inc. v. SoftView LLC</i> , Appeals 2021-005530, Reexamination Control Nos. 95/000,634 (Feb. 26, 2025)
EX1021	Board Decision, <i>Apple Inc. v. SoftView LLC</i> , Appeals 2022-000153, Reexamination Control Nos. 95/002,132 (Feb. 26, 2025)
EX1022	RESERVED
EX1023	Letter from Steven L. Caponi, Counsel for SoftView LLC, to Judge Leonard P. Stark, U.S. Dist. Ct. for the Dist. of Del., <i>SoftView LLC v. Apple Inc.</i> , No. 1:10-cv-00389-LPS (D. Del. Jan. 18, 2013), ECF No. 612.
EX1024	Board Decision, <i>Apple Inc. v. SoftView LLC</i> , Appeal 2021-005530, Reexamination Control No. 95/000,634 (Apr. 6, 2022)
EX1025	Board Decision, <i>Motorola Mobility LLC. v. SoftView LLC</i> , Appeal 2022-000153, Reexamination Control No. 95/002,132 (Apr. 6, 2022)
EX1026	RESERVED
EX1027	RESERVED
EX1028	U.S. Patent No. 9,519,729 to Rohrabough et al. (“the ’729 patent”)
EX1029	Terminal Disclaimer of the ’729 patent
EX1030	U.S. Patent No. 8,533,628 to Rohrabough et al. (“the ’628 patent”)
EX1031	RESERVED
EX1032	RESERVED

EX1033	<i>Ex Parte</i> Reexamination Control No. 90/009,995
EX1034	<i>Apple Inc. v. SoftView LLC</i> , Reexamination Control No. 95/000,635
EX1035	World Wide Web Consortium (W3C) Working Draft, <i>Level 1 Document Object Model Specification</i> , W3C (July 20, 1998)
EX1036	Lisa Rein, <i>Structuring Graphical Data</i> , XML.com (June 22, 1998)
EX1037	Definition of Cascading Style Sheets (CSS) from W3C Proposed Recommendation, <i>Cascading Style Sheets, Level 2</i> , W3C (Mar. 24, 1998)
EX1038	Markman Order, <i>SoftView LLC v. Apple Inc. et al.</i> , No. 1:10-cv-00389 (D. Del. Sept. 4, 2013)
EX1039	Transcript of Oral Hearing, <i>Kyocera Corp. v. Softview, Inc.</i> , No. IPR2013-00004 & IPR2013-00007, Paper 52 (P.T.A.B. Mar. 18, 2014)
EX1040	W3C, <i>11 Tables</i> , W3C (Wayback Machine Capture on Aug. 16, 2000)
EX1041	W3C, <i>16 Frames</i> , W3C (Wayback Machine Capture on Aug. 16, 2000)
EX1042	W3C, <i>Paragraphs</i> , W3C (Wayback Machine Capture on June 6, 1997)
EX1043	W3C, <i>HTML 3.2 Reference Specification publication history</i> , W3C (Jan. 14, 1997)
EX1044	Specification for the Simple Vector Format v.1.1 (Wayback Machine Capture on Oct. 19, 1996) (“SVF”)
EX1045	SoftSource, <i>Changes to SVF</i> , SoftSource (Wayback Machine Capture on July 13, 2002)
EX1046	W3C Recommendation, <i>Extensible Markup Language (XML) 1.0</i> , W3C (dated Feb. 10, 1998; Wayback Machine Capture dated Feb. 13, 1998)

EX1047	Right of Appeal Notice, <i>Apple Inc. v. SoftView LLC</i> , Reexamination Control No. 95/000,634 (Dec. 16, 2025)
EX1048	Right of Appeal Notice, <i>Motorola Mobility LLC. v. SoftView LLC</i> , Reexamination Control No. 95/002,132 (Dec. 16, 2025)
EX1049	Amended Joint Claim Construction Chart, <i>SoftView LLC v. Apple Inc.</i> , No. 1:10-cv-00389-LPS (D. Del. Oct. 22, 2012), ECF No. 496.
EX1050	<i>Marc Andreessen</i> , BRITANNICA, https://www.britannica.com/biography/Marc-Andreessen (last visited Jan. 27, 2026)
EX1051	<i>History of the Mozilla Project</i> , MOZILLA, https://www.mozilla.org/en-US/about/history/ (last visited Jan. 27, 2026)
EX1052	OpenGL Programming Guide, Second Edition, Addison-Wesley, 1997 (available at https://laplace.physics.ubc.ca/People/ajpenner/GPGPU/redbook.pdf)
EX1053	RESERVED
EX1054	RESERVED
EX1055	RESERVED
EX1056	RESERVED
EX1057	Defs.’ Prelim. Cl. Constr. & Extrinsic Evid. Pursuant to Loc. Pat. R. 4-2, <i>SoftView LLC v. Samsung Elecs. Co.</i> , No. 2:25-cv-00246-JRG (E.D. Tex. Feb. 11, 2026)

I. INTRODUCTION

Google LLC (“Petitioner”) respectfully requests *Inter Partes* Review of claims 1-35 (“the challenged claims”) of U.S. Patent No. 10,083,154 (“the ’154 patent”) (EX1001) assigned to SoftView LLC (“Patent Owner”). The ’154 patent never should have issued—its claims are all patentably indistinct from claims that have already been adjudicated by the USPTO to be invalid. Yet Patent Owner has begun to assert the ’154 patent and other patents containing patentably indistinct claims against Petitioner’s product (but not Petitioner), seeking to create protracted and complicated litigation. This Petition should thus be instituted to correct the Examiner’s error in allowing the ’154 patent.

II. MANDATORY NOTICES

Real Parties-in-Interest: Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioner identifies Google LLC¹, Samsung Electronics Company, Ltd. and Samsung Electronics America, Inc. as the real parties-in-interest.

Related Matters: The ’154 patent is asserted in *SoftView, LLC v. Samsung Electronics Co., Ltd. et al.*, No. 2-25-cv-00246 (E.D. Tex.).

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

The '154 patent originates from U.S. Patent Application No. 15/356,456, filed on November 18, 2016, which is a continuation of application No. 12/941,106, filed on November 8, 2010, now Patent No. 9,519,729 (“the '729 patent”), which is a continuation of application No. 11/738,486, filed on Apr. 21, 2007, now Pat. No. 7,831,926 (“the '926 patent”), which is a continuation of application No. 09/878,097, filed on June 8, 2001, now Pat. No. 7,210,099 (“the '099 patent”), which is a continuation-in-part of application No. 09/828,511, filed on April 7, 2001, which claims priority to provisional application No. 60/217,345, filed on July 11, 2000, and provisional application No. 60/211,019, filed on June 12, 2000. (EX1001, Cover.)

The '729 patent as well as related U.S. Patent Nos. 8,533,628 (“the '628 patent”), 7,844,889 (“the '889 patent”), and 7,461,353 (“the '353 patent”) are all also at issue in *SoftView, LLC v. Samsung Electronics Co., Ltd. et al.*, No. 2-25-cv-00246 (E.D. Tex.).

All 88 issued claims of the '926 patent have been canceled in post-grant proceedings. (EX1011, 41-48; EX1014.) Claims 30, 31, 40, 41, 43, 52, 55, 59, 72, and 75 of the '926 patent were canceled in IPR2013-00004, which was joined by IPR2013-00257. (EX1011, 41-42.) Claims 1-28, 32-39, 42, 44-51, and 69 of the '926 patent were canceled in *Ex Parte* Reexamination No. 90/009,995. (EX1011, 43-45.) Claims 29, 53, 54, 56-58, 60-68, 70, 71, 73, 74, and 76-88 of the '926 patent were canceled in *Inter Partes* Reexamination No. 95/000,635. (EX1011, 46-48.)

Inter Partes Reexamination No. 95/002,126 was terminated after the previous proceedings had canceled all the claims. (EX1014.)

The '353 patent is a divisional of application No. 09/878,097, filed on June 8, 2001, now the '099 patent. (EX1015, Cover.) The '353 patent was subject to IPR2013-00007 (joined by IPR2013-00256), where claims 1, 33, 36, 43, 48, 51, 52, 58, 59, 66, 118, 138, 139, 149, 183, 252, 283, and 317 were found unpatentable. (EX1015, 51.) The '353 patent was subject to *Ex Parte* Reexamination No. 90/009,994, where claims 5, 23-27, 60, 61, 122, 137, 140, 141, 147, 156, 174-177, 238-242, 248, 259, and 274-277 were amended and claims 320-322 were added. (EX1015, 53.) The '353 patent is currently subject to *Inter Partes* Reexamination Nos. 95/000,634 and 95/002,132, during which the Federal Circuit clarified 37 C.F.R. § 42.73(d)(3)(i) “applies to new claims or amended claims, but not to previously issued claims.” (*SoftView LLC v. Apple Inc.*, 108 F.4th 1366, 1369 (Fed. Cir. 2024); EX1019-EX1021; EX1023-EX1025; EX1047; EX1048.)

The '353 and the '926 patents have previously been involved in district court litigations. On May 10, 2010, SoftView filed a complaint for patent infringement of the '353 patent against Apple and AT&T. *SoftView LLC v. Apple Inc. et al.*, No. 1:10-cv-00389-LPS, Dkt. No. 1 (D. Del. filed May 10, 2010). In December 2010, SoftView filed an amended complaint to add the '926 patent, which had issued on November 9, 2010 (EX1011, 1). *SoftView LLC v. Apple Inc. et al.*, No. 1:10-cv-

00389-LPS, Dkt. No. 27 (D. Del. filed Dec. 03, 2010). On September 30, 2011, SoftView filed an amended complaint to add more defendants. *SoftView LLC v. Apple Inc. et al.*, No. 1:10-cv-00389-LPS, Dkt. No. 108 (D. Del. filed Sep. 30, 2011). On July 26, 2012, the litigation was split into the following 9 cases:

- *SoftView LLC v. Apple Inc. et al.*, No. 1:10-cv-00389-LPS (D. Del.)
- *SoftView LLC v. HTC Corp. et al.*, No. 1:12-cv-00984-LPS (D. Del.)
- *SoftView LLC v. Huawei Technologies Co. Ltd. et al.*, No. 1:12-cv-00985-LPS (D. Del.)
- *SoftView LLC v. LG Electronics Inc. et al.*, No. 1:12-cv-00986-LPS (D. Del.)
- *SoftView LLC v. Samsung Electronics Co. Ltd. et al.*, No. 1:12-cv-00987-LPS (D. Del.)
- *SoftView LLC v. Dell Inc. et al.*, No. 1:12-cv-00988-LPS (D. Del.)
- *SoftView LLC v. Kyocera Corp. et al.*, No. 1:12-cv-00989-LPS (D. Del.)
- *SoftView LLC v. Motorola Mobility LLC et al.*, No. 1:12-cv-00990-LPS (D. Del.)
- *SoftView LLC v. Sony Ericsson Mobile Communications (USA) Inc. et al.*, No. 1:12-cv-00991-LPS (D. Del.)

SoftView LLC v. Apple Inc. et al., No. 1:10-cv-00389-LPS, Dkt. No. 440 (D. Del. filed Jul. 26, 2012). On September 4, 2013, the cases were stayed pending the Board's review of the '926 and '353 patents. *SoftView LLC v. Apple Inc. et al.*, No. 1:10-cv-00389-LPS, Dkt. No. 1105 (D. Del. filed Sep. 4, 2013). On October 29, 2015, after the Board found unpatentable various claims of the '926 and '353 patents, all the cases were dismissed; SoftView agreed to dismissal of the unpatentable claims with prejudice, and the remaining claims were dismissed without prejudice. *SoftView LLC v. Apple Inc. et al.*, No. 1:10-cv-00389-LPS, Dkt. No. 1128 (D. Del. filed Oct. 29, 2015).

Petitioner is simultaneously filing IPR petitions challenging the '729, '628, and '889 patents.

Counsel and Service Information: Lead counsel is Naveen Modi (Reg. No. 46,224). Backup counsel are Daniel Zeilberger (Reg. No. 65,349) and Maksim Mints (Reg. No. 78,507). Service information is Paul Hastings LLP, 2050 M Street NW, Washington, D.C. 20036, Tel.: 202.551.1700, Fax: 202.551.1705, email: PH-Google-SoftView-IPR@paulhastings.com. Petitioner consents to electronic service.

III. PAYMENT OF FEES UNDER 37 C.F.R. §§ 42.15 AND 42.103

The PTO is authorized to charge any fees due during this proceeding to Deposit Account No. 50-262.

IV. GROUNDS FOR STANDING

Petitioner certifies under 37 C.F.R. § 42.104(a) that the '154 patent is available for review and Petitioner is not barred or estopped from requesting review on the grounds identified herein.

V. PRECISE RELIEF REQUESTED AND GROUNDS RAISED

Petitioner respectfully requests review of claims 1-35 of the '154 patent and cancellation of these claims as unpatentable. The challenged claims should be found unpatentable on the following ground:

Ground 1: Claims 1-35 are unpatentable under pre-AIA 35 U.S.C. § 103 as being obvious in view of Zaurus (EX1004), Pad_1 (EX1005) and Pad_2 (EX1007) (collectively “Pad”), and SVG (EX1008).

Ground 2: Claims 1-35 are unpatentable under pre-AIA 35 U.S.C. § 103 as being obvious in view of Zaurus, Pad, SVG, and further in view of SVF (EX1044).

The earliest possible priority date of the '154 patent is June 12, 2000. (EX1001, 1-2.) However, Patent Owner's contentions against Samsung assert that the '154 patent is entitled to a priority date of April 7, 2001 (EX1010, 5).² Regardless of which of the two dates is applied, the above references are prior art, and indeed the Board has previously credited that Zaurus, Pad, SVG, and SVF

² Petitioner does not concede that the '154 patent is entitled to either of these dates.

qualify as prior art to (i) the '926 patent to which the '154 patent claims priority and (ii) the '353 patent that is a divisional of the '099 patent to which the '154 patent claims priority. (Section II; EX1013, 2-3; EX1017, 2-3.) For example, Zaurus was published in 1997, and thus qualifies as prior art under at least § 102(b). (EX1004, 3.) Pad_1 was published on October 2, 1996 and thus qualifies as prior art under at least § 102(b). (EX1005, 84.) Pad_2 was published in April 1997 and thus qualifies as prior art under at least § 102(b). (EX1007, 1.) SVG was published on October 29, 1998, or at least by November 3, 1998, and thus qualifies as prior art under at least § 102(b). (EX1008, 1, 16.) SVF was published on January 16, 1995, and thus qualifies as prior art under at least § 102(b). (EX1044 (Captured by the Wayback Machine on October 19, 1996); *see also* EX1045 (January 16, 1995 release date).)

VI. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the art at the time of the alleged invention of the '154 patent ("POSITA") would have had an undergraduate degree in electrical engineering, computer engineering, computer science or a related field along with two years of work experience in the field of web development, graphics rendering,

and/or mobile applications. (EX1002, ¶¶13-14.)³ More education can substitute for practical experience and vice versa. (*Id.*)

VII. OVERVIEW OF THE '154 PATENT

The '154 patent is directed to improving the readability of web pages on small screen devices by converting HTML and Cascading Style Sheets (CSS) into a resolution-independent, scalable representation that can be zoomed and panned while preserving the page's original layout and functionality. (EX1001, Abs., 1:65-2:62.) The '154 patent relies on conventional browsers such as Mozilla for pre-rendering such as parsing, grouping objects, making bounding boxes, and defining page layout. (EX1001, FIG. 5 (150-154), 17:50-61; *see also id.* at 15:50-17:50.) The '154 patent utilizes the SVF (Simple Vector Format) considered by the W3C (World Wide Web Consortium) to translate content into a scalable vector representation. (EX1001, 4:56-5:3, 6:13-28, 6:61-7:5, 9:1-6, 12:23-31; EX1002, ¶¶15-19.)

³ Petitioner submits the declaration of Dr. Benjamin Goldberg, an expert in the field of the '154 patent. (EX1002, ¶¶2-6; EX1006.)

VIII. RELATED PATENTS

A. The '926 patent

The '926 patent is a parent of the '154 patent. (Section II.) In IPR2013-00004 challenging the '926 patent, the Board held claims 30, 31, 40, 41, 43, 52, 55, 59, 72, and 75 unpatentable over two grounds: Zaurus, Pad, and SVF; and Zaurus, Tsutsumitake, Hara, and SVG. (EX1013, 2-3.) The Board issued its decision on March 27, 2014 (EX1013), and the certificate cancelling the claims issued on January 15, 2016 (EX1011, 41) after the Federal Circuit affirmed the Board's decision.

In *Ex Parte* Reexamination No. 90/009,995, the Office canceled claims 1-28, 32-39, 42, 44-51, and 69 after raising rejections based on the Pad++ browser and SVF. (EX1033, 3, 22, 45.) In *Inter Partes* Reexamination No. 95/000,635, the Office canceled claims 29, 53, 54, 56-58, 60-68, 70, 71, 73, 74, and 76-88 after raising rejections based on the Pad++ browser. (EX1034, 3, 9.) Accordingly, the PTO canceled all 88 issued claims of the '926 patent. (EX1011, 41-48; EX1014.)

B. The '353 patent

The '353 patent is a divisional of the '099 patent to which the '154 patent claims priority. (Section II.) In IPR2013-00007 challenging the '353 patent, the Board held claims 1, 33, 36, 43, 48, 51, 52, 58, 59, 66, 118, 138, 139, 149, 183, 252, 283, and 317 unpatentable over four grounds: (1) Zaurus and Pad; (2) Zaurus, Pad,

and SVG; (3) Zaurus, Tsutsumitake, and Hara; and (4) Zaurus, Tsutsumitake, Hara, and SVG. (EX1017, 2-3.) The Board issued its decision on March 27, 2014 (EX1017), and the certificate cancelling the claims issued on January 12, 2016 (EX1015, 51) after the Federal Circuit affirmed the Board's decision.

In *Ex Parte* Reexamination No. 90/009,994, the Office raised rejections based on the Pad++ browser and SVF, after which Patent Owner amended and added claims. (EX1015, 53.) During *Inter Partes* Reexamination Nos. 95/000,634 and 95/002,132, the Federal Circuit affirmed the Board's finding that the claims amended and added during *Ex Parte* Reexamination No. 90/009,994 were patentably indistinct from the claims canceled during IPR2013-00007. (EX1019, 4-5; 15-16.)

IX. THE CHALLENGED CLAIMS SHOULD BE CANCELED BECAUSE THEY ARE PATENTABLY INDISTINCT FROM UNPATENTABLE CLAIMS

Patent owner estoppel bars obtaining a claim that is patentably indistinct from unpatentable claims. 37 C.F.R. § 42.73(d)(3)(i); *SoftView LLC v. Apple Inc.*, 108 F.4th 1366 (Fed. Cir. 2024). In addition, issue preclusion bars re-litigation of patentability issues for claims that are patentably indistinct from those previously held unpatentable. *See, e.g., Google LLC v. Hammond Dev. Int'l, Inc.*, 54 F.4th 1377, 1381 (Fed. Cir. 2022). As the Federal Circuit recently explained in a similar posture, “[t]o conclude otherwise would allow a patentee to assert a claim substantially similar to a claim previously found unpatentable by the PTAB.” *Apple*

Inc. v. Smart Mobile Techs. LLC, No. 2024-1352, at 9 (Fed. Cir. Jan. 21, 2026) (nonprecedential). Yet that is precisely what SoftView has done with the '154 patent.

The claims of the '154 patent are patentably indistinct from the unpatentable claims of the '353 and '926 patents, as reflected in the table below. The Office has already canceled all 88 claims of the '926 patent across multiple proceedings. (EX1011, 41-48; EX1014.) As for the '353 patent, the Office canceled 20 claims in an IPR (EX1015, 51), amended claims in an EPR (EX1015, 53), and intends to cancel many other claims in *Inter Partes* Reexaminations (EX1047; EX1048). And another petitioner specifically asserted the Zaurus, Pad, and SVG references against claims of the '926 and '353 patents, which the Board found unpatentable in 2014, the Federal Circuit affirmed in 2015, and the USPTO canceled in January 2016. (EX1013; EX1017; EX1015, 51; EX1011, 41.)

In November 2016, Patent Owner filed a patent application that issued as the '154 patent. (EX1001, 1.) Given that the '154 patent is a continuation of the '729 patent, which cited all four of the asserted references, and the same Examiner examined both patents, the asserted references should have been applied against the '154 patent. (EX1001, 1; EX1028, 3-4; *see* MPEP § 609.02(I).) The Examiner appreciated the similarity of the claims by rejecting the '154 claims as “not patentably distinct” from the '729 patent (EX1003, 95), which itself included a

terminal disclaimer against the '926, '353, and '889 patents (EX1029). Patent Owner confirmed this substantive similarity by filing a terminal disclaimer. (EX1003, 224; *see* MPEP § 804.) Instead of applying the necessary estoppel, issue preclusion, and/or art rejections, the Examiner allowed Patent Owner to erroneously obtain the patentably indistinct claims of the '154 patent in 2018. (EX1001, 1.)

In this proceeding, Petitioner applies those same Zaurus, Pad, SVF, and SVG references to the challenged claims and shows, consistent with the prior cancellations, that the challenged claims, which are not patentably distinct, are likewise unpatentable. The table below shows how the challenged claims are disclosed and/or suggested by previously canceled claims of the related '353 and '926 patents. (*See also* EX1002, ¶20, Appendices A-E (Dr. Goldberg describing this same correspondence).)⁴

'154 Patent	'353 Patent (EX1002, Appendix B)	'926 Patent (EX1002, Appendices C-E)
[1.pre]-[1.d], [13.pre], [26.pre]-[26.d]	[36.pre]-[36.a], [52], [317.pre]-[317.d]	(IPR2013-00004) ⁵ [52.pre]-[52.d] (90/009,995) [1.pre]-[1.e] (95/000,635)

⁴ The table below is exemplary and is not intended to be exhaustive.

⁵ For the '926 patent, Petitioner identifies the proceeding in which the limitation at issue was addressed.

Petition for *Inter Partes* Review
Patent No. 10,083,154

		[79.pre]-[79.f]
[1.e], [13.a], [26.e]	[36.c], [66], [317.e]	(IPR2013-00004) [30.g], [30.j], [41] (90/009,995) [1.g], [1.i.2.a], [13], [14] (95/000,635) [29], [65], [66], [79.i.2.a]
[1.f], [13.b], [26.f]- [26.f.1]	[36.c], [66]	(IPR2013-00004) [52.f], [52.i] (90/009,995) [21.b] (95/000,635) [79.h], [79.i.2], [78]
[1.f.1], [13.b.1]	[58.b], [138.b]	(IPR2013-00004) [72.b] (90/009,995) [16.b] (95/000,635) [79.h]
[1.f.2], [13.b.2], [26.f.2]	[58.c], [138.c]	(IPR2013-00004) [72.c] (90/009,995) [18.b] (95/000,635) [79.h]
[1.f.3], [13.b.3], [35.a]- [35.b]	[59], [139]	(IPR2013-00004) [72.e] (90/009,995) [18.c] (95/000,635) [73]
[1.f.4.a], [13.b.4.a]	[58.e.3], [138.e.3]	(IPR2013-00004) [72.e.3] (90/009,995) [18.d] (95/000,635) [70.a]-[70.e]
[1.f.4.b], [13.b.4.b], [35.c]	[58.e.1], [58.e.2]	(IPR2013-00004) [72.e.1], [72.e.2]

Petition for *Inter Partes* Review
Patent No. 10,083,154

		(90/009,995) [18.e] (95/000,635) [70.a]-[70.e]
[1.g], [13.c]	[43], [33]	(IPR2013-00004) [30.h] (90/009,995) [1.h] (95/000,635) [71.a]-[71.c], [79.h]
[1.g.1], [13.c.1], [26.f.3]	[58.d], [138.d]	(IPR2013-00004) [72.d] (90/009,995) [16.c] (95/000,635) [70.a]-[70.e]
[1.g.2], [13.c.2], [26.f.4]	[58.e], [138.e]	(IPR2013-00004) [72.e] (90/009,995) [16.d] (95/000,635) [70.a]
[1.g.2.a], [13.c.2.a], [26.f.4.a]	[58.e.1]-[58.e.3]	(IPR2013-00004) [72.e.1]-[72.e.3] (90/009,995) [16.d] (95/000,635) [70.a]-[70.e]
[1.g.2.b], [13.c.2.b], [26.f.4.b]	[58.e.2], [138.e.2]	(IPR2013-00004) [72.e.2] (90/009,995) [16.d] (95/000,635) [70.a]-[70.e]
[1.g.2.c], [13.c.2.c], [26.f.4.c]	[58.e.3], [138.e.3]	(IPR2013-00004) [72.e.3] (90/009,995) [16.d]

Petition for *Inter Partes* Review
Patent No. 10,083,154

		(95/000,635) [70.a]-[70.e]
[1.h.1], [13.d.1], [26.g.1]	[1.d]	(IPR2013-00004) [75.b], [75.c] (90/009,995) [25.a]-[25.c] (95/000,635) [65]
[1.h.2], [13.d.2], [26.g.2]	[36.e], [317.h]	(IPR2013-00004) [52.i] (90/009,995) [1.i.2] (95/000,635) [79.i.2]
[1.h.3], [13.d.3], [26.g.3]	[36.e], [66]	(IPR2013-00004) [52.i] (90/009,995) [1.i.2] (95/000,635) [79.i.2]
[2], [14], [27]	[36.pre]	(IPR2013-00004) [30.pre] (90/009,995) [2] (95/000,635) [53]
[3], [15], [28]	[317.pre]	(IPR2013-00004) [52.pre] (90/009,995) [3] (95/000,635) [54]
[4], [16], [29]	[317.h]	(IPR2013-00004) [31] (90/009,995) [4] (95/000,635) [82]

Petition for *Inter Partes* Review
Patent No. 10,083,154

[5], [17], [30]	[36.e]	(IPR2013-00004) [43.c] (90/009,995) [5] (95/000,635) [57]
[6], [18], [31]	[36.e]	(IPR2013-00004) [43.c] (90/009,995) [5] (95/000,635) [84]
[7.a]-[7.c], [19.a]-[19.c], [32.a]-[32.c]	[33], [43]	(IPR2013-00004) [43.b], [43.c] (90/009,995) [15.b], [15.c] (95/000,635) [71.a]-[71.c]
[8], [20], [33]	[36.b], [36.c]	(IPR2013-00004) [30.f], [30.g] (90/009,995) [1.f], [1.g] (95/000,635) [79.g], [79.h]
[9], [22], [34]	[66]	(IPR2013-00004) [30.f] (90/009,995) [1.f] (95/000,635) [29]
[10.a]-[10.c], [21.a]- [21.c], [23.a]-[23.c]	[51], [59]	(IPR2013-00004) [41] (90/009,995) [13], [18.c] (95/000,635) [65], [73]
[11.a]-[11.b], [24.a]- [24.b]	[36.c]	(IPR2013-00004) [52.f] (90/009,995)

		[21.b] (95/000,635) [77]
[12], [25]	[58.e.2]	(IPR2013-00004) [72.e.2] (90/009,995) [26.a]-[26.d], [27] (95/000,635) [63]

X. CLAIM CONSTRUCTION

For IPR proceedings, the Board applies the claim construction standard set forth in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (*en banc*). See 83 Fed. Reg. 51,340-59 (Oct. 11, 2018). Under *Phillips*, claim terms are typically given their ordinary and customary meanings, as would have been understood by a POSITA, at the time of the invention, having taken into consideration the language of the claims, the specification, and the prosecution history of record. *Phillips*, 415 F.3d at 1313; *see also id.* at 1312-16.

In *SoftView LLC v. Apple Inc. et al.*, No. 1:10-cv-00389-LPS, Dkt. No. 108 (D. Del. filed Sep. 30, 2011), the court and parties construed various terms of the '353 and '926 patents. (EX1023; EX1038; EX1049.) Likewise, during IPR2013-00004 and IPR2013-00007, the Board further construed certain terms of the '353 and '926 patents. (EX1013, 7-14; EX1017, 8-13.) In the parallel litigation, Samsung offered constructions “in a manner consistent with how they were construed” in the

previous litigation. (EX1057, 2.) SoftView offered constructions that either correspond to Samsung's or assert that no construction is necessary. (EX1009.)

In any event, the Board only construes the claims when necessary to resolve the underlying controversy. *Toyota Motor Corp. v. Cellport Sys., Inc.*, IPR2015-00633, Paper No. 11 at 16 (P.T.A.B. Aug. 14, 2015) (citing *Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999)). The analysis below is consistent with the prior constructions and those offered by Samsung and SoftView in the parallel litigation. Given the close correlation between the prior art references and the challenged claims of the '154 patent, the Board need not construe any terms of the challenged claims to resolve the underlying controversy, as any reasonable construction reads on the prior art. (EX1002, ¶¶25-27.)

XI. DETAILED EXPLANATION OF GROUNDS

A. Ground 1: Zaurus, Pad, and SVG Render Claims 1-35 Obvious

The Board has previously found (and been affirmed by the Federal Circuit) that there is motivation to combine and a reasonable expectation of success for a Zaurus-Pad-SVG combination. (EX1017, 2-3.)

1. Zaurus (EX1004)

Zaurus discloses a handheld device with an Internet/HTML browser that supports stylus/touch input and switching between reduced and magnified displays. (EX1004, 10-16, 20-21, 40, 46-47; EX1002, ¶21.)

2. Pad (EX1005 and EX1007)

Pad++ (“Pad”) discloses an HTML browser with zoom/pan capabilities, including fit-to-width zoom behavior and visible-object determination (e.g., R-tree). (EX1007, 86-96, 108, 141; EX1005, 30, 63.) Pad is designed for small-screen systems including handheld computers/PDAs. (EX1007, 139-141; EX1005, 81-82; EX1002, ¶22.)

3. SVG (EX1008)

SVG discloses standards-based vector graphics designed for devices including small mobile devices and requiring zoom/pan “with true scalability.” (EX1008, 1, 12-14; EX1002, ¶23.)

4. Motivation to Combine

A POSITA would have been motivated to combine Zaurus, Pad, and SVG to improve small-screen navigation using known zoomable browsing and scalable graphics techniques. (EX1035; EX1037; EX1043; EX1046; EX1002, ¶29.) Zaurus already supports browsing with pen/touch input and switching between a “reduced view” and “magnified view,” showing that small-screen viewing modes and zoom-related interaction were known and implemented on the handheld platform. (EX1004, 46-47; EX1002, ¶30.) Pad teaches a more flexible solution, including continuous zoom and pan that “allows users to zoom in and out of pages, enabling control of how much context is viewed at any time,” and Pad targets “small screens,

such as handheld computers (i.e. PDA's)," making it a natural enhancement to Zaurus's magnify/reduce approach. (EX1017, 18-24; EX1013, 18-23; EX1007, 86-96, 136-142; EX1002, ¶¶30-31; *see also* EX1050, 1-2; EX1051, 1.) SVG further teaches a standard for zooming and panning with true scalability across devices including small mobile devices, providing an incentive to incorporate scalable vector graphics so zoomed content remains readable and functional as the user navigates at different scales. (EX1008, 1, 14; EX1002, ¶32.) Thus, the Zaurus-Pad-SVG combination would have predictably improved handheld browsing by pairing Zaurus's handheld HTML browser with Pad's zoomable navigation techniques and SVG's scalable graphics standard. (EX1017, 23-24; EX1002, ¶32.)

5. Reasonable Expectation of Success

A POSITA would have had a reasonable expectation of success in combining these references, and the Board rejected arguments that technical difficulties would prevent porting Pad to Zaurus. (EX1017, 18-19; EX1013, 18-19; EX1002, ¶33.) Zaurus already demonstrates the relevant handheld hardware/UI capability (stylus/touch input and switching between reduced and magnified display states), supporting that integrating enhanced zooming behavior would have been a predictable adaptation to an existing handheld browsing platform. (EX1004, 15-16, 20-21, 46-47; EX1002, ¶34.) Pad discloses zoom and layout related mechanisms designed for interactive navigation (including smooth animated zooming) and

identifies handheld computers/PDAs as a target environment, reinforcing that a POSITA would have expected Pad's techniques to function on PDA devices such as Zaurus. (EX1005, 22-23, 54-56, 71; EX1013, 18; EX1017, 18; EX1002, ¶¶34-35.) SVG likewise is directed to scalable zooming/panning techniques suitable for small/mobile devices and defines a vector-based model whose scaling behavior is predictable. (EX1008, 1, 13-15; EX1002, ¶¶34-35.) Together, the references describe compatible browser/graphics techniques suitable for small-screen devices, and a POSITA would have expected the Zaurus-Pad-SVG combination to function as intended. (EX1035; EX1037; EX1043; EX1046; EX1002, ¶¶33-35.)

6. Claim 1

- i) [1.pre] "A mobile hand-held device, comprising:"**
- ii) [1.a] "a processor;"**
- iii) [1.b] "a wireless communications device, to facilitate wireless communication with a network that supports access to the Internet;"**
- iv) [1.c] "a touch-sensitive display; and"**
- v) [1.d] "flash memory, operatively coupled to the processor, in which a plurality of instructions are stored comprising a plurality of software components including an HTML rendering engine, wherein the instructions are configured to be executed by the processor to enable the mobile hand-held device to,"**

In the Zaurus-Pad-SVG combination, Zaurus is a "Mobile Business Tool" ("mobile hand-held device") that is "highly portable with compact size and

lightweight, browsing can be done by accessing the Internet.” (EX1004, 9-10; *see also id.*, 11-15.) Zaurus includes a CPU (“processor”). (EX1004, 18.) Zaurus includes adapters that support digital cellular for data communication to access the Internet (“a wireless communications device, to facilitate wireless communication with a network that supports access to the Internet”). (EX1004, 5-7, 11-13.) Zaurus further describes a display that receives touch inputs (“touch-sensitive display”). (EX1004, 15 (“Touch the screen with a pen. Let's try various operations by touching the screen with a pen.”); EX1002, ¶36.)

Zaurus includes flash memory (“flash memory”). (EX1004, 6, 18, 43.) The flash memory is used in connection with the Internet browser by placing the browser's cache in the “main body flash” or “card flash” (“operatively coupled to the processor”). (EX1004, 43, 48, 52.) Zaurus is pre-installed with an Internet browser (“in which a plurality of instructions are stored comprising a plurality of software components...instructions are configured to be executed by the processor to enable the mobile hand-held device to” [perform web functions]”). (EX1004, 39 (“Connect to the Internet, and open Zaurus Plaza (the Zaurus home page).”).) The Internet browser supports HTML 3.2 to display content such as text and images in GIF and JPEG format (“HTML rendering engine”). (EX1004, 40 (“It can display images not only in GIF format, but also in the commonly found JPEG format on a page in 16-gradation grayscale.”).) The Web pages can be displayed frame by frame,

which a POSITA would have understood are implemented using standard HTML `<frame>` markup (e.g., `<frame>/<frameset>`). (EX1004, 40 (“Pages composed of frames can also be viewed by displaying them frame by frame.”); *see also id.*, 38, 48-49; EX1041; EX1002, ¶37.)

In the combined system, Zaurus’s pre-installed browser is an executable software component stored in flash memory and run by the CPU to provide the disclosed Internet browsing functions, including the rendering of HTML content, images, hyperlinks, and multi-region page structures. (EX1004, 18, 39-41, 43.) Zaurus’s disclosure that it “incorporates HTML 3.2 specifications” and supports “pages composed of frames” would have been understood by a POSITA to require the browser’s “rendering engine” to interpret and process the underlying HTML structure and corresponding elements to generate the displayed page regions and frame-by-frame presentation. (EX1004, 40, 48-49; EX1041; EX1002, ¶38.)

- vi) **[1.e] “receive an HTML document including HTML code and cascading style sheet (CSS) code and content associated with the HTML document, the HTML code comprising a plurality of HTML elements including at least one HTML paragraph element, at least one HTML image element, and at least one HTML hyperlink element,”**

Zaurus discloses downloading HTML web pages that contain multiple HTML elements such as text, images, and links (“receive an HTML document including HTML code ... and content associated with the HTML document”). (EX1004, 10

(“download the information you want, and view it later at your own pace”), 40 (“The Internet browser in the MI-100 series incorporates HTML 3.2 specifications in important places, and therefore has rich expressive capabilities.”).) Zaurus displays blocks of text on Web pages (“at least one HTML paragraph element”). (EX1004, 10-11.) A POSITA would have understood that text is included in a “paragraph” encoded as HTML code <p>. (EX1042, 1 (“The text up to the next <p> element is treated as being part of the current paragraph.”); EX1043, 2, 10.) Zaurus further describes displaying pages frame by frame and images in GIF and JPEG format (“at least one HTML image element”). (EX1004, 38, 40 (“It can display images not only in GIF format, but also in the commonly found JPEG format on a page in 16-gradation grayscale...Pages composed of frames can also be viewed by displaying them frame by frame.”), 48-49.) Zaurus further describes hyperlinks for selecting links that cause retrieval of a page (“at least one HTML hyperlink element”). (EX1004, 10 (“A separate page which is linked is displayed by touching the underlined characters.”); EX1002, ¶39.)

Pad discloses fetching HTML content, parsing it, handling images, and creating hyperlinks. (EX1005, 5 (“Retrieve a URL over the internet in the background”), 15 (“HTML items know about the internet and will automatically fetch a file from a URL (Universal Resource Locator) as well as in-line images”), 16-19.) Pad further shows text hyperlinks and the user’s selection of those links.

(EX1007, 108 (“in the Pad++ HTML browser, some words are shown in blue. They are ‘hotwords’ - positioning the pointer over the word causes it to change color to red. When you click on the word, the appropriate document is placed on the Pad++ surface”), 111-113 (illustrating text acting as a hyperlink); EX1002, ¶40.)

SVG reflects that CSS support (including CSS positioning/styling) was a web-standard mechanism the Web community expected to be leveraged in HTML-based Web content. (EX1008, 5, 6, 7 (“All text/font attributes from CSS should be supported. Complete support for CSS is what the Web community will expect.”), 11 (“SVG should use CSS positioning”).) A POSITA designing a mobile handheld web device would have combined Zaurus’s mobile HTML browser (that supports image display and hyperlinks) with Pad’s receipt and parsing of HTML (including links and images) and SVG’s CSS styling for Web content. Combining these teachings would be a predictable and routine operation of an HTML browser that receives and renders web pages. In the relevant timeframe, it was likewise conventional that web page presentation was governed not only by the HTML element structure, but also by associated styling information (including CSS) that specified layout and presentation attributes for structured documents. (EX1037; EX1002, ¶41.)

SVG confirms that browser-based rendering environments were expected to be “compatible with and/or leverage” standards including “DOM” and “CSS,” and

that attributes “which make sense to be part of a style should be expressed in CSS,” reinforcing that CSS was treated as the standard mechanism for expressing style/layout information for web content. (EX1008, 4-7.) SVG further reflects that “complete support for CSS is what the Web community will expect” and that “SVG should use CSS positioning,” which a POSITA would have understood as confirming that CSS code (including positioning/layout properties) would be received and interpreted in connection with rendering HTML-related content. (EX1008, 6-7, 11.) In addition, SVG’s repeated requirement for “DOM access” confirms that HTML elements were treated as addressable objects in a standardized object model, consistent with receiving and identifying a plurality of HTML elements (e.g., paragraphs, images, hyperlinks) from the HTML code before rendering them. (EX1008, 15; EX1035; EX1002, ¶¶42-43.)

- vii) **[1.f] “process the HTML document with the HTML rendering engine to render a first representation of the HTML document having an interpreted page layout, functionality, and design of the content associated with the HTML document that is in accordance with the HTML code and CSS code, wherein rendering the first representation of the HTML document includes,”**

In the Zaurus-Pad-SVG combination, Zaurus describes processing HTML content with a browser that processes and visually renders web pages (“process the HTML document with the HTML rendering engine”). (EX1004, 10, 39-41, 43, 46; Sections XI.A.6.i-vi.) Zaurus’s browser can “display...on a page” based on the

interpreted HTML (“render a first representation”). (EX1004, 40.) A POSITA would have understood that the Zaurus HTML engine processes the HTML structure (frames, images, and text) and renders a page layout that is consistent with the HTML specification. (EX1002, ¶44.)

Zaurus discloses the ability to switch from a reduced view to a magnified view, as well as a left and right scrolling control and a vertical scrolling bar, to view material not currently on the screen. (EX1004, 43-47.) The selectable hyperlink is a functional element that responds to user input. (EX1004, 4, 10, 39.) Zaurus further discloses compatibility with client side clickable maps, so that by clicking inside a displayed map, one can jump to the page which corresponds to that portion and maintain the primary features of the page’s appearance. (EX1004, 40.) Pad discloses an “HTML browser” and then illustrates how links are rendered in blue or red, how following a link loads the target document with a two-column layout, and how users zoom on the content (“the HTML document having an interpreted page layout, functionality, and design of the content associated with the HTML document that is in accordance with the HTML code”). (EX1007, 86-93; EX1002, ¶45.)

As explained above with respect to limitation [1.e], the combined Zaurus-Pad-SVG device receives HTML documents with CSS information that includes the positioning and spacing of the objects. (EX1008, 5, 6, 7, 11.) The authors of SVG, W3C, described CSS as a standard. (EX1037.) A POSITA would have been

motivated to incorporate CSS support (“CSS code”) because its positioning information would enable more accurate and visually consistent rendering. Processing standard HTML and CSS encoded graphical information to produce an interactive visual representation was well known and would be the same rendering operation performed by an HTML engine that converts HTML and CSS code into a formatted page layout. (EX1002, ¶46.)

A POSITA would have understood that “render[ing] a first representation ... having an interpreted page layout, functionality, and design” is the ordinary operation of an HTML rendering engine: it interprets the document structure (e.g., text blocks/paragraphs, inline images, anchors/hyperlinks, and frames) and produces a formatted presentation in accordance with the document’s markup and associated style information. (EX1004, 39-41, 43, 48-49; EX1041; EX1042; EX1037.) SVG confirms that, in web rendering environments, CSS and DOM were expected to be leveraged together for layout and styling, such that rendering “in accordance with the HTML code and CSS code” was a predictable and routine implementation choice for an HTML engine seeking accurate and visually consistent rendering. (EX1008, 4-7, 11, 15; EX1035.) Pad further confirms that conventional rendering engines were capable of converting structured page content into rendered graphical representations that preserve page semantics (including links) while supporting interactive browsing operations. (EX1007, 86-93; EX1005, 60-63; EX1002, ¶47.)

(1) [1.f.1] “parsing the HTML document to identify the plurality of HTML elements;”

In the Zaurus-Pad-SVG combination, Zaurus describes handling HTML content with a browser that processes and visually renders web pages. (EX1004, 10, 38-41, 43, 46, 48-49; Sections XI.A.6.i-vi.) For example, displaying pages “composed of frames...frame by frame” (EX1004, 40; *see also id.*, 38, 48-49) requires parsing the specific HTML element for frames. A POSITA would have understood that the Zaurus browser parses the HTML document to identify its defined HTML elements (e.g., paragraphs, images, hyperlinks, frames) for rendering (“parsing the HTML document to identify the plurality of HTML elements”). (EX1002, ¶48.)

Pad discloses that the HTML data is parsed (“parsing the HTML document”). (EX1005, 60 (“When the html data is fetched, it is parsed and the HTML item is created which contains a method for rendering the page.”).) In the Zaurus-Pad-SVG combination, Pad creates and enumerates HTML anchors as separate items, which demonstrates the identification of HTML elements; Zaurus shows those elements (links, frames, maps) in a handheld browser; and SVG’s DOM framework confirms element-level identification (“to identify the plurality of HTML elements”). (EX1004, 38, 40, 48-49; EX1005, 61-65; EX1008, 15.) Pad describes that HTML documents are represented as graphical objects on a zoomable surface, and identifies the anchors of HTML items. (EX1005, 62, (“Returns all the anchors that are part of

this HTML item”), 63 (“If script is specified, it gets evaluated when the html source has loaded”); EX1007, 17, 20, 108, 111-113.) A POSITA starting with the Zaurus browser would have used Pad’s parsing and anchor element identification techniques to parse the HTML document for rendering. (EX1002, ¶49.)

SVG also confirms the standard methodology for identifying elements in a web context: the Document Object Model (DOM). (EX1035.) SVG mandates “DOM access” and compatibility with the DOM to expose graphic objects. (EX1008, 4, 15 (“DOM access... SVG should provide for complete DOM support for all attributes and elements.”).) A POSITA would have understood that the DOM is the standard interface for parsing both HTML and XML documents into a tree of addressable objects. Thus, SVG’s disclosure of DOM-based element identification reinforces the motivation to use standard parsing techniques to “identify the plurality of HTML elements” (e.g., anchors, images) within the Zaurus browser’s HTML rendering engine. SVG also describes XML, DOM, and CSS, which involves parsing elements and attributes into objects for rendering. (EX1008, 3-4, 6-8, 15.) The authors of SVG, W3C, described that XML was a standard Extensible Markup Language (XML) in 1998. (EX1046.) In the Zaurus-Pad-SVG combination, a POSITA would have incorporated the known parsing techniques of Pad and SVG into the Zaurus browser to parse the HTML document. (EX1002, ¶50.)

(2) [1.f.2] “logically grouping content associated with HTML elements into HTML objects”

In the Zaurus-Pad-SVG combination, Pad treats HTML content as items grouped under an HTML/page item, and explains that group items “are used just for creating structure” and “may be hierarchical,” and that an “HTML item is a group” that may contain anchors/members. (EX1005, 60-64 (hyperlinks), 64-65 (images), 78 (text), 40-41, 59, 62; *see also* EX1007, 89-96; EX1002, ¶51.) SVG likewise discloses DOM-based objects and grouping/layering collections of objects for scripting and layout. (EX1008, 4-6, 9-11, 14, 15; EX1035; EX1002, ¶52.) Thus, starting with Zaurus’s HTML browser, applying Pad’s grouped HTML-item structure and SVG’s DOM/grouping model to group parsed content into objects would have been predictable and routine. (EX1004, 40; EX1002, ¶¶52-53.) Accordingly, starting with Zaurus’s handheld HTML browser context (text/images/links/frames), applying Pad’s group-based HTML item structure and SVG’s DOM/grouping model to group content associated with HTML elements into HTML objects would have been predictable and routine. (EX1004, 38-41, 48-49; EX1005, 59-65; EX1008, 9, 14-15; EX1002, ¶53.)

(3) [1.f.3] “generating page layout information including a bounding box for each HTML object; and”

In the Zaurus-Pad-SVG combination, Zaurus contemplates pages with boundaries and Pad computes and uses bounding boxes for items (including

group/HTML items) during layout, navigation, and event handling, and its grid-based layout illustrates page layout computations. (EX1004, 48; EX1005, 2, 16-17, 22, 34, 38-41, 54-57, 59-60, 65-66, 73, 81; EX1007, 91; EX1002, ¶¶54-58.) SVG likewise contemplates DOM-scriptable graphics and utilities to query an object's bounds, consistent with maintaining per-object bounding boxes during rendering. (EX1008, 4-6, 10-11, 14-15.) Accordingly, a POSITA would have generated “page layout information including a bounding box for each HTML object” in the combined rendering pipeline. (EX1004, 38-40, 48-49; EX1005, 60; EX1007, 15; EX1002, ¶¶54-58.)

Zaurus explains that “a single page is divided into a certain number of boundaries so that the information displayed within the boundaries can be freely switched.” (EX1004, 48.) Pad's disclosed HTML/browser environment uses bounding boxes not only for display-fitting (e.g., centering and scaling views based on an item's bounding box), but also for layout computations and interaction (e.g., event routing based on whether an event occurs within a bounding box). (EX1005, 16-17, 22, 41, 54-55, 59-60; EX1002, ¶55.) SVG similarly contemplates per-object geometry tracking in a DOM-scriptable graphics environment by providing “utility methods (e.g., query an object's bounds ...),” which a POSITA would have understood as the conventional maintenance of per-object bounds/bounding boxes in the rendering engine for layout, selection, and scripting. (EX1008, 15; EX1002,

¶56.) In addition, Zaurus’s frame-by-frame viewing and frame selection interface presupposes that the browser identifies spatial regions corresponding to frame portions and rendered objects to support display of a selected region and user navigation among regions. (EX1004, 40, 48-49; EX1002, ¶57.) Thus, in the combined rendering pipeline (Zaurus HTML browsing + Pad itemization/layout + SVG DOM/geometry utilities), generating page layout information including a bounding box for each HTML object would have been the ordinary course of operation to support layout, visibility, hit-testing, and view-fitting/zoom behaviors. (EX1004, 38-43, 48-49; EX1005, 16-17, 22, 54-55, 59-60; EX1008, 15; EX1002, ¶58.)

(4) [1.f.4.a] “storing information that links each HTML object with its corresponding page layout information,”

In the Zaurus-Pad-SVG combination, Pad associates each item (including HTML/group items) with its geometry and configuration data (e.g., bounding boxes and grid-position pixel offsets) addressable by tag/id, and SVG’s DOM model assigns unique IDs and exposes per-element attributes/bounds. (EX1005, 2, 16-17, 54-55, 60; EX1008, 9, 14-15.) Given Zaurus’s handheld browser context (frames, caching/offline storage, clickable maps, and reduced/magnified views), the combination would have stored “information that links each HTML object with its

corresponding page layout information” for rendering and interaction. (EX1004, 6, 38-43, 46-49; EX1002, ¶59.)

Zaurus’s disclosed browsing functionality (including cached browsing/offline viewing, hyperlink activation by touching underlined text, and client-side clickable maps) depends on maintaining associations between logical page constructs (e.g., links/regions) and the rendered geometry presented on the screen so that user inputs (touch/stylus taps) can be mapped to the correct logical target and corresponding action. (EX1004, 6, 10, 40-41, 43.) A POSITA would have understood that these interactions require the rendering engine to retain, for rendered objects, identifiers and associated geometric/layout metadata (e.g., bounds and position) sufficient to (i) detect whether a user input hits a rendered region and (ii) dispatch the corresponding action (e.g., follow a link or activate a map region). (EX1004, 10, 40; EX1002, ¶60.)

Pad provides conventional mechanisms for precisely this association: items have identity via tags/ids and support storing “application-specific information” with an item (e.g., link targets, URLs, or element identifiers), while item geometry and placement are available through bounding box and placement/view APIs. (EX1005, 2, 17, 30, 54-55, 60; EX1007, 107-108.) For example, Pad’s tag/id mechanisms and per-item query commands support linking an item’s logical identity to its geometry and configuration data (including bounds and placement), while event bindings rely

on those same associations to invoke item-specific behaviors when user events occur. (EX1005, 17, 54, 82; EX1002, ¶61.)

SVG similarly teaches and/or suggests maintaining element-to-geometry associations in a DOM-addressable vector graphics environment: groups/objects have unique IDs and associated attributes (including transformations and visibility) accessible via scripting, and the environment provides utility methods to query object bounds, supporting retrieval of per-object geometry for selection and interaction. (EX1008, 9, 14-15; EX1035.) SVG further indicates that links may be assigned “to any graphic object or group,” reinforcing that rendered vector objects remain linked to logical link targets and can be activated as interactive objects. (EX1008, 14; EX1002, ¶62.) Given that (i) Pad discloses per-object stored bounding boxes addressable by tag/id (and applying to HTML items), (ii) SVG discloses DOM access and utilities to query an object’s bounds for selection and zoom/pan, and (iii) the Zaurus handheld browser’s includes offline storage, clickable maps, and reduced/magnified views, it would have been obvious and routine to store information linking each HTML object (or corresponding DOM element/item) to its page layout information (including bounds and transforms), to enable rendering and interaction. (EX1004, 40-43, 46-49; EX1005, 2, 17, 54-55, 60; EX1008, 9, 14-15; EX1002, ¶¶63-65.)

(5) [1.f.4.b] “wherein the page layout information further includes information from which a page layout location of each of the bounding boxes can be determined;”

In the Zaurus-Pad-SVG combination, Zaurus’s frame-by-frame viewing and frame-region selection presupposes knowing/storing layout regions/locations on the page. (EX1004, 39-40, 48-49.) Further, the stored page layout information in the combined system includes information from which the layout location of each bounding box can be determined. Zaurus’s frame display and frame selection interface reflects that multi-region page structures are rendered with spatially defined regions, and that the user can select which region/frame portion is displayed, which presupposes that the system knows where those regions lie in the page coordinate space. (EX1004, 40, 48-49; EX1002, ¶66.)

Pad provides coordinate-expressed bounding boxes (e.g., x1 y1 x2 y2), view parameters (xview, yview, zoom), and grid cell pixel offsets/width/height; information from which the page/screen location of each bounding box can be determined. (EX1005, 29-30, 41, 54-55; EX1007, 108.) Pad represents bounding boxes using numeric coordinates (e.g., “x1 y1 x2 y2”) and provides view parameters (e.g., “xview yview zoom”) that map page coordinates to the displayed viewport, which together provide information from which the location of each item’s bounding box can be computed in either page coordinates or screen coordinates for the current view. (EX1005, 29-30.) Pad’s layout mechanisms (including grid cell

offsets/width/height) similarly provide per-region pixel offsets and sizes, reinforcing that per-object (and per-region) page/screen locations are determinable from stored layout data. (EX1005, 54-55; EX1002, ¶67.)

SVG similarly provides viewport positioning and coordinate transforms for mapping object bounds into page coordinates. (EX1008, 9-15.) SVG describes coordinate systems and viewport establishment (including relationship to CSS positioning) and supports transformations (including translation) via 2×3 matrices; a POSITA would have understood that applying these coordinate systems and transforms maps local object-bounds into absolute page locations within the viewport. (EX1008, 11-12; EX1002, ¶68.)

Thus, the combined teachings provide page layout information from which each bounding box location can be determined. (EX1004, 39-40, 43; EX1005, 17, 55; EX1008, 11-12.) Combining Pad's coordinate bounds and view parameters with SVG's viewport/transform framework and Zaurus's multi-region browsing behavior yields page layout information from which the page layout location of each bounding box can be determined. (EX1004, 40, 48-49; EX1005, 29-30, 54-55; EX1008, 11-12; EX1002, ¶69.)

- viii) [1.g] “translate the first representation of the HTML document to generate a scalable vector representation of the HTML document wherein generating the scalable vector representation includes,”

In the Zaurus-Pad-SVG combination, Zaurus renders HTML 3.2 pages (including frames, images, and links) into an initial interpreted layout (a first representation). (EX1004, 9, 10, 39-41; EX1002, ¶70.)

Pad’s zoomable HTML/browser architecture fetches/parses HTML and represents it as renderable items (text, images, anchors) using scalable rendering primitives and zoom parameters. (EX1005, 6, 34-35, 53-56, 60-64, 81-82; EX1007, 85-86, 91-93.) Pad’s zoomable HTML browsing architecture demonstrates converting HTML content into renderable objects on a 2D surface (e.g., text items, image items, and anchor/link items) that can be re-rendered at different magnifications, which a POSITA would have understood as translating the browser’s initial HTML interpretation into a scalable representation suitable for zoom. (EX1007, 85-86, 91-93; EX1005, 60-65, 81-82.) Pad describes that HTML content is fetched and parsed to create an “HTML item” with a rendering method, and that anchors and images are instantiated as items within that HTML item/group, supporting a representation of the page as a set of addressable objects rather than a fixed raster snapshot. (EX1005, 60-65.) A POSITA would have recognized that such object-based representations are naturally compatible with a scalable vector

representation because object geometry (e.g., positions, bounds, and transforms) can be scaled and re-rendered without the pixelation artifacts that occur when scaling raster images. (EX1036; EX1002, ¶¶71-72; *see also* EX1017, 15 (“there appears to be little dispute that [Pad’s inventor] Bederson discloses vectorized content”).)

In Pad, default HTML bindings “lay them out with scale,” and multi-column web layouts are supported using grid items that arrange items in rows and columns while treating them “as a group,” with configurable spans and padding. (EX1005, 53-56, 60.) Pad also discloses an explicit rendering pipeline using vector primitives (e.g., drawline/drawpolygon/drawtext), consistent with building a scalable representation from page content rather than storing a fixed bitmap. (EX1005, 6.) In the Pad web browser demonstration, the system is described as “a zooming version of Mosaic and Netscape,” and items render using “zoomaction” behavior that executes when an item “grows or shrinks,” reinforcing that the browser re-renders scalable objects across magnifications. (EX1007, 85-86; EX1005, 81-82.) A POSITA would have understood these details as a blueprint for converting an initial HTML interpretation into a scalable rendering representation suitable for SVG as a standardized vector target. (EX1008, 1, 5, 13-14; EX1002, ¶72-73.)

SVG is directed to a Web-standard Scalable Vector Graphics format intended to be widely implemented in browsers and suitable for display on a wide range of device resolutions, including small mobile devices. (EX1008, 1.) SVG defines

scalable vector support for text, image data, links, CSS, coordinate systems, transforms, and zoom/pan with “true scalability,” making it a natural vector target format for the translation step. (EX1008, 7-8, 10-15; EX1036.) SVG states that it is intended to replace many uses of raster graphics and to be “displayable on a wider range of device resolutions from small mobile devices” while being “more interactive,” and it further requires zoom/pan with “true scalability” such that objects and attributes (including text and images) grow/shrink uniformly with zoom level. (EX1008, 1, 13-14.) SVG also identifies that its feature set should enable “reasonable conversion” of existing graphics into SVG, reinforcing that conversion/translation to a vector format was a contemplated and routine approach for enabling scalable display. (EX1008, 5; EX1002, ¶¶73-74.)

In the combination, translating/converting the first HTML representation into an SVG-based scalable vector representation to enable resolution-independent zooming would have been a predictable and routine design choice. (EX1008, 5-6, 14; EX1004, 9, 40.) That “scalable vector representation” would also include one or more vectors, e.g., for each HTML object a stored (x, y) translation offset (vector) represented by (X2, Y2) relative to an origin point (X1, Y1), where (X1, Y1) is the page-level origin (“primary datum”) and (X2, Y2) is the object’s anchor/placement point (“object datum”), as part of the object’s placement/transform data. (EX1002, ¶¶75-78.) In view of (i) Zaurus’s mobile browsing context on a small screen, (ii)

Pad's demonstrated zoomable HTML browser in which users zoom to read content laid out in columns, and (iii) SVG's browser-implemented scalable vector format intended for mobile devices with true scalability and web-style integration (links/CSS/DOM/coordinates/transforms), a POSITA would have translated the first representation of the HTML document into an SVG-based scalable vector representation to achieve resolution-independent zooming while preserving page layout and functionality. (EX1004, 39-41, 46-47; EX1007, 85-93; EX1008, 1, 5, 11, 13-15; EX1002, ¶¶75-78.)

**(1) [1.g.1] “defining a primary datum
corresponding to the interpreted page layout;”**

In the Zaurus-Pad-SVG combination, generating the scalable vector representation includes defining an origin. Pad discloses a fixed global coordinate system for the Pad++ surface, including an origin (0,0) used as the page-level reference for placing items, and conversion between window coordinates and Pad coordinates. (EX1005, 5, 33, 12, 13, 23, 34, 54; EX1002, ¶¶79-80.) Pad's view parameters (xview, yview, zoom) operate on that fixed coordinate space for navigation/rendering; the “primary datum” is the page-level origin (0,0) (or equivalent global reference point) of that underlying coordinate system. (EX1005, 8, 30; EX1007, 107.)

The “padxy” routine maps window coordinates to Pad coordinates, demonstrating a fixed underlying coordinate system with an origin and axes.

(EX1005, 5 (“padxy[52] Convert a window point (x, y) to pad coordinates”), 33.)

The coordinates form frames that are further defined by a bounding box to which the unit square is mapped, so that “all coordinates are interpreted relative to the frame.” (EX1005, 12 (“coordinates on the Pad++ surface”), 13 (“you can zoom into the position”), 23 (“If the flag -objectcoords is specified, then all coordinates are returned in the item’s local coordinate system... If not specified, then all coordinates are returned in the global coordinate system.”), 34.) The grids return a “pixel offset” relative to the corner and width and height, which demonstrates fixing the origin from which the layout positions are measured. (EX1005, 54; EX1007, 107 (“a view onto the infinite Pad++ surface”); EX1002, ¶80.)

A POSITA would have understood that the origin and axes of Pad’s fixed global coordinate system define the page layout space (“corresponding to the interpreted page layout”). The view (xview, yview, zoom) operates on that coordinate system for navigation/rendering and does not define the underlying datum itself. Consistent with SVG’s viewport/user-coordinate establishment, Pad’s fixed global coordinate reference (origin/axes) provides the “primary datum” corresponding to the interpreted page layout. (EX1005, 5, 30, 33; EX1008, 11; EX1002, ¶82.)

SVG likewise describes coordinate systems/viewports (including relationship to CSS positioning) and zoom/pan with true scalability, which a POSITA would

have understood defines a fixed origin point at an X, Y coordinate for the drawing/page coordinate space (“primary datum”). (EX1008, 11, 14; EX1002, ¶¶81-82.) Accordingly, in view of Zaurus’s rendering context and Pad/SVG coordinate systems, it would have been obvious to define a page-level origin/reference point (0,0) in the underlying coordinate system as the “primary datum corresponding to the interpreted page layout,” and to express object layout locations relative to that fixed reference. (EX1005, 29; EX1008, 11; EX1002, ¶83.)

(2) [1.g.2] “for each HTML object,”

**(a) [1.g.2.a] “defining an object datum
corresponding to a layout location datum
for the HTML object’s bounding box;”**

In the Zaurus-Pad-SVG combination, each object has a defined placement reference point used predictably for placement; the “object datum” is the object’s anchor reference point (defined by Pad’s -anchor), and the “layout location datum” is the absolute placement coordinate (the “positioning point” / “placement point” specified by Pad’s -place in Pad coordinates) in the page/layout coordinate system. In Pad’s placement model, the “object datum” corresponds to the object’s anchor reference point (as set by “anchor”), and the “layout location datum” corresponds to placement/positioning coordinates (as set by “-place”) at which the anchor point is placed. (EX1005, 47-50; *see also* EX1005, 38 (dx/dy applied to the item

transformation, i.e., the “-place itemconfigure option,” and reporting position at the item’s anchor point); EX1002, ¶84.)

Pad discloses per-item bounding boxes and anchor placement datums, with coordinate frames mapped to bounding boxes and bounding-box layout data, such that each object has a predictable reference point used to position it within the layout. (EX1005, 2-3, 22, 29, 34-35, 38, 47-50, 53; EX1002, ¶¶84-85.) Pad discloses anchor-based placement using a positioning point and an anchor reference on the object: “AnchorPos tells how to position the object relative to the positioning point,” explaining that selecting a particular anchor places a defined point of the object at the positioning point. (EX1005, 47.) Pad further shows that item location can be expressed in coordinates that are meaningful for display positioning (e.g., returning the coordinates/bounding box as it appears for the current view, including for a “sticky item” that appears at the same screen location). (EX1005, 17.)

SVG likewise discloses per-object coordinates/transformations (including local user coordinates) and querying object bounds, supporting per-object datums tied to layout locations. (EX1008, 9, 11, 14; EX1002, ¶86.) Thus, after parsing/grouping, each element can be materialized as a per-object vector item with an anchor/bounding-box datum and per-object coordinates/bounds consistent with Pad and SVG. (EX1005, 2-3, 17, 34-35, 47; EX1008, 3, 9, 11, 14, 16; EX1002, ¶87.) Defining each object’s reference relative to its bounding box (anchor/frame)

provides the baseline for positioning and transformations used for zoom/pan. (EX1005, 6, 12-13, 24, 38, 47-50; EX1008, 11, 15; EX1002, ¶¶87-89.) Thus, for each object, the system defines (i) the object's internal reference point ("object datum") and (ii) the object's placement location in the page coordinate system ("layout location datum"), with the object datum positioned at that layout location for rendering and later zoom/pan transformations (i.e., the anchor reference point is placed at the placement/layout coordinate). (EX1002, ¶¶87-89.)

(b) [1.g.2.b] “generating a vector from the primary datum to the object datum for the HTML object’s bounding box; and”

As explained above, the Zaurus-Pad-SVG combination renders obvious the primary datum and the object datum corresponding to the object's bounding box. (See Sections XI.A.6.viii.1-viii.2(a).) Pad discloses object locations via bounding-box coordinates and view/object translations, including use of “vector” parameters (e.g., “moveto”/”lineto”), anchor point positions, and bounding box scripts that return two-element vectors, with local and global coordinates enabling translation computations. (EX1005, 17-23, 30-32, 38, 65-66, 82; EX1007, 97; EX1002, ¶91.) SVG likewise discloses coordinate systems/viewports and 2x3 matrix transformations sufficient for translation (i.e., vector offsets), supporting vectors from origin to object positions. (EX1008, 11; EX1002, ¶92.) Consistent with that disclosure, the “object datum” is a predictable reference point for the object (e.g.,

the object's anchor point) used for placement relative to its bounding box and at its layout location (i.e., bounding box/bounds information from which the HTML object's bounding box and its "layout location datum" can be determined). Accordingly, because Pad places each item by positioning its anchor point at the "-place" coordinates in the global Pad coordinate system, the item's position is represented as a translation vector (x, y) from the page-level origin/reference ("primary datum") to the item's anchor/reference point ("object datum"). (EX1005, 38, 47-50, 82; EX1008, 11, 15; EX1002, ¶93.) Pad's "slide" command confirms this translation-vector model by adding dx/dy to the item's transformation (i.e., its "-place itemconfigure option") and returning the resulting (x, y) position at the item's anchor point. (EX1005, 38; EX1002, ¶93.)

Pad further confirms the "vector from the primary datum to the object datum" concept: it reports the (x, y) position at an item's anchor point, showing that each object has a stable reference point whose position is represented numerically in the page coordinate space. (EX1005, 38, 47.) Pad also distinguishes local item coordinates from global coordinates, providing the standard mathematical basis for expressing a displacement between a page-level reference and an item-level reference (i.e., a vector/difference). (EX1005, 23.) In addition, Pad's bounding-box computation supports returning coordinates as two-element vectors specifying the lower-left and upper-right corners, reinforcing that bounding geometry and

placement are encoded in vector/coordinate form. (EX1005, 65.) These disclosures align with SVG's 2×3 transforms, where translation is represented as an (x, y) offset component applied to place the object datum within the page/viewport coordinate system. (EX1008, 11-12; EX1002, ¶¶94-95.)

(c) [1.g.2.c] “creating a reference that links the HTML object to the vector that is generated; and”

The Zaurus-Pad-SVG combination teaches maintaining an association between logical elements and rendered geometry: Zaurus's hyperlinks and client-side clickable maps require mapping from HTML-defined regions to rendered touch-selectable areas. (EX1004, 6, 10, 40; EX1002, ¶¶96-97.) Pad supplies persistent item identity (tags/ids), storage for application-specific data, and event bindings, which are mechanisms for linking an HTML object (or DOM node) to its corresponding vector item/transform. (EX1005, 2, 17, 30, 60, 82; EX1007, 107-108; EX1002, ¶98.)

SVG discloses DOM linkage and linking on any graphic content. SVG requires “complete DOM support for all attributes and elements” and other techniques such as querying an object's bounds. (EX1008, 15.) SVG contemplates that it “should be possible to assign a link to any graphic object or group.” (EX1008, 14.) SVG further describes 2×3 transformation matrices and scaling attributes to enable vector associations from the origin to the object. (EX1008, 9 (“Groups

should have the following attributes (among others), all of which are accessible and/or controllable via scripting: i. Unique ID”), 10, 11 (“Arbitrarily nested 2x3 matrix transformations should be available. This facility is necessary to ensure compatibility with existing artwork and authoring tools...2x3 matrices are sufficient to achieve any types of translation, scaling, rotation and skewing”), 12 (“These transformations should apply to all object types (shapes, images and text) in a uniform and consistent manner”).) A POSITA would have understood that these requirements enable an association between DOM elements and the corresponding vector object to support events, linking, and testing (“creating a reference that links the HTML object to the vector that is generated”). (EX1002, ¶99.)

Pad supplies mechanisms for linking logical objects to rendered geometry/transforms. Items have identifiers/tags and a “place to store application-specific information with an item,” which a POSITA could use to associate an HTML element (or DOM node) with its corresponding vector item and transform (e.g., storing a URL, element identifier, or map-area data). (EX1005, 2.) Pad also supports item-specific event binding, where commands are associated with items so that when a specified event sequence occurs for an item, the corresponding command is invoked; a mechanism to connect rendered objects to interactive behavior. (EX1005, 17.) Pad further teaches that “Any object can be turned into a hyperlink,” reinforcing that link functionality is tied to the rendered object identity.

(EX1007, 108.) In the combination, Pad teaches vector items having persistent identity, tags, and other information for application specific associations; and SVG teaches that vector graphics in the web stack should be DOM-addressable with links on objects. (EX1004, 40; EX1005, 2, 17; EX1008, 15.) These mechanisms support Zaurus's disclosed interactions (including client-side clickable maps), which require mapping between logical HTML-defined regions and rendered touch-selectable geometry to dispatch the appropriate action. (EX1004, 40.) Reimplementing the same behavior in a vectorized renderer as taught by Pad and SVG would keep the same mapping. A POSITA would have utilized these attributes to store the DOM node identifier or URL on the vector item, creating a shared tag or reference across all vector items belonging to the HTML element. SVG's DOM addressability and ability to assign links to any graphic object/group is consistent with maintaining these references at the vector-graphics layer. (EX1008, 14-15; EX1002, ¶¶100-101.)

- ix) **[1.h.1] “render the scalable vector representation of the HTML document on the touch-sensitive display using a first scale factor to display the HTML document at a first zoom level under which the HTML document is displayed to fit across a width of the touch-sensitive display,”**

Zaurus discloses a touch-sensitive mobile device that renders Web pages (“touch-sensitive display”). (EX1004, 15 (“Touch the screen with a pen. Let’s try various operations by touching the screen with a pen”).) Zaurus describes zooming controls to switch from reduced to magnified display (“first scale factor...first zoom

level”). (EX1004, 46-47 (switching between reduced and magnified displays) ; EX1002, ¶102.)

Pad discloses scaling and zooming a vector representation using a selectable scale factor (“render the scalable vector representation...first scale factor...first zoom level”). Pad discloses numeric view and zoom settings that controls the rendering of vector items. (EX1005, 29 (“Returns a list with four elements giving the bounding box of string if it is drawn with the drawtext command. The list has the form ‘x1 y1 x2 y2’ such that the text is within the region bounded by x1 on the left, x2 on the right, y1 on the bottom, and y2 on the top.”), 30 (“Returns the current view of the main window in ‘xview yview zoom’ form. Here, (xview, yview) specifies the point at the center of the window, and zoom specifies the magnification.”), 34 (“A frame is a bounding box, and all coordinates are specified within the unit square where the unit square is mapped to the frame.”).) Pad discloses zoom capabilities that demonstrate that rendering is vector based and controlled by a zoom factor. (EX1005, 13 (“maxZoom...gives 16 orders of magnitude of zooming”); EX1002, ¶103.)

Pad discloses fitting the rendered content across the display (“under which the HTML document is displayed to fit across a width of the touch-sensitive display”). Pad can compute an image item’s bounding box (“x1 y1 x2 y2”), then adjust the view so the item (or its bounding box) “fills the specified amount of screen” via a center

command or set the view magnification using a move command for “magnification of zoom.” (EX1005, 17, 22, 30, 32, 54 (“The bounding box (in pixels) is returned for the space occupied by the grid position indicated by column and row.”).) In Pad’s browser example, selecting a text element “zooms in so that that text fills the screen horizontally and appears near the top of the screen vertically.” (EX1007, 54; EX1002, ¶104.)

SVG describes vector graphics that are intended to be scaled for rendering Web pages (“render the scalable vector representation”). SVG emphasizes uniform transformation and scaling of shapes, images, and text on a view port established via CSS positioning. (EX1008, 12.) SVG further describes zoom and pan with scalability of text and images. (EX1008, 1 (“The W3C has chartered a Scalable Vector Graphics working group to produce a specification for an SVG format, written as a modular XML tagset.”), 11 (“SVG should use CSS positioning for establishing a viewport for an outermost SVG element...SVG will support local user coordinates”), 13, 14 (“An SVG processor should support zoom and pan on graphics, with true scalability. Thus, all objects and attributes (including such things as text and linewidths) should grow/shrink uniformly with the zoom level.”); (EX1002, ¶105.)

A POSITA would have been motivated to combine the Zaurus touch web device with zoomed views, Pad’s zooming and rendering of vector content, and

SVG's scalable and transformable web graphics. (EX1002, ¶106.) On the Zaurus small screen that already supported reduce and magnify operations, usability would depend on presenting an entire page or region legibly without excessive panning. Pad teaches zoom to fit operations driven by a bounding box in a numeric zoom factor, and shows an HTML browsing scenario where content is zoomed so that it fills the width of the screen horizontally. (EX1002, ¶106.) SVG confirmed that vector graphics could be scaled and transformed uniformly to establish zoom-based vector rendering for web content. (EX1002, ¶106.) A POSITA would have been motivated to incorporate Pad's zoom to fit vector rendering into the Zaurus browser to improve readability and navigation on a handheld display. (EX1002, ¶106.)

- x) **[1.h.2] “wherein the scalable vector representation of the HTML document is configured to enable a user to view the HTML document at one or more user-defined zoom levels by rendering the scalable vector representation on the touch-sensitive display using one or more respective scale factors in response to associated user inputs made via the touch-sensitive display, and”**
- xi) **[1.h.3] “wherein the interpreted page layout, functionality, and design of the content associated with the HTML document is preserved at each of the first zoom level and the one or more user-defined zoom levels.”**

Zaurus discloses user selectable reduce and magnify to control zooming while browsing (“wherein the scalable vector representation of the HTML document is configured to enable a user to view the HTML document at one or more user-defined

zoom levels...in response to associated user inputs made via the touch-sensitive display”). (EX1004, 14, 15 (“Touch the screen with a pen. Let's try various operations by touching the screen with a pen.”), 16, 46 (“Switch between reduced display and magnified display”), 47 (“To switch the screen from a reduced view to a magnified view, touch Reduce ▼ followed by Magnify. To switch back from a magnified view to a reduced view, touch Magnify ▼ followed by Reduce.”); EX1002, ¶107.)

Pad discloses iterative zoom that can be set by a parameter to render HTML content (“by rendering the scalable vector representation on the touch-sensitive display using one or more respective scale factors”). (EX1005, 13, 24-28, 29 (“Returns the current magnification of tagOrld for this specific render (it could be rendered multiple times if visible through different portals).”), 30 (“Returns the current view of the main window in “*xview yview zoom*” form. Here, (*xview*, *yview*) specifies the point at the center of the window, and *zoom* specifies the magnification.”); EX1007, 4 (“In Pad++, clicking and holding on the middle mouse button performs a smooth animated zoom in.”); EX1002, ¶108.)

SVG discloses zooming with scalability of text and images to preserve functionality and design across zoom levels (“wherein the interpreted page layout, functionality, and design of the content associated with the HTML document is preserved at each of the first zoom level and the one or more user-defined zoom

levels”). (EX1008, 10, 11 (“Transformed objects should have the option of exhibiting true scalability (where all attributes should scale uniformly, and linewidths, images and text should scale along with the sizes of the shapes)”), 11 (“These transformations should apply to all object types (shapes, images and text) in a uniform and consistent manner...SVG should use CSS positioning for establishing a viewport for an outermost SVG element...SVG will support local user coordinates”), 13, 14 (“It should be possible to assign a link to any graphic object or group.”); EX1002, ¶109.)

A POSITA would have implemented the combination to enable smooth zooming of Web content on handheld screens without compromising legibility or functionality. (EX1002, ¶110.) Zaurus discloses the handheld browser that already supported touch inputs to perform user-selected magnify and reduce operations (“in response to associated user inputs made via the touch-sensitive display”). Pad illustrates scalability at a variety of magnifications that preserves the layout, functionality, and design of the content displayed in the HTML browser. (EX1007, 86-96.) In the combination, SVG provides the scalable vector representation designed to zoom by numeric scale factors and Pad provides zoom parameters supporting the preservation of the content at a variety of zoom levels (“wherein the scalable vector representation of the HTML document is configured to enable a user to view the HTML document at one or more user-defined zoom levels by rendering”).

the scalable vector representation on the touch-sensitive display using one or more respective scale factors”). (EX1002, ¶110.) SVG’s uniform scaling would preserve layout, design, and interactivity while Pad’s techniques would parse HTML content into interactive objects with anchors that retained their link functionality after zooming (“wherein the interpreted page layout, functionality, and design of the content associated with the HTML document is preserved at each of the first zoom level and the one or more user-defined zoom levels”). (EX1002, ¶110.) A POSITA would have been motivated to combine these teachings to achieve the benefit of resolution-independent zooming, preserved page layout and design (i.e., uniform scaling), and continued functionality of links at various zoom levels. (EX1002, ¶110; *see also* EX1039, 39:17-40:1 (SoftView admitting that functionality includes clicking on a hyperlink).) In the resulting combination, there would be scalable vector representation rendered at user-defined zoom levels via touch-driven scale factors while preserving layout, functionality (i.e., links), and design. (EX1002, ¶110.)

7. Claim 2: “The mobile hand-held device of claim 1, wherein the mobile hand-held device is a mobile phone.”

Zaurus discloses a mobile handheld with cellular connectivity to access the Internet. (EX1004, 7-8 (adapters), 10 (portable), 24 (modem).) Pad is compatible with “a small home computer” and is “especially attractive for systems which have small screens, such as handheld computers (i.e. PDA’s).” (EX1007, 141.) SVG

supports for “small mobile devices.” (EX1008, 1, 4.) In view of the aforementioned teachings, a POSITA would have implemented the combination on “a mobile phone.” (EX1002, ¶111.)

8. Claim 3: “The mobile hand-held device of claim 1, wherein the mobile hand-held device is a hand-held computer.”

Zaurus discloses a mobile handheld computer with cellular connectivity to access the Internet, and it would have been obvious to implement the combined HTML to vector translation on the handheld computer as explained above. (EX1004, 10 (portable), 12 (device), 15 (touch screen), 18 (specifications); EX1007, 141; EX1008, 1, 4; EX1002, ¶112.)

9. Claim 4: “The mobile hand-held device of claim 1, wherein execution of the instructions further enables the user to zoom in on a user-selectable portion of a view of the HTML document in response to a user interface input made via the touch-sensitive display.”

Pad discloses zooming by enabling the user to click on text so that it fills the screen (“enables the user to zoom in on a user-selectable portion of a view of the HTML document”). (EX1005, 22, 30; EX1007, 4, 54, 61, 108.) SVG discloses supporting zooming and that event handling should be “at least as extensive as what is available for HTML (e.g., mouseovers, mouseclicks).” (EX1008, 14.) Zaurus describes enabling the user to touch the screen to select between reducing and magnifying displayed content (“in response to a user interface input made via the touch-sensitive display”). (EX1004, 10, 14-16, 39-41, 46-47.) In the Zaurus-Pad-

SVG combination, the zooming techniques of Pad and SVG would be incorporated into the magnification option of Zaurus so that a user would tap a portion of the HTML page to zoom in on the HTML document. (EX1002, ¶113.)

10. Claim 5: “The mobile hand-held device of claim 1, wherein the views of the HTML document are rendered in real-time to effect zooming operations.”

Pad discloses “animated zoom” in response to user input, which demonstrates continuous re-rendering as the user zooms. (EX1007, 4, 13, 22, 31.) Pad also discloses continuous zoom navigation, which demonstrates that zooming is a real-time operation. (EX1007, 66-70, 87-96.) Pad specifically discloses performing “animation” in “milliseconds.” (EX1005, 22-23, 32-33.) SVG discloses “[r]easonable display performance” and “fast performance.” (EX1008, 7.) Zaurus discloses touch driven HTML browsing. (EX1004, 10-11, 41.) In the Zaurus-Pad-SVG combination, a POSITA would have understood that high performing zoom that animates in milliseconds would enable the HTML document to be rendered in real-time. (EX1002, ¶114.)

11. Claim 6: “The mobile hand-held device of claim 5, wherein the zooming operations are effected in a fraction of a second.”

As explained above with respect to claim 5, a POSITA would have understood that high performing zoom that animates in milliseconds enables zooming operations within a fraction of a second. (EX1002, ¶115.)

12. Claim 7

- i) [7.a] “The mobile hand-held device of claim 1, wherein execution of the instructions performs further enables the device to:”**
- ii) [7.b] “generate a vector-based display list associated with the scalable vector representation; and”**

Limitation “performs further enables” is understood to mean “further enables.” A POSITA would have understood that a “display list” is a well-known, conventional computer graphics term for the internal list of commands or objects used to define the output. (EX1052, 195-196; *see also* EX1001, 19:42-44.) Pad discloses a “display list” associated with text, lines, polygons, and images for computing bounding boxes, which is a vector display list representation of objects and their layout. (EX1005, 65.) SVG discloses the scalable vector representation context for Web content. (EX1008, 3.) In the Zaurus-Pad-SVG combination, the browser would generate the display list to be associated with scalable vector representation. (EX1002, ¶116.)

- iii) [7.c] “employ the vector-based display list to render views of the HTML document on the touch-sensitive display at different scale factors to effect zooming operations.”**

Pad discloses rendering at various zoom levels so that content fills the screen, which demonstrates using the “display list” of vectorized objects to implement zooming. (EX1007, 31-36, 61, 93.) SVG discloses zooming and transformations

with uniform scaling. (EX1008, 3, 6, 7.) Zaurus discloses touch driven HTML browsing. (EX1004, 10-11, 40-41.) In the Zaurus-Pad-SVG combination, the list of vectorized objects would be used to render the content at various scales to implement the zooming features. (EX1002, ¶117.)

13. Claim 8: “The mobile hand-held device of claim 1, wherein the HTML document is an HTML parent document for a Web page.”

In the Zaurus-Pad-SVG combination, Zaurus describes downloading and displaying HTML 3.2 web pages, which a POSITA would have understood to be part of a top level “parent” document. (EX1004, 40; EX1002, ¶118.)

14. Claim 9: “The mobile hand-held device of claim 1, wherein the HTML elements includes an HTML table having a plurality of cells, each cell having at least one of associated text content and associated image content, and the CSS code includes CSS code defining how the table and its associated content is to be rendered.”

Zaurus discloses downloading and displaying HTML web pages by implementing HTML 3.2 to define and display HTML tables, images, and text. (EX1004, 40.) HTML 3.2 is the W3C HTML standard that includes the <TABLE> element and associated attributes for defining table layout. (EX1002, ¶119; EX1040.) A browser that “incorporates HTML 3.2” would handle HTML-based Web content that includes HTML code defining the layout of an “HTML table” such as <TABLE> elements. (EX1002, ¶119; EX1040.) Pad discloses a grid to arrange objects in rows and columns (“an HTML table having a plurality of cells”) as well

as loading text and images (“each cell having at least one of associated text content and associated image content”). (EX1005, 31, 60-61.) SVG discloses handling CSS for presenting Web content to apply properties to text and graphic objects (“the CSS code includes CSS code defining how the table and its associated content is to be rendered”). (EX1008, 1, 3, 7-8, 11-14.) In the Zaurus-Pad-SVG combination, given that the Zaurus browser already browsed HTML pages containing tables and SVG disclosed supporting CSS, a POSITA would have relied on CSS attributes for the tables (e.g., cell alignment, borders, widths, and font styling) to define how the table is to be rendered. (EX1002, ¶119.) The combination would apply CSS styling to the tables (HTML structures) on the resulting device to enable consistent page rendering. (EX1002, ¶¶119.)

15. Claim 10

- i) [10.a] “The mobile hand-held device of claim 1, wherein the content associated with the HTML document includes a plurality of images that are referenced using respective Universal Resource Locators (URLs) in the HTML document, and wherein execution of the instructions further enables the device to:”
- ii) [10.b] “for each of the plurality of images;”
 - (1) [10.b.1] “retrieve the image via the URL for the image; and”

Zaurus discloses retrieving HTML Web pages with images referenced with URLs, and selectable options such as for loading home pages (“the content”

associated with the HTML document includes a plurality of images that are referenced using respective Universal Resource Locators (URLs) in the HTML document”). (EX1004, 10, 39, 45-47.) Pad discloses HTML items that fetch content and images from URLs. (EX1005, 17, 29-30, 54-61.) SVG discloses linking to external media via URL and embedded images in Web content. (EX1008, 11, 13, 15-16.) In the Zaurus-Pad-SVG combination, Zaurus’s browser would incorporate Pad’s URL based HTML and image fetching and SVG’s linked and embedded graphics to retrieve the images referenced by URLs (“for each of the plurality of images; retrieve the image via the URL for the image”). (EX1002, ¶120.) A POSITA would have understood that for each image reference in an HTML page, the resulting device would retrieve the image via its URL to load (or cache) the page. (EX1002, ¶120.)

(2) [10.b.2] “generate a respective bounding box for the image; and”

Zaurus provides the mobile HTML browser that renders images. (EX1004, 39, 45-47.) Pad discloses generating bounding boxes for items and coordinates for views; and coordinate frames as bounding boxes for text and images to be scaled so that the height of the image is scaled to the height of the frame. (EX1005, 5, 6, 17-18, 34-35.) SVG discloses treating images as objects to which transformations could be applied, as well as links to external media. (EX1008, 9, 11, 15.) In the Zaurus-Pad-SVG combination, the image objects would have their geometry and

coordinates defined by a bounding box (“generate a respective bounding box for the image”). (EX1002, ¶121.)

iii) **[10.c] “employ the respective bounding boxes to scale images in the rendered views of the HTML document.”**

Zaurus discloses reduced and magnified page views. (EX1004, 46-47.) Pad discloses scaling text and images so that the height of the image is scaled to the height of the frame for optimizing image rendering to enable real time zooming of images. (EX1005, 34-35.) Pad further discloses zooming and scaling by using each item’s bounding box to draw the object. (EX1005, 5, 6, 16, 17.) SVG discloses transformations that can be applied to images for scalability for zoom and pan. (EX1008, 11-13, 15.) In the Zaurus-Pad-SVG combination, starting with the reduced and magnified views of Zaurus, a POSITA would have incorporated the rendering techniques of Pad and SVG where each image is an object associated with a bounding box (“employ the respective bounding boxes to scale images in the rendered views of the HTML document”). (EX1002, ¶122.) In the resulting combination, each image would be retrieved via its URL, a bounding box would be computed for each image, and the bounding box would be used to scale the image during zoom. (EX1002, ¶122.)

16. Claim 11

- i) **[11.a] “The mobile hand-held device of claim 1, wherein the HTML document includes a reference to an XML document containing XML content, and wherein execution of the instructions further enables the mobile hand-held device to:”**
- ii) **[11.b] “retrieve the XML document and processing the XML content in connection with rendering the first representation of the HTML document, wherein the XML content is used to define one or more of the interpreted page layout, functionality, and design of the content associated with the HTML document.”**

Zaurus in view of Pad and SVG discloses and/or suggests this limitation. (EX1002, ¶123.) Zaurus discloses an HTML browser. (EX1004, 10, 40.) SVG discloses that XML-based vector graphics are designed to be implemented in browsers and coordinated with CSS and links. (EX1008, 1, 4-7, 9, 14-15.) The authors of SVG, W3C, described that XML was a standard Extensible Markup Language (XML) in 1998. (EX1046.) A POSITA would have understood that supporting XML naming and tags would involve referencing external XML documents in the HTML document (“wherein the HTML document includes a reference to an XML document containing XML content”). (EX1002, ¶123.)

Pad discloses routine resource references in HTML and integrating fetched results into the rendered representation. (EX1005, 5 (“Retrieve a URL over the internet in the background”), 62 (“creating an html typically starts a network communication process for fetching the URL”).) SVG discloses the SVG supports

CSS, DOM, and compatibility with various browsers. (EX1008, 3, 4.) Given that both Zaurus and Pad disclosed fetching resources via URL, a POSITA would have understood that SVG supports retrieving XML with style rules and the DOM while rendering. (EX1002, ¶124.) In the resulting combination, rendering HTML would have involved referencing and retrieving XML content for handling the layout, appearance, or interactions of the page (“retrieve the XML document and processing the XML content in connection with rendering the first representation of the HTML document”). (EX1002, ¶124.)

SVG further discloses that SVG is “an application of XML,” which means that SVG files are XML documents intended to be retrieved and rendered by a browser (“wherein the XML content is used to define one or more of the interpreted page layout, functionality, and design of the content associated with the HTML document”). (EX1008, 4; EX1002, ¶125.)

- 17. Claim 12: “The mobile hand-held device of claim 1, wherein the primary datum is an origin in a two-dimensional XY coordinate system and the vectors from the primary datum to the object datums for the HTML object bounding boxes comprise XY coordinates in the coordinate system.”**

Zaurus describes rendering HTML on a mobile device. (EX1004, 45-47.) Pad describes APIs and view control that uses x and y coordinates; centering the view at x and y; and returning the x and y coordinates for the views. (EX1005, 17-19, 29-32.) SVG discloses a coordinate system with coordinates, and transformation

matrices. (EX1008, 10, 11) In the Zaurus-Pad-SVG combination, the coordinates and the matrices mathematically presuppose an origin in a two dimension XY space (“the primary datum is an origin in a two-dimensional XY coordinate system”). (EX1002, ¶126.)

Pad discloses a frame as a bounding box and maps coordinates into the box for scaling images and text. (EX1005, 34-35.) SVG describes positioning objects with coordinates and using matrices to scale images and text. (EX1008, 10-12.) A POSITA would have understood that Pad’s bounding boxes would be represented by x and y vectors from an origin in the coordinate system as taught by SVG’s XY systems with transform vectors. (EX1002, ¶127.) Representing bounding boxes as XY vectors from an origin in a 2D system would have been obvious from Pad’s bounding boxes and SVG’s transformations and would have been incorporated into the Zaurus browser. (EX1002, ¶127.) In the resulting combination, a POSITA would have understood that representing each object’s layout position as a vector from an origin to the object’s bounding box is the conventional approach to store a coordinate for an object and apply the matrix to scale and compute the object’s position on the page. (“the vectors from the primary datum to the object datums for the HTML object bounding boxes comprise XY coordinates in the coordinate system”). (EX1002, ¶127.)

18. Claims 13-35

Claims 13-35⁶ are rendered obvious based on the combination of Zaurus, Pad, and SVG for the same reasons as provided above with respect to claims 1-12. (EX1002, ¶¶128-179.) In particular, all of the claim language in claims 13-35 is substantively addressed in the analysis above for claims 1-12:

Claim 13-35	Claim 1-12
[13.pre]	[1.pre]-[1.d]
[13.a]	[1.e]
[13.b]	[1.f]
[13.b.1]	[1.f.1]
[13.b.2]	[1.f.2]
[13.b.3]	[1.f.3]
[13.b.4.a]	[1.f.4.a]
[13.b.4.b]	[1.f.4.b]
[13.c]	[1.g]
[13.c.1]	[1.g.1]
[13.c.2]	[1.g.2]
[13.c.2.a]	[1.g.2.a]
[13.c.2.b]	[1.g.2.b]
[13.c.2.c]	[1.g.2.c]
[13.d.1]	[1.h.1]
[13.d.2]	[1.h.2]
[13.d.3]	[1.h.3]
14	2
15	3
16	4
17	5
18	6
[19.a]	[7.a]
[19.b]	[7.b]

⁶ The language of claims 13-35 is provided in Appendix A.

[19.c]	[7.c]
20	8
[21.a]	[10.a]
[21.b]	[10.b]
[21.b.1]	[10.b.1]
[21.b.2]	[10.b.2]
[21.c]	[10.c]
22	9
[23.a]	[10.a]
[23.b]	[10.b]
[23.b.1]	[10.b.1]
[23.b.2]	[10.b.2]
[23.c]	[10.c]
[24.a]	[11.a]
[24.b]	[11.b]
25	12
[26.pre]-[26.d]	[1.pre]-[1.d]
[26.e]	[1.e]
[26.f]	[1.f]
[26.f.1]	[1.f]
[26.f.2]	[1.f.2]
[26.f.3]	[1.g.1]
[26.f.4]	[1.g.2]
[26.f.4.a]	[1.g.2.a]
[26.f.4.b]	[1.g.2.b]
[26.f.4.c]	[1.g.2.c]
[26.g]	[1.h.1]-[1.h.3]
[26.g.1]	[1.h.1]
[26.g.2]	[1.h.2]
[26.g.3]	[1.h.3]
27	2
28	3
29	4
30	5
31	6
[32.a]	[7.a]
[32.b]	[7.b]
[32.c]	[7.c]

33	8
34	9
[35.a]-[35.b]	[1.f.3]
[35.c]	[1.f.4.b]

B. Ground 2: Zaurus, Pad, SVG, and SVF Render Claims 1-35 Obvious

As discussed above, the Zaurus-Pad-SVG combination renders claims 1-35 obvious. However, to the extent Patent Owner argues, or the Board finds, that certain features directed to the “datum” that are recited in limitations [1.g] and [12] (and similar limitations in limitations [13.c], [25], and [26.f]) are not met by the Zaurus-Pad-SVG combination, those features nonetheless would have been obvious in view of SVF. (EX1002, ¶180.) The Board has previously found (and been affirmed by the Federal Circuit) that there is motivation to combine and a reasonable expectation of success for incorporating SVF into the combination. (EX1013, 2-3.)

1. SVF

The Simple Vector Format (SVF) includes interactive features such as layers to manage object visibility, hyperlinks that allow users to click regions to perform actions, and notifications that trigger specific behaviors when the viewer reaches a certain zoom level. (EX1044, 1, 4-7.) The format relies on a coordinate system where the origin (0,0) starts in the lower-left corner, with coordinates specified as positive integers. (EX1044, 1-2; EX1044, 5-7; EX1044, 1; EX1002, ¶24.)

2. Motivation to Combine

A POSITA would have been motivated to incorporate SVF into the Zaurus-Pad-SVG combination to optimize the zooming capabilities of the combination. (EX1002, ¶181.) SVF was also based on basic W3C software standards that a POSITA would have been motivated to combine with Zaurus-Pad-SVG. (*See e.g.*, EX1001, 4:67-5:3; EX1008; EX1035; EX1037; EX1044; EX1046.) In the combination, SVF would make the zooming features more efficient and responsive for the user and improve the user experience. (EX1044, 1-2, 6-7; EX1002, ¶181.)

While the Zaurus-Pad-SVG combination defines reference points and an origin for layout and zooming, a POSITA would have been motivated to incorporate SVF's "coordinate system" that defines that the "origin (0,0) is in the lower left corner with increasing x coordinates moving to the right and increasing y coordinates moving up." (EX1044, 1-2.) A POSITA would have been motivated to incorporate this integer-based coordinate system to conduct the vector calculations for rendering the objects. Furthermore, while the Zaurus-Pad-SVG combination mentions utilizing grid arrangements, a POSITA would have been motivated to utilize SVF's support for "Scale and base point" data so that when "original coordinate system needed to be recreated" from the stored values, SVF would enable mapping the layout of an HTML Web page onto the integer-based vector grid. (EX1002, ¶182.)

3. Reasonable Expectation of Success

A POSITA would have had a reasonable expectation of success in implementing SVF in the Zaurus-Pad-SVG combination. (EX1002, ¶183.) As an initial matter, the '154 patent explains that “SVF has been under consideration by the W3C (World Wide Web Consortium) for adoption as a standard for vector content on the World Wide Web.” (EX1001, 4:64-68.) Zaurus, Pad, and SVG were also based on W3C standards that a POSITA would have understood to be interoperable. (*See e.g.*, EX1008; EX1035; EX1037; EX1044; EX1046.) SVF includes standard C++ code, such as “svfout.c” to handle the output of commands, which a POSITA would have integrated into SVF’s rendering logic and Pad’s “C++ core” of Pad. (EX1007, 140-141; EX1044, 2; EX1002, ¶183.) The technical architecture of SVF is compatible with the Zaurus’s “32-bit RISC” processor because SVF utilizes “integer” command parameters and coordinates. (EX1004, 18; EX1044, 1; EX1002, ¶183.)

4. Claims 1-35

i) [1.g.1] / [13.c.1] / [26.f.3]⁷

SVF defines a global coordinate system where the “origin (0,0) is in the lower left corner” and objects in the SVF file are drawn using integer coordinates relative to this defined coordinate system, which corresponds to the “primary datum” for the “page layout.” (EX1044, 1; EX1002, ¶184.) SVF defines the “image’s size (or extents) so the program viewing it will know how to display it initially” (“interpreted page layout”). (EX1044, 2.) SVF’s coordinate system is used to create a scalable representation because the “size of the image can change,” and the format supports “Scale and base point” parameters. (EX1044, 2, 7.) SVF’s “Notifications” trigger actions based on “magnification” (EX1044, 6), further demonstrating that SVF supports operations for scalable viewing. (EX1002, ¶184.)

ii) [1.g.2.a] / [13.c.2.a] / [26.f.4.a]

SVF commands use specific coordinates to position each object. SVF supports HTML-equivalent objects: “text,” “basic drawing objects” like “circles” and “arcs,” and “hyperlinks.” (EX1044, 1.) SVF also discloses “embedding SVF images in HTML documents” (EX1044, 7.) SVF objects can be “Filled regions” or

⁷ The claim language is provided above in Ground 1 and is not repeated in Ground 2.

specific shapes like “rectangles.” (EX1044, 1, 4.) These shapes define the boundaries (“bounding box”) of an image or graphic. The “MoveTo” command “moves the current point to a new location.” (EX1044, 2.) This “current point” is the reference point (“object datum”) for subsequent drawing commands. For example, “Polylines start drawing from the current point.” (EX1044, 2.) The coordinates specified in these commands correspond to the “layout location datum” derived from the original layout. (EX1044, 1; EX1002, ¶185.)

iii) [1.g.2.b] / [13.c.2.b] / [26.f.4.b]

SVF is defined as a “Simple Vector Format” where “objects are drawn in the order they occur in the file,” which would involve handling a list of vectors. (EX1044, 1; EX1002, ¶186.) Specifying a coordinate pair (X, Y) for a “MoveTo” command defines a vector from the origin (0,0) (“primary datum”) to that point (X, Y) (“object datum”). (EX1044, 1, 2.) Using integers for coordinates allows the commands to be mathematically scaled vectors. (EX1002, ¶186.) SVF also discloses “Relative lines and polylines,” where “each point describes an offset from the current point,” which is a generation of vectors between datums. (EX1044, 3; EX1002, ¶186.)

iv) [1.g.2.c] / [13.c.2.c] / [26.f.4.c]

SVF discloses a hyperlink that defines a region (“vector”) where the user can click to “perform an action” (“HTML object”), which is a “reference that links the

HTML object to the vector.” (EX1044, 1, 5.) SVF allows defining a “2 dimensional” link region, which is “chosen by picking inside the region.” (EX1044, 5.) Furthermore, SVF discloses grouping “objects” into “Layers” for controlling visibility, which creates a link between the “objects” and their “vectors.” (EX1044, 1, 6; EX1002, ¶187.)

v) [12] / [25]

SVF defines its coordinate system as a two-dimensional XY system where “the origin (0,0) is in the lower left corner with increasing x coordinates moving to the right and increasing y coordinates moving up” and coordinates are specified as “positive integers.” (EX1044, 1.) In the combination, the coordinates are part of the “vector” and the origin (“primary datum”) as required by the claim. (EX1002, ¶188.) The coordinates would enable the calculation of scale factors for zooming operations, such as those configured by the “Notifications” logic. (EX1044, 6; EX1002, ¶188.)

XII. DISCRETIONARY DENIAL IS NOT APPROPRIATE

In view of the Patent Office’s interim process regarding discretionary denial, Petitioner will substantively address this issue during discretionary denial briefing.

XIII. CONCLUSION

Petitioner requests institution of IPR for claims 1-35 of the '154 patent based on the grounds specified in this petition.

Respectfully submitted,

Dated: March 5, 2026

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

CERTIFICATE OF COMPLIANCE

Pursuant to 37 C.F.R. § 42.24(d), the undersigned certifies that the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 10,083,154 contains, as measured by the word-processing system used to prepare this paper, 13,980 words. This word count does not include the items excluded by 37 C.F.R. § 42.24 as not counting towards the word limit.

Respectfully submitted,

Dated: March 5, 2026

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

CERTIFICATE OF SERVICE

I hereby certify that on March 5, 2026, I caused a true and correct copy of the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 10,083,154 and supporting exhibits to be served via express mail on the Patent Owner at the following correspondence address of record as listed on Patent Center:

LAW OFFICE OF R. ALAN BURNETT
4108 131ST AVE. SE
BELLEVUE, WA 98006

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)

APPENDIX A (CLAIM LISTING)

[1.pre]	A mobile hand-held device, comprising:
[1.a]	a processor;
[1.b]	a wireless communications device, to facilitate wireless communication with a network that supports access to the Internet;
[1.c]	a touch-sensitive display; and
[1.d]	flash memory, operatively coupled to the processor, in which a plurality of instructions are stored comprising a plurality of software components including an HTML rendering engine, wherein the instructions are configured to be executed by the processor to enable the mobile hand-held device to,
[1.e]	receive an HTML document including HTML code and cascading style sheet (CSS) code and content associated with the HTML document, the HTML code comprising a plurality of HTML elements including at least one HTML paragraph element, at least one HTML image element, and at least one HTML hyperlink element;
[1.f]	process the HTML document with the HTML rendering engine to render a first representation of the HTML document having an interpreted page layout, functionality, and design of the content associated with the HTML document that is in accordance with the HTML code and CSS code, wherein rendering the first representation of the HTML document includes,
[1.f.1]	parsing the HTML document to identify the plurality of HTML elements;
[1.f.2]	logically grouping content associated with HTML elements into HTML objects;
[1.f.3]	generating page layout information including a bounding box for each HTML object; and
[1.f.4.a]	storing information that links each HTML object with its corresponding page layout information,
[1.f.4.b]	wherein the page layout information further includes information from which a page layout location of each of the bounding boxes can be determined;
[1.g]	translate the first representation of the HTML document to generate a scalable vector representation of the HTML document wherein generating the scalable vector representation includes,

[1.g.1]	defining a primary datum corresponding to the interpreted page layout;
[1.g.2]	for each HTML object,
[1.g.2.a]	defining an object datum corresponding to a layout location datum for the HTML object's bounding box;
[1.g.2.b]	generating a vector from the primary datum to the object datum for the HTML object's bounding box; and
[1.g.2.c]	creating a reference that links the HTML object to the vector that is generated; and
[1.h.1]	render the scalable vector representation of the HTML document on the touch-sensitive display using a first scale factor to display the HTML document at a first zoom level under which the HTML document is displayed to fit across a width of the touch-sensitive display,
[1.h.2]	wherein the scalable vector representation of the HTML document is configured to enable a user to view the HTML document at one or more user-defined zoom levels by rendering the scalable vector representation on the touch-sensitive display using one or more respective scale factors in response to associated user inputs made via the touch-sensitive display, and
[1.h.3]	wherein the interpreted page layout, functionality, and design of the content associated with the HTML document is preserved at each of the first zoom level and the one or more user-defined zoom levels.
[2]	The mobile hand-held device of claim 1, wherein the mobile hand-held device is a mobile phone.
[3]	The mobile hand-held device of claim 1, wherein the mobile hand-held device is a hand-held computer.
[4]	The mobile hand-held device of claim 1, wherein execution of the instructions further enables the user to zoom in on a user-selectable portion of a view of the HTML document in response to a user interface input made via the touch-sensitive display.
[5]	The mobile hand-held device of claim 1, wherein the views of the HTML document are rendered in real-time to effect zooming operations.
[6]	The mobile hand-held device of claim 5, wherein the zooming operations are effected in a fraction of a second.
[7.a]	The mobile hand-held device of claim 1, wherein execution of the instructions performs further enables the device to:

[7.b]	generate a vector-based display list associated with the scalable vector representation; and
[7.c]	employ the vector-based display list to render views of the HTML document on the touch-sensitive display at different scale factors to effect zooming operations.
[8]	The mobile hand-held device of claim 1, wherein the HTML document is an HTML parent document for a Web page.
[9]	The mobile hand-held device of claim 1, wherein the HTML elements includes an HTML table having a plurality of cells, each cell having at least one of associated text content and associated image content, and the CSS code includes CSS code defining how the table and its associated content is to be rendered.
[10.a]	The mobile hand-held device of claim 1, wherein the content associated with the HTML document includes a plurality of images that are referenced using respective Universal Resource Locators (URLs) in the HTML document, and wherein execution of the instructions further enables the device to:
[10.b]	for each of the plurality of images;
[10.b.1]	retrieve the image via the URL for the image; and
[10.b.2]	generate a respective bounding box for the image; and
[10.c]	employ the respective bounding boxes to scale images in the rendered views of the HTML document.
[11.a]	The mobile hand-held device of claim 1, wherein the HTML document includes a reference to an XML document containing XML content, and wherein execution of the instructions further enables the mobile hand-held device to:
[11.b]	retrieve the XML document and processing the XML content in connection with rendering the first representation of the HTML document, wherein the XML content is used to define one or more of the interpreted page layout, functionality, and design of the content associated with the HTML document.
[12]	The mobile hand-held device of claim 1, wherein the primary datum is an origin in a two-dimensional XY coordinate system and the vectors from the primary datum to the object datums for the HTML object bounding boxes comprise XY coordinates in the coordinate system.
[13.pre]	A method performed by a mobile hand-held device including a touch-sensitive display and an HTML rendering engine, comprising:

[13.a]	receiving an HTML document including HTML code and cascading style sheet (CSS) code and content associated with the HTML document, the HTML code comprising a plurality of HTML elements including at least one HTML paragraph element, at least one HTML image element, and at least one HTML hyperlink element;
[13.b]	processing the HTML document with the HTML rendering engine to render a first representation of the HTML document having an interpreted page layout, functionality, and design of the content associated with the HTML document that is in accordance with the HTML code and CSS code, wherein rendering the first representation of the HTML document includes,
[13.b.1]	parsing the HTML document to identify the plurality of HTML elements;
[13.b.2]	logically grouping content associated with HTML elements into HTML objects;
[13.b.3]	generating page layout information including a bounding box for each HTML object; and
[13.b.4.a]	storing information that links each HTML object with its corresponding page layout information,
[13.b.4.b]	wherein the page layout information further includes information from which a page layout location of each of the bounding boxes can be determined;
[13.c]	translating the first representation of the HTML document to generate a scalable vector representation of the HTML document wherein generating the scalable vector representation includes,
[13.c.1]	defining a primary datum corresponding to the interpreted page layout;
[13.c.2]	for each HTML object,
[13.c.2.a]	defining an object datum corresponding to a layout location datum for the HTML object's bounding box;
[13.c.2.b]	generating a vector from the primary datum to the object datum for the HTML object's bounding box; and
[13.c.2.c]	creating a reference that links the HTML object to the vector that is generated; and
[13.d.1]	render the scalable vector representation of the HTML document on the touch-sensitive display using a first scale factor to display the HTML document at a first zoom level under which the HTML document is displayed to fit across a width of the touch-sensitive display,

[13.d.2]	wherein the scalable vector representation of the HTML document is configured to enable a user to view the HTML document at one or more user-defined zoom levels by rendering the scalable vector representation on the touch-sensitive display using one or more respective scale factors in response to associated user inputs made via the touch-sensitive display, and
[13.d.3]	wherein the interpreted page layout, functionality, and design of the content associated with the HTML document is preserved at each of the first zoom level and the one or more user-defined zoom levels.
[14]	The method of claim 13, wherein the mobile hand-held device is a mobile phone.
[15]	The method of claim 13, wherein the mobile hand-held device is a hand-held computer.
[16]	The method of claim 13, further comprising enabling the user to zoom in on a user-selectable portion of a view of the HTML document in response to a user interface input made via the touch-sensitive display.
[17]	The method of claim 13, wherein the views of the HTML document are rendered in real-time to effect zooming operations.
[18]	The method of claim 17, wherein the zooming operations are effected in a fraction of a second.
[19.a]	The method of claim 13, further comprising:
[19.b]	generating a vector-based display list associated with the scalable vector representation; and
[19.c]	employing the vector-based display list to render views of the HTML document on the touch-sensitive display at different scale factors to effect zooming operations.
[20]	The method of claim 13, wherein the HTML document is an HTML parent document for a Web page.
[21.a]	The mobile hand-held device of claim 20, wherein the HTML document includes references to a plurality of images, each reference comprising a respective Universal Resource Locators (URLs), and wherein execution of the instructions performs further operations comprising:
[21.b]	for each of the plurality of images;
[21.b.1]	retrieving the image via the URL for the image; and
[21.b.2]	generating a respective bounding box for the image; and

[21.c]	employing the respective bounding boxes to scale images in the rendered views of the HTML document.
[22]	The method of claim 13, wherein the HTML elements includes an HTML table having a plurality of cells, each cell having at least one of associated text content and associated image content, and the CSS code includes CSS code defining how the table and its associated content is to be rendered.
[23.a]	The method of claim 13, wherein the content associated with the HTML document includes a plurality of images that are referenced using respective Universal Resource Locators (URLs) in the HTML document, further comprising:
[23.b]	for each of the plurality of images;
[23.b.1]	retrieving the image via the URL for the image; and
[23.b.2]	generating a respective bounding box for the image; and
[23.c]	employing the respective bounding boxes to scale images in the rendered views of the HTML document.
[24.a]	The method of claim 13, wherein the HTML document includes a reference to an XML document containing XML content, further comprising:
[24.b]	retrieving the XML document and processing the XML content in connection with rendering the first representation of the HTML document, wherein the XML content is used to define one or more of the interpreted page layout, functionality, and design of the content associated with the HTML document.
[25]	The method of claim 13, wherein the primary datum is an origin in a two-dimensional XY coordinate system and the vectors from the primary datum to the object datums for the HTML object bounding boxes comprise XY coordinates in the coordinate system.
[26.pre]	A mobile hand-held device, comprising:
[26.a]	a processor;
[26.b]	a wireless communications device, to facilitate wireless communication with a network that supports access to the Internet;
[26.c]	a touch-sensitive display; and

[26.d]	flash memory, operatively coupled to the processor, in which a plurality of instructions are stored comprising a plurality of software components including an HTML rendering engine, wherein the instructions, when executed by the mobile hand-held device, enable the mobile hand-held device to,
[26.e]	receive an HTML document comprising HTML-based content including HTML code and cascading style sheet (CSS) code and content associated with the HTML document;
[26.f]	produce scalable content by,
[26.f.1]	processing the HTML-based content with the rendering engine to render an interpreted page layout, functionality, and design of the content associated with the HTML document in accordance with the HTML code and CSS code;
[26.f.2]	logically grouping selected content into HTML objects, each HTML object including associated display content;
[26.f.3]	defining a primary datum corresponding to the interpreted page layout; and,
[26.f.4]	for each HTML object,
[26.f.4.a]	defining an object datum corresponding to a layout location datum for the HTML object's associated display content;
[26.f.4.b]	generating a vector from the primary datum to the object datum for the HTML object; and
[26.f.4.c]	creating a reference that links the HTML object to the vector that is generated; and
[26.g]	employ at least one of the scalable content or data derived therefrom to,
[26.g.1]	render a view of the HTML document on the touch-sensitive display at a first zoom level under which the HTML document is displayed to fit across a width of the touch-sensitive display; and
[26.g.2]	render views of the HTML document on the touch-sensitive display in response to associated user inputs to enable the HTML document to be viewed at various zoom levels and panned views
[26.g.3]	while preserving the interpreted page layout, functionality, and design of the content associated with the HTML document at each zoom level and panned view including the first zoom level.
[27]	The mobile hand-held device of claim 26, wherein the mobile hand-held device is a mobile phone.

[28]	The mobile hand-held device of claim 26, wherein the mobile hand-held device is a hand-held computer.
[29]	The mobile hand-held device of claim 26, wherein execution of the instructions further enables the user to zoom in on a user-selectable portion of a view of the HTML document in response to a user interface input made via the touch-sensitive display.
[30]	The mobile hand-held device of claim 26, wherein the views of the HTML document are rendered in real-time to effect zooming operations.
[31]	The mobile hand-held device of claim 30, wherein the zooming operations are effected in a fraction of a second.
[32.a]	The mobile hand-held device of claim 26, wherein execution of the instructions further enables the device to:
[32.b]	generate a vector-based display list associated with the scalable content; and
[32.c]	employ the vector-based display list to render views of the HTML document on the touch-sensitive display at different scale factors to effect zooming operations.
[33]	The mobile hand-held device of claim 26, wherein the HTML document is an HTML parent document for a Web page.
[34]	The mobile hand-held device of claim 26, wherein the HTML-based content includes an HTML table, and the CSS code includes CSS code defining how the table is to be formatted in the interpreted page layout page of the HTML document.
[35.a]	The mobile hand-held device of claim 26, wherein execution of the instructions further enables the device to:
[35.b]	for each HTML object, generate a bounding box for the HTML object; and
[35.c]	employ the bounding boxes to interpret the interpreted page layout of the HTML document.