

Patent No. 10,410,316 UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Goodnotes Limited,

Petitioner,

v.

MyScript SAS,

Patent Owner.

Case: IPR2026-00388

U.S. Patent No. 10,410,316

**EXPERT DECLARATION OF MONTY G. MYERS IN SUPPORT
OF PETITION FOR INTER PARTES REVIEW
OF U.S. PATENT NO. 10,410,316**

TABLE OF CONENTS

I.	DECLARATION BACKGROUND	1
II.	SUMMARY OF OPINIONS.....	1
III.	PROFESSIONAL BACKGROUND AND QUALIFICATIONS	2
IV.	BASIS OF MY OPINIONS AND MATERIALS CONSIDERED	4
V.	METHODOLOGY	6
VI.	LEGAL FRAMEWORK.....	8
	A. Standard of Proof	8
	B. Anticipation.....	8
	C. Obviousness	11
VII.	LEVEL OF ORDINARY SKILL	15
VIII.	‘316 Patent CLAIM CONSTRUCTION and claim terms	17
	A. Claim Term: "recognizing handwriting input" (claims 1, 4, 7)_	18
	B. Claim Term: "region of overlap of ranges of elements of the digital ink" (claims 1, 4, 7)	18
IX.	OVERVIEW OF THE ‘316 PATENT	19
	A. The ‘316 Patent Claims.....	29
	i. Claim 1 - ‘316 Patent	29
	ii. Claim 2:.....	30
	iii. Claim 3:.....	30
	iv. Claim 4:.....	30
	v. Claim 5:.....	31
	vi. Claim 6:.....	31
	vii. Claim 7:.....	32
	viii. Claim 8:.....	33
	ix. Claim 9:.....	33
X.	SUMMARY OF THE PRIOR ART	33
	A. Kang (US 2014/0300609 A1)	33
	B. Clary (US 8,115,748 B2)	39
	C. Li (US 2010/0066691 A1)	45
XI.	PROSECUTION HISTORY ERROR	49

EXPERT DECLARATION OF MONTY G. MYERS IN SUPPORT OF PETITION FOR
INTER PARTES REVIEW OF U.S. PATENT NO. 10,410,316

XII.	ANTICIPATION ANALYSIS	53
A.	The Kang Prior Art Reference (US 2014/0300609 A1).....	53
i.	‘316 Independent Claims 1, 4, and 7	54
ii.	‘316 Dependent Claims 2, 5, and 8.....	65
iii.	‘316 Dependent Claims 3, 6, and 9.....	66
B.	The Clary Prior Art Reference (US 8,115,748 B2).....	67
i.	‘316 Independent Claims 1, 4, and 7	69
ii.	‘316 Dependent Claims 2, 5, and 8.....	79
iii.	‘316 Dependent Claims 3, 6, and 9.....	80
XIII.	OBVIOUSNESS ANALYSIS.....	81
A.	Alternatively, the ‘316 Patent Claims Are Rendered Obvious in Light of the Li Publication in View of or Combination with Clary.....	81
i.	POSITA’s Motivation to Combine	83
ii.	‘316 Independent Claims 1, 4, and 7	85
iii.	‘316 Dependent Claims 2, 5, and 8.....	95
iv.	‘316 Dependent Claims 3, 6, and 9.....	96
XIV.	RESERVATION OF RIGHT TO UPDATE AND AMEND	98

I. DECLARATION BACKGROUND

1. My name is Monty G. Myers, I am over 18 years of age, and I am of sound mind and competent to testify to the matters stated in this declaration. I make this declaration based upon my own personal knowledge and, if called upon to testify, would testify competently to the matters stated herein.

2. I have been retained on behalf of the Petitioner Goodnotes Limited (“Goodnotes”), to provide my opinions on certain issues related to U.S. Patent No. 10,410,316 (“the ‘316 patent”) in connection with the above-captioned inter partes review (“IPR”) proceeding. The ‘316 Patent is **Exhibit 1001** and is made a part hereof by reference. In particular, I have been asked to provide my insights, analysis, and opinions regarding whether the claims of the ‘316 Patent are anticipated by or obvious in light of the prior art references identified below.

II. SUMMARY OF OPINIONS

3. In my expert opinion from the perspective of a POSITA (defined below), based on the materials referenced in this declaration and my training and experience, the claims of the ‘316 Patent are anticipated and/or obvious in light of the prior art.

4. Below, I set forth the reasons supporting my opinions. I also present related opinions concerning the ‘316 Patent, the knowledge and skill level of a

POSITA (defined below) as of the Priority Date for the ‘316 Patent, the background of the technologies related to the ‘316 Patent, and other related topics.

III. PROFESSIONAL BACKGROUND AND QUALIFICATIONS

5. I am the founder and CEO of Eureka Software Solutions Inc. (“Eureka”). Now in its 40th year, Eureka is a recognized project-oriented, custom software solution delivery and management company based in Austin, Texas. Eureka has been involved in delivering complex technology and business solutions for a client list that includes members of the Global 1000, Fortune 500, notable private companies, numerous high-profile start-up organizations, and Government agencies. Eureka specializes in business-critical, distributed software and technology solutions and was a pioneer in developing Web-based applications. Most of Eureka’s projects involve substantial aspects of integration with and among legacy systems and other third-party systems/technologies. Eureka has a highly successful track record of providing enterprise-class solutions on a variety of technology platforms such as Microsoft/.Net, Java/J2EE, Open-Source technologies, embedded systems, mobile systems, and others and has been closely involved in assisting a number of its clients with developing and refining their software and technology product strategies and approaches. It has enjoyed significant business success and has been recognized on several “fastest growing

company” lists.

6. As the leader of Eureka, I have negotiated hundreds of software project and technology-oriented agreements, most of which involved substantial intellectual property and confidentiality aspects, software licensing and use rights terms and conditions, and critical project performance terms and conditions. From a project execution standpoint, I have helped organize, plan, and execute numerous technology projects and have personally written and overseen the development of millions of lines of custom software source code for solutions in a wide range of industries and purposes. I have experience working with large and small organizations as they develop, acquire, and sell software solutions and I am familiar with the considerations surrounding such activities.

7. I began my formal computer programming education in middle/high school, and my software development career spans over 47 years. I received a Bachelor of Science degree with cum laude honors in Engineering from Texas A&M University in 1986 and performed substantial graduate level course work at The University of Texas at Austin. I am a licensed Professional Engineer in the State of Texas, and I was the primary inventor named on patents in the area of load and performance testing for distributed Internet software applications/systems.

8. Over the last several years, I have served as both a consulting and

testifying expert on over 200 litigation matters. These past engagements include several high-profile matters, some with industry-shaping consequences, involving notable clients such as Apple, Samsung, Chevron, American Airlines, Ford, General Mills, and more. My professional biography is **Exhibit 1003** and is made a part hereof by reference, reflects the expert witness engagements in which I have been involved and includes matters in which I have testified over the past four years.

9. A material portion of my litigation experience has involved alleged patent infringement cases, where I was asked to review and analyze the patents at issue to provide an opinion related to the validity of those patents based on anticipation and/or obviousness.

10. My employer, Eureka, is being compensated for my expert services on an hourly basis. My billing rate is \$690.00 per hour for this matter, and the billing rate for my assistants is \$390.00 per hour. Due to the litigation schedule and for efficiency, I utilized and directed members of my professional staff to assist me with my work and analysis. My compensation is in no way connected to the results of the findings in this matter, or the ultimate outcome of the matter.

IV. BASIS OF MY OPINIONS AND MATERIALS CONSIDERED

11. Generally, the findings and opinions expressed herein are based upon a review of: the asserted patent, the prosecution history of the asserted patent, prior

art publications/patents, recent asserted infringement contentions by third-parties against Goodnotes, and other general/public industry information. Throughout my declaration below, I provide a more detailed description of the specific nature of some of the information and observations that I received from each of these sources and how they support my ultimate conclusions. I specifically identified and analyzed information, facts, and data that I (as well as, I believe, other similarly positioned experts in my fields of expertise) would normally use as the basis for forming opinions on the subject matter of this declaration.

12. Further, additional information regarding the materials and documentation considered are listed throughout the discussion and analysis presented below. I also provide below the list of exhibits that I have considered in preparing this Declaration. I have reviewed and considered the following exhibits and all other documents mentioned in my declaration.

EXHIBIT	DESCRIPTION
Ex. 1001	U.S. Patent No. 10,410,316 (“’316 patent”)
Ex. 1003	Biography of Monty G. Myers
Ex. 1004	File and History of U.S. Patent No. 10,410,316 (“’316 Patent History”)
Ex. 1006	Kang - US20140300609A1
Ex. 1007	Clary - US8115748
Ex. 1008	Li - 2010/0066691
Ex. 1012	District Court 4-3 Joint Claim Construction Statement

13. In connection with my testimony at deposition or hearing, I may rely on the documents or information identified in this declaration.

14. I reserve the right to develop materials and exhibits as appropriate for use in helping to explain and demonstrate my analysis at trial.

V. METHODOLOGY

15. The findings and opinions developed and expressed in this declaration are based upon the application of the sum total of my knowledge, skill, experience, training, education, and expertise (developed over about 45 years) to the review of the information and materials made available to me in this matter. As such, I present myself as a skilled and expert witness in this matter. My areas of expertise—applied software, software development/engineering, forensic software investigation—are generally considered to be technical and specialized in nature. While my background and experience are extensive and broad, of particular applicability to this matter is my experience with: software engineering/development, software UI/UX design (including text/handwritten data entry, editing, and manipulation), software usability, and the development, integration, and use of digital handwriting recognition capabilities in the course of various software development projects.

16. In forming my opinions regarding the invalidity of the ‘316 Patent, I have considered the materials identified herein, including an analysis of the ‘316

Patent in light of the patents' claims, specifications, prosecution histories and certain identified prior art.

17. I am not an attorney and do not have formal study in the law regarding patents. However, counsel for Goodnotes has explained the law relating to patent (in)validity (as outlined below in the Legal Framework Section), and I have relied on that explanation as needed as the backdrop of my analysis and review.

18. I have used all of the aforementioned materials and my understanding of the law provided by counsel for Goodnotes to, among other things, (1) analyze the plain meaning and scope of the '316 Patent claims; and (2) demonstrate that the '316 Patent claims are anticipated by the prior art references discussed herein and/or are obvious in light of the prior art references discussed herein in combination with the knowledge of a person of ordinary skill in the art and/or other prior art references discussed herein.

19. The foregoing methodologies are of the nature and kind that others in the software industry would conduct and are based on the application of my extensive and specialized experience and technical expertise in the software arena. Based on the particular situation and circumstances related to this matter, the methodology and techniques that I have applied represent a reasonable and reliable way to analyze and determine the technical issues in this matter.

20. Furthermore, in my role as an expert witness, I endeavor to offer my findings and opinions in order to assist in understanding certain technical issues that may be relevant to its consideration and determination in this matter. While I may have been advised of certain legal standards potentially relevant to this matter and have considered such standards in developing and forming my opinions, it is not my intent to offer any legal conclusions.

VI. LEGAL FRAMEWORK

A. Standard of Proof

21. Although patents are presumed to be valid, this presumption can be overcome by clear and convincing evidence showing anticipation, obviousness, or lack of enablement. I understand from counsel that the “clear and convincing” standard means that it is substantially more likely than not that something is true.

B. Anticipation

22. In general, a patent claim is invalid if it is anticipated by a single prior art system or reference. For a claim to be “anticipated,” each and every limitation of the claim must be present in a single prior art system or reference, either expressly or inherently. In determining whether the single item of prior art anticipates the claim, one considers not only what is expressly disclosed in the particular item of prior art, but also what is inherently present or disclosed in that prior art or what

inherently results from its practice. Claim limitations that are not expressly found in a prior art system or reference are inherent if the prior art necessarily functions in accordance with, or includes, the claim limitations, or if the missing element or feature would be the natural result of following what the prior art teaches to persons of ordinary skill in the art. It is acceptable to examine evidence outside the prior art reference (extrinsic evidence), including experimental testing, in determining whether a feature, while not expressly discussed in the reference, is necessarily present in it. Mere probabilities are not enough, but it is not required that persons of ordinary skill actually recognized the inherent disclosure at the time the prior art was first known or used.

23. I understand that there are a number of different ways that anticipation can occur. One of the ways that a patent claim may be found invalid as anticipated is if a product embodying the invention was on sale in the United States more than a year prior to the filing date of the application to which the patent claim is entitled to priority. I have been informed that a sale is invalidating if the invention is the subject of a commercial sale or offer for sale, and the invention is ready for patenting. An actual sale is not required; an attempt to sell is sufficient so long as it is sufficiently definite that another party could make a binding contract by simple acceptance. I am informed that conception, alone, does not equal “ready for

patenting,” but, instead, that the product is ready for patenting if there has been reduction of the invention to practice or if there is proof that the inventor had prepared drawings and descriptions of the invention that are sufficiently specific to enable a person skilled in the art to practice the invention, even if the invention has not yet been reduced to practice.

24. A second way that a claim may be anticipated is if the claimed invention was “in public use” in this country more than one year prior to the date of the application for the patent in the United States. Public knowledge of the invention or an enabling disclosure is not required; only public use is required. Public use occurs where an inventor, having made his device, gives or sells it to another, to be used by such person or entity, without limitation or restriction, or injunction of secrecy, and it is so used. The public use itself need not be enabling, that is, it need not outwardly describe the full structure of the invention for all to see and recognize; it needs simply embody that invention

25. Another way that a claim may be anticipated is if the claimed invention was “described in a printed publication” anywhere in the world more than one year prior to the filing date of the application to which the patent claims priority in the United States. To anticipate, however, the printed publication must also enable one skilled in the art to make and use the claimed invention. The printed publication

must also be reasonably accessible to that portion of the public most likely to use it (although it is not necessary that the publication be available to every member of the public). The date that a printed publication becomes prior art is the date that it becomes available to the public. Thus, for prior art purposes, publications may include not only such things as books, periodicals or newspapers, but also publications that are not as widely available to the public, such as trade catalogues, journal articles or scholarly papers that are distributed or available to those skilled in the field of the invention.

C. Obviousness

26. I understand that even though a prior art item may not fully anticipate a claim of a patent, the claim is invalid as obvious when the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. The relevant inquiry requires consideration of four factors (although not necessarily in the following order): the scope and content of the prior art; the differences between the prior art and the claim at issue; the level of ordinary skill in the art at the time of the invention; and objective factors indicating obviousness or non-obviousness.

27. A prior art reference may be considered if it discloses information

designed to solve any problem or need addressed by the patent or if the reference discloses information that has obvious uses beyond its main purpose that a person having ordinary skill in the art would reasonably examine to solve any problem or need addressed by the patent. The combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results.

28. When a work is available in one field of endeavor, design incentives and other market forces can prompt variations of it, either in the same field or a different one. If a person of ordinary skill can implement a predictable variation, its patentability is likely barred for obviousness. For the same reason, if a technique has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond his or her skill.

29. The obviousness analysis need not seek out precise teachings directed to the specific subject matter of the challenged claim, for a court can take account of the inferences and creative steps that a person of ordinary skill in the art would employ at the time of the invention. Indeed, often, it will be necessary to look to interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge

possessed by a person having ordinary skill in the art, all in order to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue. Importantly, the question is not whether the combination was obvious to the patentee but whether the combination was obvious to a person with ordinary skill in the art. Under the correct analysis, any need or problem known in the field of endeavor at the time of invention and addressed by the patent can provide a reason for combining the elements in the manner claimed.

30. The obviousness analysis cannot be confined by a formalistic conception of the words “teaching, suggestion, and motivation,” or by overemphasis on the importance of published articles and the explicit content of issued patents.

31. In determining whether the subject matter of a patent claim is obvious, neither the particular motivation nor the avowed purpose of the patentee controls. What matters is the objective reach of the claim. If the claim extends to what is obvious, it is invalid. One of the ways in which a patent’s subject matter can be proved obvious is by noting that there existed at the time of invention a known problem for which there was an obvious solution encompassed by the patent’s claims.

32. A person of ordinary skill attempting to solve a problem will not be led

only to those elements of prior art designed to solve the same problem. Common sense teaches that familiar items may have obvious uses beyond their primary purposes, and in many cases a person of ordinary skill will be able to fit the teachings of multiple patents together like pieces of a puzzle. A person of ordinary skill is also a person of ordinary creativity, not an automaton.

33. When there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp. If this leads to the anticipated success, it is likely the product not of innovation but of ordinary skill and common sense. In that instance the fact that a combination was obvious to try might show that it was obvious.

34. One should be aware of the potential for distortion caused by hindsight bias and must be cautious of arguments reliant upon ex post reasoning.

35. Before deciding the issue of obviousness, I understand that one must also consider certain factors commonly referred to as secondary considerations of non-obviousness, which, if established, may indicate that the invention would not have been obvious. Those factors include: (1) Were products covered by the claim commercially successful due to the merits of the claimed invention rather than due to advertising, promotion, salesmanship, or features of the product other than those

found in the claim? (2) Was there long felt need for a solution to the problem facing the inventors, which was satisfied by the claimed invention? (3) Did others try, but fail, to solve the problem solved by the claimed invention? (4) Did others copy the claimed invention? (5) Did the claimed invention achieve unexpectedly superior results over the closest prior art? (6) Did others in the field praise the claimed invention or express surprise at the making of the claimed invention? and (7) Did others accept licenses under the patents because of the merits of the claimed invention?

36. These factors are only relevant to obviousness if there is a connection, or nexus, between them and the invention covered by the patent claims. Even if one could conclude that some of the above indicators have been established and there is the required nexus to the invention, those factors should be considered along with all the other evidence in the matter in determining whether the claimed invention would have been obvious as of the invention date.

VII. LEVEL OF ORDINARY SKILL

37. I was asked to give an opinion as to the educational and/or vocational qualifications of one of ordinary skill in the subject matter taught by the '316 Patent, also known as a person of ordinary skill in the art at the time of the claimed invention(s) (hereinafter a "POSITA"). I understand that the factors that may be

considered in determining level of ordinary skill in the art include: (1) the educational level of the inventor; (2) the type of problems encountered in the art; (3) prior art solutions to those problems; (4) the rapidity with which innovations are made; (5) the sophistication of the technology; and (6) the educational level of active workers in the field.

38. I have reviewed the '316 Patent. The generally relevant field of art is interactive computer graphical user interface operation utilizing a freeform pointing device or method (such as a mouse, stylus, pen, human finger, or similar). Based on my review, my knowledge of interactive computer graphical user interface operation utilizing freeform pointing devices, and the factors listed *supra*, it is my opinion that a POSITA would have had at least: a bachelor's degree in computer science or engineering, or similar technical field, or equivalent work experience, and two or more years working on or with interactive computer graphical user interfaces utilizing freeform input devices to make human-driven inputs using software programs such as digital drawing programs, computer aided drafting (CAD) programs, handwriting recognition programs, or other software programs that capture and process such freeform human-driven inputs.

39. In reaching this opinion as to the qualifications of the hypothetical POSITA, I have considered the types of problems encountered in the art, the prior

art solutions to those problems, the velocity with which innovations are made, the sophistication of the technology, and the educational level of active workers in the field. Additionally, I am in a position to understand this definition of a POSITA and had at least ordinary skill in the art as of the Priority Date (December 8, 2015) and several years prior. For purposes of this declaration, I applied this definition of a POSITA and conducted my analysis of the '316 Patent claims and prior art described herein from the perspective of a POSITA.

VIII. '316 PATENT CLAIM CONSTRUCTION AND CLAIM TERMS

40. I have been informed by counsel that in an *inter partes* review, claim terms are given their plain and ordinary meaning as understood by a person of ordinary skill in the art in the context of the entire patent, including the specification and prosecution history. For the purposes of my analysis, I have applied the plain and ordinary meaning of the claim terms, which I believe are clear based on the intrinsic evidence.

41. At the time of this writing, I am aware that Goodnotes and patent holder have been involved in claim construction discussions in the course of litigation related to the '316 Patent. It is my understanding that the parties have now limited the '316 Patent terms to be construed by the district court to just two (2) terms as follows:

A. Claim Term: "recognizing handwriting input" (claims 1, 4, 7)

42. I have reviewed the parties' competing constructions for "recognizing handwriting input." (EX. 1012, Page 6, Term No. 6.) Patent owner proposed that this claim term should be construed by the Court to mean "recognition of the creation of digital content by users through use of their hands either directly onto a digital or digitally connected medium or via an input tool." (EX. 1012, Page 6, Term No. 6.) Goodnotes proposed that this claim term should be construed by the Court to mean, "detecting handwriting input to derive segmented units for subsequent display and editing operations." (EX. 1012, Page 6, Term No. 6.) I have evaluated the prior art under both proposed constructions. As detailed in my analysis below, my opinions regarding anticipation and obviousness remain unchanged regardless of which construction is applied, as the prior art satisfies both the broader creation-recognition standard and the narrower derivation of segmented units standard. It is my opinion that a POSITA would understand that Goodnotes' proposed construction was the plain and ordinary meaning of this claim term.

B. Claim Term: "region of overlap of ranges of elements of the digital ink" (claims 1, 4, 7)

43. I have reviewed the parties' competing constructions for "region of overlap of ranges of elements of the digital ink." (EX. 1012, Page 7, Term No. 7.) Patent Owner proposed that this claim term should be construed by the Court to

mean "an area where the determined ranges of elements overlap." (EX. 1012, Page 7, Term No. 7.) Goodnotes proposed that this claim term should be construed by the Court to mean "an area where adjacent segmented elements overlap." (EX. 1012, Page 7, Term No. 7.) I have evaluated the prior art under both proposed constructions. As detailed below, my opinions regarding anticipation and obviousness remain unchanged regardless of which construction is applied. Both constructions are functionally consistent with the prior art's disclosure of geometric boundary resolution when spatial areas of adjacent handwritten units physically occupy the same coordinate space. Furthermore, the '316 Patent specification confirms that positioning a cursor on the boundary or edge of one character's range—when that boundary falls inside the overlapping space—constitutes displaying the cursor 'within a region of overlap.' (See, e.g., '316 Patent, Fig. 18B and Description, 17:42). A POSITA would therefore understand that snapping an edit position to an edge that is geometrically engulfed by an adjacent overlapping range satisfies this limitation under either construction. It is my opinion that a POSITA would understand that Goodnotes' proposed construction was the plain and ordinary meaning of this claim term.

IX. OVERVIEW OF THE '316 PATENT

44. The patent at issue is U.S. Patent No. 10,410,316, entitled "System and

method for beautifying digital ink” (hereafter, “the ’316 Patent”). The ’316 Patent was filed on December 22, 2015, and issued on September 10, 2019. It claims priority to European Application No. 15290303.5 filed on December 8, 2015.

45. The ’316 Patent relates generally to handwriting recognition systems and methods using computing device interfaces, and more specifically to systems and methods for beautifying digital ink rendered from handwriting input and for providing editing of that digital ink on the computing device interface. (’316 Patent, EX. 1001, Background/Summary, 1:13). The patent describes a system intended to operate on a wide variety of computing devices, including desktops, laptops, tablets, smartphones, and other devices that accept handwriting or gesture input through an input surface or related input tool. (’316 Patent, EX. 1001, Background/Summary, 1:23-1:35; Description, 6:47-7:32).

46. The specification further describes the disclosed functionality as being carried out through cooperation between an application running on the device and a handwriting-recognition system. In the examples, the handwriting-recognition system operates through multiple processing stages that parse the handwritten input, spatially groups strokes to perform segmentation, and provides the resulting geometrical framework, while the application uses that information to render, transform, and edit the displayed digital ink. (’316 Patent, EX. 1001, Description,

8:6-10:4, 13:64-16:22). This division of labor is important because the issued claims also tie the editing functions to information “determined through the recognition of the handwriting input.”

47. The Background section identifies the central problem the patent is trying to solve. According to the patent, typed text is relatively easy to edit because characters are displayed in a uniform and predictable manner, often with the aid of a displayed cursor. By contrast, handwritten input rendered as “digital ink” is inherently irregular. Characters may vary in size, slope, spacing, and alignment, and users may write virtually anywhere on the input surface rather than following the strict layout conventions of typed text. (‘316 Patent, EX. 1001, Background/Summary, 1:52-2:10). The patent explains that this makes direct editing of digital ink difficult, especially at the character level. (‘316 Patent, EX. 1001, Background/Summary, 1:66-2:10).

48. The patent also criticizes prior approaches as insufficient in two related respects. First, conventional systems could normalize or scale handwriting to some extent, but they did not use recognition-derived geometry in a way that faithfully preserved the original handwritten appearance while making the ink easier to interact with. (‘316 Patent, EX. 1001, Background/Summary, 2:10). Second, prior systems generally did not provide character-level interaction within the handwritten

signal itself. Instead, editing often had to occur only after handwriting had been converted into more conventional text output, or only at broader units such as the word level. ('316 Patent, EX. 1001, Background/Summary, 1:66-2:28). The patent therefore presents its contribution as a recognition-aware handwriting system that both beautifies the digital ink and makes that digital ink directly editable.

49. The specification describes the first part of that contribution as a “structuring transformation.” The system renders initial handwritten input as digital ink, obtains geometrical information about that handwriting through recognition, compares that geometrical information to an alignment structure of the display interface, and then transforms the digital ink so that it is scaled and aligned in a more orderly way. ('316 Patent, EX. 1001, Abstract; Background/Summary, 2:31-3:38). The patent gives examples involving line patterns, baseline and mid-line information, ratio comparisons, and scaling to make handwriting fit more naturally within a display structure while retaining the style of the user’s handwriting. ('316 Patent, EX. 1001, Description, 13:64-15:50).

50. The patent also explains that the input area and the rendering or editing area may be overlapped or separated. In an overlapped arrangement, raw input may initially appear in the same area in which the beautified or edited digital ink is later displayed. In a separated arrangement, the input may be captured in one zone while

the structured digital ink is displayed in another. (‘316 Patent, EX. 1001, Description, 14:51). This discussion provides context for why cursor management and gesture disambiguation become important in the disclosed system. (‘316 Patent, EX. 1001, Description, 17:65).

51. That beautification discussion is important context, but it is not the whole story. The portion of the specification most relevant to the claims challenged here is the second part of the patent’s disclosure: direct editing of the digital ink itself. The patent explains that, during the pipeline recognition process, the handwriting-recognition system utilizes a “segmentation expert” to group consecutive strokes into individual element hypotheses. The application then retrieves this geometric segmentation information and uses it to determine the spatial ranges of recognized elements, including characters in the disclosed examples. (‘316 Patent, EX. 1001, Description, 9:24-9:55, 16:22). The patent illustrates those ranges through character bounding boxes in one set of examples. (‘316 Patent, EX. 1001, Description, 16:22-17:5).

52. The patent then turns to edit cursor management. It explains that the cursor acts as a pointer for character-level editing operations and that the application uses the determined ranges of the segmented elements to place the cursor at range-based positions, including at a character bounding box and between consecutive

character ranges in the disclosed examples. (‘316 Patent, EX. 1001, Description, 15:50-16:22, 17:21-17:42). For non-overlapping neighboring characters, the patent describes several possible cursor placements between adjacent ranges, such as centered between the ranges or placed at the right or left border of one of the ranges. (‘316 Patent, EX. 1001, Description, 17:21). For overlapping characters, the patent describes additional strategies that use the geometry of the overlapping ranges to decide where the cursor should be placed. (‘316 Patent, EX. 1001, Description, 17:42).

53. This discussion shows that the patent is not claiming only one fixed cursor placement rule. Rather, the disclosure contemplates multiple cursor-positioning strategies, depending on, among other things, whether adjacent character ranges overlap. For non-overlapping characters, the cursor may be centered between the ranges or placed on the left or right border of one of the ranges. For overlapping characters, the patent moves from those simpler border or center rules to a geometry-based comparison of the overlapping character ranges. (‘316 Patent, EX. 1001, Description, 17:21-17:42).

54. The patent provides a concrete illustration of this issue through the handwritten word “Tree.” In that example, the handwriting-recognition system determines character ranges for the letters “T,” “r,” “e,” and “e,” and the patent then

shows how the corresponding bounding boxes can overlap once the digital ink is beautified and aligned. ('316 Patent, EX. 1001, Description, 16:61-17:42). The patent uses that example to explain why a cursor centered in the overlap area may be confusing and why the system instead may place the cursor at a border selected by comparing geometrical features of the overlapping ranges. ('316 Patent, EX. 1001, Description, 17:42). That example is significant because it ties the patent's claim language to a concrete character-level editing scenario, while still illustrating the broader element-range and overlap concepts recited in the independent claims.

55. This overlap discussion is especially important to the issued independent claims. The patent states that handwriting often causes characters to overlap horizontally, vertically, or both, and that this in turn causes overlap of the character bounding boxes. ('316 Patent, EX. 1001, Description, 16:39). The patent identifies this overlap as a source of potential user confusion if the cursor is simply rendered in the center of the shared region. ('316 Patent, EX. 1001, Description, 17:42). To address that problem, the patent proposes positioning the cursor "in consideration of the other geometrical features of the ranges of the overlapped characters," such as height, width, or area, and then placing the cursor on the border of one character range or the other within the overlap region. ('316 Patent, EX. 1001, Background/Summary, 3:62, 4:23, 4:52; Description, 17:42).

56. That passage also helps explain the dependent claims. Claim 2 narrows the element ranges to text characters and Claim 3 narrows the comparison to geometrical features including height and width. The specification's overlapping-character discussion directly mirrors those themes by using consecutive handwritten characters as the operative units and by identifying height, width, and area as the relevant geometrical criteria for cursor placement in the overlap region. ('316 Patent, EX. 1001, Background/Summary, 3:57-3:62, 4:23, 4:48-4:52; Description, 17:42).

57. The specification also addresses another practical difficulty that arises when handwriting input and editing occur in the same general area of the display: distinguishing between a user action intended to set the cursor and a user action intended to input new handwriting. The patent explains that, in an overlapped input-and-rendering arrangement, a single-point gesture such as a tap can be treated as a request to set the cursor position if certain criteria are satisfied, rather than as a new handwriting stroke. ('316 Patent, EX. 1001, Description, 17:65). This discussion reinforces that the patent is not merely concerned with beautifying handwriting after the fact, but with managing direct user interaction with the displayed digital ink.

58. The specification therefore presents a system in which recognition is not a single, instantaneous transcription event, but a process including a multi-stage

pipeline. Recognition is used to derive spatial geometry, stroke-level segmentation, and element ranges from the handwriting so that the resulting digital ink can be both beautified and edited. The patent states that segmentation information generated by the recognition process is retrieved by the application to determine the ranges of each segmented element, and it further explains that character segmentation determined by the recognition process allows management of an editing cursor position for the beautified digital ink on a character basis in the disclosed examples. ('316 Patent, EX. 1001, Description, 15:50-16:22, 18:20).

59. In my opinion, a key technical theme of the '316 Patent is not beautification in isolation, but the combination of recognition-derived segmentation and geometry with cursor-based editing of displayed digital ink. The beautification discussion gives context for why the system cares about baselines, alignment structures, and scaling. The claims are centered on displaying digital ink from handwriting input, determining ranges of digital-ink elements from segmentation information produced through recognition, and displaying an input or editing cursor in relation to those ranges, including within a region of overlap. ('316 Patent, EX. 1001, Background/Summary, 3:42-4:52; Claims 1, 4, 7).

60. That distinction matters to a POSITA considering the '316 patent. The specification is broader than the granted claims because it discusses structuring

transformation, reflow, line patterns, scaling ratios, and other beautification functions at length. But the actual claim set focuses on editing digital ink through recognition-derived ranges and cursor placement. The dependent claims further narrow that theory to text characters and to geometrical features such as height and width of the determined ranges. In other words, while the patent is titled “System and method for beautifying digital ink,” the issued claims appear to focus primarily on the editing and cursor-placement side of the disclosed system. (‘316 Patent, EX. 1001, Claims 1-9).

61. For that reason, in my view, the ’316 Patent is best understood as disclosing a recognition-aware handwriting interface in which beautification and editing are linked, but in which the challenged claims are directed more specifically to editing by means of recognition-derived ranges and cursor placement. The title and some portions of the specification emphasize beautification, but the claims target direct interaction with the displayed digital ink once recognition and segmentation have supplied usable element-level geometry, including character-level geometry in the disclosed examples and dependent claims. (‘316 Patent, EX. 1001, Background/Summary, 3:42-4:52; Description, 15:50-18:20; Claims 1-9).

62. The structure of the claim set reflects that focus. Claims 1, 4, and 7 are counterpart claims directed respectively to system, method, and computer-readable-

medium forms of the same basic concept. Claims 2, 5, and 8 then add the limitation that the digital-ink elements include text characters and that the cursor is displayed between first and second characters in relation to their determined ranges. Claims 3, 6, and 9 add that the geometrical features include height and width of the determined ranges. Thus, the claims are organized around a core cursor-management concept and then narrow that concept by character-level and geometry-based requirements. ('316 Patent, EX. 1001, Claims 1-9).

A. The '316 Patent Claims

63. I understand that the '316 Patent contains 9 claims, including 3 independent claims and 6 dependent claims. These claims are set forth below.

i. Claim 1 - '316 Patent

[1pre] A system for providing editing of digital ink on computing devices, each computing device comprising a processor and at least one non-transitory computer readable medium for recognizing handwriting input under control of the processor, the at least one non-transitory computer readable medium configured to:

[1a] cause display of, on a display interface of a computing device, digital ink in accordance with the handwriting input;

[1b] determine, from segmentation information of the handwriting

input determined through the recognition of the handwriting input, ranges of elements of the digital ink;

[1c] cause display of, on the display interface, an input cursor in relation to the determined digital ink element ranges; and

[1d] cause said display of the input cursor within a region of overlap of ranges of elements of the digital ink determined in the step of determining ranges of elements of the digital ink.

ii. Claim 2:

The system according to claim 1, wherein the elements of the digital ink include text characters, the at least one non-transitory computer readable medium configured to display the input cursor at a position between first and second characters in relation to the determined ranges of the first and second characters.

iii. Claim 3:

The system according to claim 1, wherein the geometrical features include height and width of the determined ranges.

iv. Claim 4:

[4pre] A method for providing editing of digital ink on computing devices, each computing device comprising a processor and at least one

non-transitory computer readable medium for recognizing handwriting input under control of the processor, the method comprising:

[4a] displaying, on a display interface of a computing device, digital ink in accordance with the handwriting input;

[4b] determining, from segmentation information of the handwriting input determined through the recognition of the handwriting input, ranges of elements of the digital ink;

[4c] displaying, on the display interface, an editing cursor in relation to the determined digital ink element ranges; and

[4d] causing said display of the editing cursor within a region of overlap of ranges of elements of the digital ink determined in the step of determining ranges of elements of the digital ink.

v. Claim 5:

A method according to claim 4, wherein the elements of the digital ink include text characters, the input cursor being displayed at a position between first and second characters in relation to the determined ranges of the first and second characters.

vi. Claim 6:

A method according to claim 4, wherein the geometrical features include

height and width of the determined ranges.

vii. Claim 7:

[7pre] A non-transitory computer readable medium having a computer readable program code embodied therein, said computer readable program code adapted to be executed to implement a method for providing editing of digital ink on a computing device, the computing device comprising a processor and at least one system non-transitory computer readable medium for recognizing handwriting input under control of the processor, the method comprising:

[7a] displaying, on a display interface of a computing device, digital ink in accordance with the handwriting input;

[7b] determining, from segmentation information of the handwriting input determined through the recognition of the handwriting input, ranges of elements of the digital ink;

[7c] displaying, on the display interface, an editing cursor in relation to the determined digital ink element ranges; and

[7d] causing said display of the editing cursor within a region of overlap of ranges of elements of the digital ink determined in the step of determining ranges of elements of the digital ink.

viii. Claim 8:

A non-transitory computer readable medium according to claim 7, wherein the elements of the digital ink include text characters, the input cursor being displayed at a position between first and second characters in relation to the determined ranges of the first and second characters.

ix. Claim 9:

A non-transitory computer readable medium according to claim 7, wherein the geometrical features include height and width of the determined ranges.

X. SUMMARY OF THE PRIOR ART**A. Kang (US 2014/0300609 A1)**

64. U.S. Patent Publication No. 2014/0300609, titled “Device and Method for Editing Ink Text Data,” published on October 9, 2014, to inventors Kang and Shin and assigned to DIOTEK CO., LTD. The Kang prior art reference is **Exhibit 1006** and is made a part hereof by reference. I have been informed that Kang qualifies as prior art to the ’316 Patent. The Kang publication describes a device and method for editing digital ink data displayed on a screen and is directed to making insertion, deletion, and update of digital ink more intuitive and more similar to familiar text-editing behavior. (EX. 1006, Front Page; Abstract;

Background/Summary [0003], [0009]-[0010].)

65. Kang's Background section identifies the problem it seeks to solve. Kang explains that users of mobile terminals such as smartphones increasingly input characters by touching the display screen directly, but existing systems still imposed a limitation in that "a user cannot freely edit digital ink data displayed on a display screen at his/her disposal." (EX. 1006, Background/Summary [0005]-[0008].) Kang states that its objective is to provide a device and method that edit digital ink data "in an editing manner similar to that of the text data" and that allow a user to "intuitively and conveniently" perform insertion, deletion, and update on digital ink displayed on a screen. (EX. 1006, Background/Summary [0009]-[0010].)

66. Kang's disclosed architecture is a data editor implemented as a mobile terminal or computer. The data editor includes, among other things, a stroke order rearrangement processing unit, a data generation unit, an editing unit, a cursor position determination unit, and a storage unit. (EX. 1006, Description [0061]-[0065].) Kang also discloses a user interface in which text data and digital ink data can be displayed in series along virtual lines, with digital ink entered through an input panel and text entered through a keyboard-supported text input mode. (EX. 1006, Background/Summary [0018]-[0023]; Description [0067]-[0070].)

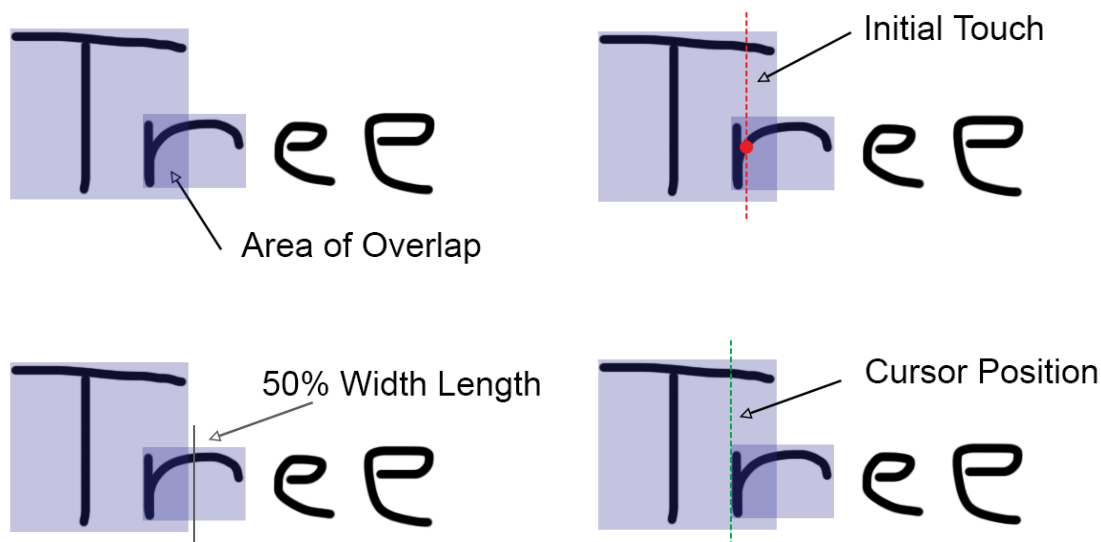
67. Kang is also explicit about the data structures it uses to establish a

geometric framework for editing. It defines “digital ink data” as data generated during a handwriting process on the basis of strokes input in a digital ink input mode, defines a “stroke” as a set of points, defines a “rearrangement stroke” as a stroke reflecting rearrangement of stroke input order, and defines a “stroke segment” as the spatial extent or range formed when points of a stroke are projected onto a virtual x-axis. Kang further defines a “stroke block” as a segmented unit composed of two or more grouped strokes, “text data” as data generated in a text input mode such as a keyboard, and “ink text data” as mixed data including digital ink data, text data, and ink space data. (EX. 1006, Description [0049]-[0060].)

68. The core processing disclosed by Kang is a processing process to reorder strokes, generate spatially grouped editable digital-ink units, and create spacing information between those units. Kang teaches rearranging the input order of two or more strokes based on a predefined criterion, including distance from an input position to a virtual reference line. (EX. 1006, Background/Summary [0012]-[0017], [0024]-[0029].) Kang then projects rearrangement strokes onto a first virtual x-axis to form stroke segments, compares overlapped length against a first reference value, and when the overlap threshold is met, spatially groups the corresponding strokes into a single segmented element known as one stroke block. Kang likewise compares distances between adjacent stroke segments or stroke-

block segments against a second reference value to decide whether ink space data should be generated between neighboring units. (EX. 1006, Background/Summary [0015]-[0017], [0027]-[0029]; Description [0053]-[0059], [0133]-[0135].)

69. Kang further discloses explicit cursor-position logic tied to this spatially grouped digital ink. The cursor position determination unit includes a sensing unit and an analysis unit. Kang states that the analysis unit determines data crossing a virtual axis formed on the basis of a touch position and then determines the cursor position from that geometric result. (EX. 1006, Background/Summary [0021]-[0023]; Description [0113]-[0115], [0138]-[0139].) In the detailed example of FIG. 9, Kang explains that the same virtual axis may cross digital ink data corresponding to the 20th, 22nd, 23rd, 24th, and 25th rearrangement strokes, and that the cursor may then be determined immediately behind the 25th rearrangement stroke or, if the touch position is in the front region of that stroke, immediately behind the 24th rearrangement stroke. Kang also explains that the front/end separation may be determined using 50% of the width length of the specific stroke or rearrangement stroke, as shown in FIG. 10. (EX. 1006, Description [0115]-[0116].)



70. The demonstrative figures above illustrate how a person of ordinary skill in the art would visualize Kang's geometric cursor-resolution process in action.

71. As shown in the first panel, the spatial ranges (represented by the shaded bounding boxes) of adjacent handwritten characters often crowd one another, creating an **"Area of Overlap."**

72. When a user attempts to edit this congested text, they provide an **"Initial Touch"** (represented by the red dot in the second panel). As taught by Kang, the system's analysis unit forms a virtual axis (the red dashed line) in the height direction of the screen based on this exact touch position. Because this single virtual axis intersects multiple overlapping stroke units simultaneously (here, the

"T" and the "r"), the edit location is ambiguous.

73. To resolve this, Kang applies geometric rules to the spatial ranges. As shown in the third panel, Kang calculates the front/end separation based on the geometric features of the strokes—specifically using **"50% of the width length"** of the stroke.

74. Because the initial touch falls in the "front region" (to the left of the 50% width line), Kang's system snaps the final **"Cursor Position"** (the green dashed line in the fourth panel) to the appropriate geometric boundary immediately behind the preceding stroke unit. A POSITA would understand this as an express, concrete mechanism for calculating and displaying an editing cursor within a region of overlapping digital ink ranges.

75. Kang additionally teaches that the same editing framework can operate across digital ink data, text data, and ink space data within one interface. Kang states that digital ink data and text data may have the same data structure and may be displayed in spaces between virtual lines along those virtual lines. (EX. 1006, Background/Summary [0018]-[0020].) Kang further explains that the editing unit can perform operations on a “variety of data including the digital ink data, the text data, and the ink space data” based on cursor position, and illustrates mixed editing in a section that includes both text data and digital ink data. (EX. 1006, Description

[0118], [0121]-[0124].)

76. In summary, Kang discloses a robust digital-ink editing system that utilizes a geometric processing process. It teaches capturing handwriting as strokes and points, rearranging stroke order, projecting strokes into spatial stroke segments, grouping sufficiently overlapped segments into segmented elements (stroke blocks), generating ink space data between separated units, displaying digital ink and text data in a common interface, and performing editing operations based on a cursor position determined from a touch-defined virtual axis. Kang also explicitly describes mechanisms for determining edit cursor placement that result in cursor placement in the overlapping regions. Kang therefore provides a detailed prior-art reference directed not merely to displaying digital ink, but to spatially organizing, positioning, and editing digital ink data in a text-like editing environment. (EX. 1006, Abstract; Background/Summary [0012]-[0023]; Description [0063]-[0070], [0115]-[0118], [0133]-[0142].)

B. Clary (US 8,115,748 B2)

77. U.S. Patent No. 8,115,748, titled “Electronically Capturing Handwritten Data,” issued on February 14, 2012, to inventor Clary and assigned to ADS Software MGMT. L.L.C. The Clary prior art reference is **Exhibit 1007** and is made a part hereof by reference. I have been informed that Clary qualifies as prior

art to the '316 Patent.

78. Clary broadly concerns pen-enabled computing devices that capture handwritten input and manipulate the resulting data, including by editing handwritten and textual content. Clary explains that once handwriting is entered into the pen-enabled computing system, the data may be stored as an “electronic ink” copy, as a recognized “text” copy, or as both. Clary also discloses that the computer program product may spatially segment handwritten data and may allow the addition, insertion, deletion, and replacement of data within a field. (EX. 1007, Background/Summary 1:42, 5:12, 6:53-6:62, 9:4-10:22.)

79. Clary’s disclosed architecture is a pen-enabled computing device with a handwriting capture interface, a central processing subsystem, a user interface, a computer program product, and an external device interface. The central processing subsystem may include a CPU, non-volatile memory, and RAM, with the non-volatile memory used for storing and/or executing the computer program product as well as storing useful data such as electronic ink. Clary also discloses a handheld-computer embodiment having an interactive display, control buttons, a speaker, a handwriting tablet, and a stylus. (EX. 1007, Description 11:39-12:6, 13:44.)

80. Clary further explains how handwritten input is represented and geometrically parsed by its central processing subsystem for later editing. The

handwriting capture interface provides a sequence of (x,y) coordinate pairs that define marks made by the stylus. The CPU and computer program product then define spatial ranges, represented as bounding boxes, around each stroke. Clary explains that each bounding box typically extends from the smallest x value to the largest x value of the relevant coordinate pairs, with y coordinates extending between successive baselines. Clary also states that, although a bounding box may encompass a single stroke, the box may be defined differently based on spatial or temporal characteristics of one or more strokes in order to bundle them together as a segmented unit for editing purposes. (EX. 1007, Background/Summary 1:42, 6:53, 9:4, 9:41, 10:4; Description 17:45-18:11.)

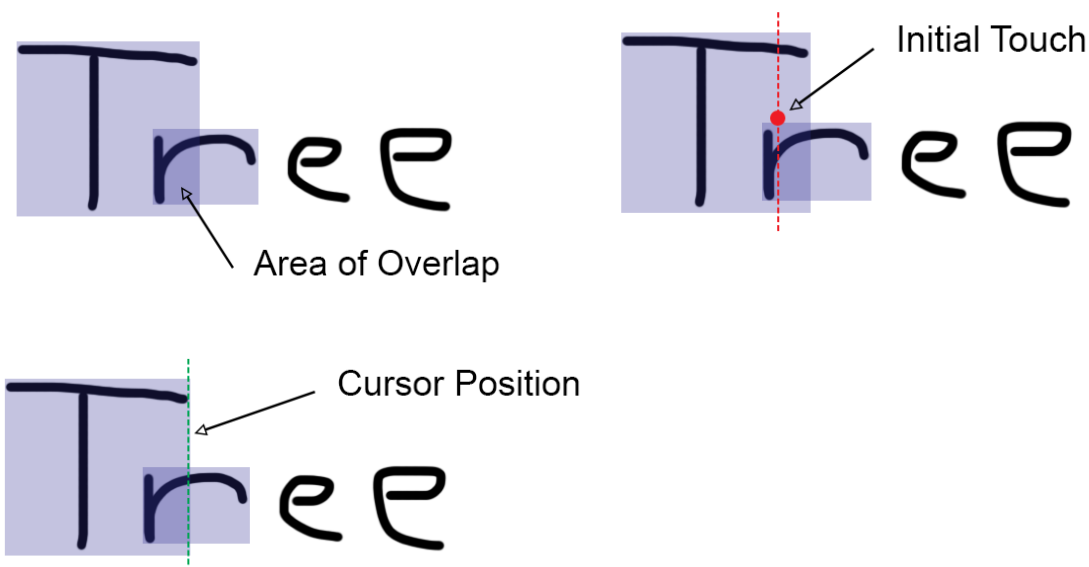
81. The detailed editing discussion in Clary is particularly relevant. Clary describes a technique for “seamlessly editing handwritten and/or textual data” by first displaying the handwritten input, then allowing the user to select an actuation position relative to that input, such as by positioning a cursor 912. Clary states that the pen-enabled computing device determines whether the cursor position is located within a spatial range (i.e., a bounding box) and, if the cursor position is located within “one or more bounding boxes 902,” the device determines the x coordinate of the center of the largest bounding box. A person of ordinary skill in the art would understand that a cursor position that overlaps “more than one” bounding boxes

would be a region of overlap. If the cursor position is to the right of that center, the cursor is shifted to the right side of the largest bounding box; if the cursor position is to the left of that center, the cursor is shifted to the left of the smallest bounding box. If the cursor position is not within a bounding box, the cursor need not be shifted. (EX. 1007, Description 17:45, 18:40.)

82. Clary also describes specific insertion, replacement, and deletion mechanics tied to this geometric framework. If text is inserted, Clary moves the strokes and the bounding boxes to the right of the insertion by an amount equal to the width of the insert. If the insertion is handwritten rather than typed, Clary forms bounding boxes about the inserted strokes and again moves the existing handwritten input to the right by the width of the insert. Clary also discloses deleting at least a portion of existing handwritten input and then moving the remaining handwritten input to the left to close the resulting gaps, including by reducing the x values of the coordinate pairs within the remaining bounding boxes. (EX. 1007, Description 19:5-19:43.)

83. Clary further states that these editing capabilities allow “seamless editing” of textual or handwritten input with either additional text or handwriting and that the resulting message remains appropriately spaced notwithstanding insertion and deletion of handwriting and/or textual data. Thus, although Clary

includes broader disclosure regarding form identification and pen-enabled data management, it also contains a detailed editing framework in which handwritten input is represented as coordinate data, spatially segmented into stroke-based bounding boxes, and edited through cursor placement and bounding-box-based spacing adjustments. (EX. 1007, Description 18:40-19:62.)



84. The demonstrative figures above illustrate how a person of ordinary skill in the art would visualize Clary's geometric cursor-resolution process in action. As shown in the first panel, the spatial ranges (represented by the shaded bounding boxes) of adjacent handwritten characters often crowd one another, creating an **"Area of Overlap."**

85. When a user attempts to edit this congested text, they provide an **"Initial Touch"** (represented by the red dot in the second panel). As taught by Clary, in one embodiment the system evaluates whether this initial cursor position is located "within one or more bounding boxes." Here, the initial touch clearly falls inside multiple distinct bounding boxes simultaneously, proving that the ranges are physically occupying the same coordinate space.

86. To resolve this ambiguity, Clary's central processing unit applies geometric rules to the overlapping spatial ranges. By calculating the x-coordinate of the center of the largest bounding box and evaluating whether the initial touch falls to the left or right of that center, Clary geometrically snaps the final **"Cursor Position"** (the green dashed line in the third panel) to the appropriate boundary—such as the left side of the smallest box or the right side of the largest box. A POSITA would understand this as an express, concrete mechanism for calculating and displaying an editing cursor within a region of overlapping digital ink ranges. In summary, Clary discloses a pen-enabled computing system that captures handwriting as electronic ink and/or recognized text, spatially segments that handwriting into geometric ranges (bounding boxes) around strokes or grouped stroke units, evaluates whether a cursor lies within one or more of those ranges, shifts the cursor based on box geometry, and edits the displayed handwritten or

textual data by insertion, deletion, replacement, and spacing changes. Clary therefore provides a detailed prior-art reference for coordinate-based handwriting capture, spatial stroke-box segmentation, bounding-box-based cursor resolution (including overlapping bounding boxes), and editing of displayed handwritten data relative to such cursor. (EX. 1007, Abstract; Background/Summary 1:42, 5:12, 6:53-6:62, 9:4-10:22; Description 11:39-12:6, 13:44, 17:45-19:62.)

C. Li (US 2010/0066691 A1)

87. U.S. Patent Publication No. 2010/0066691, titled “Input Apparatus and Computer Readable Recording Medium Recorded with Image Processing Program,” published on March 18, 2010, to inventor Li. The Li prior art reference is **Exhibit 1008** and is made a part hereof by reference. I have been informed that Li qualifies as prior art to the ’316 Patent. Li describes a handwriting input apparatus and associated program that enable easy editing and high-quality display of edited handwritten content by defining a geometric locus for handwritten characters and graphics and then moving edited content along that locus during deletion, correction, and insertion. (EX. 1008, Front Page; Abstract; Background/Summary [0003], [0012], [0020]-[0039].)

88. Li’s Background section identifies the problem it seeks to solve. Li explains that handwriting input is user-friendly and widely used, but prior

handwritten-editing systems either edited previously input text data in a limited way or failed to disclose how the edit task was concretely performed. Li specifically criticizes prior approaches that could leave an edited region with “unnatural text or image displayed” and states that its objective is to provide a handwriting input apparatus and recorded program that enable easy editing together with “high-quality display of edited content.” (EX. 1008, Background/Summary [0005]-[0012].)

89. Li’s disclosed architecture is an input apparatus implemented, in one embodiment, in a multifunctional peripheral having an operation unit, a display unit, a touch panel, a control unit, and memory. Li further discloses a module configuration including a handwriting input module, a data cache memory, a display module, a handwritten object extracting module, a handwriting locus defining module, an edit management module, and first and second databases. The input apparatus is used not only to input information but also to edit previously input information, including both characters and graphics. (EX. 1008, Description [0071]-[0075], [0078]-[0083].)

90. Li is also explicit about the recognition-based segmentation and spatial organization of handwritten content, operating as a multi-stage edit process. Li states that a handwriting recognition dictionary is stored in the first database, with handwritten object patterns for characters and/or graphics registered in advance.

The handwritten object extracting module compares the input handwritten object against those stored patterns, thereby recognizing the character and/or graphic. Recognized handwritten objects are then incorporated into a handwriting group, and the handwriting locus defining module then detects a minimum “circumscribing rectangle” (a defined spatial range) for each handwritten object in the group, detects a geometric center for each rectangle, and connects those centers with a straight or curved line to define a handwriting locus. (EX. 1008, Description [0085]-[0089], [0111]-[0112].)

91. The core editing mechanism disclosed by Li is that handwritten gestures serve as edit commands and the recognized objects are moved along the previously defined locus during execution of those commands. Li explains that the edit management module compares the input handwritten gesture against stored gesture data and recognizes edit commands represented primarily by deletion, correction, and insert commands. Li further explains that when those edit tasks are performed, the handwriting locus itself is used to move character objects as characters are deleted, replaced, or added. In the case of deletion, a character object at the gesture location is deleted and a following character object is moved forward along the locus, including until there will be a “predetermined overlap region” between circumscribing rectangles. In the case of correction, the original object is

deleted, a corrected object is input, and the corrected object is repositioned so that its circumscribing-rectangle center lies on the locus. (EX. 1008, Background/Summary [0025]-[0032]; Description [0092]-[0094], [0113]-[0118].)

92. Li also discloses insertion mechanics closely tied to this derived circumscribing-rectangle geometry. When an insert gesture is input at an insert position, Li moves character objects before or after the insert position along the locus in order to create a predetermined space. Li calculates insertion spacing using the width of a rectangle circumscribing the inserted object together with predetermined distances to adjacent objects. Li then places the inserted object so that its center is positioned on a locus connecting the centers of the adjacent objects and at a middle position therebetween. Li also discloses word-level editing examples in which deletion, correction, and insertion are carried out using the same circumscribing-rectangle and locus-based logic. (EX. 1008, Background/Summary [0033]-[0037]; Description [0123]-[0128], [0143]-[0155].)

93. Li additionally explains that the same approach can be applied not only to characters but also to graphics. The abstract itself states that the defining module defines a locus of a character and/or graphic contained in a group and that execution of an edit command is accompanied by movement of the character and graphic in the group, following the defined locus. The detailed description likewise explains

that the invention covers both characters and graphics and provides examples for editing graphical objects such as flowchart objects by detecting geometric centers and connecting those centers to define a locus. (EX. 1008, Abstract; Background/Summary [0013]-[0039]; Description [0073], [0156]-[0174].)

94. In summary, Li discloses a handwriting-editing system in which handwritten characters and graphics are recognized through a handwriting recognition dictionary, organized into groups, bounded by geometric circumscribing rectangles, and associated with geometric centers that define a locus for the group. Li then uses handwritten edit gestures to delete, correct, and insert objects while moving the affected recognized objects along that locus, including by using circumscribing-rectangle width, spacing, and overlap relationships to determine the resulting placement. Li therefore provides a detailed prior-art reference for a recognition-based process that derives handwritten objects, circumscribing-rectangle geometry, locus-based movement of edited content, and overlap-aware editing of handwritten material. (EX. 1008, Abstract; Background/Summary [0012]-[0039]; Description [0085]-[0094], [0111]-[0128], [0143]-[0155].)

XI. PROSECUTION HISTORY ERROR

95. As a technical expert performing review and analysis for patent matters

involving patent validity/invalidity and infringement/non-infringement subject matter, it is an important part of my process to obtain and review the Patent Office's so-called "file wrapper" for the patent prosecution history. It is my understanding that a patent's file wrapper is the Patent Office's complete recorded history of the prosecution process including the communications between the examiner and the inventor (or his/her representatives). For the purpose of my work for this matter, I obtained and reviewed the file wrapper for the '316 patent which is **Ex. 1004** and is made a part hereof by reference.

96. One of the notable items I reviewed in the '316 file wrapper (then application 14/978,081) was an Office Action dated October 5, 2017 (Ex. 1004). Within this Office Action, the examiner notified the Applicant's prosecuting attorney that claims 1 through 30 were all rejected and laid out his arguments that included consideration of two prior art applications, now patents, Lee (US 2016/1047723) and Clary (US 2010/0309131). (Ex. 1004) I note that Clary is one of the important prior art patents I evaluated in detail within this declaration.

97. One of the particular elements that the examiner considered in his Office Action rejection analysis was the '316 patent's required limitation of "cause/causing said display of the input cursor within a region of overlap of ranges of elements of the digital ink" ('316 claims 1, 4, & 7). (Ex. 1004) At the time of

the Office Action the examiner apparently believed the prior art disclosed this element.

98. I have also reviewed the '316 Applicant's response "REMARKS" to this office action that were provided back to the examiner for consideration. (Ex. 1004) On page 12 of the REMARKS, the Applicant asserts and represents the following based on the Clary prior art:

Accordingly, the reference describes an adjustment of a position of a cursor based on its selected position relative to bounding boxes, and adjusting the cursor position (for example, the X-value) accordingly. However, the reference **does not describe or disclose any overlapping bounding boxes**. Further, the reference does not disclose or describe displaying a cursor "within a region of **overlap of ranges of elements** of the digital ink" as claimed. Emphasis added. Accordingly, Applicant respectfully submits that the cited references, taken alone or in combination, are silent regarding at least this feature.

(Ex. 1004 at 95)

99. It is my understanding that the examiner ultimately reversed the claim rejections after receiving these Applicant remarks and representations.

100. Based upon a thorough review of the '316 patent and the relevant prior art (including particularly Clary) viewed as a POSITA at the time of the invention, it is my opinion that the specific, emphasized assertions by Applicant are not accurate.

101. First, the Applicant's statement that Clary does not describe or disclose any overlapping bounding boxes is plainly not accurate. As discussed in detail in other portions of this declaration, Clary does describe overlapping bounding boxes in multiple locations. One such instance was even shown on page 12 of the Applicant's remarks citing Clary paragraph 0066 stating "if the cursor 912 position is located **within one or more bounding boxes** 902". (Ex. 1004) In this text, Clary is fundamentally describing the algorithm for handling the situation of when the initial cursor location is placed on two overlapping bounding boxes. Clary discloses the overlapping bounding box condition and defines what should happen when they do overlap.

102. Second, the Applicant's additional assertion that Clary "does not disclose or describe displaying a cursor 'within a region of overlap of ranges of elements of the digital ink'" is likewise inaccurate and a misrepresentation in my view. (Ex. 1004) As described in much more detail in other sections of this declaration, any overlapping bounding boxes within Clary are - by how bounding boxes are expressly defined - are conditions where elements of the digital ink are overlapping. Again, a significant portion of the '316 patent is devoted to explaining how to adjust and reposition the cursor when this region of overlap condition occurs.

103. In summary, it is my opinion that the particular statements made by the Applicant to the examiner regarding Clary's lack of disclosure and/or description of the display of a cursor with a region of overlap of digital ink were not accurate. I believe a POSITA reviewing the application at the time would have understood that this element was described and disclosed in Clary. It is further my opinion that the proper consideration of Clary and what it disclosed more likely than not would have influenced the allowance of '316 claims. See also my more detailed and descriptive explanations of Clary in other sections of this declaration.

XII. ANTICIPATION ANALYSIS

104. It is my opinion that the '316 Patent claims are anticipated by multiple different prior art references as discussed in detail below.

A. The Kang Prior Art Reference (US 2014/0300609 A1)

105. In my opinion from the perspective of a POSITA, the '316 Patent claims are anticipated by Kang. As detailed below, a POSITA would have understood that Kang expressly discloses every limitation of the challenged claims, either in the same system architecture shown in FIG. 2 and described in the specification, or in the corresponding method and computer-readable-medium forms expressly recited in Kang's own claims. (Kang, Description, [0063]-[0064], [0124]-[0125], [0144]; Kang, Summary, [0024], [0030]; Kang, Claims 13, 19).

106. Kang is particularly relevant because it is not merely a handwriting-capture reference. It is directed to editing displayed digital ink data in a mixed handwriting/text environment, including insertion, deletion, update, cursor determination, stroke-order rearrangement, grouping of overlapping stroke-derived units, and spacing between adjacent handwritten units. Those are the same basic technical themes that drive the asserted '316 Patent claims. (Kang, Summary, [0012]-[0023]; Kang, Description, [0086]-[0091], [0107]-[0118], [0134]-[0140]).

107. At a practical level, Kang is solving the same basic problem as the '316 Patent: once handwriting is displayed on a screen, the system must organize that handwriting into meaningful units and then place an editing cursor at the correct position even when the writing is crowded, slanted, or overlapping. Kang addresses that problem by converting raw strokes into ordered handwritten units, assigning spatial extents to those units, and then determining cursor position by reference to the displayed units that cross a touch-defined virtual axis. (Kang, Description, [0083]-[0096], [0110]-[0118], [0134]-[0139]).

i. '316 Independent Claims 1, 4, and 7

108. Kang discloses the same core functionality in apparatus, method, and computer-readable-medium form. FIG. 2 and the accompanying description disclose a data editor implemented on a mobile terminal or computer. Kang's claim

13 separately recites a method of editing digital ink data displayed on a display, and Kang's claim 19 separately recites a computer-readable recording medium storing instructions that cause a computing device to perform the same editing operations. For that reason, Claims 1, 4, and 7 of the '316 Patent are disclosed by Kang, with Claim 1 being disclosed by Kang's system architecture, Claim 4 by Kang's method disclosures, and Claim 7 by Kang's recording-medium disclosures. (Kang, Description, [0063]-[0064], [0124]-[0125], [0144]; Kang, Claims 1, 13, 19).

(1) '316 Preamble

- a. [1pre] "A system for providing editing of digital ink on computing devices, each computing device comprising a processor and at least one non-transitory computer readable medium for recognizing handwriting input under control of the processor, the at least one non-transitory computer readable medium configured to"
- b. [4pre] A method for providing editing of digital ink on computing devices, each computing device comprising a processor and at least one non-transitory computer readable medium for recognizing handwriting input under control of the processor, the method comprising:
- c. [7pre] A non-transitory computer readable medium having a computer readable program code embodied therein, said computer readable program code adapted to be executed to implement a method for providing editing of digital ink on a computing device, the computing device comprising a processor and at least one system non-transitory computer readable medium for recognizing handwriting input under control of the processor, the method comprising:

109. Kang discloses the preamble's system for providing editing of digital ink on computing devices. Kang explains that the "data editor 100" may be implemented as "a various type of a mobile terminal or computer," and FIG. 2 shows the modules that perform the relevant work: stroke-order rearrangement, data generation, editing, input handling, cursor-position determination, and storage. In other words, Kang is not describing a passive handwriting pad. It is describing a computing system built to receive handwriting, convert that handwriting into editable displayed data, and then edit that displayed data. (Kang, Description, [0063]-[0064], [0107]-[0110], [0126]).

110. Kang also discloses the processor and non-transitory-medium aspects of the preamble. Paragraph [0125] explains that the disclosed modules may execute on "one or more processors." Paragraph [0124] explains that the storage unit 170 is a kind of memory and may be implemented using non-volatile memory, volatile memory, and other storage media. Paragraph [0144] further explains that the invention may be implemented in a computer-readable medium storing computer-readable instructions, data structures, program modules, or other data. Claim 19 then expressly recites a "computer-readable recording medium" storing an instruction set that, when executed by a computing device, performs the disclosed digital-ink editing operations. (Kang, Description, [0124]-[0125], [0144]; Kang,

Claim 19).

111. Kang further discloses processor-controlled handling of handwriting input as structured, editable handwritten data. Kang explains that digital ink data is generated during a handwriting process from input strokes, that strokes are processed into rearrangement strokes, and that grouped rearrangement strokes may generate recognizable character-level units such as Korean characters and Korean syllables. Kang also discloses that digital ink data and text data may share the same data structure. A POSITA would therefore understand Kang to disclose a processor-controlled system for recognizing and organizing handwriting input for display and editing, not merely for raw capture. (Kang, Description, [0050]-[0055], [0090]-[0091], [0117], [0124]-[0125]; Kang, Summary, [0018]-[0020]).

(2) ‘316 Claim 1/4/7 Element 1

- a. [1a] cause display of, on a display interface of a computing device, digital ink in accordance with the handwriting input;
- b. [4a] displaying, on a display interface of a computing device, digital ink in accordance with the handwriting input;
- c. [7a] displaying, on a display interface of a computing device, digital ink in accordance with the handwriting input;

112. Kang expressly discloses this limitation. Paragraph [0050] states that digital ink data is generated during a handwriting process on the basis of strokes input in a digital ink input mode such as a stylus pen. Paragraph [0068] then gives

the concrete display example: digital ink data 33 is generated from stroke 32 input on input panel 152 with stylus 38 and displayed on display panel P. Paragraph [0065] likewise explains that display 10 displays digital ink data on the screen, and paragraph [0088] explains that the data generation unit generates digital ink data for display on the screen similarly to text data.

113. In plain terms, Kang teaches that the user writes with a stylus, the system turns those pen strokes into digital ink data, and that digital ink data is displayed back to the user on the screen in accordance with the handwriting input. That is the claimed display step. (Kang, Description, [0050], [0065], [0068], [0088]).

(3) '316 Claim 1/4/7 Element 2

- a. [1b] determine, from segmentation information of the handwriting input determined through the recognition of the handwriting input, ranges of elements of the digital ink
- b. [4b] determining, from segmentation information of the handwriting input determined through the recognition of the handwriting input, ranges of elements of the digital ink;
- c. [7b] determining, from segmentation information of the handwriting input determined through the recognition of the handwriting input, ranges of elements of the digital ink;

114. Kang begins with raw handwritten strokes. It then rearranges those strokes into an order that reflects actual input position. From there, it computes a

two-dimensional “stroke region” for each rearrangement stroke, and then projects that region onto a one-dimensional virtual x-axis to create a “stroke segment” with a left end and a right end. That projected segment is the horizontal extent of the handwritten unit. If neighboring projected segments overlap enough, Kang merges them into a single “stroke block” and a corresponding “stroke block segment.” If neighboring projected units are far enough apart, Kang generates ink space data between them. (Kang, Description, [0053]-[0059], [0083]-[0085], [0091], [0093]-[0096], [0134]-[0135]).

115. The Kang process supplies the claimed ‘316 “elements,” the claimed “ranges,” and the claimed “segmentation information.” Under the proper construction of the claims, “segmentation information” encompasses the spatial grouping of strokes generated by a recognition process to separate handwriting into editable units. Kang discloses exactly this. Kang’s “elements” are the operative handwritten units: a single rearrangement stroke, or a “stroke block” where multiple strokes are spatially grouped together. Kang’s “ranges” are the spatial extents calculated for those units via projection onto the virtual x-axis (i.e., the stroke segments). (Kang, Description, [0053]-[0059], [0089]-[0091]).

116. Kang also discloses the recognition-oriented aspect of this limitation under either proposed construction of “recognizing handwriting input.” Considering

Patent Owner's construction, Kang clearly recognizes the creation of digital content by a user via an input tool, specifically capturing strokes from a stylus pen. Considering Goodnotes' construction, Kang does not merely capture raw ink; it detects this handwriting input and actively derives segmented units—calculating their geometric overlap on the x-axis and merging them into grouped editable units (stroke blocks) for subsequent display and editing operations. A POSITA would understand that Kang's active, geometric grouping of strokes to form character-level units anticipates the '316 Patent's derivation of segmentation information and ranges through a recognition process regardless of which standard is applied. (Kang, Summary, [0018]-[0020]; Kang, Description, [0090], [0095]-[0096], [0117], [0124]).

117. For the same reasons, Kang discloses the claimed “determining” step. The system is not merely storing strokes after the fact. It is actively deciding which handwritten units remain separate, which ones merge into a stroke block, and where spacing should be inserted between adjacent units. That is the determination of ranges from segmentation information. (Kang, Description, [0083]-[0085], [0091], [0093]-[0096], [0134]-[0135]).

(4) '316 Claim 1/4/7 Element 3

- a. [1c] cause display of, on the display interface, an input cursor in relation to the determined digital ink element ranges; and
- b. [4c] displaying, on the display interface, an editing cursor in relation to the determined digital ink element ranges; and
- c. [7c] displaying, on the display interface, an editing cursor in relation to the determined digital ink element ranges; and

118. Kang expressly discloses a cursor displayed in relation to the determined digital-ink ranges. Paragraph [0110] explains that the cursor position determination unit 160 determines the position of cursor C displayed on the screen and that the editing unit performs editing on the basis of that cursor position. Paragraph [0111] explains that the cursor may be placed in digital ink data, between digital ink data and text or ink space data, in text data, between text data and ink space data, in ink space data, or at the end of the most recently generated data.

119. Most importantly, paragraph [0115] explains how Kang chooses the cursor position. The analysis unit forms a virtual axis in the height direction of the screen based on the touch position, determines which displayed data crosses that axis, and analyzes which rearrangement stroke has been selected in order to determine the position of the cursor. In plain terms, Kang does not simply drop the cursor wherever the user touches. It uses the already-determined handwritten units and their displayed locations to decide where the cursor belongs. That is a cursor that is displayed “in relation to” the determined digital-ink element ranges. (Kang,

Summary, [0021]-[0022]; Kang, Description, [0110]-[0111], [0115]).

(5) ‘316 Claim 1/4/7 Element 4

- a. [1d] cause said display of the input cursor within a region of overlap of ranges of elements of the digital ink determined in the step of determining ranges of elements of the digital ink.
- b. [4d] causing said display of the editing cursor within a region of overlap of ranges of elements of the digital ink determined in the step of determining ranges of elements of the digital ink.
- c. [7d] causing said display of the editing cursor within a region of overlap of ranges of elements of the digital ink determined in the step of determining ranges of elements of the digital ink.

120. This is one of the strongest Kang disclosures. The claims require causing display of the cursor within a “region of overlap” of the determined ranges. Kang explicitly anticipates this by describing how it handles cursor placement in a crowded, overlapping handwritten region. Kang explains that when the user touches the screen, the analysis unit forms a virtual axis 92 in the height direction of the screen and determines which spatial data ranges cross that axis. In the concrete example of FIG. 9, Kang notes that the same virtual axis crosses the digital ink data corresponding to the 20th, 22nd, 23rd, 24th, and 25th rearrangement strokes simultaneously. A POSITA would understand that if a single vertical line passes through multiple distinct handwritten units, the spatial ranges (x-axis segments) of

those units are physically occupying the same space. That is an express disclosure of a region of overlap. (Kang, Description, [0115]).

121. Paragraph [0116] then explains how Kang resolves cursor placement within this overlapping region. Because the ranges of the 24th and 25th strokes overlap the touch-defined axis, Kang teaches making a choice between neighboring cursor positions. Kang explains that the cursor may be placed immediately behind the 25th rearrangement stroke, or, if the touch position is in the front region, it may instead be placed immediately behind the 24th rearrangement stroke.¹ Kang uses the geometric features of the ranges—specifically, determining separation based on 50% of the width length of the stroke—to snap the cursor to the appropriate boundary within this contested area. (Kang, Description, [0116]).

122. A POSITA would understand that Kang's logic for snapping the cursor 'immediately behind' a specific stroke within an overlapping sequence mathematically places the cursor within an area where the determined ranges of elements overlap (consistent with Patent Owner's construction) and where adjacent segmented elements overlap (consistent with Goodnotes' construction). Because

¹ For clarify, my analysis of the Kang patent informs me that Kang's use of "behind" means after or to the right of the particular stroke in left to right handwriting.

the spatial segments of the 24th and 25th strokes are explicitly described as crossing the same virtual axis, they physically overlap on the screen. Therefore, placing the cursor at the boundary (i.e., 'immediately behind') the 24th stroke drops the cursor at a geometric coordinate that is still physically inside that shared overlapping area. This edge-snapping resolution is functionally and mathematically identical to the '316 Patent's own disclosure of snapping a cursor to an inner range border within an overlapping area. (Compare Kang, Description, [0116], with '316 Patent, Fig. 18B and Description, 17:42). A POSITA reading Kang would also understand this Kang description as merely one example and there are a number of scenarios where a cursor would be placed in a region of overlap relative to the underlying determined handwriting.

123. Kang further explains that front/end separation may be determined using 50% of the width length of the specific stroke or rearrangement stroke, while also stating that the cursor position may be determined in another alternative method. A POSITA would understand the 50% width discussion as one optional geometry-based way of resolving cursor placement within that same contested area, not as the only disclosed method. In other words, the overlap problem is created by the same axis crossing multiple candidate handwritten units, and the 50% rule is one way Kang resolves that cursor placement problem. (Kang, Description, [0116]).

124. Paragraph [0118] confirms that these are real editing positions with real consequences. If the cursor is determined immediately behind the 24th rearrangement stroke, Kang deletes the digital ink data corresponding to that stroke. If the cursor is determined immediately behind the 25th rearrangement stroke, Kang deletes the data associated with that selected position, including in the disclosed grouped-stroke-block example. Paragraphs [0117] and [0134] provide additional context by explaining that overlapping stroke-derived units may be processed as one stroke block, but the strongest support for this limitation comes from the single virtual axis crossing multiple candidate items of digital ink data and the direct examples of cursor placement behind particular rearrangement strokes. (Kang, Description, [0117]-[0118], [0134]).

ii. ‘316 Dependent Claims 2, 5, and 8

125. Claims 2, 5, and 8 add the requirement that the elements of the digital ink include text characters and that the cursor is displayed at a position between first and second characters in relation to their determined ranges. Kang discloses this. Paragraph [0090] explains that Chinese and Korean characters, and letters like “i” and “j,” include two or more strokes and should be processed as a stroke block so the cursor is not placed between strokes composing one character. Paragraph [0117] explains that grouped rearrangement strokes may generate Korean character

and Korean syllable units. A POSITA would therefore understand Kang's operative handwritten units to include text characters, not just raw stroke traces. (Kang, Description, [0090], [0117]).

126. Kang also discloses cursor placement between adjacent character-forming units. As discussed above, paragraphs [0115]-[0116] explain that when the virtual axis crosses multiple adjacent items of digital ink data, including the digital ink data corresponding to the 24th and 25th rearrangement strokes, the system resolves the cursor to land immediately behind one unit or the other. The front/end separation may be determined using 50% of the width length of the specific stroke or rearrangement stroke, and Kang's stroke-block disclosure ensures that the cursor is placed between character units rather than improperly splitting a single multi-stroke character. In practical terms, Kang is doing exactly what these dependent claims require: it is using the geometry of adjacent handwritten character units to place the cursor between them in a natural editing position. (Kang, Description, [0090], [0115]-[0118]).

iii. '316 Dependent Claims 3, 6, and 9

127. Claims 3, 6, and 9 add the requirement that the geometrical features include height and width of the determined ranges. Kang expressly discloses both. Width is disclosed through the x-axis range logic. Paragraph [0054] defines a stroke

segment by the maximum and minimum values on the x-axis. Paragraphs [0056]-[0057] define overlap length and spacing distance in terms of projected stroke segments or stroke block segments. Paragraph [0116] explains that front/end separation may be determined using 50% of the width length of the specific stroke or rearrangement stroke. A POSITA would understand width to be part of the geometry Kang uses to define and resolve the determined handwritten ranges. (Kang, Description, [0054], [0056]-[0057], [0116], [0134]-[0135]).

128. Height is likewise disclosed. Paragraph [0070] explains that the height length 36 of input panel 152 and the space 37 of virtual line 156 may be matched so that the size of digital ink data may be reduced for display along the virtual line within that space. Paragraph [0115] also explains that the virtual axis used for cursor selection is formed in the height direction of the screen. Stated more simply, Kang uses horizontal measurements to define how wide the handwritten units are and vertical measurements to determine how those units are scaled, displayed, and intersected on the screen. Those are the claimed geometrical features of the determined ranges. Claims 3, 6, and 9 are therefore also anticipated. (Kang, Description, [0070], [0093]-[0094], [0115], [0134]-[0135]).

B. The Clary Prior Art Reference (US 8,115,748 B2)

129. In my opinion from the perspective of a POSITA, the '316 Patent

claims are also anticipated by Clary. As detailed below, a POSITA would have understood that Clary expressly discloses every limitation of the challenged claims, either in the same pen-enabled computing-device system described in the specification or in the corresponding computer-readable-medium form expressly recited in Clary's own claims. (Clary, Background/Summary, 1:42, 5:12-5:28, 6:53-7:12; Clary, Description, 11:39-13:44, 17:45-19:62; Clary, Claim 16).

130. Clary is particularly relevant because it is not merely a handwriting-capture reference. It is directed to capturing handwritten input, representing that input as coordinate-pair-based electronic ink, defining bounding boxes and baselines for the handwritten content, displaying a cursor relative to that handwritten content, and editing the content by insertion, deletion, replacement, and spacing adjustments. Those are the same basic technical themes that drive the asserted '316 Patent claims. (Clary, Background/Summary, 1:42, 5:12, 6:53-7:12, 9:4; Clary, Description, 17:45-19:62).

131. At a practical level, Clary is solving the same basic problem as the '316 Patent: once handwriting is displayed on a screen, the system must organize that handwriting into editable units and then place an editing cursor in the correct location even where handwritten units crowd each other. Clary addresses that problem by defining bounding boxes around strokes or bundled stroke units,

evaluating the cursor relative to those boxes, and then shifting the cursor to a box boundary when the cursor lies within one or more bounding boxes. (Clary, Description, 17:45-19:28).

i. ‘316 Independent Claims 1, 4, and 7

132. Clary discloses the same core functionality in system and computer-readable-medium form. FIG. 1 and FIG. 2 disclose a pen-enabled computing device architecture including a central processing subsystem, user interface, handwriting capture interface, and computer program product. Clary’s claim 16 separately recites a tangible computer readable medium configured to cause a computing device to detect handwriting input, define at least one bounding box, and store the handwritten input and the bounding box. For that reason, Claim 1 of the ‘316 Patent is disclosed by Clary’s described system architecture and editing functionality, Claim 4 of the ‘316 Patent is disclosed by Clary’s described editing method implemented by Clary’s described system, and Claim 7 of the ‘316 Patent is disclosed by Clary’s described tangible-computer-readable-medium used to implement Clary’s described system. (Clary, Description, 11:39-13:44, 17:45-19:62; Clary, Claim 16).

(1) ‘316 Preamble

- a. [1pre] "A system for providing editing of digital ink on computing devices, each computing device comprising a processor and at least one non-transitory computer readable medium for recognizing handwriting input under control of the processor, the at least one non-transitory computer readable medium configured to"
- b. [4pre] A method for providing editing of digital ink on computing devices, each computing device comprising a processor and at least one non-transitory computer readable medium for recognizing handwriting input under control of the processor, the method comprising:
- c. [7pre] A non-transitory computer readable medium having a computer readable program code embodied therein, said computer readable program code adapted to be executed to implement a method for providing editing of digital ink on a computing device, the computing device comprising a processor and at least one system non-transitory computer readable medium for recognizing handwriting input under control of the processor, the method comprising:

133. Clary discloses the preamble's system for providing editing of digital ink on computing devices. Paragraph 11:39 explains that device 100 generally comprises handwriting capture interface 200, central processing subsystem 300, user interface 400, computer program product 500, and external device interface 600. Paragraph 13:44 explains that one advantageous embodiment uses a handheld computer component 110 such as a modified PDA or smart phone. In other words, Clary is not describing an isolated pen tablet. It is describing a computing device system with the hardware and software needed to capture, display, and manipulate

handwritten input. (Clary, Description, 11:39, 13:44).

134. Clary also discloses the processor and medium aspects of the preamble. Paragraph 12:6 explains that the central processing subsystem 300 may comprise central processor unit 320, non-volatile memory storage 340, and RAM storage 360, and that the CPU executes the processing activities required to support the device. Paragraph 12:6 also explains that non-volatile memory stores and/or executes computer program product 500 and stores useful data such as electronic ink. Claim 16 then expressly recites a tangible computer readable medium configured to store instructions that cause a computing device to define bounding boxes and store handwritten input and the boxes. (Clary, Description, 12:6; Clary, Claim 16).

135. Clary further discloses the editing aspect of the preamble. Paragraph 18:40 explains that Clary provides a system for converting handwriting into a machine-usable format, storing handwritten input and the bounding box surrounding each stroke, and editing the handwritten element to add textual data, insert textual data between adjacent bounding boxes, delete at least one bounding box, and/or replace a bounding box with alternate data. Paragraphs 17:45 and 19:43-19:62 explain that the disclosed technique provides seamless editing of handwritten and/or textual data, including deleting existing handwritten input and moving the remaining handwritten input to close gaps. A POSITA would understand this to

disclose a system for providing editing of digital ink on a computing device. (Clary, Description, 17:45, 18:40, 19:43-19:62).

136. Finally, Clary discloses the recognition aspect of the preamble. Paragraph 1:42 explains that handwriting input may be stored as an electronic-ink copy or as a text copy in which the handwriting input is captured, recognized, and translated into corresponding text, and that in some instances the system may produce both an electronic-ink copy and a text copy of the same handwriting input. A POSITA would therefore understand Clary to disclose a processor-controlled system that can both capture handwritten input as digital ink and recognize that same input into text. (Clary, Background/Summary, 1:42).

(2) ‘316 Claim 1/4/7 Element 1

- a. [1a] cause display of, on a display interface of a computing device, digital ink in accordance with the handwriting input;
- b. [4a] displaying, on a display interface of a computing device, digital ink in accordance with the handwriting input;
- c. [7a] displaying, on a display interface of a computing device, digital ink in accordance with the handwriting input;

137. Clary expressly discloses this limitation. Paragraph 1:42 explains that the handwriting input may be stored as an electronic-ink copy, typically as a sequence of (x, y) coordinate pairs. Paragraph 17:45 explains that, in the embodiment of FIGS. 8 and 9, handwritten input 899 is initially captured by the

handwriting capture interface, which provides a sequence of coordinate pairs 900 that define marks made by the writing stylus. Paragraph 18:40 then explains that handwritten input 899 can be edited by initially displaying the handwritten input 899 to the user upon display 400 associated with the pen-enabled computing device. In plain terms, Clary teaches that the user writes, the system captures that writing as electronic ink / coordinate pairs, and the handwritten input is displayed on the device for editing. That is the claimed display step. (Clary, Background/Summary, 1:42; Clary, Description, 17:45, 18:40).

(3) ‘316 Claim 1/4/7 Element 2

- a. [1b] determine, from segmentation information of the handwriting input determined through the recognition of the handwriting input, ranges of elements of the digital ink
- b. [4b] determining, from segmentation information of the handwriting input determined through the recognition of the handwriting input, ranges of elements of the digital ink;
- c. [7b] determining, from segmentation information of the handwriting input determined through the recognition of the handwriting input, ranges of elements of the digital ink;

138. Clary begins with handwritten input captured as coordinate pairs. It then spatially segments that handwritten data and defines bounding boxes around strokes. Paragraph 6:53 explains that the computer program product may detect, capture, and store handwritten data by spatially segmenting the handwritten data

and correlating the spatial segments with the nature of the page. Paragraph 17:58 then explains that the central processing unit and computer program product define bounding boxes 902 around each stroke 904, with x-extents determined by the smallest x value 900a and largest x value 900b and y-extents running between successive baselines. Paragraph 17:58 also explains that a bounding box can be defined differently in other applications so as to bundle one or more strokes together as a unit 910 for editing purposes. (Clary, Background/Summary, 6:53; Clary, Description, 17:58).

139. The Clary disclosure supplies the claimed ‘316 “elements,” “ranges,” and “segmentation information.” Under the proper construction of the claims, “segmentation information” encompasses the spatial grouping of strokes generated by a recognition process to separate handwriting into editable units. Clary discloses exactly this. The elements are Clary’s operative handwritten editing units: individual strokes or spatially “bundled” units formed from one or more strokes. The ranges are Clary’s bounding boxes that define the physical left, right, top, and bottom boundaries for those units. (Clary, Description, 17:58-18:11).

140. Clary also discloses the recognition-oriented aspect of this limitation under either proposed construction of "recognizing handwriting input." Considering Patent Owner's construction, Clary captures and recognizes the creation of digital

content inputted by the user via a stylus. Considering Goodnotes' construction, Clary detects this handwriting input and derives segmented units by defining bounding boxes to bundle strokes into machine-usable editing units for subsequent display and editing operations. A POSITA would understand that Clary's spatial segmentation and bounding-box bundling is the exact geometric stroke-grouping required by the '316 Patent's recognition process, fully anticipating the determination of ranges from segmentation information under either proposed construction. (Clary, Background/Summary, 1:42, 6:53; Clary, Description, 17:58).

141. For the same reasons, Clary discloses the claimed "determining" step. Clary is not merely storing raw coordinates. It is determining bounding boxes around handwritten units, determining baselines for lines of handwritten input, and using those structures as the operative editing ranges for insertion, deletion, replacement, and cursor placement. That is the determination of ranges from segmentation information. (Clary, Description, 17:58-19:28).

(4) '316 Claim 1/4/7 Element 3

- a. [1c] cause display of, on the display interface, an input cursor in relation to the determined digital ink element ranges; and
- b. [4c] displaying, on the display interface, an editing cursor in relation to the determined digital ink element ranges; and

- c. [7c] displaying, on the display interface, an editing cursor in relation to the determined digital ink element ranges; and

142. Clary expressly discloses a cursor displayed in relation to determined handwritten ranges. Paragraph 18:40 explains that the handwritten input 899 is initially displayed to the user upon display 400, and that the user then selects an actuation position 910 relative to the handwritten input 899, such as by positioning cursor 912 relative to the display of the handwritten input 899. Paragraph 18:40 further explains that, depending on the location of the cursor relative to the bounding boxes 902 surrounding each stroke 904, the pen-enabled computing device may shift the cursor position somewhat. In plain terms, Clary does not place the cursor arbitrarily. It displays the cursor relative to the bounding-box ranges surrounding the handwritten input and then adjusts the cursor based on those ranges. That is a cursor displayed in relation to the determined digital-ink element ranges. (Clary, Description, 18:40).

(5) '316 Claim 1/4/7 Element 4

- a. [1d] cause said display of the input cursor within a region of overlap of ranges of elements of the digital ink determined in the step of determining ranges of elements of the digital ink.
- b. [4d] causing said display of the editing cursor within a region of overlap of ranges of elements of the digital ink determined in the step of determining ranges of elements of the digital ink.

- c. [7d] causing said display of the editing cursor within a region of overlap of ranges of elements of the digital ink determined in the step of determining ranges of elements of the digital ink.

143. This is one of the strongest Clary disclosures. The claims require causing display of the cursor within a “region of overlap” of the determined ranges. Clary explicitly anticipates this by describing how its system resolves edit locations when handwritten spatial ranges crowd or overlap one another. Paragraph 18:40 explains that the central processing unit determines whether the cursor is located within a bounding box. It then states that the system applies specific geometric rules “if the cursor 912 position is located within one or more bounding boxes 902.” A POSITA would understand that if a single (cursor coordinate is simultaneously located within multiple distinct bounding boxes, those spatial ranges are physically overlapping. This is an express disclosure of a cursor positioned within a region of overlap. (Clary, Description, 18:40).

144. In practical terms, Clary is describing the exact geometric conflict-resolution the '316 Patent claims. When the cursor falls within this overlapping region, Clary evaluates the geometry of those competing ranges to snap the cursor to the preferred location. Clary determines the x-coordinate of the center of the largest bounding box. If the cursor is to the right of that center, the device shifts the

cursor to the right side of the largest bounding box. If the cursor is to the left of that center, the device shifts the cursor to the left side of the smallest bounding box. Clary thus uses the geometric features of the overlapping ranges (centers and widths) to resolve cursor placement, anticipating the '316 Patent completely. (Clary, Description, 18:40).

145. A POSITA would understand that Clary's edge-snapping logic guarantees the cursor is displayed within an area where the determined ranges of elements overlap (consistent with Patent Owner's construction) and where adjacent segmented elements overlap (consistent with Goodnotes' construction). Because Clary's bounding boxes are physically occupying the same space, snapping the cursor to the boundary of one box (e.g., the 'left side of the smallest box') geometrically places the cursor at a coordinate that remains physically inside the overlapping portion of the larger box. This edge-snapping logic is functionally and mathematically identical to the '316 Patent's own disclosure of snapping a cursor to a range border within an overlapping area. (Compare Clary, Description, 18:40, with '316 Patent, Fig. 18B and Description, 17:42). Therefore, Clary does not teach away from the overlap; its algorithm expressly places the cursor within it.

146. Paragraphs 19:5 and 19:28 reinforce that the box geometry is the operative editing framework. Clary explains that inserted text or handwritten

insertions are placed relative to adjacent bounding boxes and that boxes to the right of the insert are moved by an amount equal to the width of the insert. Paragraphs 19:43-19:62 likewise explain deletion and gap-closing by moving the remaining handwritten input and the corresponding bounding boxes. Those disclosures confirm that the bounding boxes are not merely descriptive shapes; they are the operative ranges used to decide where insertion, deletion, and cursor placement occur. (Clary, Description, 19:5-19:62).

ii. ‘316 Dependent Claims 2, 5, and 8

147. Claims 2, 5, and 8 add the requirement that the elements of the digital ink include text characters and that the cursor is displayed at a position between first and second characters in relation to their determined ranges. Clary discloses this. Paragraph 18:11 explains that the central processing unit determines the baseline for each line so as to provide a point of reference for the characters in the respective line. Paragraphs 17:45, 18:40, and 19:5 explain that Clary inserts textual input between adjacent bounding boxes, and that cursor 912 is positioned and shifted relative to those bounding boxes. A POSITA would therefore understand the adjacent boxed handwritten units in a line to correspond to adjacent character-level handwritten units whose ranges are defined by the neighboring bounding boxes. (Clary, Description, 17:45, 18:11-19:5).

148. In practical terms, Clary is using the neighboring bounding boxes as the editing units on the handwritten line. When the cursor is evaluated relative to those adjacent boxes and the insertion position is defined between adjacent bounding boxes, the cursor is being positioned between first and second character units in relation to their determined ranges. For those reasons, Claims 2, 5, and 8 are also anticipated. (Clary, Description, 18:11-19:5; See Fig. 8 & 9).

iii. '316 Dependent Claims 3, 6, and 9

149. Claims 3, 6, and 9 add the requirement that the geometrical features include height and width of the determined ranges. Clary expressly discloses both. Width is disclosed through the x-axis definition of the bounding boxes. Paragraph 17:58 explains that each bounding box uses the smallest x value 900a and largest x value 900b of the associated coordinate pairs. Paragraphs 18:40 and 19:28 further use that width geometry by referring to the center of the largest bounding box and by defining the width of a handwritten insertion from the leftmost and rightmost inserted boxes. A POSITA would understand width to be part of the geometry Clary uses to define and operate on the handwritten ranges. (Clary, Description, 17:58, 18:40, 19:28).

150. Height is likewise disclosed. Paragraph 17:58 explains that the y coordinates of each bounding box extend between successive baselines B1, B2, and

Bn. Paragraph 18:11 further explains that the baselines provide a point of reference for the characters in the handwritten line. Stated more simply, Clary uses horizontal measurements to define how wide the handwritten units are and vertical baseline structure to define how tall they are and where they sit on the line. Those are the claimed geometrical features of the determined ranges. Claims 3, 6, and 9 are therefore also anticipated. (Clary, Description, 17:58-18:11, 18:40, 19:28).

XIII. OBVIOUSNESS ANALYSIS

A. Alternatively, the '316 Patent Claims Are Rendered Obvious in Light of the Li Publication in View of or Combination with Clary.

151. In my opinion from the perspective of a POSITA, the challenged claims of the '316 Patent would have been obvious over Li in view of Clary. Li is the proper base reference because it already discloses a handwriting-editing apparatus that recognizes handwritten characters and graphics, groups the recognized handwritten objects, determines circumscribing rectangles and geometric centers for those objects, defines a locus through those centers, and then performs deletion, correction, and insertion while moving adjacent handwritten objects so the edited result remains visually coherent. Those teachings supply the core part of the '316 theory: recognition-derived segmentation into bounded handwritten elements and geometry-based editing of those elements. (Li, Abstract, 1; Li,

Background/Summary, [0012]-[0039]; Li, Description, [0070], [0075]-[0089], [0092]-[0094], [0111]-[0128], [0135]-[0141]; Li, Claim 6).

152. Clary is not being used as a substitute recognition engine. It is being used to supply the cursor-side implementation that Li leaves less explicit. Clary explains that displayed handwritten input may be edited by positioning cursor 912 relative to bounding boxes 902, that the system checks whether the cursor lies within one or more bounding boxes, and that the cursor is then shifted to the relevant box boundary by reference to box geometry. Clary also confirms that contemporaneous pen-enabled systems used coordinate-pair electronic ink, bounded handwritten units, insertion between adjacent bounded units, deletion, replacement, and shifting of neighboring handwritten content to preserve spacing. (Clary, Background/Summary, 1:42, 5:12, 6:53-6:62, 9:25-10:22; Clary, Description, 17:58-19:62).

153. At a practical level, the combination is straightforward. Li already teaches how to recognize handwritten units, assign each such unit a circumscribing rectangle, and place edited content relative to adjacent recognized objects along a locus. Clary teaches how to present that edit position to the user as an explicit displayed cursor and how to resolve that cursor position when it falls within one or more bounded handwritten regions. A POSITA would have understood that

applying Clary's cursor technique to Li's recognized-object editing framework would make Li's edit position more visible, more precise, and more intuitive without changing Li's underlying recognition or geometry. (Li, Description, [0085]-[0089], [0111]-[0128]; Clary, Description, 18:40).

i. POSITA's Motivation to Combine

154. A POSITA would have had reason to combine Li and Clary at the time of the alleged invention. Li already teaches the central editing environment: recognized handwritten objects, circumscribing rectangles around those objects, geometric centers, loci connecting those centers, deletion and correction that preserve a predetermined overlap relationship, and insertion that positions a new character object between adjacent recognized objects. In other words, Li already gives the skilled artisan a recognition-based handwritten-editing system that depends on bounded handwritten geometry. (Li, Background/Summary, [0022]-[0037]; Li, Description, [0085]-[0089], [0111]-[0128]).

155. Clary then teaches a known and predictable way to make that kind of bounded handwritten-editing system easier to use and edit. Clary explains that the user positions cursor 912 relative to displayed handwritten input, that the system determines whether the cursor is within one or more bounding boxes, and that the cursor is shifted by reference to the largest and smallest relevant boxes. Clary also

teaches insertion, deletion, and replacement of handwritten content while moving adjacent handwritten content to maintain spacing. A POSITA would have understood that Clary's explicit cursor-and-bounding-box logic could be applied directly to Li's already-recognized circumscribing rectangles to visually indicate and resolve the edit location during the same kinds of insertion, correction, and deletion operations. (Clary, Background/Summary, 5:12, 6:62, 9:25-10:22; Clary, Description, 18:40-19:62).

156. Because Li relies on handwritten gestures to dictate the edit location, a POSITA also would have recognized the inherent imprecision of gesture-based editing when the intended edit point falls between tightly spaced or crowded handwritten characters. Clary provides a known and predictable solution to that UI problem: an explicit displayed cursor that evaluates its position relative to handwritten bounding boxes and snaps to a definitive left or right boundary when placed within a crowded region, including when the cursor lies within one or more bounding boxes. A POSITA would have understood that Clary's cursor-resolution logic could be applied directly to Li's recognized-object rectangles to provide precise, familiar visual feedback before the insertion, correction, or deletion is executed. (Li, Description, [0115], [0123]-[0128]; Clary, Description, 18:40).

157. Accordingly, a POSITA would have been motivated to combine the

references in a narrow and predictable way: keep Li's recognition-derived handwritten-object framework and apply Clary's explicit cursor technique to that framework. That combination would have improved usability without altering Li's basic operation, because both references operate on displayed handwritten content, both rely on bounded geometric regions around handwritten units, and both are directed to editing handwritten data on a computing device. It would have been the predictable application of a known UI mechanism to Li's already-disclosed recognized-object data structure. (Li, Abstract, 1; Li, Description, [0111]-[0128]; Clary, Background/Summary, 1:42, 5:12, 6:53-6:62, 9:25-10:22).

ii. '316 Independent Claims 1, 4, and 7

158. In my opinion, the combination of the Li Publication with Clary discloses every limitation of claims 1, 4, and 7. Claims 1, 4, and 7 of the '316 Patent are disclosed by this combination. Li discloses the same underlying handwriting-recognition and editing functionality as an input apparatus and also as a computer readable recording medium recorded with an image processing program for operating a computer as that apparatus. Clary, in turn, confirms that such handwriting-editing techniques were implemented on pen-enabled computing devices with processor, memory, and computer program product support. For that reason, the same obviousness analysis applies across the system, method, and

computer-readable-medium forms, with only the formal claim language changing. (Li, Background/Summary, [0013]-[0019], [0038]-[0039]; Li, Claim 6; Clary, Description, 11:39-12:6, 18:40).

(1) '316 Claim 1/4/7 Preamble

- a. [1pre] "A system for providing editing of digital ink on computing devices, each computing device comprising a processor and at least one non-transitory computer readable medium for recognizing handwriting input under control of the processor, the at least one non-transitory computer readable medium configured to"
- b. [4pre] A method for providing editing of digital ink on computing devices, each computing device comprising a processor and at least one non-transitory computer readable medium for recognizing handwriting input under control of the processor, the method comprising:
- c. [7pre] A non-transitory computer readable medium having a computer readable program code embodied therein, said computer readable program code adapted to be executed to implement a method for providing editing of digital ink on a computing device, the computing device comprising a processor and at least one system non-transitory computer readable medium for recognizing handwriting input under control of the processor, the method comprising:

159. Li already discloses the core system recited in the preamble. Li's input apparatus includes a handwritten character and/or graphic input device, a storage unit for storing handwritten data, an input direction detecting unit, an edit management unit, and a display unit for displaying the handwritten data before and

after execution of the edit command. Li also explains that a handwriting recognition dictionary is stored in the first database and that the handwritten object extracting module recognizes handwritten characters and/or graphics by comparing inputted handwritten objects against stored patterns. In other words, Li already discloses a processor-controlled handwriting-editing system that recognizes handwritten input and edits the resulting handwritten data. (Li, Background/Summary, [0013]-[0019]; Li, Description, [0075]-[0085]). Li also supplies strong support for the medium form of the claims. Li expressly states that the invention provides a computer readable recording medium recorded with an image processing program for operating a computer as the disclosed input apparatus, and claim 6 recites that same computer readable recording medium. Clary reinforces the same general computing architecture by disclosing a pen-enabled computing device with CPU, memory, and computer program product. A POSITA therefore would have understood the preamble to be disclosed or at least rendered obvious by Li's base editing system, with Clary confirming the ordinary processor-memory-program-product implementation of that sort of handwriting-editing system. (Li, Background/Summary, [0038]-[0039]; Li, Claim 6; Clary, Description, 11:39-12:6, 18:40).

(2) '316 Claim 1/4/7 Element 1

- a. [1a] cause display of, on a display interface of a computing device, digital ink in accordance with the handwriting input;
- b. [4a] displaying, on a display interface of a computing device, digital ink in accordance with the handwriting input;
- c. [7a] displaying, on a display interface of a computing device, digital ink in accordance with the handwriting input;

160. Li expressly teaches display of handwritten data in accordance with handwriting input. Li's display unit displays handwritten data that has not yet been edited and handwritten data that has been edited after execution of the edit command. Li also explains that the input apparatus is used not only to input information but also to edit previously input information and that the handwritten object is displayed as part of the editing process. In practical terms, Li teaches that the user writes by hand, the system captures that handwritten information, and the handwritten data is displayed before and after editing. (Li, Background/Summary, [0014], [0019]-[0021]; Li, Description, [0073], [0078], [0092]-[0094]).

161. Clary reinforces this same mapping by expressly describing handwriting input captured as an "electronic ink" copy and then displayed for editing on the pen-enabled computing device. The combination therefore readily teaches display of digital ink in accordance with handwriting input, with Li providing the operative displayed-handwriting editing system and Clary confirming that such displayed handwritten data was understood in the art as electronic ink.

(Clary, Background/Summary, 1:42; Clary, Description, 17:45, 18:40).

(3) ‘316 Claim 1/4/7 Element 2

- a. [1b] determine, from segmentation information of the handwriting input determined through the recognition of the handwriting input, ranges of elements of the digital ink
- b. [4b] determining, from segmentation information of the handwriting input determined through the recognition of the handwriting input, ranges of elements of the digital ink;
- c. [7b] determining, from segmentation information of the handwriting input determined through the recognition of the handwriting input, ranges of elements of the digital ink;

162. This is the limitation for which Li matters most for the combination.

The claims require determining spatial ranges of elements from segmentation information derived through the recognition process. Li discloses exactly this. Li explains that the handwritten object extracting module uses a handwriting recognition dictionary to recognize the character and/or graphic. Once recognized, the handwriting locus defining module completes the edit process by spatially segmenting the input: it detects a minimum “circumscribing rectangle” (the spatial range) and a geometric center for each recognized handwritten object in the group. That is a recognition-driven segmentation process. The system is not merely storing raw strokes; it recognizes the handwritten objects, organizes them into a group, and derives a geometric bounding range for each element based on that recognition. A POSITA would understand that Li’s derivation of circumscribing rectangles

constitutes the exact segmentation information and spatial ranges required by the claims. (Li, Description, [0085]-[0089], [0111]-[0112]).

163. Li then uses those recognized-object ranges as the operative geometry for later editing. FIGS. 11A-11C and the accompanying text explain that the handwriting group 101 is composed of characters that are handwritten objects, that a circumscribing rectangle 102 is detected for each handwritten object, and that the geometric centers of those rectangles are connected to define locus 103. A POSITA would understand those circumscribing rectangles to be the claimed ranges of the handwritten elements. Clary's bounding-box disclosure adds support that contemporaneous handwritten-ink systems likewise used bounded geometric regions around handwritten units, but Li is the primary reference because Li itself supplies the recognition step, the handwritten objects, and the circumscribing-rectangle ranges. (Li, Description, [0111]-[0112]; Clary, Background/Summary, 6:53, 9:41).

(4) '316 Claim 1/4/7 Element 3

- a. [1c] cause display of, on the display interface, an input cursor in relation to the determined digital ink element ranges; and
- b. [4c] displaying, on the display interface, an editing cursor in relation to the determined digital ink element ranges; and

- c. [7c] displaying, on the display interface, an editing cursor in relation to the determined digital ink element ranges; and

164. Li already discloses the edit position in geometric terms even though it does not dwell on a visual cursor. For deletion and correction, Li identifies the object to be acted on by detecting the circumscribing rectangle that includes the coordinate or locus where the gesture is inputted. For insertion, Li determines the width of the object to be inserted, determines the spacing needed between adjacent character objects, and then places the inserted character object so that its center lies on a locus connecting the centers of the adjacent character objects and at a middle position between them. In practical terms, Li is already determining an edit location by reference to the recognized-object ranges. (Li, Description, [0115], [0119]-[0120], [0123]-[0128]).

165. What Clary contributes is the explicit displayed-cursor implementation of that same range-based edit position. Clary explains that displayed handwritten input may be edited by positioning cursor 912 relative to the handwritten input and that, depending on the cursor's location relative to bounding boxes 902, the device may shift the cursor position. A POSITA would have understood it would have been routine to implement Li's already-disclosed range-based insert/delete/correct position as the kind of displayed cursor taught by Clary. Put differently, Li supplies

the recognized-object geometry, and Clary supplies the familiar UI mechanism for showing that edit position to the user via a cursor. (Clary, Description, 18:40).

(5) ‘316 Claim 1/4/7 Element 4

- a. [1d] cause said display of the input cursor within a region of overlap of ranges of elements of the digital ink determined in the step of determining ranges of elements of the digital ink.
- b. [4d] causing said display of the editing cursor within a region of overlap of ranges of elements of the digital ink determined in the step of determining ranges of elements of the digital ink.
- c. [7d] causing said display of the editing cursor within a region of overlap of ranges of elements of the digital ink determined in the step of determining ranges of elements of the digital ink.

166. The combination teaches causing display of the cursor within a region of overlap of the determined ranges. Li supplies the underlying recognized-object geometry for this limitation. In the deletion example, Li explains that the remaining recognized character object is moved along locus 103 until there will be a “predetermined overlap region” between the circumscribing rectangle of the moving character object and the circumscribing rectangle of another character object toward which it is heading. In the correction example, Li again explains that the corrected character object is placed so that there will be a “predetermined overlap region” between the corrected object’s circumscribing rectangle and the

circumscribing rectangle of another adjacent character object. Those passages matter because they show that Li's module configuration already performs editing in a bounded handwritten environment defined by adjacent spatial ranges, including ranges that crowd and overlap. (Li, Description, [0116], [0120]).

167. Li also teaches why a precise edit position must be resolved within that same geometric framework. In insertion, Li explains that adjacent character objects are moved to create a predetermined distance, that the width of the inserted character object's circumscribing rectangle is used to determine insert distance, and that the inserted character object is positioned on a locus connecting the centers of the adjacent character objects and at a middle position between them. In practical terms, Li is already computing an exact between-object edit location by reference to the geometric features of neighboring recognized character ranges. That is the exact same edit-location problem for which the '316 Patent uses an input cursor. (Li, Description, [0123]-[0128]).

168. What Clary contributes to the combination is the explicit displayed-cursor implementation for that overlapping-range geometry. Clary explains that when cursor 912 is located within "one or more bounding boxes 902," the system determines the center of the largest bounding box and then shifts the cursor either to the right side of the largest box or to the left of the smallest box. A POSITA

would have understood that if a single cursor coordinate is simultaneously located within multiple distinct bounding boxes, those spatial ranges are physically occupying the same space. Clary thus provides a concrete and predictable way to display and resolve Li's already-existing edit position when the relevant recognized-object ranges overlap or crowd each other. Put differently, Li's processing supplies the recognized objects, the circumscribing rectangles, and the overlap-aware adjacent-object geometry; Clary supplies the visible cursor behavior for choosing which side of that crowded geometric boundary the edit position belongs on. (Clary, Description, 18:40).

169. Furthermore, applying Clary's edge-snapping algorithm to Li's purposefully overlapping handwriting inevitably results in the cursor being displayed 'within a region of overlap.' Li explicitly teaches that adjacent character objects are moved until there is a 'predetermined overlap region' between their circumscribing rectangles. (Li, Description, [0116], [0120]). If a POSITA applies Clary's algorithm to resolve a cursor selection in that predetermined overlap region, shifting the cursor to the boundary of one rectangle places the cursor inside the adjacent, overlapping rectangle. A POSITA would recognize this geometric inevitability as identical to the '316 Patent's own proposed solution for overlapping characters. (See '316 Patent, Fig. 18B and Description, 17:42).

170. Viewed together, the references teach the claimed cursor placement within a region of overlap of ranges of elements of the digital ink. Li provides the recognition-derived rectangles and adjacent-object geometry, including the predetermined overlap regions and between-object placement context. Clary provides the explicit displayed-cursor technique for resolving a cursor position when it lies within one or more competing handwritten ranges. A POSITA implementing Li's rectangle-and-locus editing system would have had reason to use Clary's cursor logic precisely because Li's edit process already places the edit operation in a bounded handwritten environment where spatial overlap and crowding are expected, and because Clary makes that edit location visible and unambiguous before the edit is carried out. (Li, Description, [0116], [0120], [0123]-[0128]; Clary, Description, 18:40).

iii. '316 Dependent Claims 2, 5, and 8

171. In my opinion, the combination of the Li Publication with Clary discloses every limitation of claims 2, 5, and 8. Claims 2, 5, and 8 add the requirement that the elements of the digital ink include text characters and that the cursor is displayed at a position between first and second characters in relation to their determined ranges. Li discloses this character-based geometry directly. Li repeatedly refers to handwritten characters as character objects, explains that

insertion occurs between adjacent character objects, and explains that the inserted character object is placed at the middle position between the centers of the adjacent character objects. Li further explains that when multiple inserted characters would otherwise reduce visual quality, the plural characters may be split one by one into respective character objects and inserted at one insert position. In practical terms, Li is operating on text-character units and positioning new character content between neighboring character units. (Li, Description, [0123]-[0128], [0135]-[0136]).

172. Clary again supplies the explicit displayed-cursor implementation for that between-character position by disclosing a cursor evaluated relative to adjacent bounding boxes on a handwritten line and, where necessary, resolved when it falls within one or more such boxes. A POSITA would therefore have understood that the combination teaches a cursor displayed between first and second character units in relation to the determined ranges of those units, with Li providing the character-object geometry and Clary providing the displayed-cursor expression of that geometry when adjacent handwritten character ranges crowd or overlap. (Clary, Description, 18:11-19:5).

iv. ‘316 Dependent Claims 3, 6, and 9

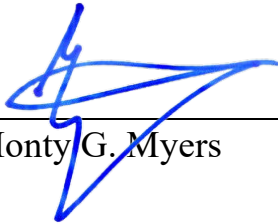
173. In my opinion, the combination of the Li Publication with Clary

discloses every limitation of claims 3, 6, and 9. Claims 3, 6, and 9 add the requirement that the geometrical features include height and width of the determined ranges. Li expressly discloses both. Width is disclosed when Li determines the width r_1 of the rectangle circumscribing the character object to be inserted, determines the necessary spacing r_2 between adjacent character objects, and uses those values to calculate insert distance r_3 . Height is disclosed in Li's treatment of alphabetic characters, where Li explains that alphabetic characters have very different vertical sizes and that certain written parts are excluded so the circumscribing rectangle tracks the principal part of the word more accurately. Li then detects a circumscribing rectangle for each principal part and a geometric center for that rectangle. Those teachings are explicit height-and-width geometry for the determined ranges. (Li, Description, [0127], [0139]-[0141]).

174. Clary corroborates the same sort of handwritten-range geometry by defining bounding boxes through smallest-x and largest-x values and by extending the boxes in the y direction between successive baselines. The combination therefore readily teaches that the relevant geometrical features include the height and width of the determined handwritten ranges. Li is again the primary source, and Clary confirms that the use of x-extent and y-extent box geometry in handwritten editing was well known. (Clary, Description, 17:58-18:11).

XIV. RESERVATION OF RIGHT TO UPDATE AND AMEND

175. All of the findings and opinions expressed above in this declaration are based upon my review, analysis, and interpretation of the information available to me at this time. I reserve the right to amend and supplement my findings and opinions in the future if and when new, additional, or revised information becomes available.



Monty G. Myers