Jill F. Kopeikin (State Bar No. 160792)
Valerie M. Wagner (State Bar No. 173146)
GCA LAW PARTNERS LLP
2570 W. El Camino Real, Suite 400
Mountain View, CA 94040
Telephone: (650) 428-3900
Fax: (650) 428-3901
jkopeikin@gcalaw.com
vwagner@gcalaw.com

(Additional Counsel Identified On Signature Page)
Attorneys for Plaintiff CYWEE GROUP LTD.

IN THE UNITED STATES DISTRICT COURT
FOR THE SOUTHERN DISTRICT OF CALIFORNIA

CYWEE GROUP LTD.,
Plaintiff,

LG ELECTRONICS, INC.,
LG ELECTRONICS U.S.A., INC.,
AND LG ELECTRONICS MOBILECOMM U.S.A., INC.,
Defendants.

CASE NO. '17CV1102 GPC RBB

CYWEE'S ORIGINAL COMPLAINT FOR PATENT INFRINGEMENT

DEMAND FOR JURY TRIAL

1. Plaintiff CyWee Group Ltd. ("Plaintiff" or "CyWee") by and through its undersigned counsel, files this Original Complaint against Defendants LG Electronics, Inc., LG Electronics U.S.A., Inc. and LG Electronics MobileComm U.S.A., Inc. ("Defendants" or "LG) as follows:

THE PARTIES

2. CyWee is a corporation existing under the laws of the British Virgin Islands with a principal place of business at 3F, No.28, Lane 128, Jing Ye 1st Road, Taipei, Taiwan 10462.

3. CyWee is a world-leading technology company that focuses on building products and providing services for consumers and businesses. CyWee has one of the most significant patent portfolios in the industry, and is a market leader in its core development areas of motion processing, wireless high definition video delivery, and facial tracking technology.

4. On information and belief, Defendant LG Electronics, Inc. ("LGE") is a company incorporated in South Korea located at LG Twin Tower, 128 Yeoui-daero, Yeongdeungpo-gu, Seoul, 150-721, South Korea.

5. On information and belief, Defendant LG Electronics U.S.A., Inc. ("LGEUSA") is a Delaware corporation with its principal place of business at 920 Sylvan Avenue, Englewood Cliffs, New Jersey 07632. On information and belief, LGEUSA is a wholly owned subsidiary of LGE. *See Innovative Display Techs. LLC v. LG Elecs., Inc., et al.*, Case No. 2:16-cv-00932-JRG, Dkt. No. 22 (E.D. Tex. Nov. 1, 2016). On information and belief, LGEUSA may be served via its registered agent for service of process: Lawyers Incorporating Service, 2710 Gateway Oaks Drive, Suite 150N, Sacramento CA 95833.

6. On information and belief, Defendant LG Electronics MobileComm U.S.A., Inc. ("LGEMU") is a California corporation with its principal place of business at 10101 Old Grove Road, San Diego, California 92131. On information

and belief, LGEMU is a wholly-owned subsidiary of LGEUSA. *See id.* On information and belief, LGEMU may be served via its registered agent for service of process: CSC – Lawyers Incorporating Service, 2710 Gateway Oaks Drive, Suite 150N, Sacramento, CA 95833.

7. Defendants LGE, LGEUSA, and LGEMU are collectively referred to as "Defendants" or "LG." LG is doing business in the United States and, more particularly, in the State of California and the Southern District of California, by designing, marketing, making, using, selling, importing, and/or offering for sale products that infringe the patent claims involved in this action or by transacting other business in this District.

JURISDICTION AND VENUE

8. This action arises under the patent laws of the United States, 35 U.S.C. § 1 *et seq.* This Court has subject matter jurisdiction pursuant to 28 U.S.C. §§ 1331 and 1338(a).

9. This Court has personal jurisdiction over each Defendant. Each Defendant has conducted and does conduct business within the State of California. Each Defendant has purposefully and voluntarily availed itself of the privileges of conducting business in the United States, State of California, and in the Southern District of California by continuously and systematically placing goods into the stream of commerce through an established distribution channel with the expectation that they will be purchased by consumers in the Southern District of California. Plaintiff's cause of action arises directly from Defendants' business contacts and other activities in the State of California and the Southern District of California.

10. Upon information and belief, each Defendant has committed acts of infringement in this District giving rise to this action and does business in this District, including making sales and/or providing service and support for their respective customers in this District. Defendants purposefully and voluntarily sold

one or more of their infringing products with the expectation that they will be purchased by consumers in this District. These infringing products have been and continue to be purchased by consumers in this District. Defendants have committed acts of patent infringement within the United States and, state of California, and the Southern District of California.

11. Venue is proper as to LGEMU under 28 U.S.C. § 1400(b) in that LGEMU is incorporated in California and, therefore, resides in this District. *TC Heartland LLC v. Kraft Food Grps. Brands LLC*, 581 U.S. ___, 2017 WL 2216934, at *8 (2017).

12. Upon information and belief, LGEMU is an agent of LGEUSA and is held out to the public as such. *See* http://www.lg.com/us/careers (last visited May 30, 2017) (referring to "LG MobileComm USA" as a "LG Mobile Unit division"). Upon information and belief, LGEMU does not have its own website but is listed as the "Media Contact" for "Mobile Phones" on the LG.com United States website. *See* http://www.lg.com/us/press-media/media-contacts (last visited May 30, 2017).

13. Further, upon information and belief, LGEMU operates under the "LG" trademark; offers, sells, services, and/or distributes only LG products; and coordinates its policies and operations with those of LGEUSA to benefit and primarily serve the interests of LGEUSA and LGEUSA's parent corporation.

14. Further, upon information and belief, support materials and documentation provided to consumers with the mobile products offered by Defendants do not delineate between LGE, LGEUSA, and LGEMU. By way of example, the User Guide for the LG G6 refers generally to "LG" without any distinction as to LGE, LGEUSA, or LGEMU, including in the limited warranty provided therein. Upon information and belief, for consumers of the products accused in this Complaint, there is no substantive distinction between LGEMU and either LGEUSA or LGE.

15. Based on the foregoing, venue is proper as to LGEUSA under 28 U.S.C. § 1400(b) in that, upon information and belief, LGEUSA has a regular and established place of business in this District—namely, the place of business of its subsidiary/agent, LGEMU—and has committed acts of infringement herein. *See Appleton v. Ronson Serv. of Ill., Inc.*, 297 F.Supp. 868, 869 (N.D. Ill. 1968) ("formal corporate separateness can be disregarded for the purpose of establishing venue . . . where a number of factors . . . in the aggregate reveal a mere cloak for the relationship of agency." (citing *Leach Co. v. Gen. Sani-Can Mfg. Corp.*, 393 F.2d 183 (7th Cir. 1968)); *see also Stanley Works v. Globemaster, Inc.*, 400 F.Supp. 1325, 1331-32 (D. Mass. 1975).

16. Venue is proper as to LGE under 28 U.S.C. § 1391(c)(3) in that it is not a resident of the United States and may, therefore, be sued in any judicial district. *Brunette Mach. Works, Ltd. v. Kockum Indus., Inc.*, 406 U.S. 706, 714 (1972). Venue is further proper as to LGE under 28 U.S.C. § 1400(b) in that, upon information and belief, LGE has a regular and established place of business in this District—namely, the place of business of its subsidiaries/agents, LGEUSA and LGEMU—and has committed acts of infringement herein.

WILLFUL INFRINGEMENT

17. LG's infringement of the patents-in-suit has been and continues to be willful. LG has had knowledge of and notice of both patents-in-suit and its infringement of those patents as a result of confidential pre-suit licensing discussions. LG also has knowledge and notice of its infringement of the patents-in-suit as a result of the complaint filed in this case, which includes two claim charts illustrating LG's infringement of the patents-in-suit. LG's infringement of the patents-in-suit has been and continues to be willful and deliberate.

BACKGROUND

18. ITRI is a Taiwanese government and industry funded research and development center. In 2007, CyWee, which was started at ITRI, was formed. Its goal was to provide innovative motion-sensing technologies, such as those claimed in the patents-in-suit. Dr. Shun-Nan Liu and Chin-Lung Li, two of the inventors of the Patents, came to CyWee from ITRI. The third inventor, Zhou "Joe" Ye joined CyWee as its President and CEO from private industry.

19. The inventors, Zhou Ye, Chin-Lung Li, and Shun-Nan Liou conceived of the claims of the patents-in-suit—U.S. Patent No. 8,441,438 (the "'438 Patent") and U.S. Patent No. 8,552,978 (the "'978 Patent")—at CyWee Group Ltd., located at 3F, No. 28, Lane 128, Jing Ye Road, Taipei.

20. Several claims of the patents-in-suit are entitled to a priority date of at least January 6, 2010 based on U.S. Provisional Application Serial No. 61/292,558, filed January 6, 2010 ("Provisional Application").

21. Before May 22, 2009, CyWee began working on the "JIL Game Phone Project" or "JIL Phone." Before July 29, 2009, CyWee developed a solution for the JIL Phone that practiced several claims of the '438 Patent. Those claims were diligently and constructively reduced to practice thereafter through the filing of the Provisional Application, and were diligently and actually reduced to practice as discussed below. Accordingly, CyWee is entitled to a priority date of at least July 29, 2009 for several claims of the '438 Patent.

22. The JIL Phone was reduced to practice by at least September 25, 2009. The JIL Phone practiced several claims of both patents-in-suit. Accordingly, CyWee is entitled to a priority date of at least September 25, 2009 for several claims of the patents-in-suit.

PATENT INFRINGEMENT OF U.S. PATENT NO. 8,441,438

23. Plaintiff repeats and re-alleges each and every allegation of paragraphs 1-22 as though fully set forth herein.

24. The ’438 Patent, titled “3D Pointing Device and Method for Compensating Movement Thereof,” was duly and legally issued by the United States Patent and Trademark Office on May 14, 2013 to CyWee Group Limited, as assignee of named inventors Zhou Ye, Chin-Lung Li, and Shun-Nan Liou.

25. CyWee is the owner of all right, title, and interest in and to the ’438 Patent with full right to bring suit to enforce the patent, including the right to recover for past infringement damages.

26. Each and every claim of the ’438 Patent is valid and enforceable and each enjoys a statutory presumption of validity separate, apart, and in addition to the statutory presumption of validity enjoyed by every other of its claims. 35 U.S.C. § 282.

27. CyWee is informed and believes, and thereupon alleges, that LG has been, and is currently directly or indirectly infringing one or more claims of the ’438 Patent in violation of 35 U.S.C. § 271, including as stated below.

28. CyWee is informed and believes, and thereupon alleges, that LG has directly infringed, literally and/or under the doctrine of equivalents, and will continue to directly infringe claims of the ’438 Patent by making, using, selling, offering to sell, and/or importing into the United States products that embody or practice the apparatus and/or method covered by one or more claims of the ’438 Patent, including but not limited to Defendants’ devices such as LG V20, LG Stylo 3, LG Stylo 3 Plus, LG G5, LG G6, LG X Mach, LG X Venture, LG X Power 2, and LG X Cam (collectively referred to as “’438 Accused Products”).

29. CyWee adopts, and incorporates by reference, as if fully stated herein, the attached claim chart for claim 14 of the ’438 Patent, which is attached hereto as

Exhibit A. The claim chart describes and demonstrates how LG infringes the ’438 Patent. In addition, CyWee alleges that LG infringes one or more additional claims of the ’438 Patent in a similar manner.

30. Defendants’ acts of infringement have caused and will continue to cause substantial and irreparable damage to CyWee.

31. As a result of Defendants’ infringement of the ’438 Patent, CyWee has been damaged. CyWee is, therefore, entitled to such damages pursuant to 35 U.S.C. § 284 in an amount that presently cannot be pled but that will be determined at trial.

32. The LG V20 includes a display screen.

33. The LG V20 includes a housing.

34. The LG V20 includes a 3-axis accelerometer.

35. The LG V20 includes a 3-axis gyroscope.

36. The LG V20 includes at least one printed circuit board (“PCB”).

37. The LG V20 includes a 3-axis accelerometer attached to a PCB.

38. The LG V20 includes a 3-axis gyroscope attached to a PCB.

39. The LG V20 includes a 3-axis accelerometer that is capable of measuring accelerations.

40. The LG V20 includes a 3-axis gyroscope that is capable of measuring rotation rates.

41. The LG V20 includes a 3-axis accelerometer that is capable of measuring accelerations using a “Sensor Coordinate System” as described in the Android developer library. *See* https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords (describing “Sensor Coordinate System”).

42. The LG V20 includes a 3-axis gyroscope that is capable of measuring rotation rates using a “Sensor Coordinate System.”

43. The LG V20 includes a processor that is capable of processing data

associated with measurement from a 3-axis accelerometer.

44. The LG V20 includes a processor that is capable of processing data associated with measurement from a 3-axis gyroscope.

45. The LG V20 runs an Android operating system.

46. The Android operating system that runs on the LG V20 uses the measurement from a 3-axis accelerometer included in the device.

47. The Android operating system that runs on the LG V20 uses the measurement from a 3-axis gyroscope included in the device.

48. The Android operating system that runs on the LG V20 uses the measurement from a 3-axis accelerometer and the measurement from a 3-axis gyroscope to calculate an attitude of the device.

49. The LG Stylo 3 includes a display screen.

50. The LG Stylo 3 includes a housing.

51. The LG Stylo 3 includes a 3-axis accelerometer.

52. The LG Stylo 3 includes a 3-axis gyroscope.

53. The LG Stylo 3 includes at least one printed circuit board ("PCB").

54. The LG Stylo 3 includes a 3-axis accelerometer attached to a PCB.

55. The LG Stylo 3 includes a 3-axis gyroscope attached to a PCB.

56. The LG Stylo 3 includes a 3-axis accelerometer that is capable of measuring accelerations.

57. The LG Stylo 3 includes a 3-axis gyroscope that is capable of measuring rotation rates.

58. The LG Stylo 3 includes a 3-axis accelerometer that is capable of measuring accelerations using a "Sensor Coordinate System" as described in the Android developer library. *See* https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords (describing "Sensor Coordinate System").

59. The LG Stylo 3 includes a 3-axis gyroscope that is capable of measuring rotation rates using a "Sensor Coordinate System."

60. The LG Stylo 3 includes a processor that is capable of processing data associated with measurement from a 3-axis accelerometer.

61. The LG Stylo 3 includes a processor that is capable of processing data associated with measurement from a 3-axis gyroscope.

62. The LG Stylo 3 runs an Android operating system.

63. The Android operating system that runs on the LG Stylo 3 uses the measurement from a 3-axis accelerometer included in the device.

64. The Android operating system that runs on the LG Stylo 3 uses the measurement from a 3-axis gyroscope included in the device.

65. The Android operating system that runs on the LG Stylo 3 uses the measurement from a 3-axis accelerometer and the measurement from a 3-axis gyroscope to calculate an attitude of the device.

66. The LG Stylo 3 Plus includes a display screen.

67. The LG Stylo 3 Plus includes a housing.

68. The LG Stylo 3 Plus includes a 3-axis accelerometer.

69. The LG Stylo 3 Plus includes a 3-axis gyroscope.

70. The LG Stylo 3 Plus includes at least one printed circuit board ("PCB").

71. The LG Stylo 3 Plus includes a 3-axis accelerometer attached to a PCB.

72. The LG Stylo 3 Plus includes a 3-axis gyroscope attached to a PCB.

73. The LG Stylo 3 Plus includes a 3-axis accelerometer that is capable of measuring accelerations.

74. The LG Stylo 3 Plus includes a 3-axis gyroscope that is capable of measuring rotation rates.

75. The LG Stylo 3 Plus includes a 3-axis accelerometer that is capable of measuring accelerations using a "Sensor Coordinate System" as described in the

Android developer library. *See* https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords (describing "Sensor Coordinate System").

76. The LG Stylo 3 Plus includes a 3-axis gyroscope that is capable of measuring rotation rates using a "Sensor Coordinate System."

77. The LG Stylo 3 Plus includes a processor that is capable of processing data associated with measurement from a 3-axis accelerometer.

78. The LG Stylo 3 Plus includes a processor that is capable of processing data associated with measurement from a 3-axis gyroscope.

79. The LG Stylo 3 Plus runs an Android operating system.

80. The Android operating system that runs on the LG Stylo 3 Plus uses the measurement from a 3-axis accelerometer included in the device.

81. The Android operating system that runs on the LG Stylo 3 Plus uses the measurement from a 3-axis gyroscope included in the device.

82. The Android operating system that runs on the LG Stylo 3 Plus uses the measurement from a 3-axis accelerometer and the measurement from a 3-axis gyroscope to calculate an attitude of the device.

83. The LG G5 includes a display screen.

84. The LG G5 includes a housing.

85. The LG G5 includes a 3-axis accelerometer.

86. The LG G5 includes a 3-axis gyroscope.

87. The LG G5 includes at least one printed circuit board ("PCB").

88. The LG G5 includes a 3-axis accelerometer attached to a PCB.

89. The LG G5 includes a 3-axis gyroscope attached to a PCB.

90. The LG G5 includes a 3-axis accelerometer that is capable of measuring accelerations.

91. The LG G5 includes a 3-axis gyroscope that is capable of measuring

rotation rates.

92. The LG G5 includes a 3-axis accelerometer that is capable of measuring accelerations using a "Sensor Coordinate System" as described in the Android developer library. *See* https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords (describing "Sensor Coordinate System").

93. The LG G5 includes a 3-axis gyroscope that is capable of measuring rotation rates using a "Sensor Coordinate System."

94. The LG G5 includes a processor that is capable of processing data associated with measurement from a 3-axis accelerometer.

95. The LG G5 includes a processor that is capable of processing data associated with measurement from a 3-axis gyroscope.

96. The LG G5 runs an Android operating system.

97. The Android operating system that runs on the LG G5 uses the measurement from a 3-axis accelerometer included in the device.

98. The Android operating system that runs on the LG G5 uses the measurement from a 3-axis gyroscope included in the device.

99. The Android operating system that runs on the LG G5 uses the measurement from a 3-axis accelerometer and the measurement from a 3-axis gyroscope to calculate an attitude of the device.

100. The LG G6 includes a display screen.

101. The LG G6 includes a housing.

102. The LG G6 includes a 3-axis accelerometer.

103. The LG G6 includes a 3-axis gyroscope.

104. The LG G6 includes at least one printed circuit board ("PCB").

105. The LG G6 includes a 3-axis accelerometer attached to a PCB.

106. The LG G6 includes a 3-axis gyroscope attached to a PCB.

107. The LG G6 includes a 3-axis accelerometer that is capable of measuring accelerations.

108. The LG G6 includes a 3-axis gyroscope that is capable of measuring rotation rates.

109. The LG G6 includes a 3-axis accelerometer that is capable of measuring accelerations using a "Sensor Coordinate System" as described in the Android developer library. *See* https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords (describing "Sensor Coordinate System").

110. The LG G6 includes a 3-axis gyroscope that is capable of measuring rotation rates using a "Sensor Coordinate System."

111. The LG G6 includes a processor that is capable of processing data associated with measurement from a 3-axis accelerometer.

112. The LG G6 includes a processor that is capable of processing data associated with measurement from a 3-axis gyroscope.

113. The LG G6 runs an Android operating system.

114. The Android operating system that runs on the LG G6 uses the measurement from a 3-axis accelerometer included in the device.

115. The Android operating system that runs on the LG G6 uses the measurement from a 3-axis gyroscope included in the device.

116. The Android operating system that runs on the LG G6 uses the measurement from a 3-axis accelerometer and the measurement from a 3-axis gyroscope to calculate an attitude of the device.

117. The LG X Mach includes a display screen.

118. The LG X Mach includes a housing.

119. The LG X Mach includes a 3-axis accelerometer.

120. The LG X Mach includes a 3-axis gyroscope.

121. The LG X Mach includes at least one printed circuit board (“PCB”).

122. The LG X Mach includes a 3-axis accelerometer attached to a PCB.

123. The LG X Mach includes a 3-axis gyroscope attached to a PCB.

124. The LG X Mach includes a 3-axis accelerometer that is capable of measuring accelerations.

125. The LG X Mach includes a 3-axis gyroscope that is capable of measuring rotation rates.

126. The LG X Mach includes a 3-axis accelerometer that is capable of measuring accelerations using a “Sensor Coordinate System” as described in the Android developer library. *See* https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords (describing “Sensor Coordinate System”).

127. The LG X Mach includes a 3-axis gyroscope that is capable of measuring rotation rates using a “Sensor Coordinate System.”

128. The LG X Mach includes a processor that is capable of processing data associated with measurement from a 3-axis accelerometer.

129. The LG X Mach includes a processor that is capable of processing data associated with measurement from a 3-axis gyroscope.

130. The LG X Mach runs an Android operating system.

131. The Android operating system that runs on the LG X Mach uses the measurement from a 3-axis accelerometer included in the device.

132. The Android operating system that runs on the LG X Mach uses the measurement from a 3-axis gyroscope included in the device.

133. The Android operating system that runs on the LG X Mach uses the measurement from a 3-axis accelerometer and the measurement from a 3-axis gyroscope to calculate an attitude of the device.

134. The LG X Venture includes a display screen.

135. The LG X Venture includes a housing.

136. The LG X Venture includes a 3-axis accelerometer.

137. The LG X Venture includes a 3-axis gyroscope.

138. The LG X Venture includes at least one printed circuit board ("PCB").

139. The LG X Venture includes a 3-axis accelerometer attached to a PCB.

140. The LG X Venture includes a 3-axis gyroscope attached to a PCB.

141. The LG X Venture includes a 3-axis accelerometer that is capable of measuring accelerations.

142. The LG X Venture includes a 3-axis gyroscope that is capable of measuring rotation rates.

143. The LG X Venture includes a 3-axis accelerometer that is capable of measuring accelerations using a "Sensor Coordinate System" as described in the Android developer library. *See* https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords (describing "Sensor Coordinate System").

144. The LG X Venture includes a 3-axis gyroscope that is capable of measuring rotation rates using a "Sensor Coordinate System."

145. The LG X Venture includes a processor that is capable of processing data associated with measurement from a 3-axis accelerometer.

146. The LG X Venture includes a processor that is capable of processing data associated with measurement from a 3-axis gyroscope.

147. The LG X Venture runs an Android operating system.

148. The Android operating system that runs on the LG X Venture uses the measurement from a 3-axis accelerometer included in the device.

149. The Android operating system that runs on the LG X Venture uses the measurement from a 3-axis gyroscope included in the device.

150. The Android operating system that runs on the LG X Venture uses the

measurement from a 3-axis accelerometer and the measurement from a 3-axis gyroscope to calculate an attitude of the device.

151. The LG X Power 2 includes a display screen.

152. The LG X Power 2 includes a housing.

153. The LG X Power 2 includes a 3-axis accelerometer.

154. The LG X Power 2 includes a 3-axis gyroscope.

155. The LG X Power 2 includes at least one printed circuit board ("PCB").

156. The LG X Power 2 includes a 3-axis accelerometer attached to a PCB.

157. The LG X Power 2 includes a 3-axis gyroscope attached to a PCB.

158. The LG X Power 2 includes a 3-axis accelerometer that is capable of measuring accelerations.

159. The LG X Power 2 includes a 3-axis gyroscope that is capable of measuring rotation rates.

160. The LG X Power 2 includes a 3-axis accelerometer that is capable of measuring accelerations using a "Sensor Coordinate System" as described in the Android developer library. *See* https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords (describing "Sensor Coordinate System").

161. The LG X Power 2 includes a 3-axis gyroscope that is capable of measuring rotation rates using a "Sensor Coordinate System."

162. The LG X Power 2 includes a processor that is capable of processing data associated with measurement from a 3-axis accelerometer.

163. The LG X Power 2 includes a processor that is capable of processing data associated with measurement from a 3-axis gyroscope.

164. The LG X Power 2 runs an Android operating system.

165. The Android operating system that runs on the LG X Power 2 uses the measurement from a 3-axis accelerometer included in the device.

166. The Android operating system that runs on the LG X Power 2 uses the measurement from a 3-axis gyroscope included in the device.

167. The Android operating system that runs on the LG X Power 2 uses the measurement from a 3-axis accelerometer and the measurement from a 3-axis gyroscope to calculate an attitude of the device.

168. The LG X Cam includes a display screen.

169. The LG X Cam includes a housing.

170. The LG X Cam includes a 3-axis accelerometer.

171. The LG X Cam includes a 3-axis gyroscope.

172. The LG X Cam includes at least one printed circuit board ("PCB").

173. The LG X Cam includes a 3-axis accelerometer attached to a PCB.

174. The LG X Cam includes a 3-axis gyroscope attached to a PCB.

175. The LG X Cam includes a 3-axis accelerometer that is capable of measuring accelerations.

176. The LG X Cam includes a 3-axis gyroscope that is capable of measuring rotation rates.

177. The LG X Cam includes a 3-axis accelerometer that is capable of measuring accelerations using a "Sensor Coordinate System" as described in the Android developer library. *See* https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords (describing "Sensor Coordinate System").

178. The LG X Cam includes a 3-axis gyroscope that is capable of measuring rotation rates using a "Sensor Coordinate System."

179. The LG X Cam includes a processor that is capable of processing data associated with measurement from a 3-axis accelerometer.

180. The LG X Cam includes a processor that is capable of processing data associated with measurement from a 3-axis gyroscope.

181. The LG X Cam runs an Android operating system.

182. The Android operating system that runs on the LG X Cam uses the measurement from a 3-axis accelerometer included in the device.

183. The Android operating system that runs on the LG X Cam uses the measurement from a 3-axis gyroscope included in the device.

184. The Android operating system that runs on the LG X Cam uses the measurement from a 3-axis accelerometer and the measurement from a 3-axis gyroscope to calculate an attitude of the device.

PATENT INFRINGEMENT OF U.S. PATENT NO. 8,552,978

185. Plaintiff repeats and re-alleges each and every allegation of paragraphs 1-184 as though fully set forth herein.

186. The ’978 Patent, titled “3D Pointing Device and Method for Compensating Rotations of the 3D Pointing Device Thereof,” was duly and legally issued by the United States Patent and Trademark Office on October 8, 2013 to CyWee Group Limited, as assignee of named inventors Zhou Ye, Chin-Lung Li, and Shun-Nan Liou.

187. CyWee is the owner of all right, title, and interest in and to the ’978 Patent with full right to bring suit to enforce the patent, including the right to recover for past infringement damages.

188. Each and every claim of the ’978 Patent is valid and enforceable and each enjoys a statutory presumption of validity separate, apart, and in addition to the statutory presumption of validity enjoyed by every other of its claims. 35 U.S.C. § 282.

189. CyWee is informed and believes, and thereupon alleges, that LG has been, and is currently directly and/or indirectly infringing one or more claims of the ’978 Patent in violation of 35 U.S.C. § 271, including as stated below.

190. CyWee is informed and believes, and thereupon alleges, that LG has directly infringed, literally and/or under the doctrine of equivalents, and will continue to directly infringe claims of the ’978 Patent by making, using, selling, offering to sell, and/or importing into the United States products that embody or practice the apparatus and/or method covered by one or more claims of the ’978 Patent, including but not limited to Defendants’ devices such as LG V20, LG Stylo 3, LG Stylo 3 Plus, LG G5, LG G6, LG X mach, LG X venture, LG X power 2, and LG X cam (collectively referred to as “’978 Accused Products”).

191. CyWee adopts, and incorporates by reference, as if fully stated herein, the attached claim chart for claim 10 of the ’978 Patent, which is attached hereto as Exhibit B. The claim chart describes and demonstrates how LG infringes the ’978 Patent. In addition, CyWee alleges that LG infringes one or more additional claims of the ’978 Patent in a similar manner.

192. Defendants’ acts of infringement have caused and will continue to cause substantial and irreparable damage to CyWee.

193. As a result of Defendants’ infringement of the ’978 Patent, CyWee has been damaged. CyWee is, therefore, entitled to such damages pursuant to 35 U.S.C. § 284 in an amount that presently cannot be pled but that will be determined at trial.

194. The LG V20 includes a 3-axis geomagnetic sensor.

195. The LG V20 includes a 3-axis geomagnetic sensor that is capable of measuring a geomagnetic field.

196. The LG V20 includes a 3-axis geomagnetic field sensor to measure a geomagnetic field using a “Sensor Coordinate System.” *See* https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords (describing “Sensor Coordinate System”).

197. The Android operating system that runs on the LG V20 uses the measurement from a 3-axis geomagnetic sensor included in the device.

198. The Android operating system that runs on the LG V20 uses the measurement from a 3-axis accelerometer, the measurement from a 3-axis geomagnetic field sensor, and the measurement from a 3-axis gyroscope to calculate an attitude of the device.

199. The Android operating system that runs on the LG V20 uses the measurement from a 3-axis accelerometer, the measurement from a 3-axis geomagnetic field sensor, and the measurement from a 3-axis gyroscope to calculate an attitude of the device that can be represented by an azimuth angle, a pitch angle, and a roll angle.

200. The LG V20 has the ability to directly control apps by moving or rotating the device (for example, racing game apps).

201. The LG V20 has the ability to run apps that can provide information based on the direction your device is facing, such as a map or navigation app.

202. The LG V20 includes a 3-axis geomagnetic sensor.

203. The LG V20 includes a 3-axis geomagnetic sensor that is capable of measuring a geomagnetic field.

204. The LG Stylo 3 includes a 3-axis geomagnetic sensor.

205. The LG Stylo 3 includes a 3-axis geomagnetic sensor that is capable of measuring a geomagnetic field.

206. The LG Stylo 3 includes a 3-axis geomagnetic field sensor to measure a geomagnetic field using a "Sensor Coordinate System." *See* https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords (describing "Sensor Coordinate System").

207. The Android operating system that runs on the LG Stylo 3 uses the measurement from a 3-axis geomagnetic sensor included in the device.

208. The Android operating system that runs on the LG Stylo 3 uses the measurement from a 3-axis accelerometer, the measurement from a 3-axis

geomagnetic field sensor, and the measurement from a 3-axis gyroscope to calculate an attitude of the device.

209. The Android operating system that runs on the LG Stylo 3 uses the measurement from a 3-axis accelerometer, the measurement from a 3-axis geomagnetic field sensor, and the measurement from a 3-axis gyroscope to calculate an attitude of the device that can be represented by an azimuth angle, a pitch angle, and a roll angle.

210. The LG Stylo 3 has the ability to directly control apps by moving or rotating the device (for example, racing game apps).

211. The LG Stylo 3 has the ability to run apps that can provide information based on the direction your device is facing, such as a map or navigation app.

212. The LG Stylo 3 includes a 3-axis geomagnetic sensor.

213. The LG Stylo 3 includes a 3-axis geomagnetic sensor that is capable of measuring a geomagnetic field.

214. The LG Stylo 3 Plus includes a 3-axis geomagnetic sensor.

215. The LG Stylo 3 Plus includes a 3-axis geomagnetic sensor that is capable of measuring a geomagnetic field.

216. The LG Stylo 3 Plus includes a 3-axis geomagnetic field sensor to measure a geomagnetic field using a "Sensor Coordinate System." *See* https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords (describing "Sensor Coordinate System").

217. The Android operating system that runs on the LG Stylo 3 Plus uses the measurement from a 3-axis geomagnetic sensor included in the device.

218. The Android operating system that runs on the LG Stylo 3 Plus uses the measurement from a 3-axis accelerometer, the measurement from a 3-axis geomagnetic field sensor, and the measurement from a 3-axis gyroscope to calculate an attitude of the device.

219. The Android operating system that runs on the LG Stylo 3 Plus uses the measurement from a 3-axis accelerometer, the measurement from a 3-axis geomagnetic field sensor, and the measurement from a 3-axis gyroscope to calculate an attitude of the device that can be represented by an azimuth angle, a pitch angle, and a roll angle.

220. The LG Stylo 3 Plus has the ability to directly control apps by moving or rotating the device (for example, racing game apps).

221. The LG Stylo 3 Plus has the ability to run apps that can provide information based on the direction your device is facing, such as a map or navigation app.

222. The LG Stylo 3 Plus includes a 3-axis geomagnetic sensor.

223. The LG Stylo 3 Plus includes a 3-axis geomagnetic sensor that is capable of measuring a geomagnetic field.

224. The LG G5 includes a 3-axis geomagnetic sensor.

225. The LG G5 includes a 3-axis geomagnetic sensor that is capable of measuring a geomagnetic field.

226. The LG G5 includes a 3-axis geomagnetic field sensor to measure a geomagnetic field using a "Sensor Coordinate System." *See* https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords (describing "Sensor Coordinate System").

227. The Android operating system that runs on the LG G5 uses the measurement from a 3-axis geomagnetic sensor included in the device.

228. The Android operating system that runs on the LG G5 uses the measurement from a 3-axis accelerometer, the measurement from a 3-axis geomagnetic field sensor, and the measurement from a 3-axis gyroscope to calculate an attitude of the device.

229. The Android operating system that runs on the LG G5 uses the

measurement from a 3-axis accelerometer, the measurement from a 3-axis geomagnetic field sensor, and the measurement from a 3-axis gyroscope to calculate an attitude of the device that can be represented by an azimuth angle, a pitch angle, and a roll angle.

230. The LG G5 has the ability to directly control apps by moving or rotating the device (for example, racing game apps).

231. The LG G5 has the ability to run apps that can provide information based on the direction your device is facing, such as a map or navigation app.

232. The LG G5 includes a 3-axis geomagnetic sensor.

233. The LG G5 includes a 3-axis geomagnetic sensor that is capable of measuring a geomagnetic field.

234. The LG G6 includes a 3-axis geomagnetic sensor.

235. The LG G6 includes a 3-axis geomagnetic sensor that is capable of measuring a geomagnetic field.

236. The LG G6 includes a 3-axis geomagnetic field sensor to measure a geomagnetic field using a "Sensor Coordinate System." *See* https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords (describing "Sensor Coordinate System").

237. The Android operating system that runs on the LG G6 uses the measurement from a 3-axis geomagnetic sensor included in the device.

238. The Android operating system that runs on the LG G6 uses the measurement from a 3-axis accelerometer, the measurement from a 3-axis geomagnetic field sensor, and the measurement from a 3-axis gyroscope to calculate an attitude of the device.

239. The Android operating system that runs on the LG G6 uses the measurement from a 3-axis accelerometer, the measurement from a 3-axis geomagnetic field sensor, and the measurement from a 3-axis gyroscope to calculate

an attitude of the device that can be represented by an azimuth angle, a pitch angle, and a roll angle.

240. The LG G6 has the ability to directly control apps by moving or rotating the device (for example, racing game apps).

241. The LG G6 has the ability to run apps that can provide information based on the direction your device is facing, such as a map or navigation app.

242. The LG G6 includes a 3-axis geomagnetic sensor.

243. The LG G6 includes a 3-axis geomagnetic sensor that is capable of measuring a geomagnetic field.

244. The LG X Mach includes a 3-axis geomagnetic sensor.

245. The LG X Mach includes a 3-axis geomagnetic sensor that is capable of measuring a geomagnetic field.

246. The LG X Mach includes a 3-axis geomagnetic field sensor to measure a geomagnetic field using a "Sensor Coordinate System." *See* https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords (describing "Sensor Coordinate System").

247. The Android operating system that runs on the LG X Mach uses the measurement from a 3-axis geomagnetic sensor included in the device.

248. The Android operating system that runs on the LG X Mach uses the measurement from a 3-axis accelerometer, the measurement from a 3-axis geomagnetic field sensor, and the measurement from a 3-axis gyroscope to calculate an attitude of the device.

249. The Android operating system that runs on the LG X Mach uses the measurement from a 3-axis accelerometer, the measurement from a 3-axis geomagnetic field sensor, and the measurement from a 3-axis gyroscope to calculate an attitude of the device that can be represented by an azimuth angle, a pitch angle, and a roll angle.

250. The LG X Mach has the ability to directly control apps by moving or rotating the device (for example, racing game apps).

251. The LG X Mach has the ability to run apps that can provide information based on the direction your device is facing, such as a map or navigation app.

252. The LG X Mach includes a 3-axis geomagnetic sensor.

253. The LG X Mach includes a 3-axis geomagnetic sensor that is capable of measuring a geomagnetic field.

254. The LG X Venture includes a 3-axis geomagnetic sensor.

255. The LG X Venture includes a 3-axis geomagnetic sensor that is capable of measuring a geomagnetic field.

256. The LG X Venture includes a 3-axis geomagnetic field sensor to measure a geomagnetic field using a "Sensor Coordinate System." *See* https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords (describing "Sensor Coordinate System").

257. The Android operating system that runs on the LG X Venture uses the measurement from a 3-axis geomagnetic sensor included in the device.

258. The Android operating system that runs on the LG X Venture uses the measurement from a 3-axis accelerometer, the measurement from a 3-axis geomagnetic field sensor, and the measurement from a 3-axis gyroscope to calculate an attitude of the device.

259. The Android operating system that runs on the LG X Venture uses the measurement from a 3-axis accelerometer, the measurement from a 3-axis geomagnetic field sensor, and the measurement from a 3-axis gyroscope to calculate an attitude of the device that can be represented by an azimuth angle, a pitch angle, and a roll angle.

260. The LG X Venture has the ability to directly control apps by moving or rotating the device (for example, racing game apps).

261. The LG X Venture has the ability to run apps that can provide information based on the direction your device is facing, such as a map or navigation app.

262. The LG X Venture includes a 3-axis geomagnetic sensor.

263. The LG X Venture includes a 3-axis geomagnetic sensor that is capable of measuring a geomagnetic field.

264. The LG X Power 2 includes a 3-axis geomagnetic sensor.

265. The LG X Power 2 includes a 3-axis geomagnetic sensor that is capable of measuring a geomagnetic field.

266. The LG X Power 2 includes a 3-axis geomagnetic field sensor to measure a geomagnetic field using a "Sensor Coordinate System." *See* https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords (describing "Sensor Coordinate System").

267. The Android operating system that runs on the LG X Power 2 uses the measurement from a 3-axis geomagnetic sensor included in the device.

268. The Android operating system that runs on the LG X Power 2 uses the measurement from a 3-axis accelerometer, the measurement from a 3-axis geomagnetic field sensor, and the measurement from a 3-axis gyroscope to calculate an attitude of the device.

269. The Android operating system that runs on the LG X Power 2 uses the measurement from a 3-axis accelerometer, the measurement from a 3-axis geomagnetic field sensor, and the measurement from a 3-axis gyroscope to calculate an attitude of the device that can be represented by an azimuth angle, a pitch angle, and a roll angle.

270. The LG X Power 2 has the ability to directly control apps by moving or rotating the device (for example, racing game apps).

271. The LG X Power 2 has the ability to run apps that can provide

information based on the direction your device is facing, such as a map or navigation app.

272. The LG X Power 2 includes a 3-axis geomagnetic sensor.

273. The LG X Power 2 includes a 3-axis geomagnetic sensor that is capable of measuring a geomagnetic field.

274. The LG X Cam includes a 3-axis geomagnetic sensor.

275. The LG X Cam includes a 3-axis geomagnetic sensor that is capable of measuring a geomagnetic field.

276. The LG X Cam includes a 3-axis geomagnetic field sensor to measure a geomagnetic field using a "Sensor Coordinate System." *See* https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords (describing "Sensor Coordinate System").

277. The Android operating system that runs on the LG X Cam uses the measurement from a 3-axis geomagnetic sensor included in the device.

278. The Android operating system that runs on the LG X Cam uses the measurement from a 3-axis accelerometer, the measurement from a 3-axis geomagnetic field sensor, and the measurement from a 3-axis gyroscope to calculate an attitude of the device.

279. The Android operating system that runs on the LG X Cam uses the measurement from a 3-axis accelerometer, the measurement from a 3-axis geomagnetic field sensor, and the measurement from a 3-axis gyroscope to calculate an attitude of the device that can be represented by an azimuth angle, a pitch angle, and a roll angle.

280. The LG X Cam has the ability to directly control apps by moving or rotating the device (for example, racing game apps).

281. The LG X Cam has the ability to run apps that can provide information based on the direction your device is facing, such as a map or navigation app.

282. The LG X Cam includes a 3-axis geomagnetic sensor.

283. The LG X Cam includes a 3-axis geomagnetic sensor that is capable of measuring a geomagnetic field.

PRAYER FOR RELIEF

WHEREFORE, Plaintiff prays for entry of judgment against Defendants as follows:

A. A judgment that Defendants have infringed and continue to infringe the ’438 Patent and ’978 Patent, directly and/or indirectly by way of inducing infringement of such patents as alleged herein;

B. That Defendants provide to CyWee an accounting of all gains, profits and advantages derived by Defendants’ infringement of the ’978 Patent and ’438 Patent, and that CyWee be awarded damages adequate to compensate them for the wrongful infringement by Defendants, including treble damages for willful infringement, in accordance with 35 U.S.C. § 284;

C. That CyWee be awarded any other supplemental damages and interest on all damages, including, but not limited to, attorney fees available under 35 U.S.C. § 285;

D. That the Court permanently enjoin Defendants and all those in privity with Defendants from making, having made, selling, offering for sale, distributing and/or using products that infringe the ’978 Patent and/or the ’438 Patent, including the ’978 Accused Products and/or ’438 Accused Products, in the United States; and

E. That CyWee be awarded such other and further relief and all remedies available at law.

DEMAND FOR JURY TRIAL

Pursuant to Federal Rule of Civil Procedure 38(b), CyWee hereby demands a trial by jury on all issues triable to a jury.

Dated: May 31, 2017

Respectfully submitted,
/s/ *Jill Kopeikin*
Jill F. Kopeikin (State Bar No. 160792)
jkopeikin@gcalaw.com
Valerie M. Wagner (State Bar No. 173146)
vwagner@gcalaw.com
GCA LAW PARTNERS LLP
2570 W. El Camino Real, Suite 400
Mountain View, CA 94040
Telephone: (650) 482-3900
Fax: (650) 428-3901

Michael W. Shore (*pro hac vice* to be filed)
mshore@shorechan.com
Alfonso G. Chan (*pro hac vice* to be filed)
achan@shorechan.com
Ari Rafilson (*pro hac vice* to be filed)
arafilson@shorechan.com
Christopher L. Evans (*pro hac vice* to be filed)
cevans@shorechan.com
Paul T. Beeler (*pro hac vice* to be filed)
pbeeler@shorechan.com
SHORE CHAN DEPUMPO LLP
901 Main Street, Suite 3300
Dallas, California 75202
Telephone: (214) 593-9110
Facsimile: (214) 593-9111

COUNSEL FOR PLAINTIFF CYWEE GROUP LTD.

TABLE OF CONTENTS – EXHIBITS TO ORIGINAL COMPLAINT

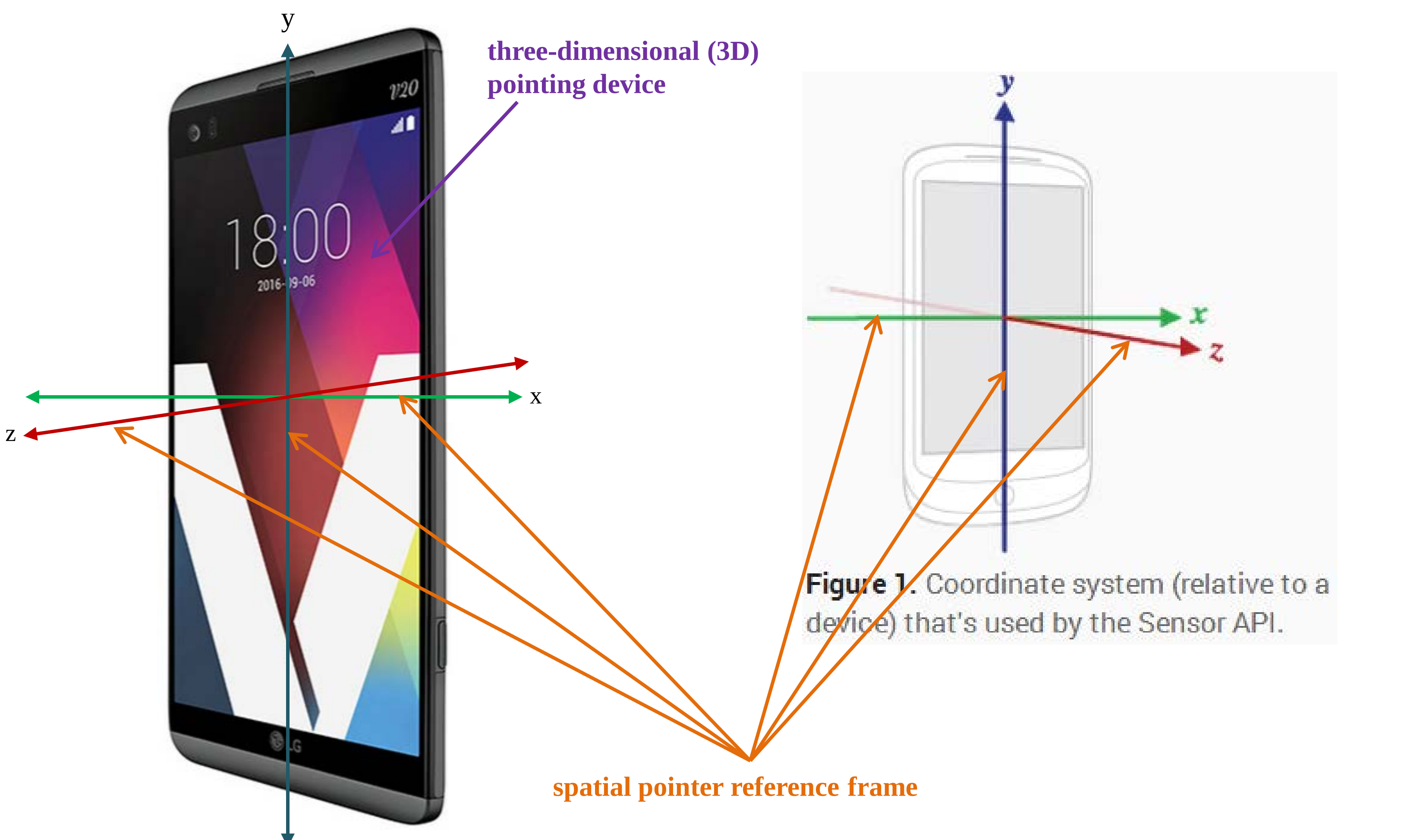
U.S. Patent No. 8,441,438

LG V20

U.S. Patent No. 8,441,438 – LG V20

Claim 14

A method for obtaining a resulting deviation including resultant angles in a **spatial pointer reference frame** of a **three-dimensional (3D) pointing device** utilizing a six-axis motion sensor module therein and subject to movements and rotations in dynamic environments in said **spatial pointer reference frame**, comprising the steps of:

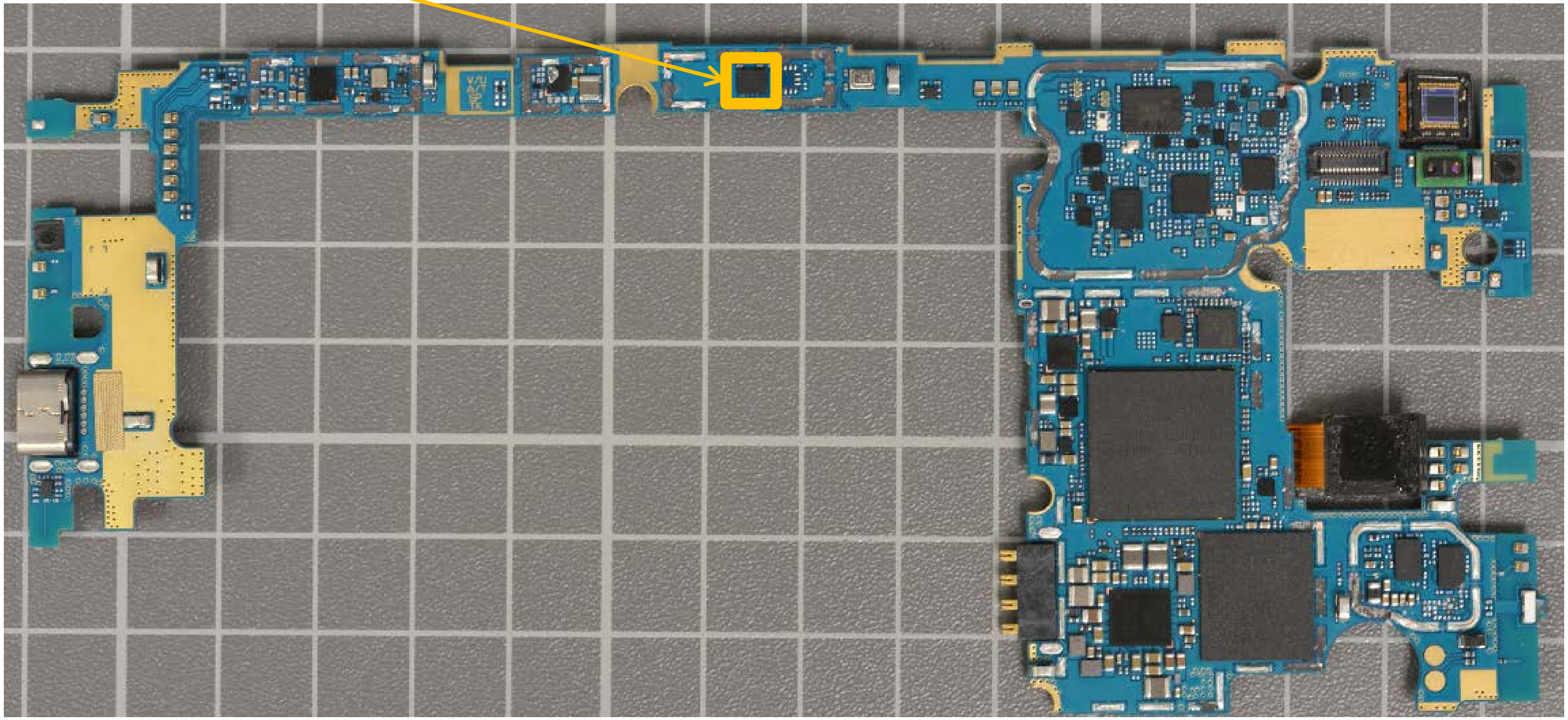


Source: http://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords

Claim 14

A method for obtaining a resulting deviation including resultant angles in a spatial pointer reference frame of a three-dimensional (3D) pointing device utilizing a **six-axis motion sensor module** therein and subject to movements and rotations in dynamic environments in said spatial pointer reference frame, comprising the steps of:

six-axis motion sensor module



Claim 14

obtaining a **previous state** of the six-axis motion sensor module; wherein the **previous state** includes an initial-value set associated with **previous angular velocities** gained from the motion sensor signals of the six-axis motion sensor module at a previous time T−1;

The previous state is obtained through an update program that includes a `predict()` function and an `update()` function. Those functions that are used to update the global variable `x0` based on `x0` (the **previous state**) associated with **previous angular velocities** `w` gained at a previous time T-1 to obtain an updated state `x0`. The updated state `x0` becomes the previous state `x0` at time T (the next iteration) of the update program to obtain the updated state `x0` at time T.

```
void Fusion::predict(const vec3_t& w, float dT) {
    const vec4_t q  = x0;
```

previous state

```
        x0 = O*q;
```

```
void Fusion::update(const vec3_t& z, const vec3_t& Bi, float sigma) {
    vec4_t q(x0);
```

```
        const vec3_t e(z - Bb);
        const vec3_t dq(K[0]*e);

        q += getF(q)*(0.5f*dq);
        x0 = normalize_quat(q);
```

next iteration

Source: https://android.googlesource.com/platform/frameworks/native/+/master/services/sensorservice/Fusion.cpp

Claim 14

obtaining a **current state** of the six-axis motion sensor module by obtaining **measured angular velocities** ω_x, ω_y, ω_z gained from the motion sensor signals of the six-axis motion sensor module at a current time T;

```
void Fusion::handleGyro(const vec3_t& w, float dT) {
    if (!checkInitComplete(GYRO, w, dT))
        return;
```

measured angular velocities

```
void Fusion::predict(const vec3_t& w, float dT) {
    const vec4_t q  = x0;
```

```
        x0 = O*q;
```

current state

Source: https://android.googlesource.com/platform/frameworks/native/+/master/services/sensorservice/Fusion.cpp

Claim 14

obtaining a **measured state** of the six-axis motion sensor module by obtaining **measured axial accelerations** Ax, Ay, Az gained from the motion sensor signals of the six-axis motion sensor module at the current time T and calculating **predicted axial accelerations** Ax′, Ay′, Az′ based on the **measured angular velocities** ω_x, ω_y, ω_z of the current state of the six-axis motion sensor module **without using any derivatives of the measured angular velocities** ωx, ωy, ωz;

measured axial accelerations

```
    vec3_t unityA = a * l_inv;
```

```
void Fusion::update(const vec3_t& z, const vec3_t& Bi, float sigma) {
    vec4_t q(x0);
    // measured vector in body space: h(p) = A(p)*Bi
    const mat33_t A(quatToMatrix(q));
    const vec3_t Bb(A*Bi);
```

```
    const vec3_t e(z - Bb);
```

measured state

measured axial accelerations

predicted axial accelerations

As shown in the code above, the predicted measurement is obtained based on the first signal set **without using any derivatives of the measured angular velocities**.

Source: https://android.googlesource.com/platform/frameworks/native/+/master/services/sensorservice/Fusion.cpp

Claim 14

said **current state** of the six-axis motion sensor module is a second quaternion with respect to said current time T;

As shown in the examples provided, the **current state** is represented by the global state variable x0, which is a quaternion with respect to the current time T.

```
vec4_t Fusion::getAttitude() const {
    return x0;
}
```

Source: https://android.googlesource.com/platform/frameworks/native/+/master/services/sensorservice/Fusion.cpp

Claim 14

comparing the second quaternion in relation to the **measured angular velocities** ω_x, ω_y, ω_z of the **current state** at current time T with the **measured axial accelerations** Ax, Ay, Az and the **predicted axial accelerations** Ax′, Ay′, Az′ also at current time T; obtaining an **updated state** of the six-axis motion sensor module by comparing the **current state** with the **measured state** of the six-axis motion sensor module; and

measured angular velocities

```
void Fusion::predict(const vec3_t& w, float dT) {
    const vec4_t q  = x0;
```

previous state

```
    x0 = O*q;
```

current state

```
void Fusion::update(const vec3_t& z, const vec3_t& Bi, float sigma) {
    vec4_t q(x0);
```

measured state

measured axial accelerations

```
    const vec3_t e(z - Bb);
    const vec3_t dq(K[0]*e);

    q += getF(q)*(0.5f*dq);
    x0 = normalize_quat(q);
```

predicted axial accelerations

updated state

Source: https://android.googlesource.com/platform/frameworks/native/+/master/services/sensorservice/Fusion.cpp

Claim 14
calculating and converting the **updated state** of the six axis motion sensor module to said **resulting deviation comprising said resultant angles** in said spatial pointer reference frame of the 3D pointing device.

The **updated state** `x0` is in quaternion form, and can easily be converted to resultant angles.

According to Android's developer library, the `getOrientation()` function "computes the device's orientation based on the rotation matrix," and returns **resultant angles** including the Azimuth, Pitch, and Roll angles.

getOrientation

Added in API level 3

```
float[] getOrientation (float[] R,
                float[] values)
```

Computes the device's orientation based on the rotation matrix.

When it returns, the array values are as follows:

- values[0]: *Azimuth*, angle of rotation about the -z axis. This value represents the angle between the device's y axis and the magnetic north pole. When facing north, this angle is 0, when facing south, this angle is π. Likewise, when facing east, this angle is π/2, and when facing west, this angle is -π/2. The range of values is -π to π.
- values[1]: *Pitch*, angle of rotation about the x axis. This value represents the angle between a plane parallel to the device's screen and a plane parallel to the ground. Assuming that the bottom edge of the device faces the user and that the screen is face-up, tilting the top edge of the device toward the ground creates a positive pitch angle. The range of values is -π to π.
- values[2]: *Roll*, angle of rotation about the y axis. This value represents the angle between a plane perpendicular to the device's screen and a plane perpendicular to the ground. Assuming that the bottom edge of the device faces the user and that the screen is face-up, tilting the left edge of the device toward the ground creates a positive roll angle. The range of values is -π/2 to π/2.

The `getRotationMatrixFromVector()` function "convert[s] a rotation vector to a rotation matrix," and the `getQuaternionFromVector()` function "convert[s] a rotation vector to a normalized quaternion." Therefore, the quaternion, `x0`, can be easily converted to its mathematically equivalent form, rotation matrix, and used by `getOrientation()` function to compute the orientation in its angular form.

Source: https://android.googlesource.com/platform/frameworks/base/+/b267554/core/java/android/hardware/SensorManager.java

TABLE OF CONTENTS – EXHIBITS TO ORIGINAL COMPLAINT

U.S. Patent No. 8,552,978

LG V20

SUBJECT TO CHANGE

Claim 10
A method for compensating rotations of a 3D pointing device, comprising:

LG V20

Claim 10

generating an **orientation output** associated with an orientation of the 3D pointing device associated with three coordinate axes of a **global reference frame associated with Earth**;

When the orientation sensor is software-based, the **orientation output** is the attitude of the device that can be represented by the azimuth, pitch, and roll angles relative to the magnetic North Pole associated with a **global reference frame associated with Earth**.

Orientation

Underlying base sensor(s): Accelerometer, Magnetometer PREFERRED Gyroscope
Trigger-mode: Continuous
Wake-up sensor: No

Note: This is an older sensor type that has been deprecated in the Android SDK although not yet in the HAL. It has been replaced by the rotation vector sensor, which is more clearly defined, requires a gyroscope, and therefore provides more accurate results. Use the rotation vector sensor over the orientation sensor whenever possible.

The orientation sensor tracks the attitude of the device. All values are angles in degrees. Orientation sensors return sensor events for all three axes at a constant rate defined by setDelay().

- azimuth: angle between the magnetic north direction and the Y axis, around the Z axis (0<=azimuth<360). 0=North, 90=East, 180=South, 270=West
- pitch: Rotation around X axis (-180<=pitch<=180), with positive values when the z-axis moves toward the y-axis.
- roll: Rotation around Y axis (-90<=roll<=90), with positive values when the x-axis moves towards the z-axis.

Source: https://source.android.com/devices/sensors/composite_sensors.html

Claim 10

generating a **first signal set** comprising axial accelerations associated with movements and rotations of the 3D pointing device in the **spatial reference frame**;

Accelerometer

Trigger-mode: Continuous
Wake-up sensor: No

All values are in SI units (m/s^2) and measure the acceleration of the device minus the force of gravity.

Acceleration sensors return sensor events for all three axes at a constant rate defined by setDelay().

- x: Acceleration on the x-axis
- y: Acceleration on the y-axis
- z: Acceleration on the z-axis

Source: https://source.android.com/devices/sensors/base_triggers.html

Sensor Coordinate System

In general, the sensor framework uses a standard 3-axis coordinate system to express data values. For most sensors, the coordinate system is defined relative to the device's screen when the device is held in its default orientation (see figure 1). When a device is held in its default orientation, the X axis is horizontal and points to the right, the Y axis is vertical and points up, and the Z axis points toward the outside of the screen face. In this system, coordinates behind the screen have negative Z values. This coordinate system is used by the following sensors:

- Acceleration sensor
- Gravity sensor
- Gyroscope
- Linear acceleration sensor
- Geomagnetic field sensor

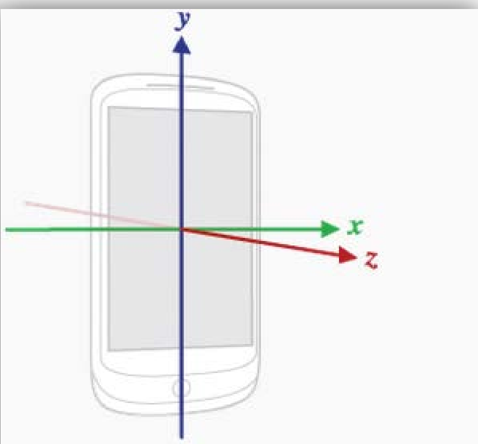


Figure 1. Coordinate system (relative to a device) that's used by the Sensor API.

Source: http://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords

Claim 10

generating a **second signal set** associated with **Earth's magnetism**;

The magnetometer (i.e., the compass) generates a second signal set associated with **Earth’s magnetism**.

Geomagnetic field

Trigger-mode: Continuous
Wake-up sensor: No

All values are in micro-Tesla (uT) and measure the geomagnetic field in the X, Y and Z axis.

Source: https://source.android.com/devices/sensors/composite_sensors.html

Claim 10

generating the **orientation output** based on the **first signal set**, the **second signal set** and the **rotation output** or based on the **first signal set** and the **second signal set**;

The Android source code shows generating the **orientation output** based on the **first signal set**, the **second signal set** and the **rotation output**.

The `handleGyro()` function passes **rotation output** `w` to the `predict()` function and the `update()` function to calculate an **orientation output**, `x0`.

```
void Fusion::handleGyro(const vec3_t& w, float dT) {
    if (!checkInitComplete(GYRO, w, dT))
        return;
```

```
void Fusion::predict(const vec3_t& w, float dT) {
    const vec4_t q  = x0;
```

```
        x0 = O*q;
```

```
void Fusion::update(const vec3_t& z, const vec3_t& Bi, float sigma) {
    vec4_t q(x0);
```

```
        const vec3_t e(z - Bb);
        const vec3_t dq(K[0]*e);

        q += getF(q)*(0.5f*dq);
        x0 = normalize_quat(q);
```

Source: https://android.googlesource.com/platform/frameworks/native/+/master/services/sensorservice/Fusion.cpp

Claim 10

generating the **orientation output** based on the **first signal set**, the second signal set and the rotation output or based on the **first signal set** and the second signal set;

The handleAcc() function passes the accelerometer measurements (**first signal set**) a to the update() function, which updates the **orientation output** x0.

```
status_t Fusion::handleAcc(const vec3_t& a, float dT) {
    if (!checkInitComplete(ACC, a, dT))
        return BAD_VALUE;
```

```
void Fusion::update(const vec3_t& z, const vec3_t& Bi, float sigma) {
    vec4_t q(x0);
```

```
        const vec3_t e(z - Bb);
        const vec3_t dq(K[0]*e);

        q += getF(q)*(0.5f*dq);
        x0 = normalize_quat(q);
```

Source: https://android.googlesource.com/platform/frameworks/native/+/master/services/sensorservice/Fusion.cpp

Claim 10

generating the **orientation output** based on the first signal set, the **second signal set** and the rotation output or based on the first signal set and the **second signal set**;

The `handleMag()` function passes the magnetometer measurements (**second signal set**) `m` to the same `update()`, which also updates the **orientation output** `x0`.

```
status_t Fusion::handleMag(const vec3_t& m) {
    if (!checkInitComplete(MAG, m))
        return BAD_VALUE;
```

```
void Fusion::update(const vec3_t& z, const vec3_t& Bi, float sigma) {
    vec4_t q(x0);
```

```
        const vec3_t e(z - Bb);
        const vec3_t dq(K[0]*e);

        q += getF(q)*(0.5f*dq);
        x0 = normalize_quat(q);
```

Source: https://android.googlesource.com/platform/frameworks/native/+/master/services/sensorservice/Fusion.cpp

Claim 10

generating a **rotation output** associated with a rotation of the 3D pointing device associated with three coordinate axes of a **spatial reference frame** associated with the 3D pointing device; and

Gyroscope

Trigger-mode: Continuous
Wake-up sensor: No

All values are in radians/second and measure the rate of rotation around the X, Y and Z axis. The coordinate system is the same as is used for the acceleration sensor. Rotation is positive in the counter-clockwise direction (right-hand rule).

Source: https://source.android.com/devices/sensors/base_triggers.html

Sensor Coordinate System

In general, the sensor framework uses a standard 3-axis coordinate system to express data values. For most sensors, the coordinate system is defined relative to the device's screen when the device is held in its default orientation (see figure 1). When a device is held in its default orientation, the X axis is horizontal and points to the right, the Y axis is vertical and points up, and the Z axis points toward the outside of the screen face. In this system, coordinates behind the screen have negative Z values. This coordinate system is used by the following sensors:

- Acceleration sensor
- Gravity sensor
- Gyroscope
- Linear acceleration sensor
- Geomagnetic field sensor

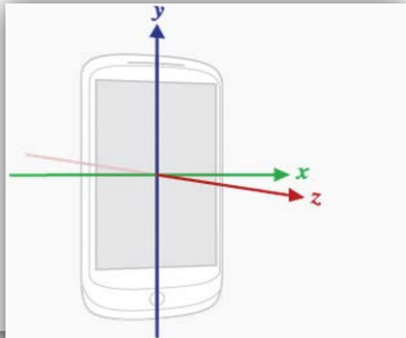


Figure 1. Coordinate system (relative to a device) that's used by the Sensor API.

Source: http://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords

Claim 10

using the orientation output and the rotation output to generate a transformed output associated with a **fixed reference frame** associated with a **display device**,

The **fixed reference frame** is defined by the horizontal and vertical axes of pixels on LG V20 Edge's **display device**.

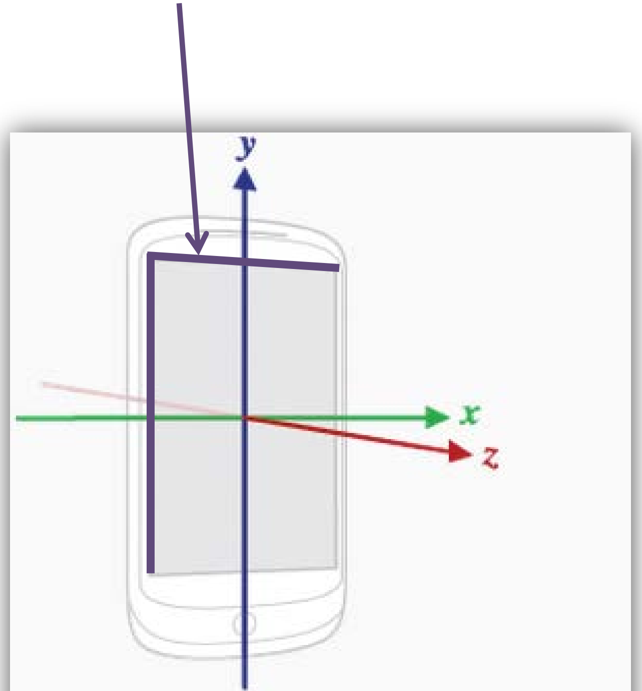


Figure 1. Coordinate system (relative to a device) that's used by the Sensor API.

Source: http://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords

Claim 10

using the orientation output and the rotation output to generate a transformed output associated with a **fixed reference frame** associated with a **display device**,

The `remapCoordinateSystem()` function transforms the orientation output (`inR`) to a transformed output (`outR`), associated with a two dimensional movement in a plane that is parallel to the screen of a **display device**.

```
    public static boolean remapCoordinateSystem(float[] inR, int X, int Y,
            float[] outR)
    {
        if (inR == outR) {
            final float[] temp = mTempMatrix;
            synchronized(temp) {
                // we don't expect to have a lot of contention
                if (remapCoordinateSystemImpl(inR, X, Y, temp)) {
                    final int size = outR.length;
                    for (int i=0 ; i<size ; i++)
                        outR[i] = temp[i];
                    return true;
                }
            }
        }
        return remapCoordinateSystemImpl(inR, X, Y, outR);
    }
```

Source: https://android.googlesource.com/platform/frameworks/base/+/master/core/java/android/hardware/SensorManager.java

```
boolean remapCoordinateSystem (float[] inR,
                int X,
                int Y,
                float[] outR)
```

Rotates the supplied rotation matrix so it is expressed in a different coordinate system. This is typically used when an application needs to compute the three orientation angles of the device (see `getOrientation(float[], float[])`) in a different coordinate system.

When the rotation matrix is used for drawing (for instance with OpenGL ES), it usually **doesn't need** to be transformed by this function, unless the screen is physically rotated, in which case you can use `Display.getRotation()` to retrieve the current rotation of the screen. Note that because the user is generally free to rotate their screen, you often should consider the rotation in deciding the parameters to use here.

Source: http://developer.android.com/reference/android/hardware/SensorManager.html

Claim 10

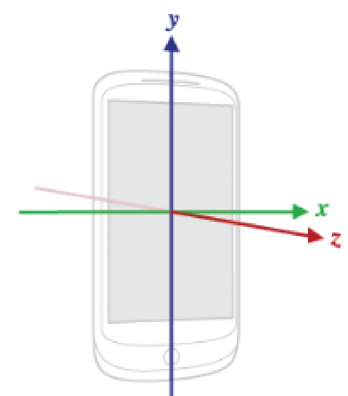
wherein the orientation output and the rotation output is generated by a **nine-axis motion sensor module**;

The LG V20 includes a 3-axis gyroscope, a 3-axis accelerometer, and a 3-axis magnetometer which form a **nine-axis motion sensor module.**

Sensor Coordinate System

In general, the sensor framework uses a standard 3-axis coordinate system to express data values. For most sensors, the coordinate system is defined relative to the device's screen when the device is held in its default orientation (see figure 1). When a device is held in its default orientation, the X axis is horizontal and points to the right, the Y axis is vertical and points up, and the Z axis points toward the outside of the screen face. In this system, coordinates behind the screen have negative Z values. This coordinate system is used by the following sensors:

- Acceleration sensor
- Gravity sensor
- Gyroscope
- Linear acceleration sensor
- Geomagnetic field sensor



Source: http://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords

Claim 10
obtaining one or more resultant deviation including a plurality of deviation angles using a plurality of **measured magnetisms Mx, My, Mz** and a plurality of predicted magnetism Mx′, My′ and Mz′ for the second signal set.

The **measured magnetisms Mx, My, Mz** are `values[0]-[2]`.

`Sensor.TYPE_MAGNETIC_FIELD_UNCALIBRATED`:

Similar to `TYPE_MAGNETIC_FIELD`, but the hard iron calibration is reported separately instead of being included in the measurement. Factory calibration and temperature compensation will still be applied to the "uncalibrated" measurement. Assumptions that the magnetic field is due to the Earth's poles is avoided.

The values array is shown below:

- values[0] = x_uncalib
- values[1] = y_uncalib
- values[2] = z_uncalib
- values[3] = x_bias
- values[4] = y_bias
- values[5] = z_bias

x_uncalib, y_uncalib, z_uncalib are the measured magnetic field in X, Y, Z axes. Soft iron and temperature calibrations are applied. But the hard iron calibration is not applied. The values are in micro-Tesla (uT).

x_bias, y_bias, z_bias give the iron bias estimated in X, Y, Z axes. Each field is a component of the estimated hard iron calibration. The values are in micro-Tesla (uT).

Hard iron - These distortions arise due to the magnetized iron, steel or permanenet magnets on the device. Soft iron - These distortions arise due to the interaction with the earth's magentic field.

Source: http://developer.android.com/reference/android/hardware/SensorEvent.html#values

Claim 10
obtaining one or more resultant deviation including a plurality of deviation angles using a plurality of **measured magnetisms Mx, My, Mz** and a plurality of **predicted magnetism** Mx′, My′ and Mz′ for the second signal set.
The **measured magnetisms**, z, and a **predicted magnetism**, Bb, are used to calculate a global variable x0 in quaternion form.

measured magnetisms

```
        update(m, Bm, mParam.magStdev);
```

```
void Fusion::update(const vec3_t& z, const vec3_t& Bi, float sigma) {
    const vec3_t Bb(A*Bi);
```

```
    const vec3_t e(z - Bb);
    const vec3_t dq(K[0]*e);

    q += getF(q)*(0.5f*dq);
    x0 = normalize_quat(q);
```

predicted magnetism

Source: https://android.googlesource.com/platform/frameworks/native/+/master/services/sensorservice/Fusion.cpp

Claim 10

obtaining one or more **resultant deviation including a plurality of deviation angles** using a plurality of measured magnetisms Mx, My, Mz and a plurality of predicted magnetism Mx′, My′ and Mz′ for the second signal set. .

The global variable `x0` is in quaternion form, and can easily be converted to resultant angles.

According to Android's developer library, the `getOrientation()` function "computes the device's orientation based on the rotation matrix," and returns **deviation angles** including the Azimuth, Pitch, and Roll angles.

getOrientation

Added in API level 3

```
float[] getOrientation (float[] R,
                float[] values)
```

Computes the device's orientation based on the rotation matrix.

When it returns, the array values are as follows:

- values[0]: *Azimuth*, angle of rotation about the -z axis. This value represents the angle between the device's y axis and the magnetic north pole. When facing north, this angle is 0, when facing south, this angle is π. Likewise, when facing east, this angle is π/2, and when facing west, this angle is -π/2. The range of values is -π to π.
- values[1]: *Pitch*, angle of rotation about the x axis. This value represents the angle between a plane parallel to the device's screen and a plane parallel to the ground. Assuming that the bottom edge of the device faces the user and that the screen is face-up, tilting the top edge of the device toward the ground creates a positive pitch angle. The range of values is -π to π.
- values[2]: *Roll*, angle of rotation about the y axis. This value represents the angle between a plane perpendicular to the device's screen and a plane perpendicular to the ground. Assuming that the bottom edge of the device faces the user and that the screen is face-up, tilting the left edge of the device toward the ground creates a positive roll angle. The range of values is -π/2 to π/2.

The `getRotationMatrixFromVector()` function "convert[s] a rotation vector to a rotation matrix," and the `getQuaternionFromVector()` function "convert[s] a rotation vector to a normalized quaternion." Therefore, the quaternion, `x0`, can be easily converted to its mathematically equivalent form, rotation matrix, and used by `getOrientation()` function to compute the orientation in its angular form.

Source: https://android.googlesource.com/platform/frameworks/base/+/b267554/core/java/android/hardware/SensorManager.java

Claim 10

obtaining one or more **resultant deviation including a plurality of deviation angles** using a plurality of measured magnetisms Mx, My, Mz and a plurality of predicted magnetism Mx′, My′ and Mz′ for the second signal set. .

Rotation vector

Underlying physical sensors: Accelerometer, Magnetometer, and Gyroscope

Reporting-mode: *Continuous*

`getDefaultSensor(SENSOR_TYPE_ROTATION_VECTOR)` *returns a non-wake-up sensor*

public static void **getRotationMatrixFromVector** (float[] R, float[] rotationVector)

Added in API level 9

Helper function to convert a rotation vector to a rotation matrix. Given a rotation vector (presumably from a ROTATION_VECTOR sensor), returns a 9 or 16 element rotation matrix in the array R. R must have length 9 or 16. If R.length == 9, the following matrix is returned:

public static float[] **getOrientation** (float[] R, float[] values)

Added in API level 3

Computes the device's orientation based on the rotation matrix.

When it returns, the array values is filled with the result:

- values[0]: *azimuth*, rotation around the -Z axis, i.e. the opposite direction of Z axis.
- values[1]: *pitch*, rotation around the -X axis, i.e the opposite direction of X axis.
- values[2]: *roll*, rotation around the Y axis.

Source: https://developer.android.com/reference/android/hardware/SensorManager.html#getOrientation(float[], float[])