

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner

v.

SOFTVIEW LLC,
Patent Owner

Patent No. 9,519,729

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 9,519,729**

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 '729 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	16
XI.	DETAILED EXPLANATION OF GROUNDS.....	17
	A. Ground 1: Zaurus, Pad, and SVG Render Claims 1-22 Obvious.....	17
	1. Zaurus (EX1004).....	17
	2. Pad (EX1005 and EX1007)	18
	3. SVG (EX1008).....	18
	4. Motivation to Combine	18
	5. Reasonable Expectation of Success.....	19
	6. Claim 1	20
	7. Claim 2: “The method of claim 1, wherein the mobile device comprises a mobile phone.”	38
	8. Claim 3: “The method of claim 1, further comprising enabling a user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input made via the touch-sensitive display.”	39

9.	Claim 4: “The method of claim 1, wherein a display of the Web page is re-rendered in real-time to effect zooming operations.”	39
10.	Claim 5	40
11.	Claim 6: “The method of claim 1, further comprising enabling a user to pan views of the Web page in real-time in response to corresponding user inputs made via the touch-sensitive display.”	43
12.	Claim 7	44
13.	Claim 8	48
14.	Claim 9	51
15.	Claim 10: “The method of claim 9, wherein under the first view the Web page width is rendered to fit across the width of the display area of the touch-sensitive display, the display area used for displaying the HTML-based Web content.”	53
16.	Claim 11: “The method of claim 1, wherein the HTML-based Web content further includes HTML code defining the layout for at least one table.”	54
17.	Claim 12: “The method of claim 1, wherein the HTML-based Web content further includes HTML code for two or more frames.”	54
18.	Claim 13	55
19.	Claim 14	56
20.	Claim 15: “The method of claim 14, wherein the mobile device comprises a mobile phone.”	58
21.	Claim 16: “The method of claim 14, further comprising enabling a user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input made via the touch-sensitive display.”	58
22.	Claim 17	59
23.	Claim 18	60

24.	Claim 19: “The method of claim 14, wherein the HTML-based Web content further includes HTML code for at least one table.”	60
25.	Claim 20: “The method of claim 14, wherein the HTML-based Web content further includes HTML code for two or more frames.”	60
26.	Claim 21	61
27.	Claim 22: “The method of claim 14, wherein the HTML-based Web content further includes HTML code for at least one form.”	66
B.	Ground 2: Zaurus, Pad, SVG, and SVF Render Claim 21 Obvious.....	66
1.	SVF	67
2.	Motivation to Combine	67
3.	Reasonable Expectation of Success	68
4.	Claim 21	69
XII.	DISCRETIONARY DENIAL IS NOT APPROPRIATE	71
XIII.	CONCLUSION.....	71

LIST OF EXHIBITS

EX1001	U.S. Patent No. 9,519,729 to Rohrbaugh et al. (“the ’729 patent”)
EX1002	Declaration of Benjamin Goldberg, Ph.D.
EX1003	Prosecution History of U.S. Patent App. No. 12/941,106
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 Rohrbaugh 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 Rohrbaugh et al. (“the ’353 patent”)
EX1016	RESERVED

Petition for *Inter Partes* Review
Patent No. 9,519,729

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	RESERVED
EX1029	RESERVED
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	RESERVED
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 1-22 (“the challenged claims”) of U.S. Patent No. 9,519,729 (“the ’729 patent”) (EX1001) assigned to SoftView LLC (“Patent Owner”). The ’729 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 ’729 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 ’729 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 ’729 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 '729 patent originates from U.S. Patent Application No. 12/941,106, filed on November 8, 2010, 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 related U.S. Patent Nos. 10,083,154 (“’154 patent”), 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 an assertion 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 '154, '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 '729 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 1-22 of the '729 patent and cancellation of these claims as unpatentable. The challenged claims should be found unpatentable on the following ground:

Ground 1: Claims 1-22 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: Claim 21 is 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 '729 patent is June 12, 2000. (EX1001, 1-2.) However, Patent Owner's contentions against Samsung assert that the '729 patent is entitled to a priority date of April 7, 2001 (EX1010, 5), and Patent Owner did not dispute the Examiner's assertion during prosecution that the claims are entitled to a priority date of June 8, 2001 (EX1003, 474, 644).² Regardless of

² Petitioner does not concede that the '729 patent is entitled to any of these dates.

which of the three dates is applied, the above references are prior art, and indeed the Board has previously credited that Zaurus, Pad, SVG, and SVF qualify as prior art to (i) the '926 patent to which the '729 patent claims priority and (ii) the '353 patent that is a divisional of the '099 patent to which the '729 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 '729 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 '729 PATENT

The '729 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., 1:59-2:59.) The '729 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), 16:36-47; *see also id.* at 14:63-16:35.) The '729 patent utilizes the SVF (Simple Vector Format) considered by the W3C (World Wide Web Consortium) to translate content into a scalable vector representation. (EX1001, 4:53-67, 6:10-24, 6:57-7:4, 8:63-9:3, 12:14-22; EX1002, ¶¶15-20.)

³ Petitioner submits the declaration of Dr. Benjamin Goldberg, an expert in the field of the '729 patent. (EX1002, ¶¶2-6; EX1006.)

VIII. RELATED PATENTS

A. The '926 patent

The '926 patent is a parent of the '729 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 '729 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 '729 patent.

In November 2010, SoftView filed a patent application that issued as the '729 patent. (EX1001, 1.) During prosecution, the Examiner rejected the '729 claims for double patenting, finding they were “not patentably distinct” from the '926 and '353 patents. (EX1003, 173.) SoftView confirmed this substantive similarity by filing a terminal disclaimer. (EX1003, 251; *see* MPEP § 804.)

The claims of the '729 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.)

Despite having these references and listing them on the patent (EX1001, 3-4), the Examiner only raised rejections based on the Pad++ browser and failed to apply

the necessary estoppel, issue preclusion, and/or the combination based on Zaurus, Pad, SVG, and SVF. Instead, SoftView was allowed to obtain the patentably indistinct claims of the '729 patent in December 2016. (EX1001, 1.)

In this proceeding, Petitioner applies those same Zaurus, Pad, SVG, and SVF 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, ¶21, Appendices A-E (Dr. Goldberg describing this same correspondence).)⁴

'729 Patent	'353 Patent (EX1002, Appendix B)	'926 Patent (EX1002, Appendices C-E)
[1.pre], [14.pre]	[118.pre]	(IPR2013-00004) ⁵ [30.pre] (90/009,995) [1.pre] (95/000,635) [79.pre]
[1.a], [14.a], [22]	[1.b], [66], [317.e]	(IPR2013-00004) [30.f], [40] (90/009,995)

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. 9,519,729

		[1.f], [12] (95/000,635) [79.g], [29], [64]
[1.b], [14.b]	[149.b], [183], [66]	(IPR2013-00004) [30.g], [30.h] (90/009,995) [1.g], [1.h] (95/000,635) [79.h], [78]
[1.c], [14.c]	[149.d], [149.e], [66]	(IPR2013-00004) [30.i], [30.h] (90/009,995) [1.i.1], [1.i.2] (95/000,635) [79.i.1], [79.i.2], [78]
[1.d], [14.d]	[149.e], [66]	(IPR2013-00004) [52.i] (90/009,995) [1.i.2.a] (95/000,635) [79.i.2], [78]
[2], [15]	[36.pre]	(IPR2013-00004) [30.pre] (90/009,995) [2] (95/000,635) [53]
[3], [16]	[317.h]	(IPR2013-00004) [31] (90/009,995) [4] (95/000,635) [82]
[4]	[36.e]	(IPR2013-00004) [43.c] (90/009,995) [5] (95/000,635) [57]

Petition for *Inter Partes* Review
Patent No. 9,519,729

[5.a]-[5.d], [17]	[149.b], [66], [317.i]	(IPR2013-00004) [30.g], [30.h], [30.i], [30.j] (90/009,995) [6.a]-[6.d] (95/000,635) [58.a]-[58.d], [78]
[6]	[149.e]	(IPR2013-00004) [52.i] (90/009,995) [10] (95/000,635) [62]
[7.a]-[7.b], [14.e]	[48], [317.k]	(IPR2013-00004) [40] (90/009,995) [12] (95/000,635) [64]
[8.a]-[8.c], [18]	[51], [52], [317.j]	(IPR2013-00004) [41] (90/009,995) [13] (95/000,635) [65]
[9.a]-[9.c]	[1.b], [118.c]	(IPR2013-00004) [75.a]-[75.c] (90/009,995) [25.a]-[25.c] (95/000,635) [63]
[10]	[118.c]	(IPR2013-00004) [75.c] (90/009,995) [25.a]-[25.c] (95/000,635) [63]
[11], [19]	[1.b]	(IPR2013-00004) [30.f] (90/009,995)

Petition for *Inter Partes* Review
Patent No. 9,519,729

		[1.f] (95/000,635) [79.g]
[12], [20]	[1.b]	(IPR2013-00004) [30.f] (90/009,995) [1.f] (95/000,635) [79.g]
[13.a]-[13.c]	[149.e]	(IPR2013-00004) [31] (90/009,995) [14] (95/000,635) [66]
[21.a]-[21.b]	[138.a]-[138.b]	(IPR2013-00004) [72.a]-[72.b] (90/009,995) [16.a]-[16.b] (95/000,635) [70.a]-[70.b]
[21.c]	[138.c]	(IPR2013-00004) [72.c] (90/009,995) [16.b] (95/000,635) [73]
[21.d]	[138.d]	(IPR2013-00004) [72.d] (90/009,995) [16.c] (95/000,635) [70.a]-[70.e]
[21.e.1]	[138.e]-[138.e.1]	(IPR2013-00004) [72.e]-[72.e.1] (90/009,995) [16.d] (95/000,635) [70.a]-[70.e]

[21.e.2]	[138.e.2]	(IPR2013-00004) [72.e.2] (90/009,995) [16.c] (95/000,635) [70.a]-[70.e]
----------	-----------	--

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 '729 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, ¶¶26-29.)

XI. DETAILED EXPLANATION OF GROUNDS

A. Ground 1: Zaurus, Pad, and SVG Render Claims 1-22 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, ¶22.)

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, ¶23.)

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, ¶24.)

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, ¶31.) 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, ¶32.) 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, ¶¶32-33; *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, ¶34.) 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, ¶34.)

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, ¶35.) 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, ¶36.) 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, ¶¶36-37.) 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, ¶¶36-37.) 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, ¶¶35-37.)

6. Claim 1

- i) [1.pre] "A method, comprising:"**
- ii) [1.a] "requesting through a mobile device having a touch-sensitive display a Web page comprising HTML-based Web content defining an original page layout, functionality, and design for viewing on a commercially available desktop browser, said HTML-based Web content including HTML code for at least one form and multiple columns and Cascading Style Sheets (CSS) content;"**

Zaurus discloses a mobile device with a touch-sensitive display that is operated with a "touch pen," with which the user can "touch [the screen] lightly" and use the pen to operate on-screen keys and icons ("a mobile device having a touch-sensitive display"). (EX1004, 12-13.) The user can "Connect to the Internet, and open Zaurus Plaza (the Zaurus home page) and other home pages using the

Internet browser,” thereby requesting Web pages through the device’s Internet browser (“requesting...a Web page”). (EX1004, 39.) The “ZAURUS PLAZA (Zaurus home page) screen” illustrates that the user can “Open another home page by specifying URL,” “Go to home page,” and “Reload currently displayed screen,” confirming that the touch-screen device is used to request and load Web pages. (EX1004, 46-47.) These requested pages are standard HTML-based Web content because the “Internet Advantages of the Internet Browser” section explains that the “Internet browser in the MI-100 series incorporates HTML 3.2 specifications in important places, and therefore has rich expressive capabilities” (“a Web page comprising HTML-based Web content”). (EX1004, 40.) Because the requested Web pages are encoded in HTML and associated CSS style information, they are provided in a particular way that the information is encoded (i.e., in that HTML/CSS “format”). (EX1004, 40; EX1008, 4, 13-14; EX1002, ¶38.)

Zaurus’s “Quick Start” section states that, when displaying a home page, “the home page is displayed in a 16-gradation grayscale” and that “a separate page which is linked is displayed by touching the underlined characters,” describing standard Web “home pages” with hyperlinks. (EX1004, 10.) Zaurus Plaza provides URLs for various reservation, shopping, and information services, showing that the device is used to access general-purpose Web sites that were designed for “commercially

available desktop browsers.” (EX1004, 40, 44-45, 48; EX1002, ¶39.) Pad describes the mobile-compatible Pad browser as a “version of” Mosaic and Netscape (EX1007, 86); Mosaic and Netscape were commercially available desktop browsers well before the earliest effective filing date of the ’729 patent (“commercially available desktop browsers”). (EX1050, 1-2.) A POSITA would have understood that these are conventional HTML Web pages rendered on the mobile touch-screen device in a defined layout, functionality, and design for desktop browsers (“HTML-based Web content defining an original page layout, functionality, and design for viewing on a commercially available desktop browser”). (EX1002, ¶39.)

In describing how to perform “Internet searches” and “reserving airline tickets,” Zaurus explains that “a space for entering characters is displayed” on the screen; the user taps the entry space to bring up a keyboard, “enter[s] characters,” and then touches “Enter,” after which “[c]haracters are entered into the entry space.” (EX1004, 49-50.) A POSITA would have recognized that these “spaces for entering characters” on Web pages are standard HTML form input fields used on search and reservation pages. (EX1002, ¶40.) Given that the browser “incorporates HTML 3.2 specifications” (EX1004, 40), such character-entry regions would be implemented using HTML form elements in the underlying HTML source (“said HTML-based Web content including HTML code for at least one form”). (EX1002, ¶40; EX1043, 14-22.)

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.) Zaurus’s description of pages divided into “boundaries” (i.e., structured page regions) supports that the Zaurus browser contemplates layout constructs that segment page content. (EX1004, 48.) A POSITA would have understood that, in HTML 3.2, a standard mechanism for implementing multi-column page layout is the HTML table model, which arranges content into “rows and columns of cells.” (EX1040, 2; EX1043, 2, 15-17; EX1002, ¶41.) Consistent with that understanding, Pad’s HTML browser discloses a multi-column layout because when the user follows an HTML link from a home page, “the relevant document is loaded into Pad++ and placed on the surface to the right of the original document, at a smaller scale,” and “[t]he Pad++ HTML browser will lay out sub-documents in two columns next to the parent document” (“multiple columns”). (EX1007, 91; EX1002, ¶42.)

SVG discloses that “Cascading Style Sheets (CSS) content” is included in HTML-based Web pages by stating that SVG must be “compatible with and/or leverage[] other relevant standards efforts, including ... CSS,” and explains that “attributes which make sense to be part of a style should be expressed in CSS.” (EX1008, 4.) SVG identifies “CSS support” as a detailed requirement and provides that “[a]ll CSS text/font attributes from the most recently approved CSS

recommendation should be honored by all conforming SVG processors,” noting that authors may “use either XSL or CSS stylesheets within SVG” and “use stylesheets to apply properties to both the text AND graphic objects/assemblies within an embedded SVG component (to provide/enforce ‘house style’ or common properties across all illustrations in a document).” (EX1008, 13.) SVG emphasizes compatibility with “devices with a wide range of resolutions and capabilities, from small mobile devices through office computer monitors to high resolution printers,” underscoring that Web-standard styling technologies like CSS were intended to span both desktop and mobile environments. (EX1008, 4; EX1002, ¶43.)

In view of SVG, a POSITA would have understood that Web pages designed for desktop browsers and retrieved by Zaurus’s and Pad’s HTML browsers would include CSS stylesheets or CSS style attributes as part of their HTML-based content. (EX1002, ¶44.) In the Zaurus-Pad-SVG combination, a mobile device with a touch-sensitive display requests Web pages, and the requested pages are standard HTML-based Web content designed for desktop browsers, including HTML form fields, multi-column layouts, and CSS styling. (EX1002, ¶44.)

- iii) **[1.b] “retrieving and processing at the mobile device the HTML-based Web content including the HTML code for the at least one form and the multiple columns and the CSS content to generate a scalable vector representation of the Web page;”**

Zaurus and Pad disclose “retrieving and processing at the mobile device the HTML-based Web content.” The Zaurus device connects to the Internet and browses “home pages using the Internet browser,” including pages saved from standard desktop tools such as PowerPoint saved in HTML. (EX1004, 39-40.) The browser “incorporates HTML 3.2 specifications in important places,” and can display images, animated GIFs, tables and frame-based pages. (EX1004, 40.) The homepage data is received by the Internet browser and stored in cache memory in the device’s main body or flash card, and “[w]hen the Internet browser loads data from a home page and displays it, the data which is being loaded is downloaded into the cache sequentially.” (EX1004, 43; EX1002, ¶45.)

Pad discloses that “Items of type html are compound items representing the specified html file,” that HTML items “automatically fetch a file from a URL...as well as in-line images,” and that when “html data is fetched, it is parsed and the HTML item is created which contains a method for rendering the page,” with HTML anchors created as separate items. (EX1005, 60.) A POSITA would have understood that Pad discloses processing HTML page content, such as that retrieved by Zaurus, and processing it into internal objects for rendering. (EX1002, ¶46.) As

explained above, the HTML-based Web content includes “the HTML code for the at least one form and the multiple columns and the CSS content.” (See Section XI.A.6.ii.)

Pad suggests processing the HTML content “to generate a scalable vector representation of the Web page.” (See also EX1017, 15 (“there appears to be little dispute that [Pad’s inventor] Bederson discloses vectorized content”).) Pad discloses “a Tcl file” that defines default bindings for HTML items which “lay them out with scale.” (EX1005, 60.) Pad supports multi-column layouts via its “grid” items, which “arrange one or more items in rows and columns and [treat] them as a group,” with commands to configure columns, rows, spans, and padding. (EX1005, 53-56.) A POSITA would have understood Pad’s rendering is scalable because items are rendered with “zoomaction” commands that execute when an item “grows or shrinks” and commands that scale text and images so that they resize smoothly with the coordinate frame. (EX1005, 34-35, 53-54, 81-82). A POSITA would have understood this as a “scalable vector representation” (i.e., scalable vector-based content: graphic content that (1) is capable of being rendered at multiple zoom levels and (2) includes one or more vectors) of the HTML page because the page is represented as positioned objects in a coordinate system (e.g., text objects, graphic primitives/shapes, and images), with their layout and geometry defined by

coordinates and geometric/vector data that can be re-rendered at different magnifications. In this context, the scalable representation is a coordinate geometry-based representation of the page elements that can be rendered at different magnifications, with element positions and geometry expressed using coordinate values and vector/geometry data rather than a single fixed-resolution screenshot of the page. This “scalable vector representation” includes one or more vectors; for example, stored (x, y) translation vectors/offsets represented by (X2, Y2) relative to an origin point (X1, Y1), that position objects’ anchor points relative to a page-level origin/reference point in the coordinate system. (EX1005, 38, 65; EX1008, 11; EX1002, ¶47.)

SVG further supports the understanding that CSS-related styling and “true scalability” in a “scalable vector representation” were known techniques consistent with Pad’s scalable object-based rendering approach. SVG discloses “CSS support” for “CSS text/font attributes.” (EX1008, 7-8, 13.) SVG associates its coordinate systems with CSS positioning for establishing viewports. (EX1008, 11.) SVG can “support zoom and pan on graphics, with true scalability,” such that “all objects and attributes (including such things as text and linewidths) should grow/shrink uniformly with the zoom level.” (EX1008, 14). SVG likewise confirms the standard methodology for identifying elements in a web context via the Document Object Model (DOM), mandating “DOM access” and “complete DOM support for all

attributes and elements.” (EX1008, 4, 15; EX1035.) A POSITA would have understood that SVG discloses HTML/CSS-styled content rendered as vector shapes, text, and images that scale smoothly with zoom (“to generate a scalable vector representation of the Web page”). (EX1002, ¶48.)

In the Zaurus-Pad-SVG combination, Zaurus discloses a mobile handheld device that retrieves and processes HTML Web pages, including pages with forms and multi-column layouts; Pad discloses parsing such HTML into internal objects and automatically laying out those objects “with scale” in a zoomable vector coordinate system, with explicit support for rows and columns; and SVG further supports CSS-related styling and “true scalability” in a scalable vector graphics model. A POSITA would have been motivated to improve Zaurus’s small-screen Web browsing by rendering those pages using known Web vector techniques like Pad and known scalable vector graphics techniques reflected in SVG to enable smooth zooming and panning, while preserving CSS-driven layout and styling. (EX1002, ¶49.) Implementing the Pad and SVG-style HTML renderer on the Zaurus device would enable the Zaurus mobile browser, like a desktop browser, to perform the HTML and CSS parsing and generate the “scalable vector representation” locally including storing/maintaining the coordinate-based layout and geometry information for page objects so the page can be rendered at multiple zoom levels. A POSITA would have understood that DOM-style parsing instantiates a tree of addressable

objects corresponding to the HTML elements to be rendered and transformed (EX1035), thereby satisfying “retrieving and processing” at the mobile device. In the combination, the mobile device would retrieve and process HTML-based Web content (including form and frame/multi-column HTML and associated CSS style information) “to generate a scalable vector representation of the Web page.” (EX1002, ¶49.)

- iv) **[1.c] “rendering the Web page on the touch-sensitive display utilizing the scalable vector representation to produce a first view of the Web page at a first zoom level that preserves the original page layout, functionality and design defined by the HTML-based Web content including the at least one form and the multiple columns and the CSS content relative to the original page layout, functionality and design defined by the HTML-based Web content including the at least one form and the multiple columns and the CSS content when rendered by the commercially available desktop browser; and,”**

As explained above, Zaurus discloses “rendering the Web page on the touch-sensitive display.” (See Section XI.A.6.ii.) Zaurus suggests that its mobile HTML browser was expected to reproduce the desktop Web experience as closely as possible by “incorporat[ing] HTML 3.2 specifications in important places” and supporting images, frames, image maps, and PowerPoint. (EX1004, 39-40; EX1002, ¶50).

Pad discloses the view is defined by a magnification factor and (x,y) position, and discloses commands such as “[z]oom around [a] specified pad position by

multiplicative factor” and animating the zoom. (EX1005, 46, 83). The Web pages are represented as “compound items representing the specified html file” with a URL property identifying the Web address and a width parameter that causes the page to be “re-laid out according to the new width.” (EX1005, 60-63.) The Pad browser then loads an HTML document and places linked documents “on the surface to the right of the original document, at a smaller scale,” with the user instructed to “[s]imply zoom in on the sub-document to read it.” (EX1007, 89-93). A POSITA would have understood that this would “produce a first view of the Web page at a first zoom level.” (EX1002, ¶50.) Pad thus supports “utilizing the scalable vector representation” for rendering because the HTML page is rendered as structured objects within Pad’s zoomable coordinate system, with page elements laid out as coordinate-defined objects whose geometry/positioning information is used when rendering the page at a selected magnification. (EX1005, 46, 60-63, 83; EX1007, 89-93; EX1002, ¶51.)

Pad illustrates that when HTML content is rendered into its zoomable space, zooming in or out preserves layout and interactivity. (EX1007, 87-93.) Pad discloses that an HTML document is displayed with “hotwords...shown in blue,” which change color on hover and act as hyperlinks just like in a desktop browser. (EX1007, 87-89.) When a link is followed, “the relevant document is loaded into Pad++ and placed on the surface...at a smaller scale,” and the user is told to

“[s]imply zoom in on the sub-document to read it,” which a POSITA would have understood is geometric zoom such that the HTML structure, links, and layout remain the same. (EX1007, 89-93; EX1002, ¶52.)

Pad further explains that HTML items are assigned anchors and other members, which a POSITA would have understood enables the hyperlink functionality and internal layout to be preserved at different views. (EX1005, 60-63; *see also* EX1039, 39:17-40:1 (Patent Owner admitting that functionality includes clicking on a hyperlink).) Pad further explains that “HTML anchors are created as separate items that can have events bound to them.” (EX1005, 60.) Because those anchor objects are part of the same scalable, coordinate-defined representation that is re-rendered when the view’s magnification changes, a POSITA would have understood that hyperlink selection behavior (and similar UI interactions) would remain available and consistent as the Web page is rendered at different zoom levels, supporting the requirement that the rendered view “preserves” the page’s functionality relative to a desktop browser. (EX1002, ¶53.)

SVG further supports the “scalable vector representation” approach including by disclosing an XML-based, vector graphics format intended to replace many uses of raster graphics on the Web and to be “displayable on a wider range of device resolutions from small mobile devices through office computer monitors to high resolution printers.” (EX1008, 1). The authors of SVG, W3C, described that XML

was a standard Extensible Markup Language (XML) in 1998. (EX1046.) SVG requires “support[ing] zoom and pan on graphics, with true scalability,” such that “all objects and attributes (including such things as text and linewidths) should grow/shrink uniformly with the zoom level.” (EX1008, 14). SVG also discloses “consistent visual results and behaviors across implementations,” so that conforming processors show the “same visual result.” (EX1008, 4; EX1002, ¶54).

SVG is designed to produce the same appearance across devices and zoom levels. (EX1008, 1, 4, 14.) SVG discloses an implementation that is “consistent so that the same visual results and behaviors exist in all conforming SVG processors,” and that SVG be usable “on devices with a wide range of resolutions and capabilities, from small mobile devices through office computer monitors to high resolution printers.” (EX1008, 4.) SVG’s features include “[c]oordinate systems, relationship to CSS positioning” and “CSS support,” which a POSITA would have understood preserves CSS-defined positioning and styling when rendering Web pages across different viewing contexts and zoom levels. (EX1008, 11, 13; EX1002, ¶55.)

SVG discloses converting elements (e.g., shapes, text, images, links, transforms, coordinates, scale, references) from static graphics files into the scalable vector representation. (EX1008, 5 (“Feature set is rich enough such that a reasonable conversion of most existing standard static graphics files into SVG is possible”), 6 (“For example, a picture of an arrow expressed in SVG might appear inline within

an XML file. The expected way this would be achieved would be by referencing the SVG namespace within the XML file.”), 7 (“Vector graphics shapes...Text data”), 8 (“Image data...The same image formats available to HTML should be available to SVG.”), 11 (“Transformations...Coordinate systems, relationship to CSS positioning”), 14 (“Zoom and pan...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...Links.”); EX1002, ¶56.)

In the combination, a POSITA would have been “utilizing the scalable vector representation” to render the Web page on the device’s HTML browser at an initial magnification. In the combination, a POSITA would have understood that (i) Zaurus provided a touch-screen mobile device running an HTML browser that reproduces desktop Web pages (EX1004, 39-40); (ii) Pad demonstrated rendering HTML pages at different zoom levels while preserving the layout, hyperlink functionality, and design (EX1007, 87-93); and (iii) SVG provides a scalable vector representation to maintain the same layout and visual design across devices and zoom levels (EX1008, 1-4, 14-15; EX1002, ¶57.)

The combination preserves the “original page layout, functionality and design” (including forms, columns, CSS) relative to a desktop browser. It would have been obvious for a POSITA to render an HTML/CSS page (including forms

and multi-column layouts) on a touch-screen mobile device using a scalable vector representation (e.g., Pad's scalable HTML object rendering with SVG's disclosure of "true scalability" vector zoom/pan behavior) such that the first view of the page on the mobile device is a scaled or zoomed version of the same layout, functionality (links), and design relative to what a conventional desktop browser would render ("preserves the original page layout, functionality and design defined by the HTML-based Web content including the at least one form and the multiple columns and the CSS content relative to the original page layout, functionality and design defined by the HTML-based Web content including the at least one form and the multiple columns and the CSS content when rendered by the commercially available desktop browser"). Stated differently, the maintained layout/functionality/design is consistent with the translated HTML/CSS-derived scalable vector representation used to render at different zoom levels. Accordingly, the combination preserves the "original page layout, functionality and design" by maintaining those capabilities and appearances in a manner consistent with the translated portion of HTML/CSS code defining those capabilities and appearances. (EX1002, ¶58.)

- v) **[1.d] “re-rendering the Web page utilizing the scalable vector representation in response to associated user inputs to iteratively zoom in and out views of the Web page at multiple different zoom levels while preserving the original page layout, functionality, and design defined by the HTML-based Web content including the at least one form and the multiple columns and the CSS content relative to the original page layout, functionality and design defined by the HTML-based Web content including the at least one form and the multiple columns and the CSS content when rendered by the commercially available desktop browser.”**

Zaurus discloses a reduced and magnified view of the same Web page on a touch-sensitive display. (EX1004, 46-47.) The “Controls on the Screen While Connected” illustrates on-screen controls to “switch between reduced display and magnified display,” with instructions that to go from reduced to magnified, the user touches “Reduce ▼” followed by “Magnify,” and to go back touches “Magnify ▼” followed by “Reduce.” (EX1004, 46-47.) Pad discloses that “clicking and holding on the middle mouse button performs an smooth animated zoom in,” and “this process can be repeated indefinitely” such that “at any point, you zoom out to see an overview of what’s been done.” (EX1007, 4-46; EX1002, ¶59.)

Pad discloses “HTML items” as compound items that “represent[] the specified html file.” (EX1005, 60.) These items automatically fetch a file from a URL and in-line images; when the HTML data is fetched, “it is parsed and the

HTML item is created” and “contains a method for rendering the page.” (EX1005, 60.) Pad explains that “HTML anchors are created as separate items that can have events bound to them.” (EX1005, 60.) A POSITA would have understood that Pad discloses a zoomable browser where all standard items scale and are redrawn at various zooms: an HTML page is represented as an HTML item; that item is rendered using Pad’s graphics engine that is vector-based and designed for smooth zoom; and zooming causes the HTML item to be re-rendered at the new zoom level by rendering the same coordinate-defined objects (including their positions/geometry) at the updated magnification. (EX1002, ¶60.) Pad confirms that HTML documents are rendered at different scales and viewed by zooming because when a link is followed, “the relevant document is loaded into Pad++ and placed on the surface to the right of the original document, at a smaller scale,” and the user is told to “simply zoom in on the sub-document to read it.” (EX1007, 91-93.)

SVG discloses “Scalable Vector Graphics,” a vector-based web rendering format with “CSS support” for displaying content “on a wider range of device resolutions from small mobile devices through office computer monitors to high resolution printers.” (EX1008, 1, 13.) SVG discloses “transformations” and “zoom and pan” with “true scalability,” which a POSITA would understand involves re-rendering objects and attributes (including images and text) (“scalable vector

representation”) for scaling uniformly at “different zoom levels.” (EX1008, 11, 13; EX1002, ¶61.) A POSITA combining Zaurus’s reduce or magnify; Pad’s HTML items (vector-rendered HTML pages on a zoomable surface); and SVG to convert HTML/CSS-defined pages into a vector representation would understand that this would enable “re-rendering the Web page...in response to associated user inputs to iteratively zoom in and out views of the Web page at multiple different zoom levels.” (EX1002, ¶61.)

In the combination, a POSITA would have found it obvious to implement a mobile browser that converts desktop-oriented HTML/CSS pages (including forms and multi-column layouts) into a scalable vector representation; and re-renders that representation at multiple zoom levels in response to touch-based zoom commands while preserving the layout, functionality, and design relative to the way the Web page appears in a standard desktop browser. (EX1002, ¶62.) Pad’s HTML browser parses HTML items into rendered objects on the zoomable Pad surface, which is then re-drawn at different scales when the user zooms; and SVG specifies a vector format for Web graphics in browsers, which supports zoom and pan with uniform scaling of graphical attributes “re-rendering the Web page utilizing the scalable vector representation.” (EX1002, ¶62.) Pad allows the user to zoom with the middle mouse button and Zaurus provides touch-screen controls to switch between reduced and magnified displays of the same Web page “in response to associated user inputs

to iteratively zoom in and out views... at multiple different zoom levels.” (EX1002, ¶62.) Zaurus discloses rendering Web pages designed for a desktop browser HTML (i.e., layout, interactivity, forms, frames, and image maps); Pad discloses preserving document structure and link functionality while arranging documents in columns and allowing the user to zoom in to read each one; and SVG defines a vector format with CSS support and “true scalability” so that styling and layout are visually consistent across zoom levels and device resolutions, including small mobile devices and desktop monitors “while preserving the original page layout, functionality, and design ... including the at least one form and the multiple columns and the CSS content relative to ... when rendered by the commercially available desktop browser.” (EX1002, ¶62.)

7. Claim 2: “The method of claim 1, wherein the mobile device comprises a mobile phone.”

Zaurus discloses a handheld mobile device and wireless connectivity suitable for Internet access. (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, ¶63.)

8. Claim 3: “The method of claim 1, further comprising enabling a user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input made via the touch-sensitive display.”

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.) Pad discloses zooming by enabling the user to click on text so that it fills the screen (“enabling a user to zoom in on a user-selectable portion of a display of the Web page”). (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.) 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, ¶64.)

9. Claim 4: “The method of claim 1, wherein a display of the Web page is re-rendered in real-time to effect zooming operations.”

Zaurus discloses touch driven HTML browsing. (EX1004, 10-11, 41.) 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, 4-48, 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.) 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, ¶65.)

10. Claim 5

- i) **[5.a] “The method of claim 1, wherein the Web page includes at least one hyperlink, the method further comprising:”**
- ii) **[5.b] “enabling a user to select a hyperlink via the touch-sensitive display; and,”**

Zaurus discloses that the Web page includes at least one hyperlink that a user can touch to select (“enabling a user to select a hyperlink via the touch-sensitive display”). (EX1004, 10 (“a separate page which is linked is displayed by touching the underlined characters.”).) Pad’s browser also displays HTML document that contain blue “hotwords” and “positioning the pointer over a hotword changes its color to red,” so “when you follow a link, the relevant document is loaded.” (EX1007, 89-91.) Pad discloses that its hyperlinks can be configured from text by setting a “from object” and a “to point,” and clicking such text causes the page to zoom to the linked region. (EX1007, 108-113.) In the combination, Zaurus discloses the touch-screen selection of a hyperlink on a mobile Web page, and Pad discloses the hyperlink functionality within the browser. (EX1002, ¶66.)

- iii) **[5.c] “in response thereto, retrieving and processing HTML-based Web content and the CSS content associated with the hyperlink to translate the HTML-based Web content and the CSS content associated with the hyperlink into an additional scalable vector representation; and”**

The Zaurus device retrieves and processes HTML Web pages by interpreting the HTML specifications (“HTML-based Web content”). (EX1004, 10 (“A separate page which is linked is displayed by touching the underlined characters.”), 40 (“The Internet browser in the MI-100 series incorporates HTML 3.2 specifications in important places, and therefore has rich expressive capabilities...Pages composed of frames can also be viewed by displaying them frame by frame.”), 48.) Pad explains that when the user “follows a link, the relevant document is loaded into Pad++ and placed on the surface to the right of the original document, at a smaller scale,” and then the “browser will lay out sub-documents in two columns next to the parent document.” (EX1007, 91.) As explained above, the combination retrieves HTML/CSS content. (*See* Section XI.A.6.ii-iii.) In the combination, a POSITA would have understood that in response to selecting a hyperlink, the browser retrieves the HTML/CSS Web content associated with that hyperlink (“in response thereto, retrieving and processing HTML-based Web content and the CSS content associated with the hyperlink”). (EX1002, ¶67.)

As explained above, the combination retrieves HTML/CSS content and generates a scalable vector representation. (*See* Section XI.A.6.iii.) SVG’s

capabilities include “Links” that can be assigned to objects and can target URLs or other views. (EX1008, 14.) Given the capability to generate an initial scalable vector representation, it would have been obvious and routine to translate the HTML and CSS content after it is retrieved in response to following a link to generate “an additional scalable vector representation.” (EX1002, ¶68.) A POSITA would have understood that the rendering engine of the combined device would “translate” the Web content into the additional scalable vector representation described by Pad and SVG (e.g., converting the HTML layout objects into SVG vector objects) to enable the zooming features. In the combination, after generating a scalable vector representation for an initial Web page, the same techniques would be used for newly-retrieved linked pages “to translate the HTML-based Web content and the CSS content associated with the hyperlink into an additional scalable vector representation.” (EX1002, ¶68.)

- iv) **[5.d] “employing the additional scalable vector representation and/or data derived therefrom to render the Web content and the CSS content associated with the hyperlink on the touch-sensitive display.”**

As explained above, the scalable vector representation is utilized for rendering the HTML/CSS content and is rendered on a touchscreen device while preserving layout and design. (See Section XI.A.6.iv.) A POSITA would have found it obvious to handle hyperlinked pages in exactly the same way: allow the user to select a

hyperlink by touch, retrieve the HTML and associated CSS for the linked page, translate that content into another scalable vector representation, and render that representation on the touchscreen display (“employing the additional scalable vector representation and/or data derived therefrom to render the Web content and the CSS content associated with the hyperlink on the touch-sensitive display”). (EX1002, ¶69.)

11. Claim 6: “The method of claim 1, further comprising enabling a user to pan views of the Web page in real-time in response to corresponding user inputs made via the touch-sensitive display.”

Zaurus discloses enabling the user to scroll the content on the screen by touching scrolling inputs on the display. (EX1004, 43 (“Characters are scrolled and input if a string of characters longer than the input item frame is entered when inputs to “open a home page” or “proxy server” address specification cells are being made. Displaying the hidden characters by scrolling inside the input item is carried out by moving the cursor by touching ← or → at the right side of the input line of the input board. At this time, please leave the input line of the input board blank. Touching → when the cursor is at the right end of the input item displays the hidden characters on the right side of the input item, and touching ← when the cursor is at the left end of the input item displays the hidden characters on the left side of the input item.”); EX1002, ¶70.)

Pad includes a pan command, and “Pad++ normally does fast pans, i.e., copying the portion of the screen that doesn't change, and re-rendering the new portion.” (EX1005, 11.) Pad enables using “smooth animated zooms,” and “at any time you can use the mouse to pan and zoom to another part of the presentation.” (EX1007, 71.) Pad further discloses to “navigate through the data by panning and zooming.” (EX1007, 73.) SVG discloses that SVG “should support zoom and pan on graphics, with true scalability.” (EX1008, 14; EX1002, ¶71.)

In the combination, it would have been obvious for a POSITA to apply known pan techniques to the known problem of navigating scalable web content on a small touch-screen mobile browser by incorporating real-time panning of the web-page view using the same pan features taught by Pad and SVG, in response to touch input on the touch-sensitive display as taught by Zaurus (“enabling a user to pan views of the Web page in real-time in response to corresponding user inputs made via the touch-sensitive display”). (EX1002, ¶72.)

12. Claim 7

- i) **[7.a] “The method of claim 1, further comprising enabling a user to zoom in on a first of the multiple columns by tapping on the first column via the touch-sensitive display,”**

Zaurus discloses an HTML browser and changing zoom level via touch. (EX1004, 47-49.) Zaurus discloses that users can “[t]ouch the screen with a pen,” and to operate the UI by “touch[ing] keys such as function select keys, icons, buttons,

and menu,” noting that it “can be operated by a light touch.” (EX1004, 15.) Users “Lightly touch Schedule among the function select keys. A schedule screen will be displayed.” (EX1004, 15.) For multi-pane web content, Zaurus discloses tapping a part of a framed page to change which frame’s content is shown by “Switching Pages with Frames,” where Zaurus explains that when a page uses frames, “a white frame icon is displayed on the upper right of the screen,” and “switching is done by touching the portion of the frame icon you would like to display. The touched portion turns grey, and information in the selected portion (child page) is displayed.” (EX1004, 48-49; EX1002, ¶73.)

Pad’s “HTML browser will lay out sub-documents in two columns next to the parent document.” (EX1007, 91.) Pad’s grid system further formalizes rows and columns of grid items that “arrange one or more items in rows and columns and treat them as a group.” (EX1005, 54.) A command “returns the bounding box ... for the space occupied by the grid position indicated by column and row,” including the cell’s width and height. (EX1005, 54.) A POSITA would have understood that the “first of the multiple columns” corresponds to the first grid column (e.g., column 0) or the first sub-document column in Pad’s HTML browser. (EX1002, ¶74.)

Pad illustrates zooming in response to input and discloses that “[w]hen you click on a piece of text, Pad++ zooms in so that that text fills the screen horizontally and appears near the top of the screen vertically.” (EX1007, 54-68.) Pad explains

that “this concept is generalised. Any object can be turned into a hyperlink.” (EX1007, 108.) Pad supports making an object active with a “from object” and “to point,” and then “clicking” that object to perform an animated zoom to the specified location. (EX1007, 108-112). A POSITA would have incorporated Pad’s click-to-zoom feature into Zaurus’s touch input tap as a straightforward substitution of known inputs for the same function. (EX1002, ¶75.) In the combination, the device would render a multi-column web layout on a touch display for “enabling a user to zoom in on a first of the multiple columns by tapping on the first column via the touch-sensitive display.” The tap-to-zoom is accomplished by employing the “scalable vector representation” to re-render the view at the new zoom level. (EX1002, ¶75.)

- ii) **[7.b] “wherein in response thereto, the display is re-rendered such that the first column is displayed to fit a width of a display area of the touch-sensitive display, the display area used for displaying Web page content and the CSS content.”**

As explained above, Pad discloses zooming so that the selected content fills the screen width. (EX1007, 54-68, 108-113.) 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.) Pad provides a “grid bbox” command to return the bounding box of a

given column and row (EX1005, 54), and a “grid columnconfigure” command to control column sizes (EX1005, 55). Pad handles HTML content as items within the bounding boxes: “Because an HTML item is a group, it may contain other members in addition to its anchors. This allows setting and retrieving of all members that are part of this HTML item.” (EX1005, 62; EX1002, ¶76.)

A POSITA would have understood that in a grid-based multi-column layout, Pad discloses obtaining the bounding box of the selected (“first”) column and invoking the “center” command to zoom so that the column’s width fills the display. (EX1002, ¶77.) Pad illustrates this functionality of zooming on columns of sub-documents because after sub-documents are laid out “in two columns next to the parent document,” the user is instructed to “[s]imply zoom in on the sub-document to read it.” (EX1007, 91-93.) A POSITA would have understood that zooming in on one of these sub-document columns using the “center to fill” command results in the selected column occupying the full width of the browser’s content area on the screen. (EX1002, ¶77.) Generating the new view at the new zoom level would require that the “display is re-rendered such that the first column is displayed to fit a width of a display area.” (EX1002, ¶77.)

The combination discloses that the zooming operation occurs in “the display area used for displaying Web page content and the CSS content.” Zaurus discloses

an HTML internet browser. (EX1004, 40.) Pad's browser fetches content from a URL, parses it, and renders the Web page with anchors as separate items that can have events bound to them and are laid out "with scale." (EX1005, 36.) SVG "support[s] zoom and pan on graphics, with true scalability," such that "all objects and attributes (including such things as text and linewidths)...grow/shrink uniformly with the zoom level." (EX1008, 14.) SVG is designed to integrate with CSS, stating that attributes which make sense to be part of style "should be expressed in CSS." (EX1008, 4-5.) In the combined browser (with its two-column layout), the zoom functionality would have been applied to the selected HTML region or column so that its bounding box fills the screen's width. (EX1002, ¶78.)

13. Claim 8

- i) **[8.a] "The method of claim 1, wherein the HTML-based Web content includes at least one image, the method further comprising"**

The combined browser renders "the HTML-based Web content [that] includes at least one image." (EX1002, ¶79.) Zaurus's browser "incorporates HTML 3.2 specifications" and "can display images not only in GIF format, but also in the commonly found JPEG format" and "small animated GIFs" all on an HTML page. (EX1004, 40.) The Internet settings screen includes an "Image Display" setting, with "Yes ... image data is displayed" and "No ... only text data is displayed," which confirms HTML pages with optional image content. (EX1004, 42.) Pad

defines image items that “appear on the display as color images,” created via allocimage, which can read GIF and JPEG image files. (EX1005, 17, 64.) Pad “automatically fetch[es] a file from a URL ... as well as in-line images,” and HTML anchors are created as separate items, which discloses support for images. (EX1005, 61-64). SVG supports that “the same image formats available to HTML should be available to SVG.” (EX1008, 8; EX1002, ¶79.)

ii) [8.b] “enabling a user to zoom on an image by tapping on the image via the touch-sensitive display,”

As explained above, the combination discloses “enabling a user to zoom...by tapping...via the touch-sensitive display.” (See Section XI.A.12.i.) Zaurus further discloses that the browser is “compatible with client-side clickable maps. By clicking inside the displayed figure, you can jump to the page which corresponds to that portion.” (EX1004, 40.) A POSITA would have understood this clicking to be tapping an image in an HTML page on a touchscreen device to trigger an action. (EX1002, ¶80.)

A POSITA would have also understood that Pad handles images as distinct, interactive objects capable of managing events. For example, Pad’s HTML anchors include specific properties such as “ismap” (which is true when the anchor is a clickable image map) and “notloaded” (which manages the state of inline images). (EX1005, 63-64; EX1002, ¶81.) This confirms that Pad’s architecture was already designed to identify images and handle their interaction states. In the combination,

a POSITA would have found it straightforward to map a tap on the image (the touchscreen analog of a mouse click) to a trigger of the zooming operation (“enabling a user to zoom on an image by tapping on the image via the touch-sensitive display”). (EX1002, ¶81.) Replacing a mouse click with a touch input tap when incorporating Pad and SVG into the Zaurus touch-screen mobile browser would have been a routine, predictable substitution to enable interaction with these defined image objects. The tap-to-zoom is accomplished by employing the “scalable vector representation” to re-render at the new zoom level. (EX1002, ¶81.)

- iii) [8.c] **“wherein in response thereto, the display is re-rendered such that the image is displayed to fit a width of a display area of the touch-sensitive display, the display area used for displaying the HTML-based Web content.”**

As explained above, the combination discloses “wherein in response thereto, the display is re-rendered such that ... is displayed to fit a width of a display area of the touch-sensitive display, the display area used for displaying the HTML-based Web content.” (See Section XI.A.12.ii.) In the combination, setting “the image” item’s bounding box to fill the screen would result in the image’s width matching the “width of a display area of the touch-sensitive display.” (EX1005, 22, 63-64.) Pad’s browser demonstrates that after zooming into an “images” folder, “[y]ou can see the pictures within the ‘images’ folder... Of course you could zoom in further to see the images in greater detail.” (EX1007, 76-80.) 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, 30, 32.) In the combination, the user would tap an image displayed on the touch device to select a zoom factor based on the display-area width, and the device would zoom that image item's bounding box to re-render the page at a zoom where the image is displayed to fit the width of the display area. (EX1002, ¶82.)

14. Claim 9

- i) **[9.a] "The method of claim 1, wherein HTML code in the HTML-based Web content of the Web page defines a Web page width, the method further comprising:"**
- ii) **[9.b] "interpreting the HTML code to determine the Web page width; and"**

Pad's HTML items are objects that represents an HTML Web content retrieved from a URL. (EX1005, 60.) Pad discloses "HTML code" such as an "url" option that "specifies the URL ... that this html page should be accessed from." Pad explains that the code "defines a Web page width" because it "specifies the width (in the current units) of the html page. The page will be re-laid out according to the new width, and the length of the page may change dependent on the new width." (EX1005, 2, 13, 63.) Because the HTML items setting the width represent "the HTML-based Web content of the Web page," a POSITA would have understood

that the browser is “interpreting the HTML code to determine the Web page width.” (EX1002, ¶83.) Zaurus’s HTML browser reinforces the concept of interpreting HTML code. (EX1004, 40.) In the combination, given Zaurus’s mobile HTML 3.2 browser and Pad’s disclosure of treating an HTML page as having a configurable page width that controls layout, it would have been obvious for a POSITA to use the parsed HTML width information as the value for that page width and to re-layout accordingly. (EX1002, ¶83.)

iii) [9.c] “rendering a view of the Web page wherein the Web page width is rendered to fit across a width of a display area of the touch-sensitive display, the display area used for displaying Web page content.”

For similar reasons explained above, the combination discloses this limitation. (See Sections XI.A.12.ii, XI.A.13.iii.) Zaurus discloses an HTML browser that renders Web pages. (EX1004, 40.) Pad discloses rendering content so that its width fills the screen horizontally. (EX1007, 54-68, 108-113.) Pad discloses the underlying mechanisms based on setting bounding boxes to fill the screen and defining the width. (EX1005, 22, 63.) Pad discloses commands for selecting various zoom levels. (EX1005, 2, 30, 83.) A POSITA would have understood that for an HTML item having a defined page width and a selectable zoom, the zoom can be set such that the width of that HTML item’s bounding box fills the screen and is equal to the width of the display area used to show the web content (“rendering a view of the Web page wherein the Web page width is rendered to fit across a width of a

display area of the touch-sensitive display”). (EX1002, ¶84.) A POSITA integrating Pad’s HTML item/zoom behavior into a mobile HTML 3.2 browser would have found it straightforward to set the page width equal to the display width and to use zoom and center functions so that the resulting html page width fits the handheld’s display area as a routine design choice of a mobile browser layout. (EX1002, ¶84.)

15. Claim 10: “The method of claim 9, wherein under the first view the Web page width is rendered to fit across the width of the display area of the touch-sensitive display, the display area used for displaying the HTML-based Web content.”

Given that claim 9 already requires interpreting HTML page-width information and rendering the page so its width “is rendered to fit across” the display, the combination discloses that the “first view” (the initial zoomed-out view of claim 1; *see* Section XI.A.6.iv.) uses that same width-to-display fitting in the display area that displays “HTML-based Web content.” (EX1002, ¶85.) In view of Zaurus’s reduced view and Pad disclosing zooming so that an object’s width fills the screen, configuring the first scalable-vector view to fit the page width to the display would have been an obvious design choice for a known zoom-to-fit width feature. (EX1002, ¶85.)

16. Claim 11: “The method of claim 1, wherein the HTML-based Web content further includes HTML code defining the layout for at least one table.”

Zaurus discloses that its browser “incorporates HTML 3.2 specifications in important places, and therefore has rich expressive capabilities.” (EX1004, 40.) HTML 3.2 is the W3C HTML standard that includes the <TABLE> element and associated attributes for defining table layout. (EX1002, ¶86; EX1040.) A browser that “incorporates HTML 3.2” would handle HTML-based Web content that includes HTML code defining the layout of “at least one table” such as <TABLE> elements. (EX1002, ¶86; EX1040.) When Zaurus retrieves and displays standard HTML 3.2 Web pages, “the HTML-based Web content” contains “HTML code defining the layout of at least one table.” Because tables are a core part of the HTML 3.2 specifications, it would also have been obvious for the scalable-vector rendering techniques of the Zaurus-Pad-SVG combination to be applied to ordinary Web pages that use tables. (EX1002, ¶86.)

17. Claim 12: “The method of claim 1, wherein the HTML-based Web content further includes HTML code for two or more frames.”

Zaurus’s HTML browser supports rendering Web pages made up of multiple frames (EX1004, 40, 48, 49), which a POSITA would have understood are implemented using standard HTML <frame> code in HTML pages. (EX1002, ¶87; EX1041.) A POSITA would have understood that when an HTML Web page is

requested and processed as explained with respect to claim 1, the underlying HTML-based Web content for such pages “includes HTML code for two or more frames.” (EX1002, ¶87.)

18. Claim 13

- i) [13.a] “The method of claim 1, wherein the Web page includes at least one paragraph, the method further comprising”
- ii) [13.b] “enabling a user to zoom on the at least one paragraph by tapping on the at least one paragraph via the touch-sensitive display,”
- iii) [13.c] “wherein in response thereto, the display is re-rendered such that the at least one paragraph is displayed to fit a width of a display area of the touch-sensitive display, the display area used for displaying the HTML-based Web content.”

For similar reasons explained above in claim 7, the combination discloses this limitation. (*See* Sections XI.A.12.i-ii.) As explained above, the combination renders HTML Web pages including text. (*Id.*) A POSITA would have understood that text is included in a “paragraph” encoded as HTML code. (EX1042, 1; EX1043, 10.) Zaurus supports user interaction by touching the display, and Pad teaches zooming on selected text so it fills the screen horizontally. (EX1004, 15-16; EX1007, 54-68.) Thus, upon a user tap on a paragraph (i.e., selecting the paragraph’s rendered text region), a POSITA would have applied Pad’s zoom-to-fit approach so that, in response, the display employs the “scalable vector representation” to re-render at the

new zoom level with the selected paragraph displayed to fit the width of the display area used for displaying the HTML-based Web content. The tap-to-zoom is accomplished by employing the “scalable vector representation” to re-render the paragraph to fit the display width at the new zoom level. (EX1002, ¶88.)

19. Claim 14

- i) [14.pre] **“A method, comprising:”**
- ii) [14.a] **“requesting through a mobile device having a touch-sensitive display a Web page comprising HTML-based Web content including HTML code for at least multiple columns and Cascading Style Sheets (CSS) content;”**

The USPTO canceled and the Zaurus-Pad-SVG combination discloses and/or suggests this limitation for the same reasons as discussed for limitation [1.a]. (EX1002, ¶89; Section XI.A.6.ii.)

- iii) [14.b] **“retrieving and processing the HTML-based Web content including the HTML code for the at least multiple columns and the CSS content at the mobile device to generate a scalable vector representation of the Web page;”**

The USPTO canceled and the Zaurus-Pad-SVG combination discloses and/or suggests this limitation for the same reasons as discussed for limitation [1.b]. (EX1002, ¶90; Section XI.A.6.iii.)

- iv) **[14.c] “rendering the Web page on the touch-sensitive display utilizing the scalable vector representation to produce a first view of the Web page at a first zoom level that preserves the original page layout, functionality and design defined by the HTML-based Web content including the at least multiple columns and the CSS content relative to the original page layout, functionality and design defined by the HTML-based Web content including the at least multiple columns and the CSS content when rendered by a commercially available desktop browser; and”**

The USPTO canceled and the Zaurus-Pad-SVG combination discloses and/or suggests this limitation for the same reasons as discussed for limitation [1.c].
(EX1002, ¶91; Section XI.A.6.iv.)

- v) **[14.d] “re-rendering the Web page utilizing the scalable vector representation in response to associated user inputs to iteratively zoom in and out views of the Web page at multiple different zoom levels while preserving the original page layout, functionality, and design defined by the HTML-based Web content including the multiple columns and the CSS content relative to the original page layout, functionality and design defined by the HTML-based Web content including the at least multiple columns and the CSS content when rendered by the commercially available desktop browser; and”**

The USPTO canceled and the Zaurus-Pad-SVG combination discloses and/or suggests this limitation for the same reasons as discussed for limitation [1.d].
(EX1002, ¶92; Section XI.A.6.v.)

- vi) **[14.e] “in response to a tap on a first of the multiple columns via the touch-sensitive display, re-rendering the Web page on the touch-sensitive display such that the first of the multiple columns is displayed to fit a width of a display area of the touch-sensitive display, the display area used for displaying the HTML-based Web content.”**

The USPTO canceled and the Zaurus-Pad-SVG combination discloses and/or suggests this limitation for the same reasons as discussed for claim 7. (EX1002, ¶93; Section XI.A.12.)

- 20. Claim 15: “The method of claim 14, wherein the mobile device comprises a mobile phone.”**

The USPTO canceled and the Zaurus-Pad-SVG combination discloses and/or suggests this claim for the same reasons as discussed for claim 2. (EX1002, ¶94; Section XI.A.7.)

- 21. Claim 16: “The method of claim 14, further comprising enabling a user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input made via the touch-sensitive display.”**

The USPTO canceled and the Zaurus-Pad-SVG combination discloses and/or suggests this claim for the same reasons as discussed for claim 3. (EX1002, ¶95; Section XI.A.8.)

22. Claim 17

- i) [17.a] “The method of claim 14, wherein the Web page includes at least one hyperlink, the method further comprising:”**
- ii) [17.b] “enabling a user to select a hyperlink via the touch-sensitive display; and, in response thereto,”**
- iii) [17.c] “retrieving and processing HTML-based Web content associated with the hyperlink to translate the HTML-based Web content into an additional scalable vector representation; and”**
- iv) [17.d] “employing the additional scalable vector representation and/or data derived therefrom to render the HTML-based Web content associated with the hyperlink on the touch-sensitive display.”**

The USPTO canceled and the Zaurus-Pad-SVG combination discloses and/or suggests this limitation for the same reasons as discussed for claim 5. (EX1002, ¶96; Section XI.A.10.)

23. Claim 18

- i) **[18.a] “The method of claim 14, wherein the HTML-based Web content includes at least one image, the method further comprising enabling a user to zoom on an image by tapping on the image via the touch-sensitive display,”**
- ii) **[18.b] “wherein in response thereto, the display is re-rendered such that the image is displayed to fit the width of the display area of the touch-sensitive display, the display area used for displaying the HTML-based Web content.”**

The USPTO canceled and the Zaurus-Pad-SVG combination discloses and/or suggests this claim for the same reasons as discussed for limitations [8.a]-[8.c]. (EX1002, ¶97; Section XI.A.13.)

24. Claim 19: “The method of claim 14, wherein the HTML-based Web content further includes HTML code for at least one table.”

The USPTO canceled and the Zaurus-Pad-SVG combination discloses and/or suggests this limitation for the same reasons as discussed for claim 11. (EX1002, ¶98; Section XI.A.16.)

25. Claim 20: “The method of claim 14, wherein the HTML-based Web content further includes HTML code for two or more frames.”

The USPTO canceled and the Zaurus-Pad-SVG combination discloses and/or suggests this claim for the same reasons as discussed for claim 12. (EX1002, ¶99; Section XI.A.17.)

26. Claim 21

- i) [21.a] “The method of claim 14, wherein processing the HTML-based Web content comprises:”**
- ii) [21.b] “parsing HTML-based code corresponding to the Web page to determine an original page layout of the HTML-based Web content on the Web page;”**

In the Zaurus-Pad-SVG combination, Zaurus processes and renders HTML 3.2 Web pages, including pages composed of frames, images, links, and other structural elements. (EX1004, 10, 38-41, 43, 46, 48-49; Sections XI.A.1.i-ii.) A POSITA would have understood that rendering such HTML content requires parsing the HTML to identify the elements that define layout (e.g., frames, images, links), consistent with Pad’s disclosure that when HTML is fetched “is parsed” and the HTML item is created and contains a method for rendering the page, and with SVG’s requirement for DOM access and complete DOM support for all attributes and elements for rendering/scripting. (EX1005, 60-65; EX1007, 17, 20; EX1008, 4, 15; EX1046; EX1002, ¶¶100-102.)

- iii) [21.c] “logically grouping selected content into objects;”**

In the Zaurus-Pad-SVG combination, the parsed HTML elements are represented as discrete items and organized into hierarchical groups (“objects”). 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; EX1002, ¶103.) Pad’s HTML browser further illustrates grouped document/object structure when linked sub-documents are loaded and laid out next to the parent document. (EX1007, 89-96; EX1002, ¶103.) SVG likewise discloses DOM-based objects and grouping/layering collections of objects for scripting and layout. (EX1008, 4, 5-6, 10-11, 14, 15, 9; EX1002, ¶104.) 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, ¶105.)

iv) [21.d] “defining a primary datum corresponding to the original page layout; and,”

In the Zaurus-Pad-SVG combination, generating the scalable vector representation includes defining an origin. (EX1002, ¶106.) 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, 12, 13, 23, 33, 34, 54; EX1002, ¶¶106-108.) 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; EX1002, ¶¶106-108.) SVG likewise describes coordinate

systems/viewports (including relationship to CSS positioning) and zoom/pan with true scalability, supporting a page-level coordinate reference (“primary datum”) corresponding to the layout. (EX1008, 11, 14; EX1002, ¶¶108-109.) 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 original page layout,” and to express object layout locations relative to that fixed reference. (EX1005, 29; EX1008, 11; EX1002, ¶110.)

v) [21.e] “for each object,”

(1) [21.e.1] “defining an object datum corresponding to a layout location datum for the object’s associated display content; and”

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. (EX1002, ¶111.) 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 id.*, 38 (slide: dx/dy applied to the

item transformation, i.e., the “-place itemconfigure option,” and reporting position at the item’s anchor point); EX1002, ¶111.)

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, ¶112.) 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; EX1002, ¶112.)

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, ¶113.) 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,

¶114.) 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, ¶115.) 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, ¶¶114-116.)

(2) [21.e.2] “generating a vector from the primary datum to the object datum for the object.”

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.26.iv-v.1; EX1002, ¶117.) Pad discloses object locations via bounding-box coordinates and view/object translations, including use of “vector” parameters (e.g., “moveto” or “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, ¶¶118-121.) 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, ¶118.) 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, ¶120.) 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, ¶¶121-122.)

27. Claim 22: “The method of claim 14, wherein the HTML-based Web content further includes HTML code for at least one form.”

The USPTO canceled and the Zaurus-Pad-SVG combination discloses and/or suggests this limitation for the same reasons as discussed for limitation [1.a]. (EX1002, ¶123; Section XI.A.6.ii.)

B. Ground 2: Zaurus, Pad, SVG, and SVF Render Claim 21 Obvious

As discussed above, the Zaurus-Pad-SVG combination renders claims 1-22 obvious. However, to the extent Patent Owner argues, or the Board finds, that certain features directed to the “datum” that are recited in claim 21 are not met by the Zaurus-Pad-SVG combination, those features nonetheless would have been obvious in view of SVF. (EX1002, ¶124.) 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-SVF 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, ¶125.)

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, ¶125.) 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, 4:64-68; EX1008; EX1035; EX1037; EX1043; 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, ¶125.)

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 explicitly 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; EX1002, ¶126.) 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, ¶126.)

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, ¶127.) As an initial matter, the ’729 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; EX1043; 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, ¶127.) 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, ¶124.)

4. Claim 21

i) [21.d]⁶

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, ¶128.) 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, ¶128.)

⁶ The claim language is provided above in Ground 1 and is not repeated in Ground 2.

ii) [21.e]-[21.e.1]

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 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, ¶129.)

iii) [21.e.2]

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, ¶130.) 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, ¶130.) 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, ¶130.)

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 1-22 of the ’729 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. 9,519,729 contains, as
measured by the word-processing system used to prepare this paper, 13,955 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. 9,519,729 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 method, comprising:
[1.a]	requesting through a mobile device having a touch-sensitive display a Web page comprising HTML-based Web content defining an original page layout, functionality, and design for viewing on a commercially available desktop browser, said HTML-based Web content including HTML code for at least one form and multiple columns and Cascading Style Sheets (CSS) content;
[1.b]	retrieving and processing at the mobile device the HTML-based Web content including the HTML code for the at least one form and the multiple columns and the CSS content to generate a scalable vector representation of the Web page;
[1.c]	rendering the Web page on the touch-sensitive display utilizing the scalable vector representation to produce a first view of the Web page at a first zoom level that preserves the original page layout, functionality and design defined by the HTML-based Web content including the at least one form and the multiple columns and the CSS content relative to the original page layout, functionality and design defined by the HTML-based Web content including the at least one form and the multiple columns and the CSS content when rendered by the commercially available desktop browser; and
[1.d]	re-rendering the Web page utilizing the scalable vector representation in response to associated user inputs to iteratively zoom in and out views of the Web page at multiple different zoom levels while preserving the original page layout, functionality, and design defined by the HTML-based Web content including the at least one form and the multiple columns and the CSS content relative to the original page layout, functionality and design defined by the HTML-based Web content including the at least one form and the multiple columns and the CSS content when rendered by the commercially available desktop browser.
[2]	The method of claim 1, wherein the mobile device comprises a mobile phone.
[3]	The method of claim 1, further comprising enabling a user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input made via the touch-sensitive display.
[4]	The method of claim 1, wherein a display of the Web page is re-rendered in real-time to effect zooming operations.

[5.a]	The method of claim 1, wherein the Web page includes at least one hyperlink, the method further comprising:
[5.b]	enabling a user to select a hyperlink via the touch-sensitive display; and, in response thereto,
[5.c]	retrieving and processing HTML-based Web content and the CSS content associated with the hyperlink to translate the HTML-based Web content and the CSS content associated with the hyperlink into an additional scalable vector representation; and
[5.d]	employing the additional scalable vector representation and/or data derived therefrom to render the Web content and the CSS content associated with the hyperlink on the touch-sensitive display.
[6]	The method of claim 1, further comprising enabling a user to pan views of the Web page in real-time in response to corresponding user inputs made via the touch-sensitive display.
[7.a]	The method of claim 1, further comprising enabling a user to zoom in on a first of the multiple columns by tapping on the first column via the touch-sensitive display,
[7.b]	wherein in response thereto, the display is re-rendered such that the first column is displayed to fit a width of a display area of the touch-sensitive display, the display area used for displaying Web page content and the CSS content.
[8.a]	The method of claim 1, wherein the HTML-based Web content includes at least one image, the method further comprising
[8.b]	enabling a user to zoom on an image by tapping on the image via the touch-sensitive display,
[8.c]	wherein in response thereto, the display is re-rendered such that the image is displayed to fit a width of a display area of the touch-sensitive display, the display area used for displaying the HTML-based Web content.
[9.a]	The method of claim 1, wherein HTML code in the HTML-based Web content of the Web page defines a Web page width, the method further comprising:
[9.b]	interpreting the HTML code to determine the Web page width; and
[9.c]	rendering a view of the Web page wherein the Web page width is rendered to fit across a width of a display area of the touch-sensitive display, the display area used for displaying Web page content.

[10]	The method of claim 9, wherein under the first view the Web page width is rendered to fit across the width of the display area of the touch-sensitive display, the display area used for displaying the HTML-based Web content.
[11]	The method of claim 1, wherein the HTML-based Web content further includes HTML code defining the layout for at least one table.
[12]	The method of claim 1, wherein the HTML-based Web content further includes HTML code for two or more frames.
[13.a]	The method of claim 1, wherein the Web page includes at least one paragraph, the method further comprising
[13.b]	enabling a user to zoom on the at least one paragraph by tapping on the at least one paragraph via the touch-sensitive display,
[13.c]	wherein in response thereto, the display is re-rendered such that the at least one paragraph is displayed to fit a width of a display area of the touch-sensitive display, the display area used for displaying the HTML-based Web content.
[14.pre]	A method, comprising:
[14.a]	requesting through a mobile device having a touch-sensitive display a Web page comprising HTML-based Web content including HTML code for at least multiple columns and Cascading Style Sheets (CSS) content;
[14.b]	retrieving and processing the HTML-based Web content including the HTML code for the at least multiple columns and the CSS content at the mobile device to generate a scalable vector representation of the Web page;
[14.c]	rendering the Web page on the touch-sensitive display utilizing the scalable vector representation to produce a first view of the Web page at a first zoom level that preserves the original page layout, functionality and design defined by the HTML-based Web content including the at least multiple columns and the CSS content relative to the original page layout, functionality and design defined by the HTML-based Web content including the at least multiple columns and the CSS content when rendered by a commercially available desktop browser; and

[14.d]	re-rendering the Web page utilizing the scalable vector representation in response to associated user inputs to iteratively zoom in and out views of the Web page at multiple different zoom levels while preserving the original page layout, functionality, and design defined by the HTML-based Web content including the multiple columns and the CSS content relative to the original page layout, functionality and design defined by the HTML-based Web content including the at least multiple columns and the CSS content when rendered by the commercially available desktop browser; and
[14.e]	in response to a tap on a first of the multiple columns via the touch-sensitive display, re-rendering the Web page on the touch-sensitive display such that the first of the multiple columns is displayed to fit a width of a display area of the touch-sensitive display, the display area used for displaying the HTML-based Web content.
[15]	The method of claim 14, wherein the mobile device comprises a mobile phone.
[16]	The method of claim 14, further comprising enabling a user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input made via the touch-sensitive display.
[17.a]	The method of claim 14, wherein the Web page includes at least one hyperlink, the method further comprising:
[17.b]	enabling a user to select a hyperlink via the touch-sensitive display; and, in response thereto,
[17.c]	retrieving and processing HTML-based Web content associated with the hyperlink to translate the HTML-based Web content into an additional scalable vector representation; and
[17.d]	employing the additional scalable vector representation and/or data derived therefrom to render the HTML-based Web content associated with the hyperlink on the touch-sensitive display.
[18.a]	The method of claim 14, wherein the HTML-based Web content includes at least one image, the method further comprising enabling a user to zoom on an image by tapping on the image via the touch-sensitive display,
[18.b]	wherein in response thereto, the display is re-rendered such that the image is displayed to fit the width of the display area of the touch-sensitive display, the display area used for displaying the HTML-based Web content.

[19]	The method of claim 14, wherein the HTML-based Web content further includes HTML code for at least one table.
[20]	The method of claim 14, wherein the HTML-based Web content further includes HTML code for two or more frames.
[21.a]	The method of claim 14, wherein processing the HTML-based Web content comprises:
[21.b]	parsing HTML-based code corresponding to the Web page to determine an original page layout of the HTML-based Web content on the Web page;
[21.c]	logically grouping selected content into objects;
[21.d]	defining a primary datum corresponding to the original page layout; and,
[21.e]	for each object,
[21.e.1]	defining an object datum corresponding to a layout location datum for the object's associated display content; and
[21.e.2]	generating a vector from the primary datum to the object datum for the object.
[22]	The method of claim 14, wherein the HTML-based Web content further includes HTML code for at least one form.