tion for a plurality of objects, including at least one of text objects, graphic layout objects, or graphic image objects included in the Web page;

defining a primary datum corresponding to the original page layout; and,

for each object,

defining an object datum corresponding to the layout location for the object;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to the vector that is generated.

123. The method of claim **118**, further comprising enabling the Web page to be displayed at different resolutions by scaling the scalable content to re-render the display in response to associated user inputs,

wherein the original page layout, functionality, and design of the Web page content are preserved at each of the different resolutions.

124. The method of claim **118**, further comprising returning the display of the Web page to a previous view in response to a corresponding user input.

125. The method of claim **118**, further comprising enabling a user to pan a view of the Web page in response to a corresponding user input.

126. The method of claim **125**, further comprising enabling the view of the Web page to be panned in real-time.

127. The method of claim **118**, wherein the page layout of the Web page is defined to have an original aspect ratio, and wherein the scalable content is scaled when rendered so as to produce a display of the Web page having a different aspect ratio.

128. The method of claim **118**, further comprising enabling a user to zoom on a column of the Web page via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is rendered to fit across the display.

129. The method of claim **128**, wherein the corresponding user input comprises tapping on the column via the display.

130. The method of claim **128**, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

131. The method of claim **118**, wherein the Web content includes at least one image, the method further comprising enabling a user to zoom on an image via a corresponding user input, wherein in response thereto, the display is re-rendered such that the image is rendered to fit across the display.

132. The method of claim **131**, wherein the corresponding user input comprises tapping on the image via the display.

133. The method of claim **118**, further comprising enabling a user to zoom on a paragraph of the Web content via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is rendered to fit across the display.

134. The method of claim **133**, wherein the corresponding user input comprises tapping on the paragraph via the display.

135. The method of claim **133**, wherein the content of the paragraph is reformatted to fit characteristics of the display when the display is re-rendered.

136. The method of claim **118**, wherein the Web page includes text, layout attributes, and images, the method further comprising:

receiving content corresponding to the text and layout attributes via a first connection; and

receiving content corresponding to at least one image via a second connection.

137. The method of claim **118**, further comprising:

generating a vector-based display list associated with the scalable content; and

employing the display list to re-render the display at different scale factors to zoom the Web page.

138. The method of claim **118**, further comprising:

parsing markup language code corresponding to the retrieved Web page to determine the original page layout of the content on the Web page;

logically grouping selected content into objects;

defining a primary datum corresponding to the original page layout; and,

for each object,

defining an object datum corresponding to a layout location datum for the object's associated display content;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to the vector that is generated.

139. The method of claim **138**, further comprising:

generating a bounding box for each object, the bounding box representing a portion of a rendered display page occupied by the object's associated group of content.

140. The method of claim **139**, further comprising:

mapping the object vectors and associated bounding boxes to a virtual display in memory.

141. The method of claim **140**, further comprising:

enabling a user to view the Web page at a user-selectable zoom level and panned view by,

determining a first scale factor and offset in response to one or more corresponding user inputs defining a user-selectable zoom level and panned view corresponding to a rendered display of the Web page desired by a user; and

determining a virtual display bounding box for the virtual display associated with the first scale factor and offset;

identifying object bounding boxes having at least a portion falling within the virtual display bounding box; and,

for each of such object bounding boxes,

retrieving content associated with that object bounding box;

applying an appropriate scale factor to the content associated with that object bounding box to produce scaled content; and

rendering the portion of scaled content within the virtual display bounding box to render the content on the display.

142. The method of claim **118**, wherein the scalable content includes scalable text content, the method further comprising scaling a scalable font to render the scalable text content.

143. The method of claim **118**, wherein the method is facilitated, at least in part, via execution of Java-based instructions.

144. The method of claim **118**, wherein the device comprises a mobile phone.

145. The method of claim **118**, wherein the device comprises a hand-held device.

146. The method of claim **118**, further comprising accessing the Internet via a wireless connection to retrieve the Web page.

147. The method of claim **118**, wherein a portion of the scalable content comprises vector-based content.

148. The method of claim 118, wherein the device comprises one of a desktop computer, notebook computer or laptop computer.

149. A method, comprising:

rendering a browser interface on a hand-held device via which a user is enabled to request access to a Web page comprising HTML-based Web content defining an original page layout, functionality, and design of content on the Web page;

retrieving the Web page via the hand-held device, and processing HTML-based Web content to produce scalable content; and

employing at least one of the scalable content or data derived therefrom to,

render the Web page on a display of the hand-held device; and

re-render the display in response to associated user inputs to enable the Web page to be browsed at various zoom levels and panned views while preserving the original page layout, functionality, and design of the Web page content at each zoom level and panned view.

150. The method of claim 149, wherein the hand-held device comprises a mobile phone.

151. The method of claim 149, wherein the hand-held device comprises one of a Personal Digital Assistant (PDA) or hand-held computer.

152. The method of claim 149, further comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input.

153. The method of claim 152, wherein the user interface input enables the user to define an area of a current view of the Web page on which to zoom in on.

154. The method of claim 149, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

155. The method of claim 149, wherein the Web page includes at least one hyperlink, the method further comprising:

enabling the user to select the hyperlink via the display; and, in response thereto,

retrieving and processing HMTL-based Web content associated with the hyperlink to produce additional scalable content; and

employing at least one of the additional scalable content or data derived therefrom to render the Web content associated with the hyperlink on the display.

156. The method of claim 149, wherein at least a portion of the scalable content comprises scalable vector-based content.

157. The method of claim 149, further comprising returning the display of the Web page to a previous view in response to a corresponding user input made via the display.

158. The method of claim 149, further comprising enabling a user to pan a view of the Web page in response to a corresponding user input made via the display.

159. The method of claim 149, further comprising enabling the view of the Web page to be panned in real-time.

160. The method of claim 149, wherein the page layout of the Web page is defined to have an original aspect ratio, and wherein said at least one of scalable content or data derived therefrom is scaled to render a display of the Web page having a different aspect ratio.

161. The method of claim 149, further comprising enabling a user to zoom on a column of the Web content via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is enlarged.

162. The method of claim 161, wherein the corresponding user input comprises tapping on the column via the display.

163. The method of claim 161, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

164. The method of claim 161, wherein the display is re-rendered such that content corresponding to the selected column is rendered to fit across the display.

165. The method of claim 149, wherein the Web content includes at least one image, the method further comprising enabling a user to zoom on an image via a corresponding user input, wherein in response thereto, the display is re-rendered such that the image is enlarged.

166. The method of claim 165, wherein the corresponding user input comprises tapping on the image via the display.

167. The method of claim 165, wherein the display is re-rendered such that the image is rendered to fit across the display.

168. The method of claim 149, further comprising enabling a user to zoom on a paragraph of the Web page via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is enlarged.

169. The method of claim 168, wherein the content of the paragraph is reformatted to fit characteristics of the display area when the display is re-rendered.

170. The method of claim 168, wherein the display is re-rendered such that content corresponding to the selected paragraph is rendered to fit across the display.

171. The method of claim 168, wherein the corresponding user input comprises tapping on the paragraph via the display.

172. The method of claim 149, wherein the Web page includes text, layout attributes, and images, the method further comprising:

receiving content corresponding to the text and layout attributes via a first connection; and

receiving content corresponding to at least one image via a second connection.

173. The method of claim 149, wherein the hand-held device includes dynamic memory having at least a portion employed for rendering purposes, the method further comprising:

building a display list via use of the scalable content and rendering display list content on a virtual display in dynamic memory; and

scaling the display list content to re-render the display of the Web page.

174. The method of claim 149, further comprising:

parsing HTML-based code corresponding to the retrieved Web page to determine the original page layout of the content on the Web page;

logically grouping selected content into objects;

defining a primary datum corresponding to the original page layout; and,

for each object,

defining an object datum corresponding to a layout location datum for the object's associated display content;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to the vector that is generated.

175. The method of claim 174, further comprising:

generating a bounding box for each object, the bounding box representing a portion of a rendered display page occupied by the object's associated group of content.

176. The method of claim 175, wherein the hand-held device includes dynamic memory having at least a portion employed for rendering purposes, the method further comprising:

mapping the object vectors and associated bounding boxes to a virtual display in the dynamic memory.

177. The method of claim 176, further comprising:

enabling a user to view the Web page at a user-selectable zoom level and panned view by,

determining a first scale factor and offset in response to one or more corresponding user inputs defining a user-selectable zoom level and panned view corresponding to a rendered display of the Web page desired by a user;

determining a virtual display bounding box for the virtual display associated with the first scale factor and offset;

identifying object bounding boxes having at least a portion falling within the virtual display bounding box; and,

for each of such object bounding boxes,

retrieving content associated with that object bounding box;

applying an appropriate scale factor to the content associated with that object bounding box to produce scaled content; and

rendering the portion of scaled content within the virtual display bounding box to render the content on the display.

178. The method of claim 149, wherein the scalable content includes scalable text content, the method further comprising scaling a scalable font to render the scalable text content.

179. The method of claim 149, wherein the original format of the Web page defines a height and width for the Web page, the method further comprising:

determining an applicable scale factor to display at least one of the width and height of the Web page across a browser display area of the display; and

employing the scale factor to render the browser display area.

180. The method of claim 149, wherein the method is facilitated, at least in part, via execution of Java-based instructions.

181. The method of claim 149, wherein a portion of the HTML-based Web content comprises XML code.

182. The method of claim 149, wherein a portion of the HTML-based Web content comprises cascaded style sheet data defining aspects of the Web page design that are preserved at each zoom level and panned view.

183. The method of claim 149, wherein a portion of the scalable content comprises vector-based content.

184. The method of claim 149, further comprising enabling a user to browse, zoom, and pan billions of Web pages in a manner that preserves the original layout, functionality, and design of the HTML-based Web page content of each Web page.

185. A method, comprising:

rendering a browser interface on a display of a device via which a user is enabled to request access to a Web page comprising HTML-based Web content defining an original page layout, functionality, and design of content on the Web page;

in response to a user request of the Web page via the browser interface,

retrieving the Web page via the device;

rendering the Web page via the device such that a full width of the Web page is rendered on the display; and

re-rendering the Web page in response to associated user inputs to the hand-held device to enable the Web page to be browsed at various zoom levels and panned views while preserving the original page layout, functionality, and design of the Web page content at each zoom level and panned view of the Web page,

wherein the method enables a user of the device to browse, zoom, and pan billions of Web pages in a manner that preserves the original layout, functionality, and design of the HTML-based Web page content of each Web page.

186. The method of claim 185, further comprising enabling a user to zoom on a column of the Web page via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is enlarged.

187. The method of claim 186, wherein the corresponding user input comprises tapping on the column via the display.

188. The method of claim 186, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

189. The method of claim 186, wherein the display is re-rendered such that content corresponding to the selected column is rendered to fit across the display.

190. The method of claim 185, wherein the Web page includes at least one image, the method further comprising enabling a user to zoom on an image via a corresponding user input, wherein in response thereto, the display is re-rendered such that the image is enlarged.

191. The method of claim 190, wherein the corresponding user input comprises tapping on the image via the display.

192. The method of claim 190, wherein the display is re-rendered such that the image is rendered to fit across the display.

193. The method of claim 185, further comprising enabling a user to zoom on a paragraph of the Web page via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is enlarged.

194. The method of claim 193, wherein the corresponding user input comprises tapping on the paragraph via the display.

195. The method of claim 193, wherein the content of the paragraph is reformatted to fit characteristics of the display when re-rendered.

196. The method of claim 193, wherein the display is re-rendered such that content corresponding to the selected paragraph is rendered to fit across the display.

197. The method of claim 185, further comprising enabling a user to pan a view of the web page while in a zoomed state under which a portion of the web page is displayed in response to a user input made via the display.

198. The method of claim 185, further comprising returning the display of the Web page to a previous view in response to a corresponding user input.

199. The method of claim 185, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

200. The method of claim 185, further comprising enabling a user to pan a view of the Web page in response to a corresponding user input made via the display.

201. The method of claim 185, further comprising enabling the view of the Web page to be panned in real-time.

202. The method of claim 185, wherein the Web page includes text, layout attributes, and images, the method further comprising:

receiving content corresponding to the text and layout attributes via a first connection; and

receiving content corresponding to at least one image via a second connection.

203. The method of claim 185, wherein a portion of the HTML-based Web content comprises XML code.

204. The method of claim 185, wherein a portion of the HTML-based Web content comprises cascaded style sheet data defining aspects of the Web page design that are preserved at each zoom level and panned view.

205. The method of claim 185, wherein the Web page is retrieved via a wireless connection to one of a mobile service provider network, local area network, or wide area network.

206. The method of claim 185, wherein the device comprises a mobile phone.

207. The method of claim 185, wherein the device comprises a hand-held device.

208. The method of claim 185, wherein the device comprises one of a desktop computer, notebook computer or laptop computer.

209. A method, comprising:

rendering a browser interface on a display via which a user of a device is enabled to request access to a Web page, the Web page comprising HTML-based Web content having an original format including HTML code defining an original page layout, functionality, and design of corresponding content on the Web page;

retrieving the Web page, via the device, and translating at least a portion of the HTML-based Web content into scalable content that supports a scalable resolution-independent representation of the Web page that preserves the original page layout, functionality and design of the content defined by its original format when scaled and rendered; and

employing the scalable content to render the Web page on the display using a first scale factor; and

enabling the Web page to be displayed at a different resolution by scaling the scalable content using a second scale factor to re-render the display,

wherein the original page layout, functionality, and design of the Web page content are preserved under both the first and second scale factors.

210. The method of claim 209, wherein the display is re-rendered in real-time.

211. The method of claim 209, wherein the device comprises a hand-held device.

212. The method of claim 209, wherein the device comprises one of a desktop computer, notebook computer or laptop computer.

213. The method of claim 209, further comprising enabling a user to pan a view of the Web page in response to a corresponding user input.

214. The method of claim 213, further comprising enabling the view of the Web page to be panned in real-time.

215. The method of claim 209, further comprising enabling a user to browse, zoom, and pan billions of Web pages in a manner that preserves the original layout, functionality, and design of the HTML-based Web page content of each Web page.

216. A machine-readable medium having a plurality of instructions tangibly stored thereon, which when executed enable a device to perform operations comprising:

rendering a browser interface via which a user is enabled to request access to a Web page hosted by an Internet Web site, the Web page comprising HTML-based Web content having an original format defining an original width and height of the Web page and an original page layout, functionality, and design of content on the Web page;

retrieving the Web page via the wireless communication means, and translating at least a portion of the HTML-based Web content from its original format into scalable content that supports a scalable resolution-independent representation of the Web page that preserves the original page layout, functionality and design of the content defined by its original format when scaled and rendered; and

scaling the scalable content to render the Web page on the display such that a width of the Web page is rendered to fit across the display.

217. The machine-readable medium of claim 216, wherein execution of the instructions performs further operations comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a corresponding user interface input.

218. The machine-readable medium of claim 217, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

219. The machine-readable medium of claim 216, wherein the Web page includes at least one hyperlink, and wherein execution of the instructions performs further operations comprising:

enabling the user to select the hyperlink; and, in response thereto,

retrieving and translating Web content associated with the hyperlink to produce additional scalable content; and

employing the additional scalable content to render the Web content associated with the hyperlink on the display.

220. The machine-readable medium of claim 216, wherein execution of the instructions performs further operations comprising:

parsing markup language code to determine the original page layout of display content within the Web page, wherein the original page layout defines a layout location for a plurality of objects, including at least one of text objects, graphic layout objects, or graphic image objects included in the Web page;

defining a primary datum corresponding to the original page layout; and,

for each object,

defining an object datum corresponding to the layout location for the object;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to the vector that is generated.

221. The machine-readable medium of claim 216, wherein execution of the instructions performs further operations comprising enabling the Web page to be displayed at different resolutions by scaling the scalable content to re-render the display in response to associated user inputs,

wherein the original page layout, functionality, and design of the Web page content are preserved at each of the different resolutions.

222. The machine-readable medium of claim 216, wherein execution of the instructions performs further operations comprising returning the display of the Web page to a previous view in response to a corresponding user input.

223. The machine-readable medium of claim 216, wherein execution of the instructions performs further operations

comprising enabling a user to pan a view of the Web page in response to a corresponding user input.

224. The machine-readable medium of claim **223**, wherein execution of the instructions performs further operations comprising enabling the view of the Web page to be panned in real-time.

225. The machine-readable medium of claim **216**, wherein the page layout of the Web page is defined to have an original aspect ratio, and wherein the scalable content is scaled when rendered so as to produce a display of the Web page having a different aspect ratio.

226. The machine-readable medium of claim **216**, wherein execution of the instructions performs further operations comprising enabling a user to zoom on a column of the Web page via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is enlarged.

227. The machine-readable medium of claim **226**, wherein the corresponding user input comprises tapping on the column via the display.

228. The machine-readable medium of claim **226**, wherein the display is re-rendered such that content corresponding to the selected column is rendered to fit across the display.

229. The machine-readable medium of claim **226**, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

230. The machine-readable medium of claim **216**, wherein the Web content includes at least one image, and wherein execution of the instructions performs further operations comprising enabling a user to zoom on an image via a corresponding user input, wherein in response thereto, the display is re-rendered such that the image is enlarged.

231. The machine-readable medium of claim **230**, wherein the corresponding user input comprises tapping on the image via the display.

232. The machine-readable medium of claim **230**, wherein the display is re-rendered such that the image is rendered to fit across the display.

233. The machine-readable medium of claim **216**, wherein execution of the instructions performs further operations comprising enabling a user to zoom on a paragraph of the Web content via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is enlarged.

234. The machine-readable medium of claim **233**, wherein the corresponding user input comprises tapping on the paragraph via the display.

235. The machine-readable medium of claim **233**, wherein the content of the paragraph is reformatted to fit characteristics of the display when the display is re-rendered.

236. The machine-readable medium of claim **233**, wherein the display is re-rendered such that content corresponding to the selected paragraph is rendered to fit across the display.

237. The machine-readable medium of claim **216**, wherein the Web page includes text, layout attributes, and images, and wherein execution of the instructions performs further operations comprising:

receiving content corresponding to the text and layout attributes via a first connection; and

receiving content corresponding to at least one image via a second connection.

238. The machine-readable medium of claim **216**, wherein execution of the instructions performs further operations comprising:

generating a vector-based display list associated with the scalable content; and

employing the display list to re-render the display at different scale factors to zoom the Web page.

239. The machine-readable medium of claim **216**, wherein execution of the instructions performs further operations comprising:

parsing markup language code corresponding to the retrieved Web page to determine the original page layout of the content on the Web page;

logically grouping selected content into objects;

defining a primary datum corresponding to the original page layout; and,

for each object,

defining an object datum corresponding to a layout location datum for the object's associated display content;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to the vector that is generated.

240. The machine-readable medium of claim **239**, wherein execution of the instructions performs further operations comprising:

generating a bounding box for each object, the bounding box representing a portion of a rendered display page occupied by the object's associated group of content.

241. The machine-readable medium of claim **240**, wherein execution of the instructions performs further operations comprising:

mapping the object vectors and associated bounding boxes to a virtual display in memory.

242. The machine-readable medium of claim **241**, wherein execution of the instructions performs further operations comprising:

enabling a user to view the Web page at a user-selectable zoom level and panned view by,

determining a first scale factor and offset in response to one or more corresponding user inputs defining a user-selectable zoom level and panned view corresponding to a rendered display of the Web page desired by a user; and

determining a virtual display bounding box for the virtual display associated with the first scale factor and offset;

identifying object bounding boxes having at least a portion falling within the virtual display bounding box; and,

for each of such object bounding boxes,

retrieving content associated with that object bounding box;

applying an appropriate scale factor to the content associated with that object bounding box to produce scaled content; and

rendering the portion of scaled content within the virtual display bounding box to render the content on the display.

243. The machine-readable medium of claim **216**, wherein the scalable content includes scalable text content, and wherein execution of the instructions performs further operations comprising scaling a scalable font to render the scalable text content.

244. The machine-readable medium of claim **216**, wherein at least a portion of the instructions comprise Java-based instructions.

245. The machine-readable medium of claim **216**, wherein the device comprises a mobile phone.

246. The machine-readable medium of claim **216**, wherein the device comprises a hand-held device.

247. The machine-readable medium of claim 216, wherein the Web page is accessed via a mobile service provider network.

248. The machine-readable medium of claim 216, wherein a portion of the scalable content comprises vector-based content.

249. The machine-readable medium of claim 216, wherein the device comprises one of a desktop computer, notebook computer or laptop computer.

250. The machine-readable medium of claim 216, wherein the instructions are embodied as a Web browser.

251. The machine-readable medium of claim 216, wherein execution of the instructions enables a user to browse, zoom, and pan billions of Web pages in a manner that preserves the original layout, functionality, and design of the HTML-based Web page content of each Web page.

252. A machine-readable medium having a plurality of instructions comprising a Web browser tangibly stored thereon, which when executed enable a device to perform operations comprising:

rendering a browser interface on a display associated with the device via which a user is enabled to request access to a Web page comprising HTML-based Web content defining an original page layout, functionality, and design of content on the Web page;

retrieving the Web page and processing HTML-based Web content to produce scalable content; and

employing at least one of the scalable content or data derived therefrom to,

render the Web page on the display; and

re-render the display in response to associated user inputs to enable the Web page to be browsed at various zoom levels and panned views while preserving the original page layout, functionality, and design of the Web page content at each zoom level and panned view of the Web page,

wherein the Web browser enables a user of the device to browse, zoom, and pan billions of Web pages in a manner that preserves the original layout, functionality, and design of the HTML-based Web page content of each Web page at each zoom level and panned view.

253. The machine-readable medium of claim 252, wherein the device comprises a mobile phone.

254. The machine-readable medium of claim 252, wherein the device comprises a hand-held device.

255. The machine-readable medium of claim 252, wherein execution of the instructions performs further operations comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input.

256. The machine-readable medium of claim 255, wherein the user interface input enables the user to define an area of a current view of the Web page on which to zoom in on.

257. The machine-readable medium of claim 252, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

258. The machine-readable medium of claim 252, wherein the Web page includes at least one hyperlink, and wherein execution of the instructions performs further operations comprising:

enabling the user to select the hyperlink; and, in response thereto,

retrieving and processing HMTL-based Web content associated with the hyperlink to produce additional scalable content; and

employing at least one of the additional scalable content or data derived therefrom to render the Web content associated with the hyperlink on the display.

259. The machine-readable medium of claim 252, wherein at least a portion of the scalable content comprises scalable vector-based content.

260. The machine-readable medium of claim 252, wherein execution of the instructions performs further operations comprising returning the display of the Web page to a previous view in response to a corresponding user input made via the display.

261. The machine-readable medium of claim 252, wherein execution of the instructions performs further operations comprising enabling a user to pan a view of the Web page in response to a corresponding user input made via the display.

262. The machine-readable medium of claim 252, wherein execution of the instructions performs further operations comprising enabling the view of the Web page to be panned in real-time.

263. The machine-readable medium of claim 252, wherein the page layout of the Web page is defined to have an original aspect ratio, and wherein said at least one of scalable content or data derived therefrom is scaled to render a display of the Web page having a different aspect ratio.

264. The machine-readable medium of claim 252, wherein execution of the instructions performs further operations comprising enabling a user to zoom on a column of the Web content via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is rendered to fit across the display.

265. The machine-readable medium of claim 264, wherein the corresponding user input comprises tapping on the column via the display.

266. The machine-readable medium of claim 264, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

267. The machine-readable medium of claim 252, wherein the Web content includes at least one image, and wherein execution of the instructions performs further operations comprising enabling a user to zoom on an image via a corresponding user input, wherein in response thereto, the display is re-rendered such that the image is rendered to fit across the display.

268. The machine-readable medium of claim 267, wherein the corresponding user input comprises tapping on the image via the display.

269. The machine-readable medium of claim 252, wherein execution of the instructions performs further operations comprising enabling a user to zoom on a paragraph of the Web page via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is rendered to fit across a browser display area of the display.

270. The machine-readable medium of claim 269, wherein the corresponding user input comprises tapping on the paragraph via the display.

271. The machine-readable medium of claim 269, wherein the content of the paragraph is reformatted to fit characteristics of the display area when the display is re-rendered.

272. The machine-readable medium of claim 252, wherein the Web page includes text, layout attributes, and images, and wherein execution of the instructions performs further operations comprising:

receiving content corresponding to the text and layout attributes via a first connection; and

receiving content corresponding to at least one image via a second connection.

273. The machine-readable medium of claim **252**, wherein the device includes dynamic memory having at least a portion employed for rendering purposes, and wherein execution of the instructions performs further operations comprising:

building a display list via use of the scalable content and rendering display list content on a virtual display in the dynamic memory; and

scaling the display list content to re-render the display of the Web page.

274. The machine-readable medium of claim **252**, wherein execution of the instructions performs further operations comprising:

parsing HTML-based code corresponding to the retrieved Web page to determine the original page layout of the content on the Web page;

logically grouping selected content into objects;

defining a primary datum corresponding to the original page layout; and,

for each object,

defining an object datum corresponding to a layout location datum for the object's associated display content;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to the vector that is generated.

275. The machine-readable medium of claim **274**, wherein execution of the instructions performs further operations comprising:

generating a bounding box for each object, the bounding box representing a portion of a rendered display page occupied by the object's associated group of content.

276. The machine-readable medium of claim **275**, wherein the device includes dynamic memory having at least a portion employed for rendering purposes, and wherein execution of the instructions performs further operations comprising:

mapping the object vectors and associated bounding boxes to a virtual display in the dynamic memory.

277. The machine-readable medium of claim **276**, wherein execution of the instructions performs further operations comprising:

enabling a user to view the Web page at a user-selectable zoom level and panned view by,

determining a first scale factor and offset in response to one or more corresponding user inputs defining a user-selectable zoom level and panned view corresponding to a rendered display of the Web page desired by a user;

determining a virtual display bounding box for the virtual display associated with the first scale factor and offset;

identifying object bounding boxes having at least a portion falling within the virtual display bounding box; and,

for each of such object bounding boxes,

retrieving content associated with that object bounding box;

applying an appropriate scale factor to the content associated with that object bounding box to produce scaled content; and

rendering the portion of scaled content within the virtual display bounding box to render the content on the display.

278. The machine-readable medium of claim **252**, wherein the scalable content includes scalable text content, and wherein execution of the instructions performs further operations comprising scaling a scalable font to render the scalable text content.

279. The machine-readable medium of claim **252**, wherein the original format of the Web page defines a height and width for the Web page, and wherein execution of the instructions performs further operations comprising:

determining an applicable scale factor to display at least one of the width and height of the Web page across a browser display area of the display; and

employing the scale factor to render the browser display area.

280. The machine-readable medium of claim **252**, wherein at least a portion of the instructions comprise Java-based instructions.

281. The machine-readable medium of claim **252**, wherein a portion of the HTML-based Web content comprises XML code.

282. The machine-readable medium of claim **252**, wherein a portion of the HTML-based Web content comprises cascaded style sheet data defining aspects of the Web page design that are preserved at each zoom level and panned view.

283. The machine-readable medium of claim **252**, wherein a portion of the scalable content comprises vector-based content.

284. The machine-readable medium of claim **252**, wherein the device comprises one of a desktop computer, notebook computer or laptop computer.

285. A machine-readable medium having a plurality of instructions tangibly stored thereon, which when executed enable a wireless device to perform operations comprising:

rendering a browser interface on a display of the wireless device via which a user is enabled to request access to a Web page comprising HTML-based Web content defining an original page layout, functionality, and design of content on the Web page;

in response to a user request of the Web page,

retrieving the Web page via the wireless device;

rendering the Web page such that a width of the Web page is rendered to fit across the display; and

re-rendering the Web page in response to associated user inputs to enable the Web page to be browsed at various zoom levels and panned views while preserving the original page layout, functionality, and design of the Web page content at each zoom level and panned view.

286. The machine-readable medium of claim **285**, wherein execution of the instructions performs further operations comprising enabling a user to zoom on a column of the Web page via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is enlarged.

287. The machine-readable medium of claim **286**, wherein the corresponding user input comprises tapping on the column via the display.

288. The machine-readable medium of claim **286**, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

289. The machine-readable medium of claim **286**, wherein the display is re-rendered such that content corresponding to the selected column is rendered to fit across the display.

290. The machine-readable medium of claim **285**, wherein the Web page includes at least one image, and wherein execution of the instructions performs further operations comprising enabling a user to zoom on an image via a corresponding user input, wherein in response thereto, the display is re-rendered such that the image is enlarged.

291. The machine-readable medium of claim 290, wherein the corresponding user input comprises tapping on the image via the display.

292. The machine-readable medium of claim 290, wherein the display is re-rendered such that the image is rendered to fit across the display.

293. The machine-readable medium of claim 285, wherein execution of the instructions performs further operations comprising enabling a user to zoom on a paragraph of the Web page via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is enlarged.

294. The machine-readable medium of claim 293, wherein the corresponding user input comprises tapping on the paragraph via the display.

295. The machine-readable medium of claim 293, wherein the content of the paragraph is reformatted to fit characteristics of the display when re-rendered.

296. The machine-readable medium of claim 293, wherein the display is re-rendered such that content corresponding to the selected paragraph is rendered to fit across the display.

297. The machine-readable medium of claim 285, wherein execution of the instructions performs further operations comprising enabling a user to pan a display of the web page while in a zoomed state under which a portion of the web page is displayed.

298. The machine-readable medium of claim 285, wherein execution of the instructions performs further operations comprising returning the display of the Web page to a previous view in response to a corresponding user input.

299. The machine-readable medium of claim 285, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

300. The machine-readable medium of claim 285, wherein execution of the instructions performs further operations comprising enabling a user to pan a view of the Web page in response to a corresponding user input made via the display.

301. The machine-readable medium of claim 285, wherein execution of the instructions performs further operations comprising enabling the view of the Web page to be panned in real-time.

302. The machine-readable medium of claim 285, wherein the Web page includes text, layout attributes, and images, and wherein execution of the instructions performs further operations comprising:

receiving content corresponding to the text and layout attributes via a first connection; and

receiving content corresponding to at least one image via a second connection.

303. The machine-readable medium of claim 285, wherein a portion of the HTML-based Web content comprises XML code.

304. The machine-readable medium of claim 285, wherein a portion of the HTML-based Web content comprises cascaded style sheet data defining aspects of the Web page design that are preserved at each zoom level and panned view.

305. The machine-readable medium of claim 285, wherein the wireless device is configured to connect to a mobile service provider network and retrieve the Web page via the mobile service provider network.

306. The machine-readable medium of claim 285, wherein the wireless device comprises a mobile phone.

307. The machine-readable medium of claim 285, wherein the wireless device comprises a hand-held device.

308. The machine-readable medium of claim 285, wherein the wireless device comprises one of a notebook computer or laptop computer.

309. The machine-readable medium of claim 285, wherein the instructions are embodied as a Web browser.

310. The machine-readable medium of claim 285, wherein execution of the instructions enables a user to browse, zoom, and pan billions of Web pages in a manner that preserves the original layout, functionality, and design of the HTML-based Web page content of each Web page.

311. A machine-readable medium having a plurality of instructions comprising a Web browser stored thereon, which when executed enable a device to perform operations comprising:

launching a Web browser including a browser interface via which a user is enabled to request access to a Web page, the Web page comprising HTML-based Web content having an original format including HTML code defining an original page layout, functionality, and design of corresponding content on the Web page;

retrieving, and translating at least a portion of the HTML-based Web content into scalable content that supports a scalable resolution-independent representation of the Web page that preserves the original page layout, functionality and design of the content defined by its original format when scaled and rendered; and

employing the scalable content to render the Web page on the Web browser using a first scale factor; and

enabling the Web page to be displayed at a different resolution by scaling the scalable content using a second scale factor to re-render the Web page on the Web browser,

wherein the original page layout, functionality, and design of the Web page content are preserved under both the first and second scale factors, and

wherein the Web browser enables a user of the device to browse billions of Web pages at multiple scale factors in a manner that preserves the original layout, functionality, and design of the HTML-based Web page content of each Web page at each scale factor.

312. The machine-readable medium of claim 311, wherein the display is re-rendered in real-time.

313. The machine-readable medium of claim 311, wherein the Web browser is configured to be installed on a hand-held device.

314. The machine-readable medium of claim 311, wherein the Web browser is configured to be installed on at least one of a desktop computer, notebook computer or laptop computer.

315. The machine-readable medium of claim 311, wherein execution of the instructions performs further operations comprising enabling a user to pan a view of the Web page in response to a corresponding user input.

316. The machine-readable medium of claim 315, wherein execution of the instructions performs further operations comprising enabling the view of the Web page to be panned in real-time.

317. A hand-held wireless device, comprising:

a processor,

a wireless communications interface, to facilitate wireless communication with a network that supports access to the Internet;

a display; and

non-volatile memory, operatively coupled to the processor, in which software comprising a browser is stored, the browser comprising a plurality of instructions that when executed by the processor enable the device to perform operations including,

rendering a browser interface on the display via which a user is enabled to request access to a Web page includ-

ing at least one image, at least one column, at least one hyperlink to an external reference and having a width and height;

retrieving the Web page via the wireless communications interface;

rendering the Web page on the display such that at least one of the width and height of the Web page is fully displayed; and

enabling the user to,

zoom and pan a view of the Web page;

activate a currently displayed hyperlink to an external reference while at a given zoom level and panned view, wherein in response to an activation of a hyperlink to an external reference, Web content associated with the external reference is retrieved and rendered on the display;

zoom in on an image of the Web page by tapping on the image via the display;

zoom in on a column of the Web page by tapping on the column via the display; and

zoom out to a previous view of the Web page.

318. The hand-held wireless device of claim **317**, wherein the Web page comprises HTML-based Web page content defining an original page layout, functionality, and design of the Web page content, and wherein the browser renders the Web page such that the original page layout, functionality, and design of the Web page are preserved at any selectable zoom level.

319. The hand-held wireless device of claim **318**, wherein the user is enabled to browse, zoom, and pan billions of Web pages in a manner that preserves the original layout, functionality, and design of the HTML-based Web page content of each Web page.

* * * * *

EXHIBIT 2

U.S. Patent No. 7,831,926

US007831926B2

(12) **United States Patent**
Rohrabaugh et al.

(10) **Patent No.: US 7,831,926 B2**
(45) **Date of Patent: *Nov. 9, 2010**

(54) **SCALABLE DISPLAY OF INTERNET CONTENT ON MOBILE DEVICES**

(75) Inventors: **Gary B. Rohrabaugh**, Bellingham, WA (US); **Scott A. Sherman**, Seattle, WA (US)

(73) Assignee: **SoftView LLC**, Bellingham, WA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 799 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **11/738,486**

(22) Filed: **Apr. 21, 2007**

(65) **Prior Publication Data**

US 2007/0288841 A1 Dec. 13, 2007

Related U.S. Application Data

(63) Continuation of application No. 09/878,097, filed on Jun. 8, 2001, now Pat. No. 7,210,099, which is a continuation-in-part of application No. 09/828,511, filed on Apr. 7, 2001, now abandoned.

(60) Provisional application No. 60/211,019, filed on Jun. 12, 2000, provisional application No. 60/217,345, filed on Jul. 11, 2000.

(51) **Int. Cl.**
G06F 17/00 (2006.01)

(52) **U.S. Cl.** **715/800**; 715/234; 715/243; 715/853

(58) **Field of Classification Search** 715/800, 715/238, 249, 204, 234
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,615,384 A 3/1997 Allard et al.
5,897,644 A 4/1999 Blumberg
5,952,994 A 9/1999 Ong et al.
5,956,025 A 9/1999 Goulden et al.
5,966,135 A 10/1999 Roy et al.
6,011,905 A 1/2000 Huttenlocher et al.

(Continued)

FOREIGN PATENT DOCUMENTS

GB 0009129.8 4/2000
GB 2344197 A 5/2000
JP 10-334082 A 12/1998

OTHER PUBLICATIONS

Scalable Vector Graphics (SVG) Specification W3C Working Draft Feb. 11, 1999 http://www.w3.org/TR/1999/WD-SVG-19990211/--Pub Feb. 11, 1999 by W3C pp. 1-7.

Fisher, B., G. Agelidis, J. Dill, P. Tan, G. Collaud and C. Jones. "CZWeb: Fish-Eye Views for Visualiziang the World-Wide Web", Proc. Seventh Int. Conf. on Human-Computer Interaction (HCI International '97), pp. 719-722, 1997.

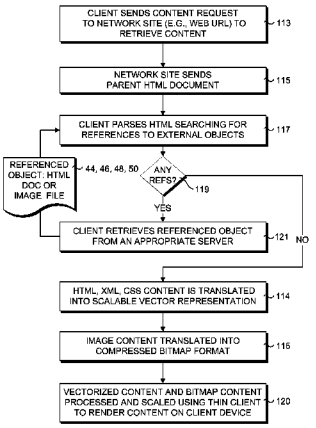
(Continued)

Primary Examiner—Doug Hutton
Assistant Examiner—Quoc A Tran
(74) *Attorney, Agent, or Firm*—Law Office of R. Alan Burnett

(57) **ABSTRACT**

Mobile devices enabled to support resolution-independent scalable display of Internet (Web) content to allow Web pages to be scaled (zoomed) and panned for better viewing on smaller screen sizes. The mobile devices employ software-based processing of original Web content, including HTML-based content, XML, cascade style sheets, etc. to generate scalable content. The scalable content and/or data derived therefrom are then employed to enable the Web content to be rapidly rendered, zoomed, and panned. Display lists may also be employed to provide further enhancements in rendering speed.

88 Claims, 22 Drawing Sheets



U.S. PATENT DOCUMENTS

6,057,854	A	5/2000	Davis, Jr. et al.
6,076,166	A	6/2000	Moshfeghi et al.
6,211,856	B1	4/2001	Choi et al.
6,300,947	B1	10/2001	Kanevsky
6,337,693	B1	1/2002	Roy et al.
6,421,733	B1	7/2002	Tso et al.
6,449,639	B1	9/2002	Blumberg
6,456,305	B1	9/2002	Qureshi et al.
6,456,308	B1	9/2002	Agranet et al.
6,457,030	B1	9/2002	Adams et al.
6,466,203	B2	10/2002	Van Ee
6,546,397	B1	4/2003	Rempell
6,553,410	B2	4/2003	Kikinis
6,556,217	B1	4/2003	Mäkipää
6,615,212	B1	9/2003	Dutta et al.
6,674,445	B1	1/2004	Chithambaram et al.
6,704,024	B2	3/2004	Robotham et al.
6,781,600	B2	8/2004	Anwar
6,829,746	B1	12/2004	Schwerdtfeger et al.
6,857,102	B1	2/2005	Bickmore et al.
6,886,034	B2	4/2005	Blumberg
6,925,595	B1	8/2005	Whitledge et al.
6,978,418	B1	12/2005	Bain et al.
6,996,533	B2	2/2006	Ikeda et al.
7,023,572	B2	4/2006	Tuli
7,055,095	B1	5/2006	Anwar
7,072,984	B1	7/2006	Polonsky et al.
7,219,309	B2	5/2007	Kaasila et al.
7,308,649	B2	12/2007	Ehrich et al.
7,450,114	B2	11/2008	Anwar
2001/0015719	A1	8/2001	Van Ee et al.
2001/0047428	A1	11/2001	Hunter
2002/0023110	A1	2/2002	Fortin et al.
2002/0030699	A1	3/2002	Van Ee
2002/0112237	A1	8/2002	Ketts
2002/0158908	A1	10/2002	Vaajala et al.
2004/0049598	A1	3/2004	Tucker et al.
2005/0144256	A1	6/2005	Blumberg

OTHER PUBLICATIONS

Benjamin B. Bederson et al., Pad++: A Zoomable Graphical Sketchpad For Exploring Alternate Interface Physics, Sep. 19, 1995, pp. 1-30, http://www.cs.unm.edu/pad++.

Benjamin B. Bederson et al., A Zooming Web Browser, SPIE 1996, http://www.cs.umd.edu/hcil/jazz/learn/papers/spie-96-webbrowser.pdf, entirety (pages not numbered) .

Specification for Simple Vector Format (SVF) v1.1 Jan. 16, 1995, pp. 1-9.

Specification for Simple Vector Format (SFV) v2.0 Dec. 6, 2000, http://www.svf.org/spec.html, pp. 1-20.

SVF XML http://www.svf.org/svfxml.html, pp. 1-18.

Changes to SVF (Date unknown), pp. 1-2.

Scalable Vector Graphics (SVG) 1.0 Specification, W3C Candidate Recommendation Nov. 2, 2000, pp. 1-513.

Introduction to SVG, part of WD-SVG-19990211, http://www.w3.orgfTR/1999/ WD-SVG-19990211/intro.html#Document . . ., pp. 1-3.

W3C Scalable Vector Graphics (SVG)—History, http://www.w3.org/Graphics/SVG/History, pp. 1-18.

Janus Boye, SVG Brings Fast Vector Graphics to Web, Jul. 29, 1999 http://www.irt.org/articles/js176/, pp. 1-5.

Vector Markup Language (VML), World Wide Web Consortium Note May 13, 1998, Note-VML-19980513, http://www.w3.org/TR/1998/Note-VML_19980513, pp. 1-47.

Vector Markup Language, http://en.wikipedia.org/wiki/Vector_Markup_Language, pp. 1-2.

Precision Graphics Markup Language (PGML), World Wide Web Consortium Note Apr. 10, 1998, NOTE-PGML-19980410, http://www.w3.org/TR/1998/NOTE-PGML-19980410, pp. 1-31.

PCT International Search Report for International Application No. PCT/US01/40920, International Searching Authority, European Patent Office, Dec. 27, 2001, pp. 1-8.

PCT International Preliminary Exam Report for International Application No. PCT/US01/40920, International Preliminary Examining Authority, European Patent Office, Jun. 20, 2002, pp. 1-5.

Dennis, "Webraska Joins W3C for Next Generation Internet Graphics," Newsbytes, May 8, 2000, entirety (pages not numbered).

Fox et al. "Experience With Top Gun Wingman: A Proxy-Based Graphical Web Browser for the 3Com PalmPilot" Proceedings of Middleware '98, 1998, entirety (pages not numbered).

"Tadpole Technology: ILOG technology within Tadpole's Cartesia Redline system making repairs simpler," M2 Presswire, Jan. 1999, entirety (pages not numbered).

Thryft, "Wireless link ramped for 'always on' Internet," EETimes, Mar. 23, 1999, pp. 1-4.

Bharadvaj et al., "An Active Transcoding Proxy to Support Mobile Web Access," Proc. Of the IEEE Symposium on Reliable Distributed Systems, 1998, entirety (pages not numbered).

Björk, et al., "WEST: A Web Browser for Small Terminals," Proceedings of the 12 ACM symposium on User Interface Software and Technology (1999), entirety (pages not numbered).

Holmquist, "The Zoom Browser," Human IT Tridskrift för studier av IT ur ett humanvetenskaplight perspektive, Mar., 1998, pp. 1-18.

Chapple, "Rethinking the role of an embedded Internet client in digital set-top boxes," WebTV Workshop Submission Paper, Jun. 1998, http://www.w3.org/Architecture/1998/06/Workshop/paper11/, pp. 1-4.

Graham, "Mobile SVG at BitFlash Inc.", May 2001, http://www.w3.org/Talks/2002/1007-Di-Helsinki/bitflash/index.html, pp. 1-4.

Koncz, "Working with Autodesk Mapguide," Nov. 2000, http://www.directionsmag.com/article.php? article_id=77&trv=1, pp. 1-6.

Schaffer et al. "Navigating Hierarchically Cluster Networks through Fisheye and Full-Zoom Methods," Mar. 24, 1988—pp. 1-20.

Holmquist "Flip Zooming: Focus+Context Visualization of Linearly Ordered Discrete Visual Structures,"Published 2000, pp. 27-53.

Scalable Vector Graphics (SVG) Specification, W3C Working Draft Feb. 11, 1999 WD-SVG-19990211, http://www.w3.org/TR/1999/WD-SVG-19990211/ (HTML format—initial page).

Scalable Vector Graphics (SVG) Specification, W3C Working Draft Apr. 12, 1999 WD-SVG-19990412, http://www.w3.org/TR/1999/WD-SVG-19990412/ (HTML format—initial page).

Jones et al. "From the large screen to the small screen—retaining the designer's design for effective user interaction,"IEEE Colloquim on Issues for Networked Interpersonal Communicators, May 1997, pp. 3/1-3/4.

Pacchiano, "Adobe's New View," Computer Shopper, vol. 19, No. 4, Apr. 1999, p. 346.

Official USPTO Actions for U.S. Appl. No. 09/878,097. (Entirety—pages not numbered).

Official USPTO Actions for U.S. Appl. No. 11/045,649. (Entirety—pages not numbered).

Official USPTO Actions for U.S. Appl. No. 11/045,757. (Entirety—pages not numbered).

Official USPTO Actions for U.S. Appl. No. 11/738,932. (Entirety—pages not numbered).

Official USPTO Action for U.S. Appl. No. 11/868,124 Mailed Aug. 24, 2010 pp.1-15.

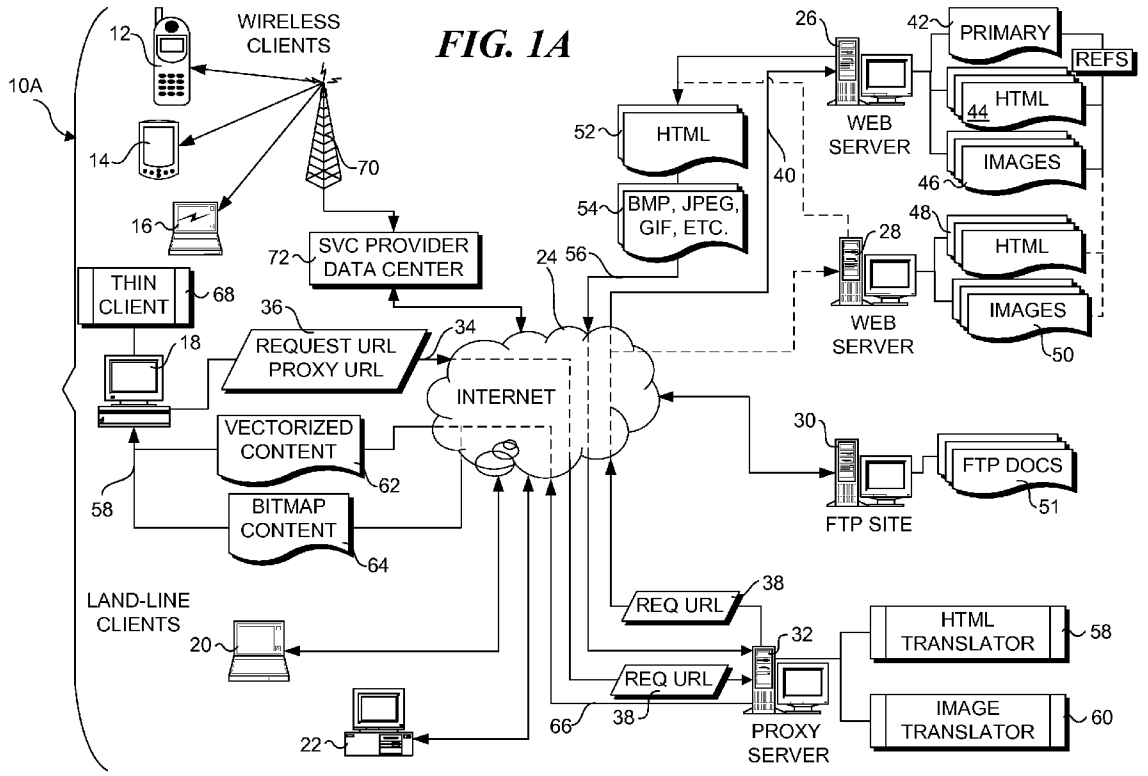


FIG. 1A
10A
WIRELESS CLIENTS
12
14
16
70
72
SVC PROVIDER DATA CENTER
THIN CLIENT
68
18
36
REQUEST URL PROXY URL
34
INTERNET
24
56
VECTORIZED CONTENT
62
58
BITMAP CONTENT
64
LAND-LINE CLIENTS
20
22
26
WEB SERVER
42
PRIMARY
REFS
HTML
44
IMAGES
46
40
52
HTML
54
BMP, JPEG, GIF, ETC.
28
48
HTML
IMAGES
50
WEB SERVER
30
FTP SITE
FTP DOCS
51
38
REQ URL
32
REQ URL
38
66
PROXY SERVER
HTML TRANSLATOR
58
IMAGE TRANSLATOR
60

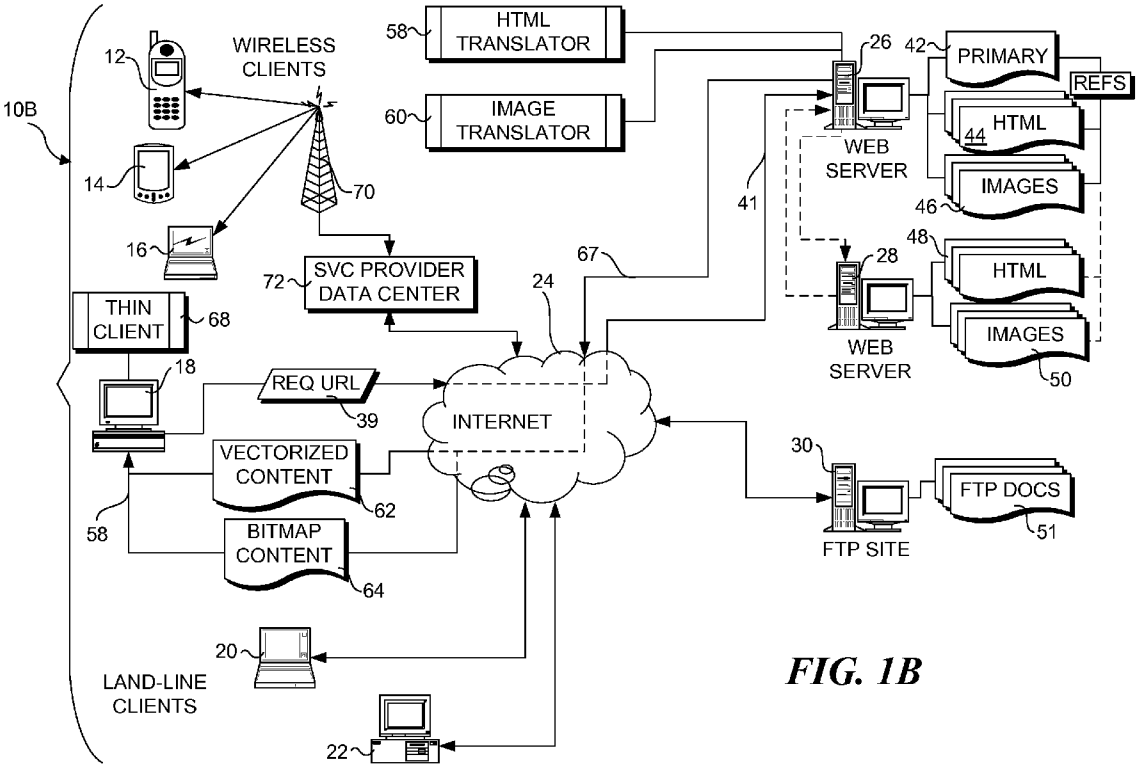

10B
12
WIRELESS CLIENTS
14
16
70
58
HTML TRANSLATOR
60
IMAGE TRANSLATOR
72
SVC PROVIDER DATA CENTER
THIN CLIENT
68
18
REQ URL
39
INTERNET
VECTORIZED CONTENT
62
58
BITMAP CONTENT
64
20
LAND-LINE CLIENTS
22
24
67
26
WEB SERVER
41
42
PRIMARY
REFS
HTML
44
IMAGES
46
48
28
WEB SERVER
HTML
IMAGES
50
30
FTP SITE
FTP DOCS
51

FIG. 1B

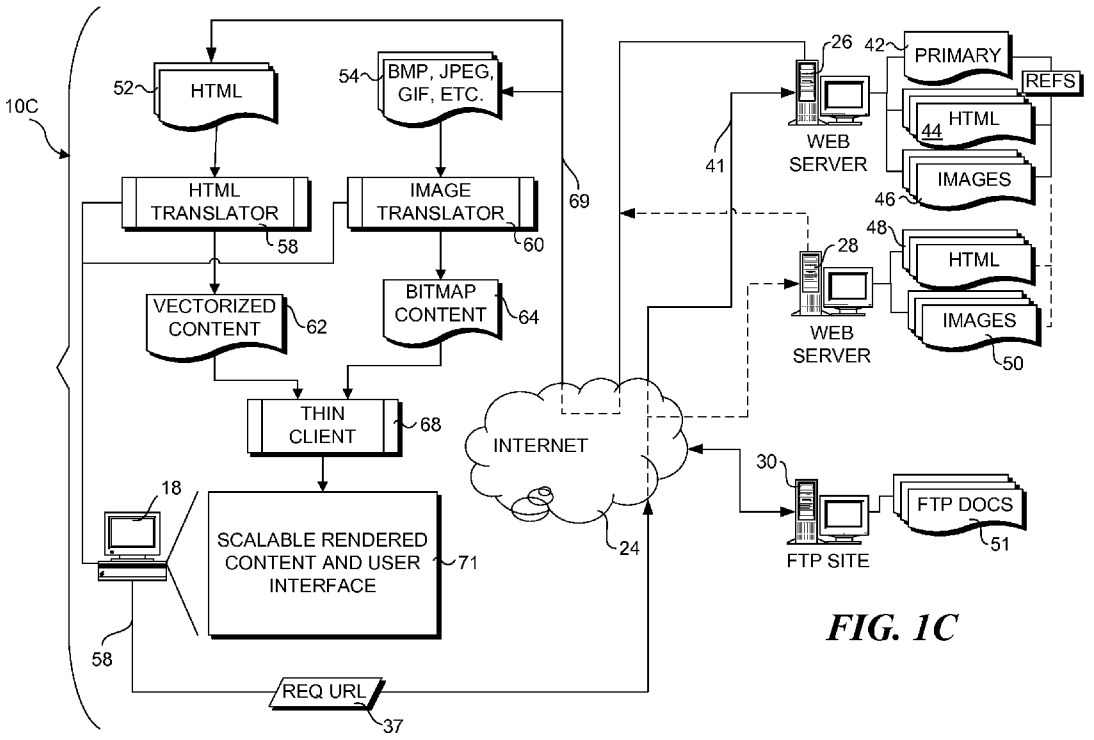

10C
52
HTML
54
BMP, JPEG, GIF, ETC.
HTML TRANSLATOR
58
IMAGE TRANSLATOR
60
69
VECTORIZED CONTENT
62
BITMAP CONTENT
64
THIN CLIENT
68
18
SCALABLE RENDERED CONTENT AND USER INTERFACE
71
58
REQ URL
37
INTERNET
24
26
WEB SERVER
41
42
PRIMARY
REFS
HTML
44
IMAGES
46
28
48
HTML
IMAGES
50
WEB SERVER
30
FTP DOCS
51
FTP SITE

FIG. 1C

FIG. 2A

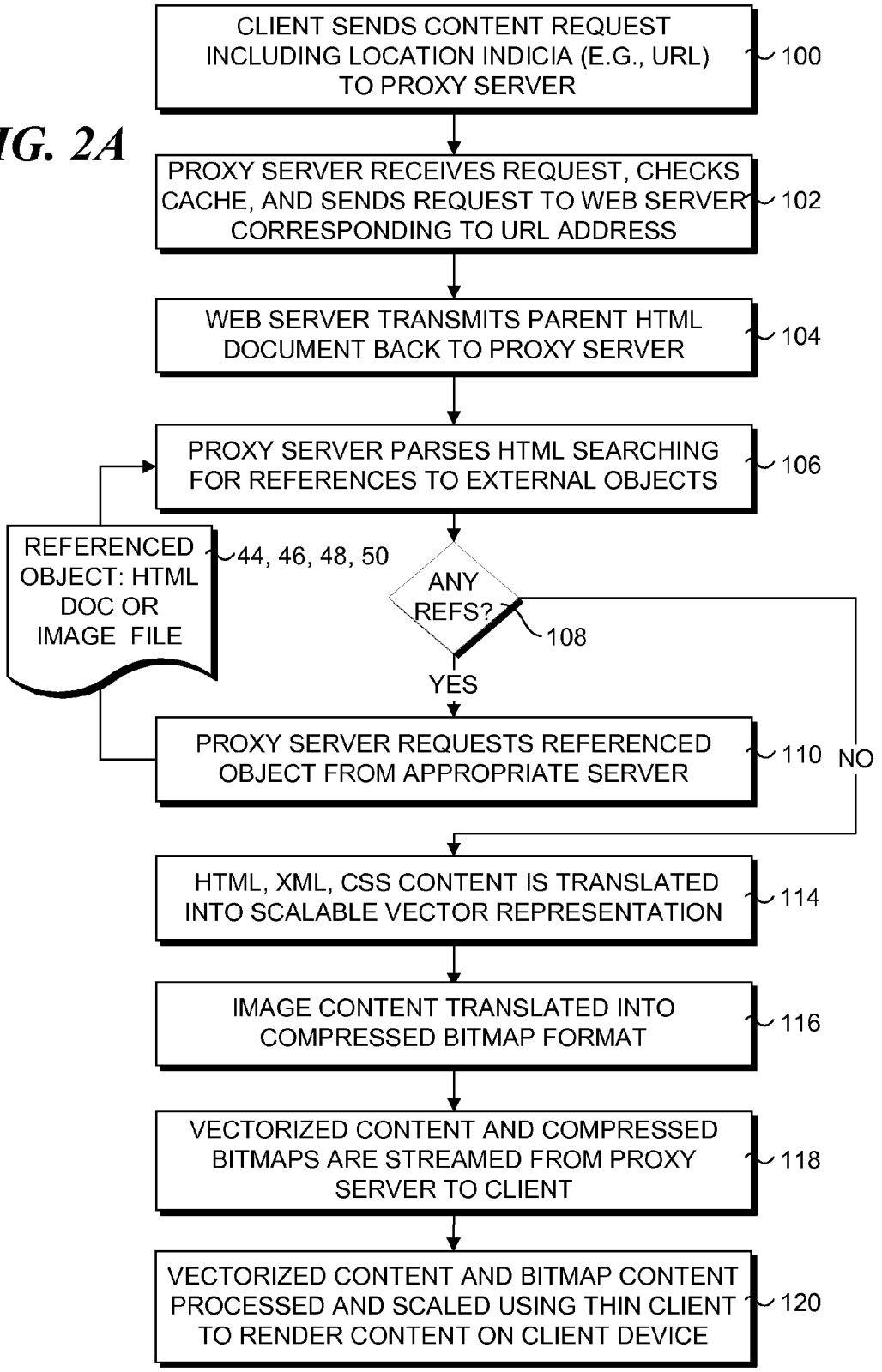

CLIENT SENDS CONTENT REQUEST INCLUDING LOCATION INDICIA (E.G., URL) TO PROXY SERVER
100
PROXY SERVER RECEIVES REQUEST, CHECKS CACHE, AND SENDS REQUEST TO WEB SERVER CORRESPONDING TO URL ADDRESS
102
WEB SERVER TRANSMITS PARENT HTML DOCUMENT BACK TO PROXY SERVER
104
PROXY SERVER PARSES HTML SEARCHING FOR REFERENCES TO EXTERNAL OBJECTS
106
REFERENCED OBJECT: HTML DOC OR IMAGE FILE
44, 46, 48, 50
ANY REFS?
108
YES
NO
PROXY SERVER REQUESTS REFERENCED OBJECT FROM APPROPRIATE SERVER
110
HTML, XML, CSS CONTENT IS TRANSLATED INTO SCALABLE VECTOR REPRESENTATION
114
IMAGE CONTENT TRANSLATED INTO COMPRESSED BITMAP FORMAT
116
VECTORIZED CONTENT AND COMPRESSED BITMAPS ARE STREAMED FROM PROXY SERVER TO CLIENT
118
VECTORIZED CONTENT AND BITMAP CONTENT PROCESSED AND SCALED USING THIN CLIENT TO RENDER CONTENT ON CLIENT DEVICE
120

FIG. 2B

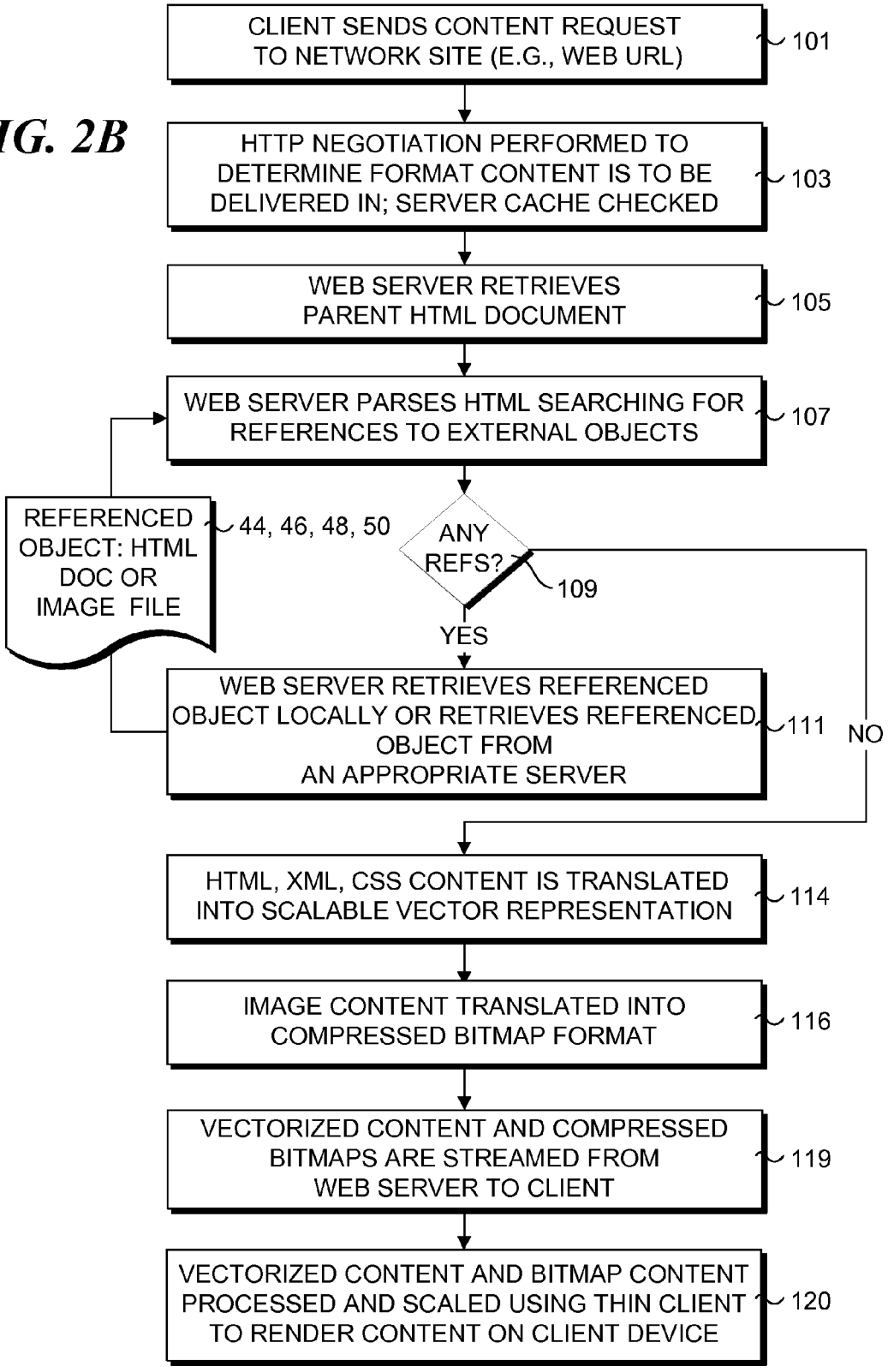
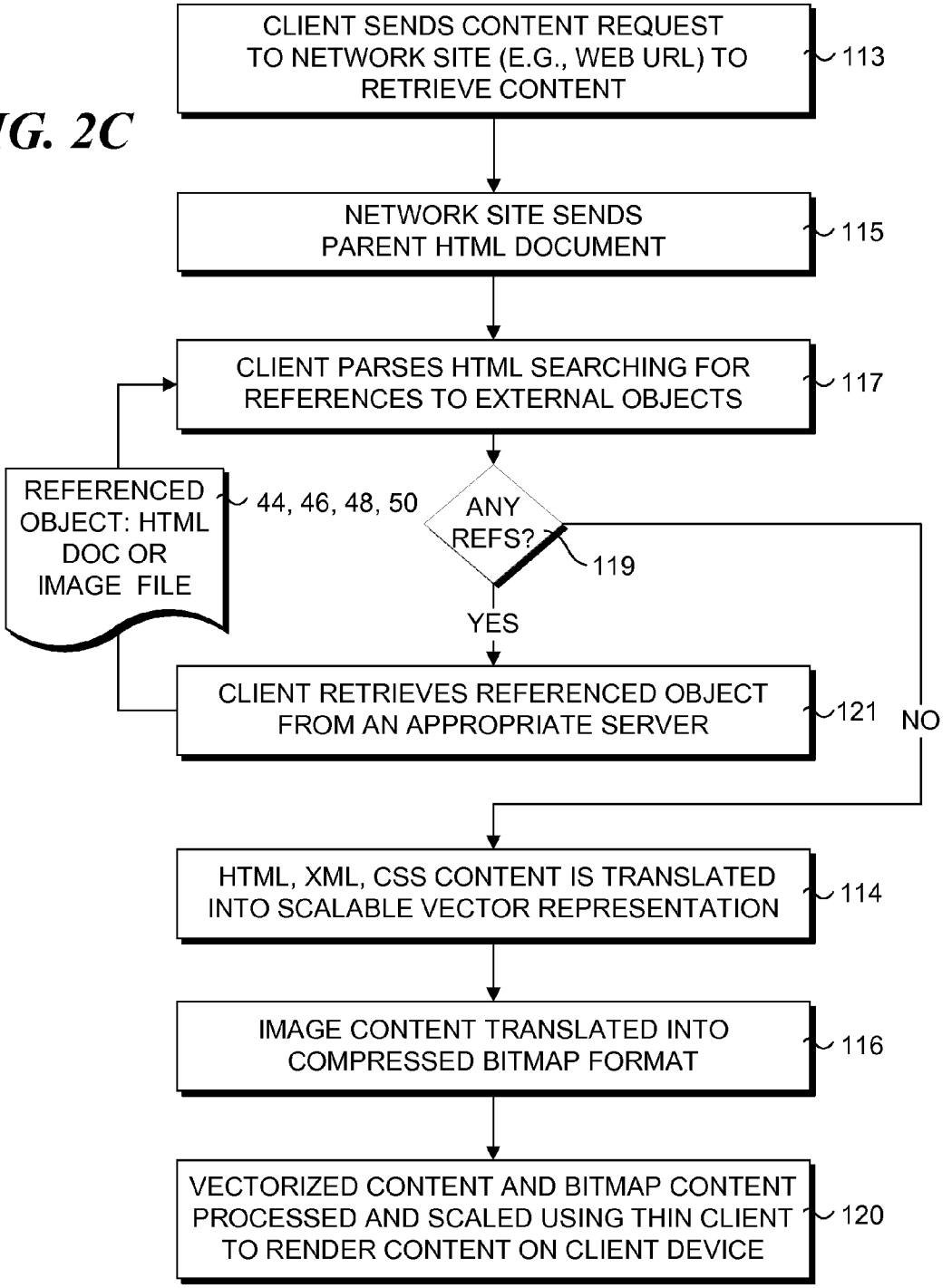

CLIENT SENDS CONTENT REQUEST TO NETWORK SITE (E.G., WEB URL)
101
HTTP NEGOTIATION PERFORMED TO DETERMINE FORMAT CONTENT IS TO BE DELIVERED IN; SERVER CACHE CHECKED
103
WEB SERVER RETRIEVES PARENT HTML DOCUMENT
105
WEB SERVER PARSES HTML SEARCHING FOR REFERENCES TO EXTERNAL OBJECTS
107
REFERENCED OBJECT: HTML DOC OR IMAGE FILE
44, 46, 48, 50
ANY REFS?
109
YES
NO
WEB SERVER RETRIEVES REFERENCED OBJECT LOCALLY OR RETRIEVES REFERENCED OBJECT FROM AN APPROPRIATE SERVER
111
HTML, XML, CSS CONTENT IS TRANSLATED INTO SCALABLE VECTOR REPRESENTATION
114
IMAGE CONTENT TRANSLATED INTO COMPRESSED BITMAP FORMAT
116
VECTORIZED CONTENT AND COMPRESSED BITMAPS ARE STREAMED FROM WEB SERVER TO CLIENT
119
VECTORIZED CONTENT AND BITMAP CONTENT PROCESSED AND SCALED USING THIN CLIENT TO RENDER CONTENT ON CLIENT DEVICE
120

FIG. 2C


CLIENT SENDS CONTENT REQUEST TO NETWORK SITE (E.G., WEB URL) TO RETRIEVE CONTENT
113
NETWORK SITE SENDS PARENT HTML DOCUMENT
115
CLIENT PARSES HTML SEARCHING FOR REFERENCES TO EXTERNAL OBJECTS
117
REFERENCED OBJECT: HTML DOC OR IMAGE FILE
44, 46, 48, 50
ANY REFS?
119
YES
NO
CLIENT RETRIEVES REFERENCED OBJECT FROM AN APPROPRIATE SERVER
121
HTML, XML, CSS CONTENT IS TRANSLATED INTO SCALABLE VECTOR REPRESENTATION
114
IMAGE CONTENT TRANSLATED INTO COMPRESSED BITMAP FORMAT
116
VECTORIZED CONTENT AND BITMAP CONTENT PROCESSED AND SCALED USING THIN CLIENT TO RENDER CONTENT ON CLIENT DEVICE
120

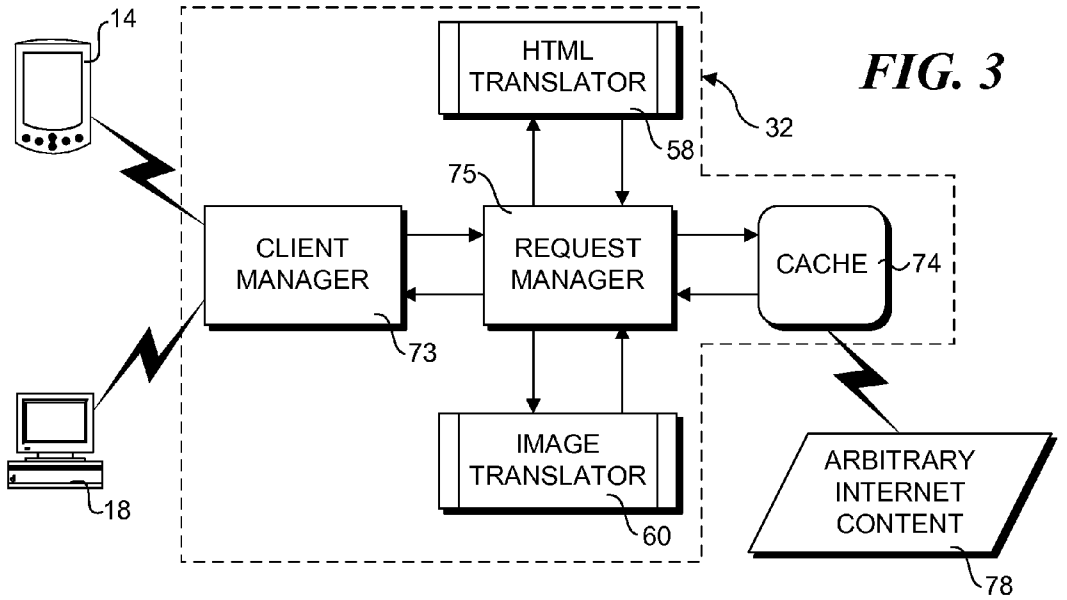

14
HTML TRANSLATOR
32
58
75
CLIENT MANAGER
REQUEST MANAGER
CACHE
74
73
IMAGE TRANSLATOR
60
ARBITRARY INTERNET CONTENT
78
18

FIG. 3

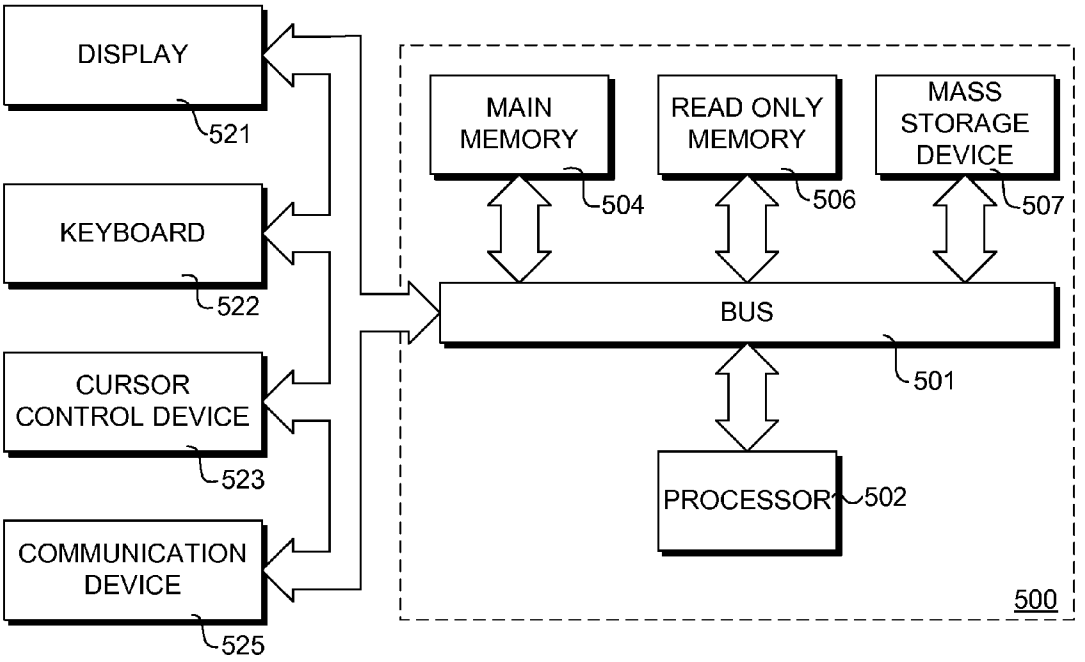

DISPLAY
521
KEYBOARD
522
CURSOR CONTROL DEVICE
523
COMMUNICATION DEVICE
525
MAIN MEMORY
504
READ ONLY MEMORY
506
MASS STORAGE DEVICE
507
BUS
501
PROCESSOR
502
500

FIG. 10

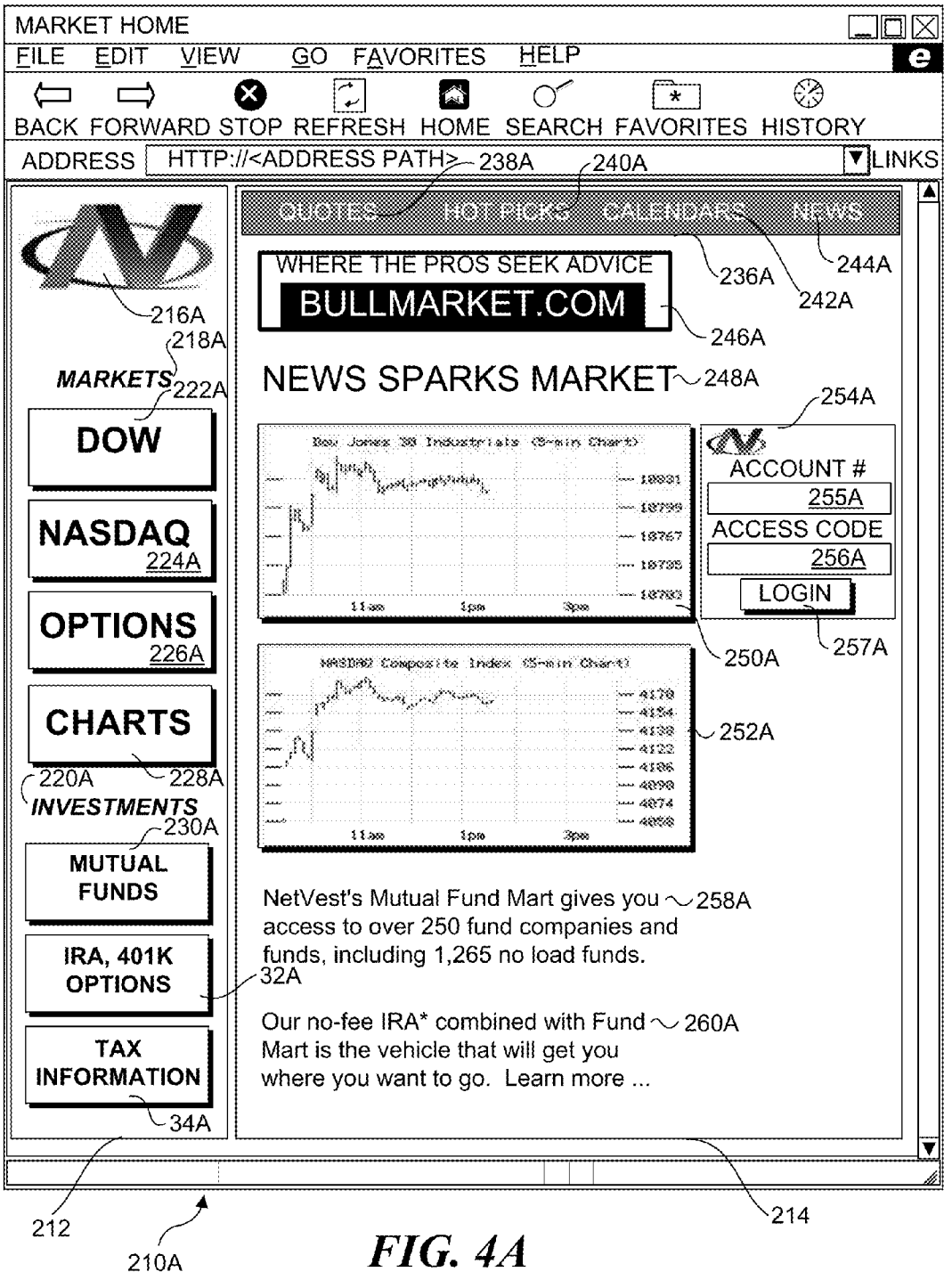

MARKET HOME
FILE EDIT VIEW GO FAVORITES HELP
BACK FORWARD STOP REFRESH HOME SEARCH FAVORITES HISTORY
ADDRESS HTTP://<ADDRESS PATH>
LINKS
238A
240A
QUOTES
HOT PICKS
CALENDARS
NEWS
244A
WHERE THE PROS SEEK ADVICE
BULLMARKET.COM
236A
242A
246A
216A
218A
MARKETS
222A
NEWS SPARKS MARKET
248A
254A
DOW
NASDAQ
224A
OPTIONS
226A
CHARTS
228A
220A
INVESTMENTS
230A
ACCOUNT #
255A
ACCESS CODE
256A
LOGIN
257A
250A
252A
MUTUAL FUNDS
IRA, 401K OPTIONS
32A
TAX INFORMATION
34A
NetVest's Mutual Fund Mart gives you access to over 250 fund companies and funds, including 1,265 no load funds.
258A
Our no-fee IRA* combined with Fund Mart is the vehicle that will get you where you want to go. Learn more ...
260A
212
210A
214

FIG. 4A

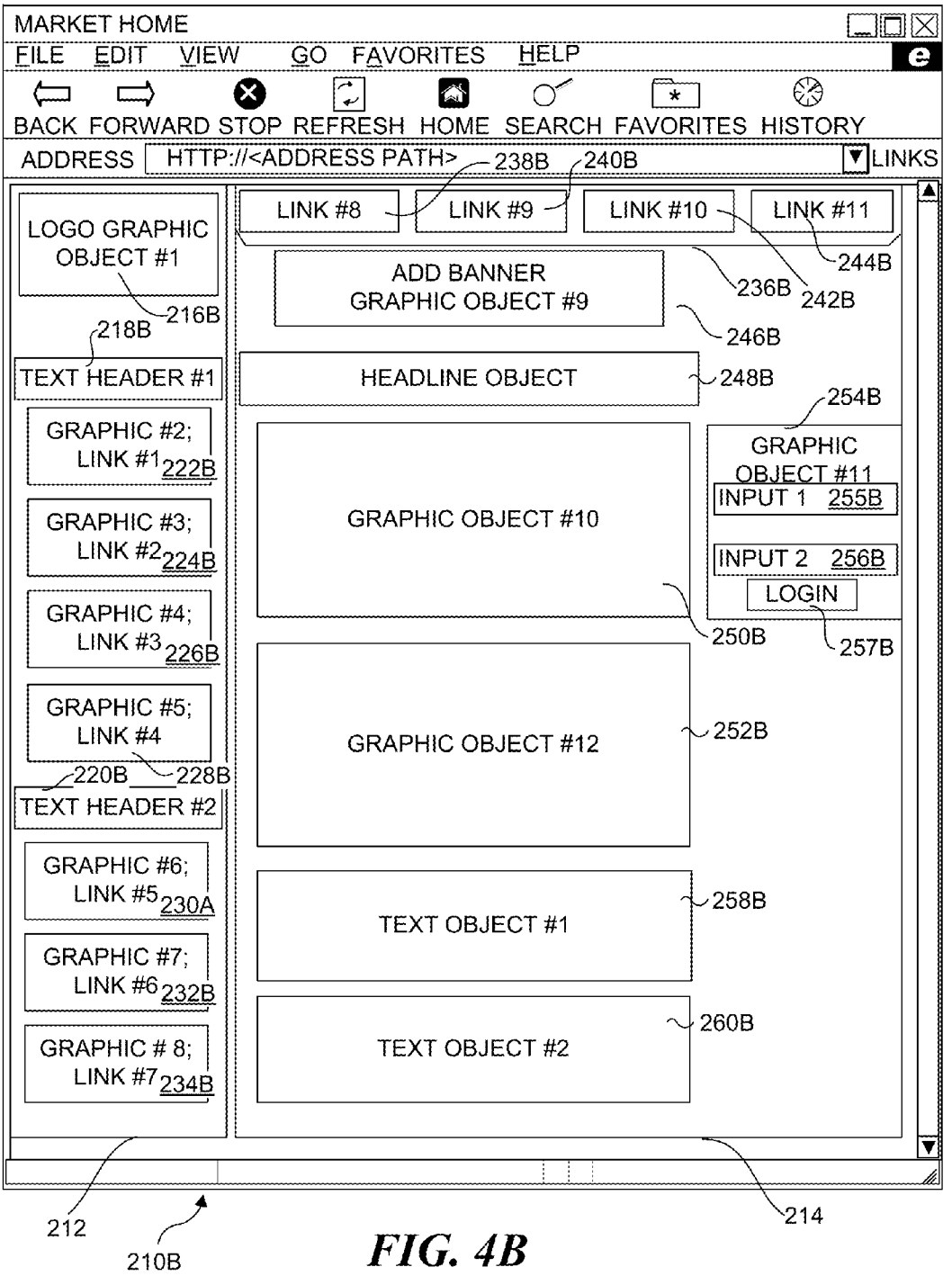

MARKET HOME
FILE EDIT VIEW GO FAVORITES HELP
BACK FORWARD STOP REFRESH HOME SEARCH FAVORITES HISTORY
ADDRESS HTTP://<ADDRESS PATH>
LINKS
LOGO GRAPHIC OBJECT #1
216B
218B
TEXT HEADER #1
GRAPHIC #2; LINK #1 222B
GRAPHIC #3; LINK #2 224B
GRAPHIC #4; LINK #3 226B
GRAPHIC #5; LINK #4
220B
228B
TEXT HEADER #2
GRAPHIC #6; LINK #5 230A
GRAPHIC #7; LINK #6 232B
GRAPHIC # 8; LINK #7 234B
LINK #8
LINK #9
LINK #10
LINK #11
238B
240B
244B
ADD BANNER GRAPHIC OBJECT #9
236B
242B
246B
HEADLINE OBJECT
248B
254B
GRAPHIC OBJECT #10
GRAPHIC OBJECT #11
INPUT 1 255B
INPUT 2 256B
LOGIN
250B
257B
GRAPHIC OBJECT #12
252B
TEXT OBJECT #1
258B
TEXT OBJECT #2
260B
212
210B
214

FIG. 4B

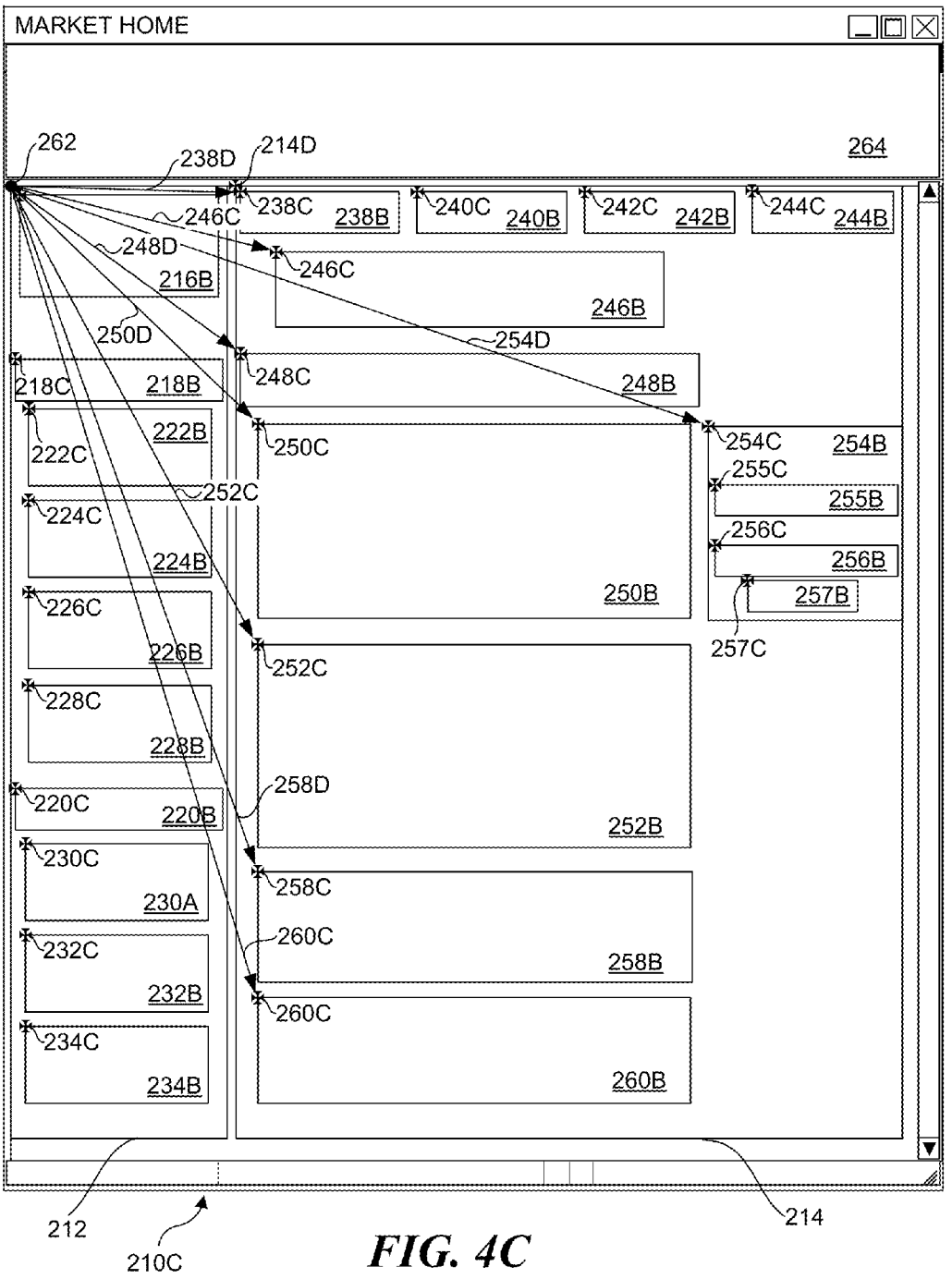

MARKET HOME
262
238D
214D
264
238C
238B
240C
240B
242C
242B
244C
244B
246C
248D
216B
246C
246B
250D
254D
218C
218B
248C
248B
222B
222C
250C
254C
254B
255C
255B
252C
224C
224B
256C
256B
250B
257B
226C
226B
257C
252C
228C
228B
220C
220B
258D
252B
230C
230A
258C
260C
258B
232C
232B
260C
234C
234B
260B
212
210C
214

FIG. 4C

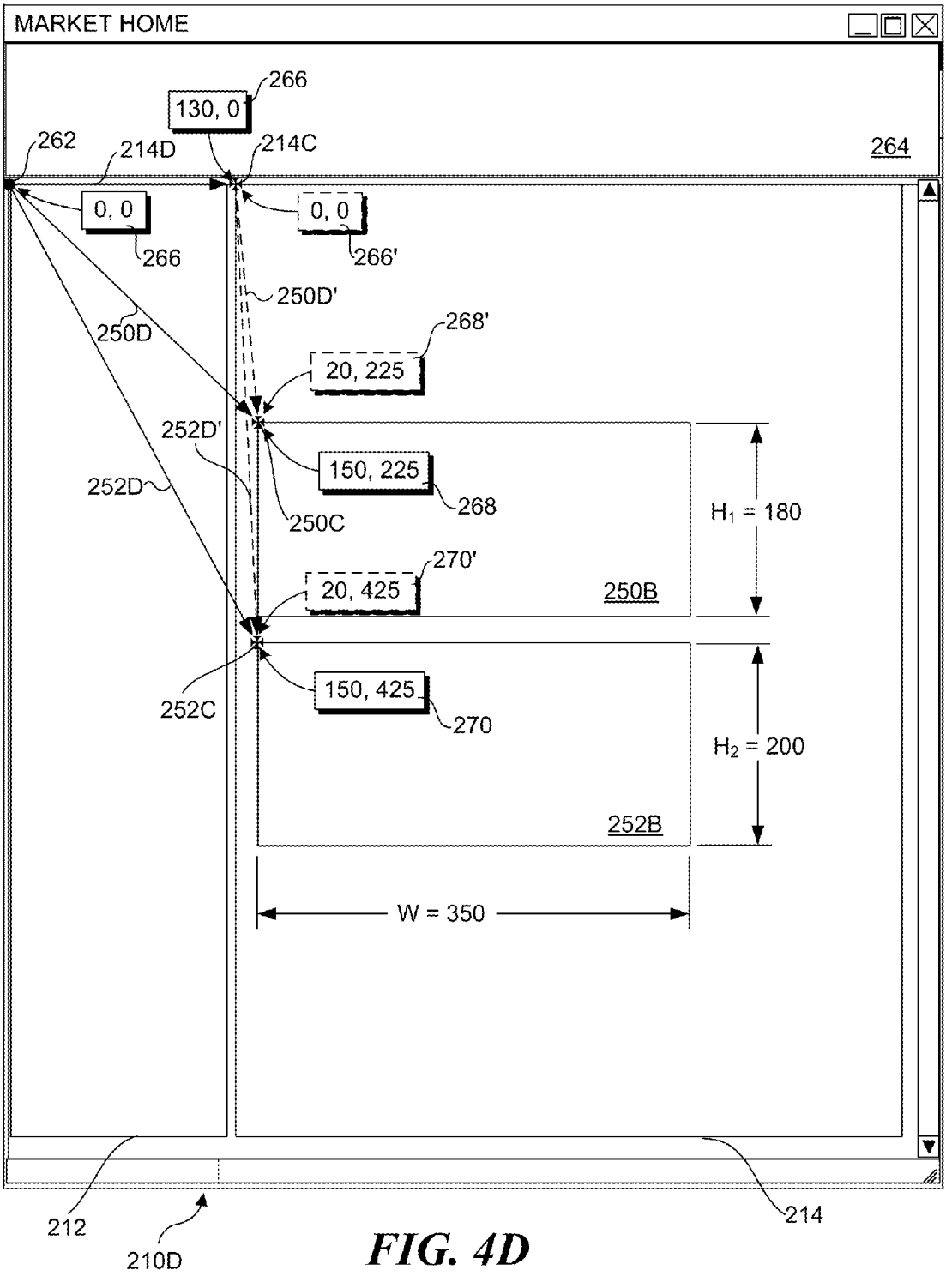
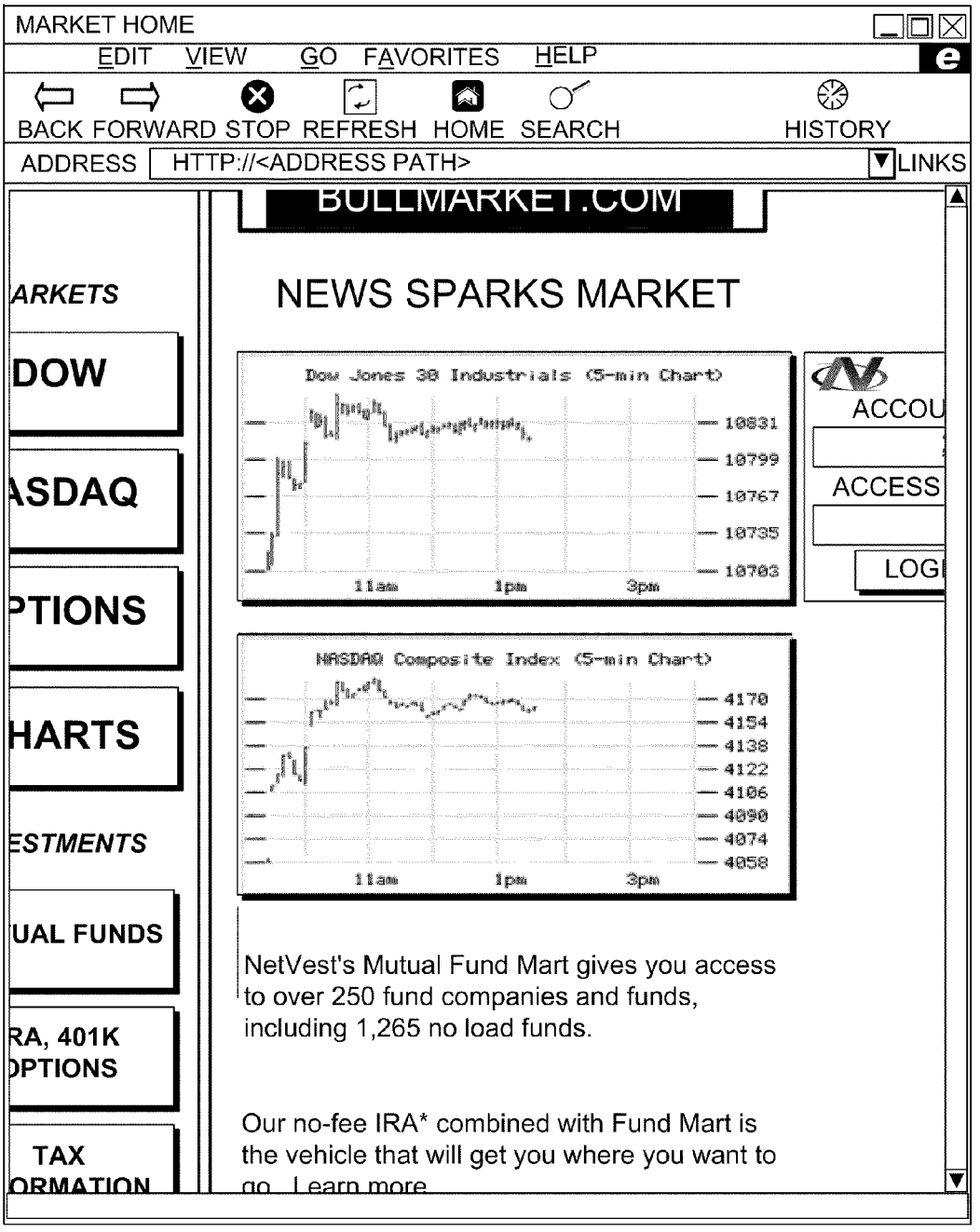

MARKET HOME
266
130, 0
262
214D
214C
264
0, 0
266
0, 0
266'
250D'
250D
268'
20, 225
252D'
252D
150, 225
268
250C
H1 = 180
270'
20, 425
250B
252C
150, 425
270
H2 = 200
252B
W = 350
212
210D
214

FIG. 4D


MARKET HOME
EDIT VIEW GO FAVORITES HELP
BACK FORWARD STOP REFRESH HOME SEARCH HISTORY
ADDRESS HTTP://<ADDRESS PATH>
LINKS
BULLMARKET.COM
ARKETS
DOW
ASDAQ
PTIONS
HARTS
ESTMENTS
UAL FUNDS
RA, 401K
PTIONS
TAX
ORMATION
NEWS SPARKS MARKET
Dow Jones 30 Industrials (5-min Chart)
10831
10799
10767
10735
10703
11am
1pm
3pm
ACCOU
ACCESS
LOGI
NASDAQ Composite Index (5-min Chart)
4170
4154
4138
4122
4106
4090
4074
4058
11am
1pm
3pm
NetVest's Mutual Fund Mart gives you access to over 250 fund companies and funds, including 1,265 no load funds.
Our no-fee IRA* combined with Fund Mart is the vehicle that will get you where you want to go. Learn more

210E

FIG. 4E

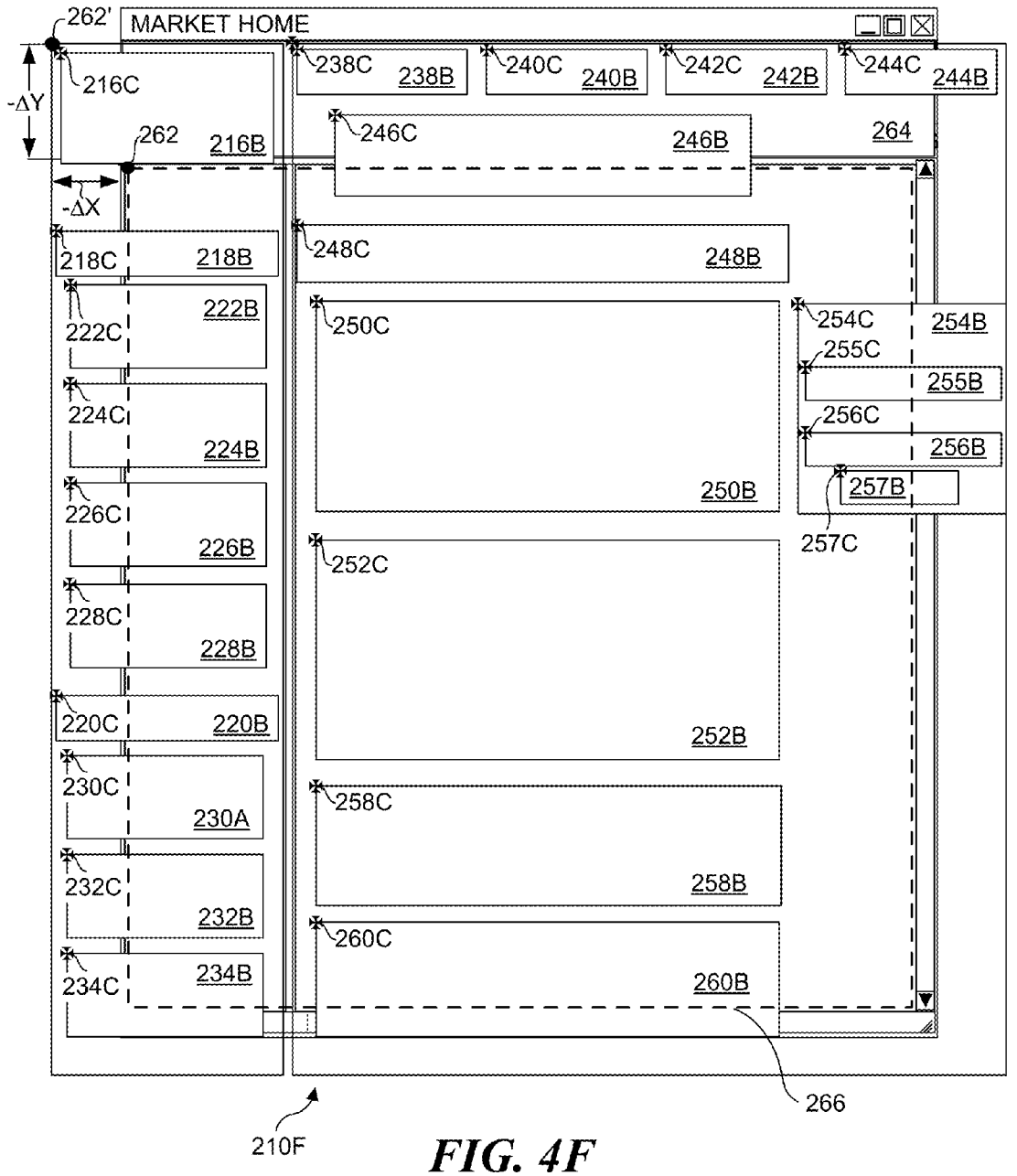

262'
MARKET HOME
-ΔY
216C
262
216B
238C
238B
240C
240B
242C
242B
244C
244B
246C
246B
264
-ΔX
218C
218B
248C
248B
222C
222B
250C
254C
254B
255C
255B
224C
224B
256C
256B
226C
226B
250B
257B
257C
228C
228B
252C
220C
220B
252B
230C
230A
258C
232C
232B
258B
260C
234C
234B
260B
266
210F

FIG. 4F

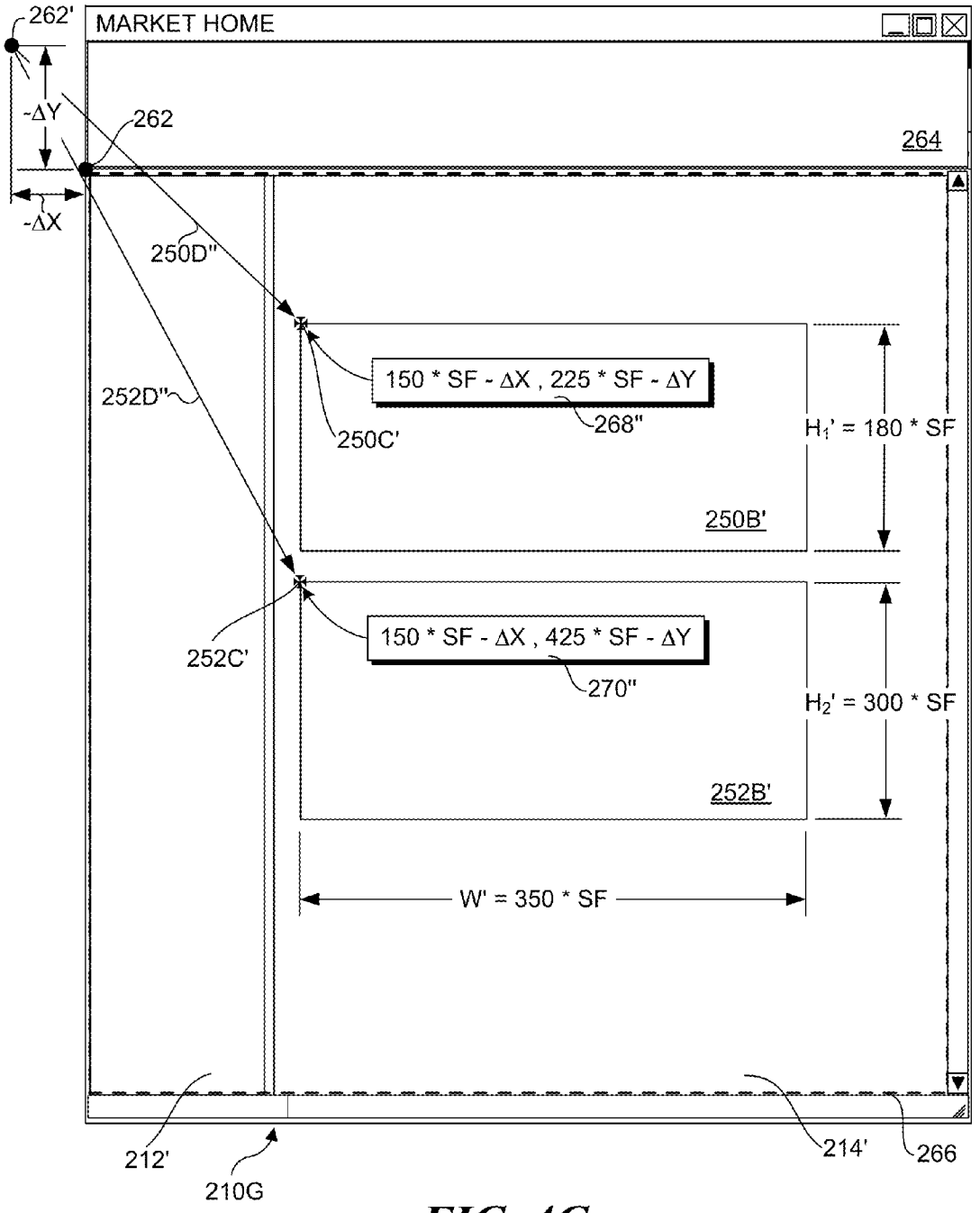

262'
MARKET HOME
-ΔY
262
264
-ΔX
250D"
252D"
150 * SF - ΔX , 225 * SF - ΔY
268"
250C'
H1' = 180 * SF
250B'
252C'
150 * SF - ΔX , 425 * SF - ΔY
270"
H2' = 300 * SF
252B'
W' = 350 * SF
212'
210G
214'
266

FIG. 4G

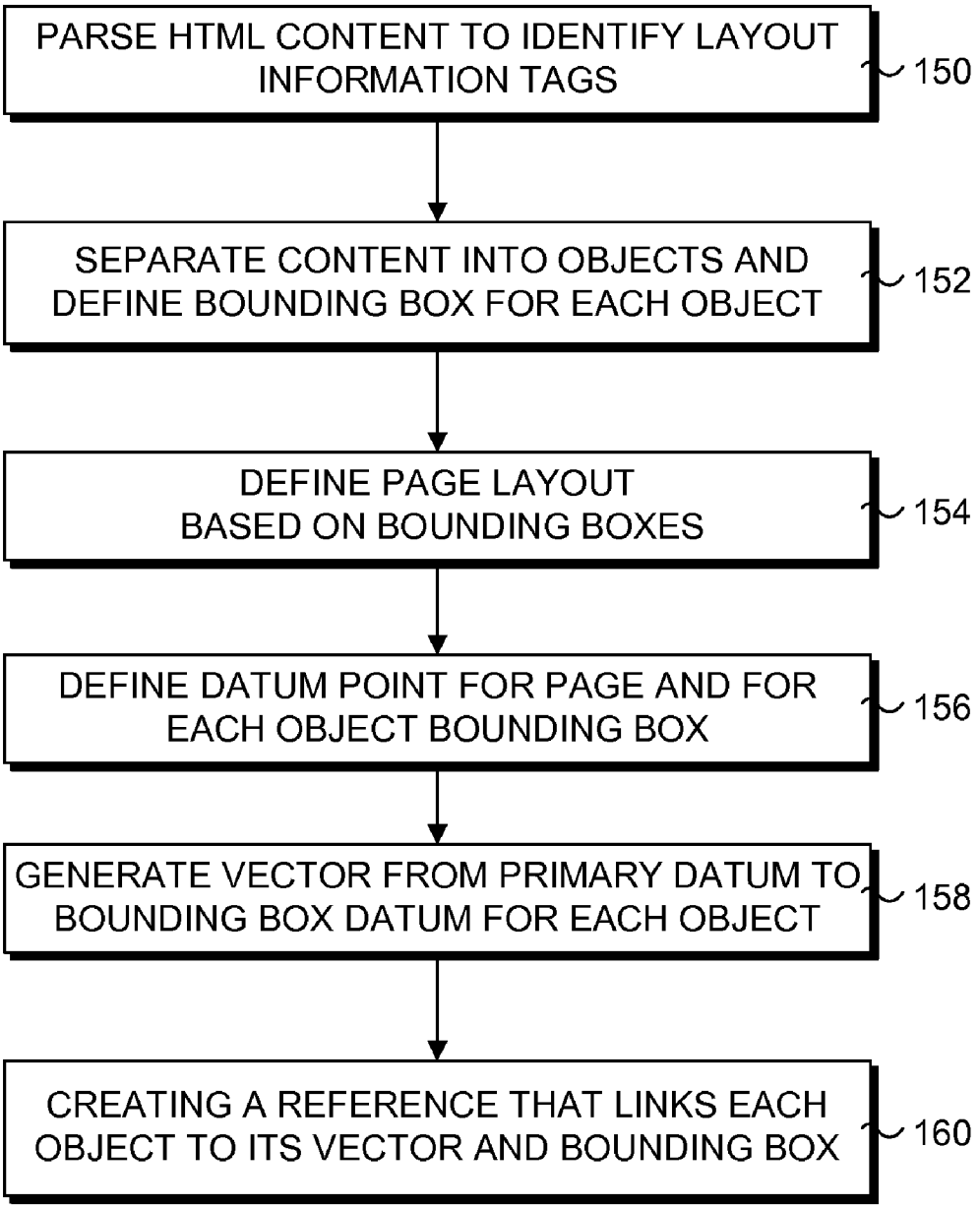

PARSE HTML CONTENT TO IDENTIFY LAYOUT INFORMATION TAGS
150
SEPARATE CONTENT INTO OBJECTS AND DEFINE BOUNDING BOX FOR EACH OBJECT
152
DEFINE PAGE LAYOUT BASED ON BOUNDING BOXES
154
DEFINE DATUM POINT FOR PAGE AND FOR EACH OBJECT BOUNDING BOX
156
GENERATE VECTOR FROM PRIMARY DATUM TO BOUNDING BOX DATUM FOR EACH OBJECT
158
CREATING A REFERENCE THAT LINKS EACH OBJECT TO ITS VECTOR AND BOUNDING BOX
160

FIG. 5

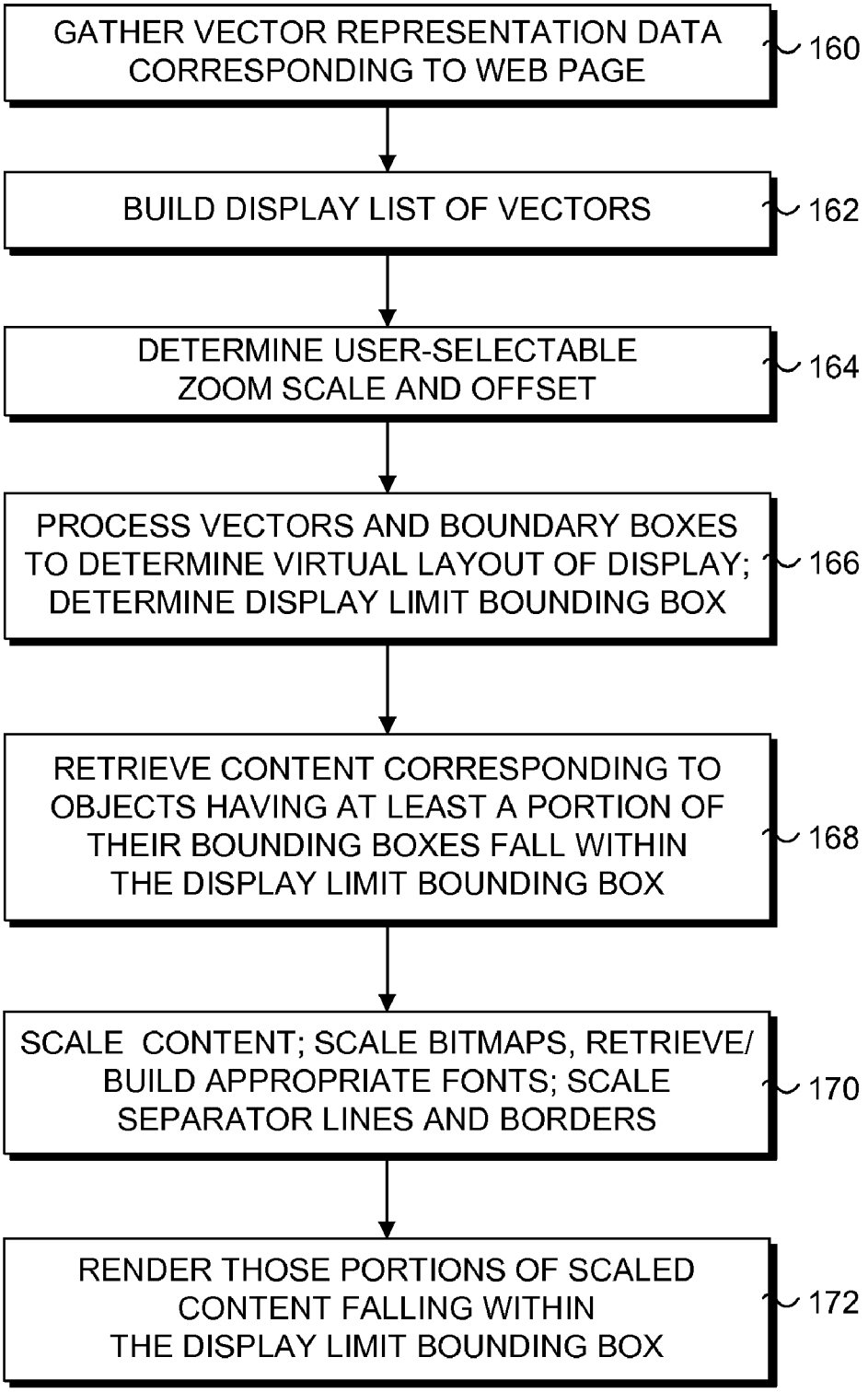

GATHER VECTOR REPRESENTATION DATA CORRESPONDING TO WEB PAGE
160
BUILD DISPLAY LIST OF VECTORS
162
DETERMINE USER-SELECTABLE ZOOM SCALE AND OFFSET
164
PROCESS VECTORS AND BOUNDARY BOXES TO DETERMINE VIRTUAL LAYOUT OF DISPLAY; DETERMINE DISPLAY LIMIT BOUNDING BOX
166
RETRIEVE CONTENT CORRESPONDING TO OBJECTS HAVING AT LEAST A PORTION OF THEIR BOUNDING BOXES FALL WITHIN THE DISPLAY LIMIT BOUNDING BOX
168
SCALE CONTENT; SCALE BITMAPS, RETRIEVE/ BUILD APPROPRIATE FONTS; SCALE SEPARATOR LINES AND BORDERS
170
RENDER THOSE PORTIONS OF SCALED CONTENT FALLING WITHIN THE DISPLAY LIMIT BOUNDING BOX
172

FIG. 6

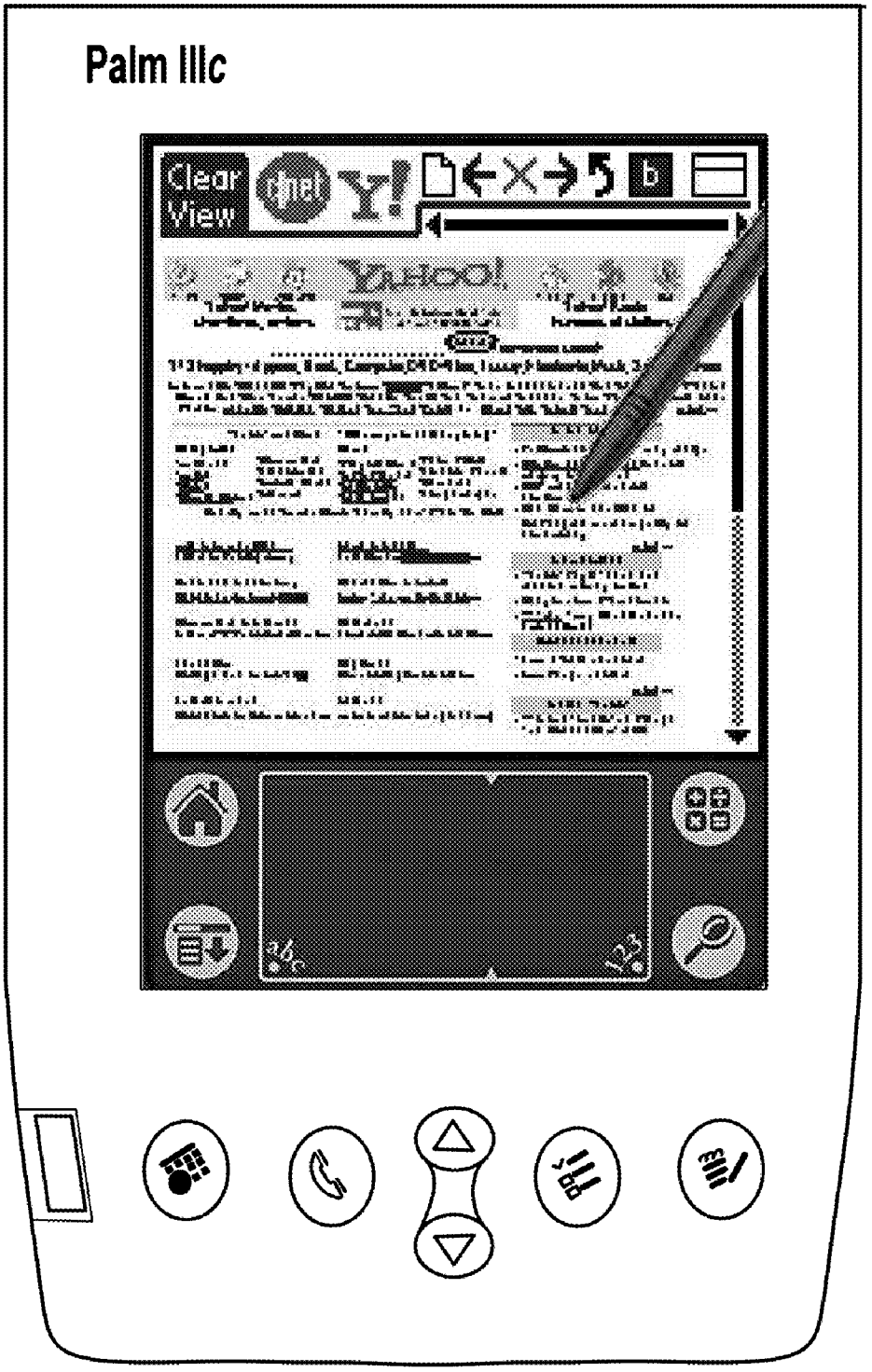


FIG. 7A

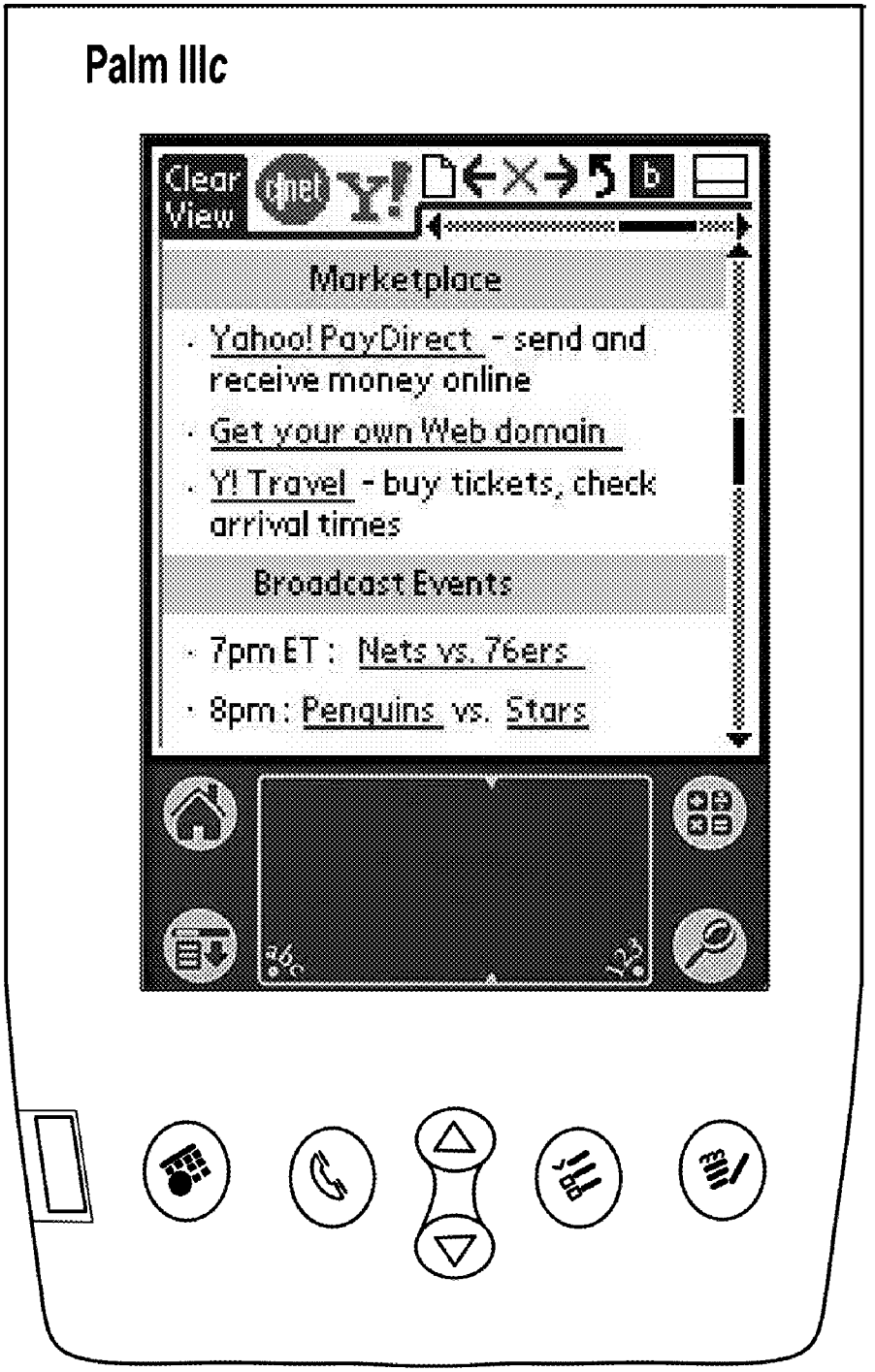

Palm IIIc
Clear View
Marketplace
· Yahoo! PayDirect - send and receive money online
· Get your own Web domain
· Y! Travel - buy tickets, check arrival times
Broadcast Events
· 7pm ET : Nets vs. 76ers
· 8pm : Penguins vs. Stars

FIG. 7B

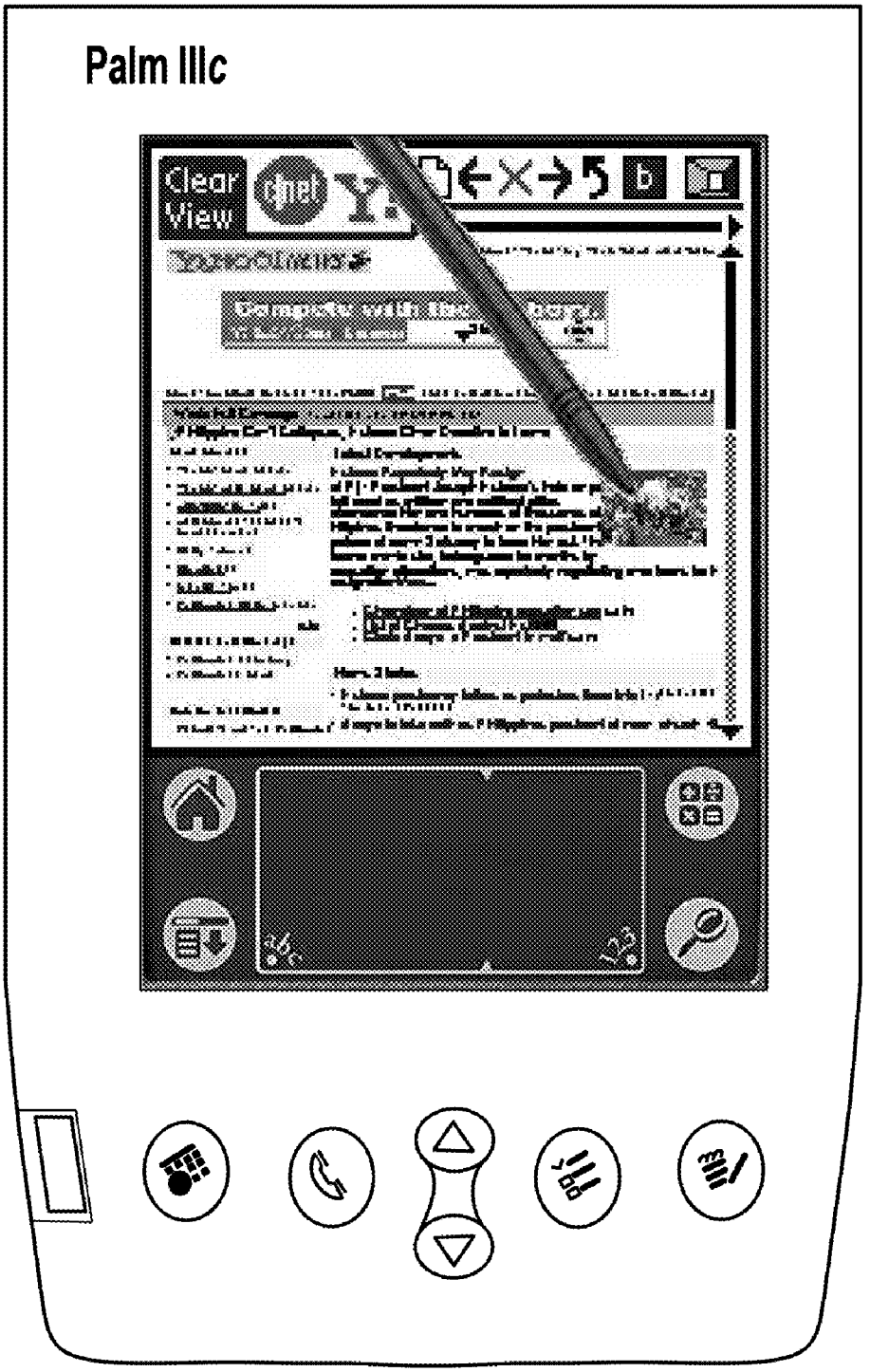


FIG. 8A

FIG. 8B

FIG. 9A

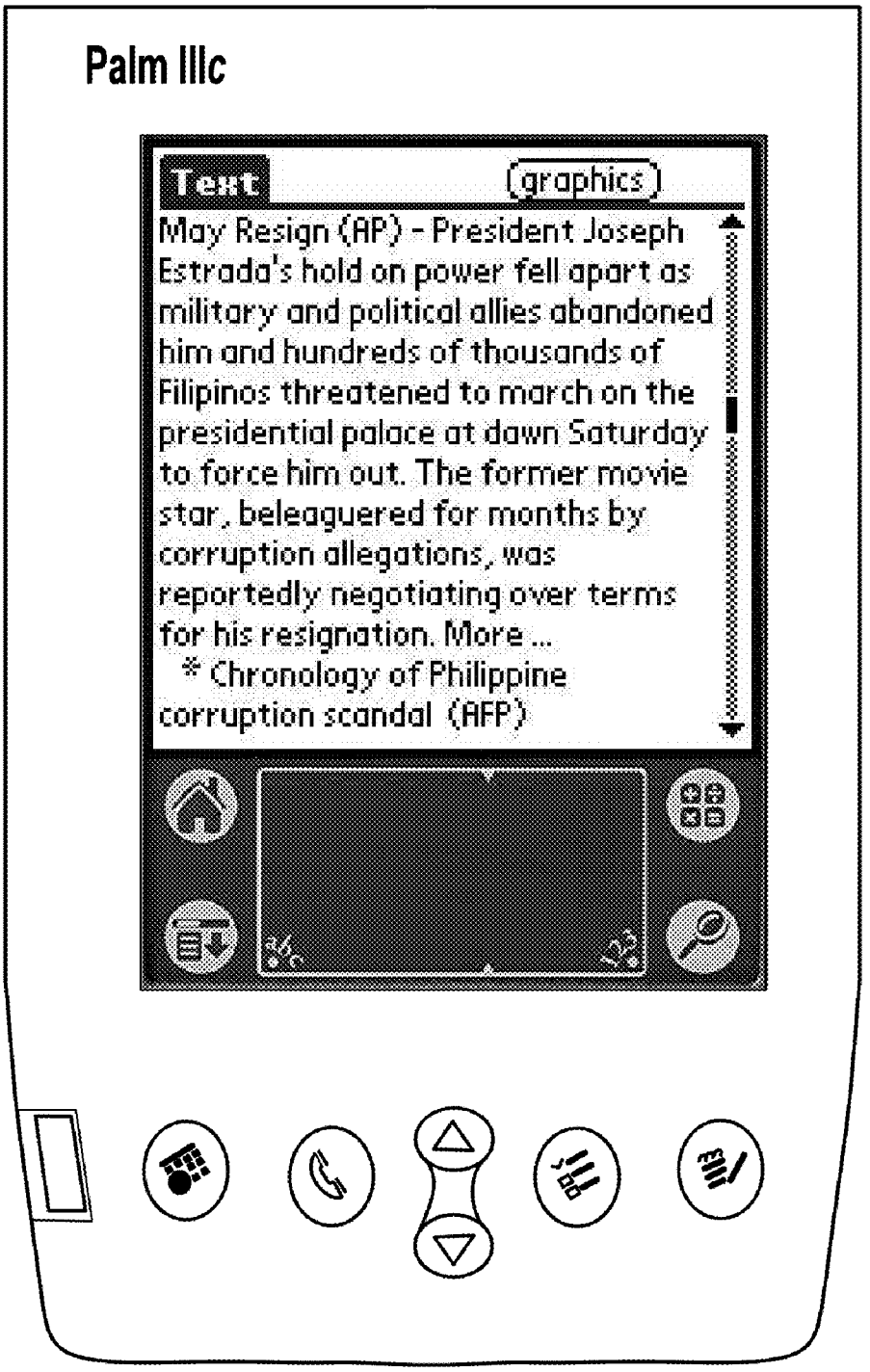

Palm IIIc
Text
graphics
May Resign (AP) - President Joseph Estrada's hold on power fell apart as military and political allies abandoned him and hundreds of thousands of Filipinos threatened to march on the presidential palace at dawn Saturday to force him out. The former movie star, beleaguered for months by corruption allegations, was reportedly negotiating over terms for his resignation. More ...
* Chronology of Philippine corruption scandal (AFP)

FIG. 9B

SCALABLE DISPLAY OF INTERNET CONTENT ON MOBILE DEVICES

RELATED APPLICATIONS

This application is a Continuation of U.S. Non-provisional application Ser. No. 09/878,097, filed Jun. 8, 2001, (issued as U.S. Pat. No. 7,210,099) entitled "RESOLUTION INDEPENDENT VECTOR DISPLAY OF INTERNET CONTENT," which is a Continuation-in-Part of U.S. Non-provisional application Ser. No. 09/825,511, filed Apr. 7, 2001, (Abandoned) entitled "RESOLUTION INDEPENDENT VECTOR DISPLAY OF INTERNET CONTENT," the benefit of the filing dates of which is claimed under 35 U.S.C. §120. U.S. Non-provisional application Ser. No. 09/878,097 further claims the benefit of the filing dates of U.S. Provisional Application No. 60/211,019, filed Jun. 12, 2000, entitled "METHOD AND SYSTEM FOR RESOLUTION INDEPENDENT DISPLAY OF HTML AND XML CONTENT" and U.S. Provisional Application No. 60/217,345, filed Jul. 11, 2000, entitled "METHOD AND SYSTEM FOR SELECTION, RETRIEVAL, AND CONVERSION OF COMPUTER CONTENT TO VECTOR FORMAT FOR RESOLUTION INDEPENDENT DISPLAY," under 35 U.S.C. §119(e). The disclosure of each of the foregoing applications is incorporated by reference in its entirety herein for all purposes.

This application also contains subject matter related to Divisionals (of Ser. No. 09/878,097) U.S. Non-provisional application Ser. Nos. 11/045,649 (issued as U.S. Pat. No. 7,584,423) entitled METHOD, PROXY AND SYSTEM TO SUPPORT FULL-PAGE WEB BROWSING ON HANDHELD DEVICES, and 11/045,757 (issued as U.S. Pat. No. 7,461,353) entitled SCALABLE DISPLAY OF INTERNET CONTENT ON MOBILE DEVICES, both filed Jan. 28, 2005. This application also contains subject matter related to U.S. Non-provisional application Ser. Nos. 11/735,477 and 11/735,482, both filed on Apr. 15, 2007, 11/738,932 filed on Apr. 23, 2007, 11/868,124 filed on Oct. 5, 2007, and 12/326,092 filed on Dec. 1, 2008.

COPYRIGHT NOTICE

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention relates generally to viewing of Internet content on mobile devices, and more particularly concerns to novel processing of Internet and World Wide Web content to scalable forms for resolution-independent rendering and zoom- and pan-enabling the display of content on mobile devices.

2. Description of the Related Art

Text only Internet information browsers began as a project at the CERN, European Organization for Nuclear Research, facility in Geneva Switzerland. From its inception the intent was to provide a mesh or web of access to data with a common user interface. Browsers moved from the academic environment when NCSA, the National Center for Supercomputing

Applications at the University of Illinois in Urbana-Champaign developed Mosaic, an Internet information browser and World Wide Web client.

Internet content is stored in multiple file formats. These formats include HTML (Hyper Text Markup Language) and XML (eXtended Markup Language) as well as graphic file format GIF (Graphics Interchange Format) and JPEG (Joint Photographic Experts Group). These four file formats constitute the majority of Internet content. Font size and resizing display area for content can alter the size of the display of Internet content in existing browsers. The majority of Internet content displays as a flat single resolution with no browser support for zoom.

Much of the Internet content has been designed for display on desktop computers with a single target resolution. Even though HTML has the ability to adapt to changes in screen resolution, major Internet content providers have chosen to create their Web pages using fixed resolution structures, such as tables. This gives them the ability to control the look and feel of their Web sites. This fixed resolution approach has evolved to the point that the fixed resolution layout of Web pages has become the most common method to brand or uniquely identify Web sites. While this fixed resolution approach is good for site branding and product differentiation it does present a daunting technical problem for display of Internet content (designed for desktop computers) on small screen, low resolution, or different aspect ratio devices, such as cell phones and hand held computers.

BRIEF SUMMARY OF THE INVENTION

In accordance with aspects of the invention, mobile devices enabled to support resolution-independent scalable display of Internet (Web) content to allow Web pages to be scaled (zoomed) and panned for better viewing on smaller screen sizes are disclosed. The mobile devices employ novel processing of original Web content, including HTML-based content, XML, cascade style sheets, etc. to generate scalable content. The scalable content and/or data derived therefrom are then employed to enable the Web content to be rapidly rendered, zoomed, and panned. Display lists may also be employed to provide further enhancements in rendering speed.

According to further aspects, the mobile devices employ touch-sensitive display screens that enable users to provide various inputs to control display of content within Web pages. Exemplary user inputs include tap-based inputs to selectively zoom in on columns, images, and paragraphs. Users can also define a window to zoom in on via the touch-sensitive display.

Other features of the present invention will be apparent from the accompanying drawings and from the detailed description that follows.

BRIEF DESCRIPTION OF THE DRAWINGS

The appended claims set forth the features of the invention with particularity. The invention, together with its advantages, may be best understood from the following detailed description taken in conjunction with the accompanying drawings of which:

FIG. **1**A is a block schematic diagram illustrating a first exemplary system infrastructure in accordance with the present invention in which content translation services are performed by a third-party proxy service that translates content requested from a client that is retrieved from one or more network resources into a scalable vector representation and delivers the translated content to the client;

FIG. **1**B is a block schematic diagram illustrating a second exemplary system infrastructure in which the translation of content is performed at a content provider's web site and delivered directly to the requesting client;

FIG. **1**C is a block schematic diagram illustrating a third exemplary system infrastructure in which content received from one or more network sources is translated into a scalable vector representation at the client;

FIG. **2**A is a flowchart illustrating how data is retrieved, processed and transferred in accordance with the system infrastructure of FIG. **1**A;

FIG. **2**B is a flowchart illustrating how data is retrieved, processed and transferred in accordance with the system infrastructure of FIG. **1**B;

FIG. **2**C is a flowchart illustrating how data is retrieved, processed and transferred in accordance with the system infrastructure of FIG. **1**C;

FIG. **3** is a block schematic diagram illustrating an exemplary architecture corresponding to the proxy server of FIG. **1**A;

FIG. **4**A is a representation of an exemplary web page has displayed on a conventional browser;

FIG. **4**B is a schematic diagram illustrates various objects that are generated based on the HTML code of the web page of FIG. **4**A;

FIG. **4**C is a schematic diagram illustrating a set of vectors and bounding boxes corresponding to the objects generated in FIG. **4**B;

FIG. **4**D is a schematic diagram illustrating how various vectors and bounding boxes may be defined in accordance with the invention;

FIG. **4**E is a representation of the web page of FIG. **4**A after it has been offset and scaled in accordance with the invention;

FIG. **4**F is a schematic diagram illustrating new datum points and bounding boxes corresponding to the scaled and offset web page;

FIG. **4**G is a schematic diagram illustrating new vectors and bounding box parameters for a pair of objects in the scaled and offset web page;

FIG. **5** is a flowchart illustrating the logic used by the invention when translating content into a scalable vector representation of that content;

FIG. **6** is a flowchart illustrating client-side operations that are performed to create a rendered display page based on the translated content the client receives and user-input;

FIGS. **7**A and **7**B are representations of a nominal and a zoomed in column view of an exemplary web page as they might appear on a Palm device;

FIGS. **8**A and **8**B are representation of nominal and zoomed in view of an exemplary graphic image as they might appear on the Palm device;

FIGS. **9**A and **9**B are representations of a nominal and zoomed in view of a text portion of a web page as they might appear on the Palm device; and

FIG. **10** illustrates an exemplary computer system that may be used for implementing various aspects of embodiments of the invention.

DETAILED DESCRIPTION OF THE INVENTION

Apparatus and methods are described for creating resolution independent vector display of Internet content to allow it to be scaled (zoomed) larger and smaller for better viewing or to fit any resolution or screen size. In addition, infrastructure and methods are provided for delivering such content to clients.

In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without some of these specific details. In other instances, well-known structures and devices are shown in block diagram form.

The present invention includes various operations, which will be described below. The operations of the present invention may be performed by hardware components or may be embodied in machine-executable instructions, which may be used to cause a general-purpose or special-purpose processor or logic circuits programmed with the instructions to perform the operations. Alternatively, the operations may be performed by a combination of hardware and software.

The present invention may be provided as a computer program product that may include one or more machine-readable mediums having stored thereon instructions, which may be used to program a computer (or other electronic devices) to perform a process according to the present invention. The machine-readable medium may include, but is not limited to, floppy diskettes, optical disks, CD-ROMs, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, magnetic or optical cards, flash memory, or other type of media/machine-readable medium suitable for storing electronic instructions. Moreover, the present invention may also be downloaded as a computer program product, wherein the program may be transferred from a remote computer (e.g., a server) to a requesting computer (e.g., a client) by way of data signals embodied in a carrier wave or other propagation medium via a communication link (e.g., a modem or network connection). Accordingly, herein, a carrier wave shall be regarded as comprising a machine-readable medium.

Client Overview

According to one embodiment, an ultra-thin client-side viewer provides the graphics, linking, caching, and function handling capabilities necessary for extending the web to almost any platform. It is designed as a lightweight browser (micro-browser) running directly on device operating systems. In alternative embodiments, the client-side viewer may be deployed as a standard browser plug-in, or Java applet for extending browser functionality. In one embodiment, the client-side viewer attains its small size and efficiency by taking advantage of the power of SVF (Simple Vector Format) to describe almost any current web content. SVF files can be handled with a tiny fraction of the client code required by normal web browsers because current browsers must interpret a large and growing number of file types and their idiosyncrasies. SVF was originally designed to handle a superset of the most commonly used file formats in the complex world of CAD. It can accommodate not only new graphical functions, but the storage and transfer of almost any foreseeable new functional capability. SVF has been under consideration by the W3C (World Wide Web Consortium) for adoption as a standard for vector content on the World Wide Web.

By working tightly with a server-side content translator, web content and functionality can be passed seamlessly to the end user platform without any degradation in the look or feel of the output. In addition, because the resulting file graphics are handled as vectors, the end user can control real time changes in the size of text and graphics as well as what portion of the file is viewable in the display. This "zoom and pan" capability, familiar to CAD and other vector content software users, adds dramatically to the usability of non-standard display sizes. For very small displays, real time zooming and panning allows the user to see graphics and text at sizes that make them easily readable, and then "back up" to view an

entire page for context or pan in any direction for navigation. Because the client-side viewer manipulates vectors, there is no loss in quality as the display is zoomed. The graphics rendering engine within the client is so efficient that file manipulation happens in a fraction of a second. There is no perceptible wait for the user as the file is resized, or the window is repositioned. Content created for one display resolution now can be sized, real time, for any other display without degradation. Besides making small displays eminently usable, this technology extends web content into some surprising new arenas. For example, it enables normal desktop displays to be effective for individuals with visual impairment, or for content designed for 640×480 standard PC monitors to be shown without degradation on web billboards now appearing in cities like Seattle and San Francisco.

With a client of such extraordinary power packed in a tiny footprint, end user device manufacturers can free up valuable memory space for pre-fetching, caching and pre-loading content, dramatically improving performance for use in low bandwidth and portable applications. In the example of a wireless handheld device where expensive flash memory must be used instead of more cost effective bulk storage technology, the difference between consuming 10's of megabytes of flash memory with a standard browser versus running the client-side viewer described herein is dramatic.

Those "saved" megabytes of memory are now available for impressive interfaces, caching of often used content, and pre-fetching of intelligently selected linked files or pre-loading of content for targeted applications. For example, in a mapping application, the map tiles surrounding the viewed map could be downloaded and stored while the user was working with the initial tile, enabling an experience remarkably free from the current frustrations of waiting for a new map to be transferred for even the smallest change in magnification or coverage. If the user knows ahead of time what city they will visit on a business trip, maps and additional travel information in great detail could also be pre-loaded using a high bandwidth connection at home or in the office before heading out to shop or conduct business in the city. Additionally, SVF is a more efficient way to store web content. Resulting content files are reduced in size by anywhere from 20 to 80 percent over their source. SVF is also very compressible. With target file size reduction in the range of 90%, SVF files can take up as little as $1/10^{th}$ the space of the web files in current use. This means that pre-translated content can be moved up to 10 times the rate of current web pages, and as much as 10 times as many pages, maps, stock charts, etc. can be stored for instant retrieval on the hand held platform as can be handled with current web technology.

When used on content created natively in SVF, additional capability can be extended to the client-side viewer.

Graphing the performance of stocks over time is only one use of SVF's ability to handle streams of data. Handling the output from financial systems, transactional systems, ERP packages, and CRM systems becomes easier and more flexible. Of course, systems integrators don't have to use these powerful capabilities to start with. If the target system provides web interfaces, these can be viewed, as designed, with no additional software to write, and no changes to the design or layout of the interface.

Server Overview

Enabling the client-side viewer to be so small and powerful is the server-side content translator. The server-side content translator rapidly translates Web content to SVF, compresses and encrypts the SVF results if desired, and transfers the vector formatted results to the client-side viewer. Alternatively, SVF files can be cached or stored in a file system for fetching and transfer at a later time. Pre-translated or cached content transfers are significantly faster as no conversion overhead is incurred, and file sizes are reduced using the more efficient SVF. Combine that with standard compression algorithms selectable for use with the client-side viewer for additional performance improvements.

During the translation process, and in the process of serving cached, pre-translated, or native SVF content, output files are "streamed" to the client-side viewer. Although this does not decrease the total time for file transfer, it can significantly improve the effective system performance for the end user. Content can be selectively streamed, with text and links coming through first, followed by graphic images and other content, for example. Should the user be accessing a link, rather than having interest in the entire file served, links can be selected early in the transfer and the next file download started immediately. In addition to streaming, the server-side content converter may also layer the content by type. This means that text can be put in one layer, links in another, GIF images in another, Javascript in another and so on. Layers can be turned on or off depending upon client capabilities, making files for less capable clients, or for users interested in a reduced functionality, higher transfer performance mode to be handled automatically.

All operational modes may be controlled through an administrative interface or accessible through a straightforward API (Application Program Interface). Furthermore, the system works with existing firewalls and within standard security protocols. In more secure modes, the server-side content converter and the client-side viewer may operate using Public/Private key authentication and encryption.

Exemplary System Infrastructures

In the following paragraphs, a description of three exemplary system infrastructures is provided. Schematic illustrations of these system infrastructures are shown in FIGS. **1**A, **1**B, and **1**C. It is noted that like-numbered components in these Figures perform substantially the same function. Therefore, any discussion of the functions of a component with reference to one or more of the infrastructures generally may apply to the other infrastructures as well, unless specifically noted otherwise.

A first of exemplary system infrastructure **10**A for implementing the invention is shown in FIG. **1**A. Infrastructure **10**A enables various clients, including wireless devices such as a cellular phone **12**, a wireless-enabled PDA **14**, and a wireless-enabled laptop computer **16**, as well as landline computers **18**, **20**, and **22**, to request content that is accessible via a network such as the Internet **24** to be retrieved from selected network resources, including web servers **26** and **28** and an FTP site **30**, wherein the content is translated into a scalable vector representation (e.g., SVF, also referred to herein as "vectorized content") through use of a proxy server **32** and sent to the requesting client. Upon being received by the client, the vectorized content is processed and rendered using a thin client to enable a user to view the content on the client device.

With reference to the flowchart of FIG. **2**A, the foregoing process is initiated by a client in a block **100**, wherein the client submits a request to proxy server **32** to retrieve and convert selected content. As depicted by a transfer path **34**, this comprises sending data **36**, which includes content network location indicia from which the content can be retrieved and proxy server network location information by which the content request may be delivered to over Internet **24** to proxy server **32**. Typically, it will be desired to retrieve a particular web page. Accordingly, the content network location indicia will comprise a URL (uniform resource locator) for the web

page. Similarly, the proxy server network location information may also comprise a URL corresponding to a network access point for the proxy server. Optionally, the location information may comprise a network IP address for one or both of the content location and the proxy server location. If the content is to be retrieved from an Internet resource, the request will typically be sent using the HyperText Transfer Protocol (HTTP) over the TCP/IP transport.

Next, in a block **102**, the request is received by the proxy server and the proxy server checks its cache to see if it already has the request content in its cache. If it does, it sends this cached content back to the client. If it does not have the requested content cached, the proxy server sends out a request to retrieve the content from the network resource. For illustrative purposes, it will be assumed for the present example that the desired content comprises a web page that is stored on web server **26**. Typically, when the requested content comprises a web page, the content may be retrieved using conventional web content retrieval techniques, such as that employed by various modern browser clients, including Netscape Navigator and Internet Explorer. This generally comprises providing routing information, such as the URL for the web page (URL **38**) to routing services provided by Internet **24**, which routes the request to an appropriate network resource (e.g., web server **26**), as depicted by a transfer path **40**.

Typically, the URL will correspond to a web page whose content is stored by the web server in an HTML (HyperText Markup Language) document comprising HTML code and embedded text content, in addition to other optional content languages, that may contain references to other objects (e.g., HTML documents and graphic image files) stored locally to the server or stored on a remote server. For example, the HTML content corresponding to a single-frame web page is often stored in a single file, while multiple-frame web pages may comprise content that is stored in a single file or in multiple files. These files may be stored locally on the web server (e.g., on one of the server's hard disks), or on a local storage device connected to the web server via a local area network (LAN), such as a network attached storage (NAS) filer. Optionally, some of the web page's content may comprise one or more documents that are stored at remote locations that may be accessed via a WAN (wide area network) or the Internet.

HTML is a standardized language that describes the layout of content on a web page, and attributes of that content. This layout and attribute information is defined by sets of tags contained in HTML code corresponding to the page. The tags define various HTML layout and display information, including tables, paragraph boundaries, graphic image positions and bounding box sizes, typeface styles, sizes, and colors, borders, and other presentation attributes. A portion or all of a web page's text content may be contained in the parent HTML document corresponding to the URL. In addition to basic HTML, web page documents may contain XML (eXtensable markup language) code, as well as scripting language code, such as javascript. However, for simplicity, any documents containing web page content other than only graphic content that are discussed herein will be referred to as HTML documents.

In addition to HTML and other markup and scripting language content, it is very common for web pages to include graphical content. In general, graphical content is usually stored in an image file or files that are external from the parent HTML document for the web page. For example, the parent HTML document may contain one or more embedded image tags that reference the location where those images are stored. As before, the graphic images may be stored locally, or may be stored on remote servers that are accessed by the web server via a WAN, or the Internet. These files will typically comprise data stored in one of several well-known graphic formats, including bitmap files (BMP), GIF (Graphics Interchange Format) files, and JPEG (Joint Photographic Experts Group) files.

In response to receiving the request for content, web server **26** begins sending a parent HTML document **42** back to proxy server **32** in a block **104**. In a block **106**, the HTML content of the parent HTML document is parsed to search for references to external objects such as HTML frames and graphics. In a decision block **108**, a determination is made to whether any references are found. For each reference to an external object that is found, proxy server **32** requests to have the object retrieved from an appropriate network resource (e.g., a web server) in a block **110**, and data corresponding to the object is transmitted back to the proxy server, as depicted by locally accessible HTML documents **44** and graphic images **46**, as well as remotely accessible HTML documents **48** and graphic images **50**, which may be accessed via web server **28**. If the external object is a graphic image, there is no further processing of the object at this point. If the object is an HTML document, the functions provided by blocks **106** and **108** are repeated. Generally, this set of processing functions is repeated iteratively until all of the external objects are retrieved. However, as described below, there will be some instances in which certain objects will be retrieved at a later point in time. In addition to content stored on web servers that are accessed using HTTP, content may also be retrieved from various network sites using the File Transfer Protocol (FTP), such as FTP documents **51**, which are accessed via FTP server **30**.

In general, HTML documents and graphic files will be sent as packetized data streams using HTTP over one or more TCP/IP network connections, wherein the data streams will usually be asynchronous. Retrieval of HTML documents and graphic files corresponding to the embedded references will usually require additional transfer time. Furthermore, graphic content oftentimes comprises significantly larger file sizes than HTML content, leading to significant transfer times in some instances. For simplicity, the transfer of the various HTML documents and graphic files for the content request are depicted by HTML documents **52** and graphic documents **54**, which are transferred over a transfer path **56**.

When the HTML documents and graphic content are received by proxy server **32**, a scalable vector representation of the web page is generated in a block **114** by an HTML translator **58**. In brief, HTML translator **58** translates HTML, XML, and cascaded style sheet (CSS) layout content into a scalable vector representation, such as SVF. Details of the HTML translation process are contained below. In addition, the graphic images are converted into a compressed bitmap format in a block **116** by a graphics translator **60**. The vectorized content **62** and compressed bitmaps **64** are then streamed back to the client (i.e., computer **18**) in a block **118**, as depicted by a transfer path **66**. In one embodiment, the content portions are sent in separate streams using multiple connections. In another embodiment, the content portions are sent via a multiplexed stream using a single connection. As the vectorized content and compressed bitmap data are received by the client device, they are processed by a thin client **68** running on the client device, whereby a representation of the original web page content may be rendered on the client device's display screen at various user-selectable scaled resolutions and pan offsets in a block **120**, thereby

enabling a user to more clearly see an overview or details in the web page. Further details of the client side processing are provided below.

As discussed above, wireless clients may also access the vectorized network (e.g., web site) content provided via proxy server **24**. The majority of this process is identical to that described above for land-line clients (e.g., computers **18**, **20**, and **22**), except for provisions required for sending data to and receiving data from wireless devices. In general, most wireless devices will access the Internet via a wireless service provider (i.e., a wireless telecommunications carrier) that is particular to that wireless device. Accordingly, a portion of the transmission path to and from proxy server **24** will comprise infrastructure provided by that service provider and/or shared with other service providers. For simplicity, this infrastructure is shown as a cellular tower **70** and a service provider data center **72**, although it will be understood by those skilled in the art that the connection path may comprise additional infrastructure components, including appropriate gateways and routers, that enable wireless devices to access proxy server **24**.

In some implementations, there will be no special formatting/protocol services that need to be performed by proxy service **24**—from the viewpoint of the proxy service, it will be immaterial whether the client is a land-based or wireless client; the special handling provisions for wireless devices will be handled entirely by the service providers infrastructure transparently at both ends of the communications path. In other instances, it may be desired or necessary to reformat the data content delivered to the wireless device at the proxy service. This will generally be dependent on the particular wireless protocol used, and what services are provided by the service provider for the wireless client.

Currently, in the United States, wireless clients generally access Internet **24** by using the Wireless Application Protocol (WAP). In Japan, the most popular access means is NTT DoCoMo's i-Mode wireless protocol. In addition to these wireless standards, new standards are anticipated to be in force in the near future, including NTT DoCoMo's FOMA (Freedom of Mobile Multimedia Access), which is transported over W-CDMA (Wideband Code Division Multiple Access), and CDMA-2000. For the purposes of the invention herein, it will be understood that those skilled in the mobile telecommunications arts will be knowledgeable about any particular format and/or transport protocol requirements that pertain to the particular protocol that is to be used.

A second exemplary system infrastructure **10**B for implementing the invention is shown in FIG. **1**B. As will be readily recognized, much of infrastructure **10**B is similar to infrastructure **10**A; however, rather than have a separate proxy server perform the proxy functions (retrieve and translate content), these functions are performed on machines operated by the web site in infrastructure **10**B.

The logic implemented by the invention when providing content to a client using infrastructure **10**B is illustrated in the flowchart of FIG. **2**B, wherein the process begins in a block **101** in which the client sends a content request **39** directly to the network site (e.g., web server **26**), as depicted by a transfer path **41**. In a block **103**, HTTP negotiations are performed to determine the format the content is to be delivered in. For example, the request may contain indicia identifying the type of content requested, such as an SVF MIME type (e.g., image/vnd.svf). This is to inform the web server that the request is for specially-formatted content rather than conventional content. The server first checks to see if it already has cached the requested content. If it has, it sends the content to the requesting client; otherwise, it retrieves the parent HTML document in a block **107**. It then performs processing steps in blocks **107**, **109**, and **111** to retrieve content referenced through embedded tags in a manner substantially similar to that discussed above with reference to respective blocks **106**, **108**, and **110**. The primary difference in this instance is that the web server does not receive requests from or send documents to a proxy server—rather, the content is retrieved and processed at the web server, wherein the retrieved content may be stored local to the web server or retrieved from a remote server in a manner similar to that described above.

As before, the retrieved HTML documents are translated into scalable vector representations by HTML translator **58** in a block **114**, while the graphic images are translated into a compressed bitmap format by image translator **60** in a block **116**, as depicted by vectorized content **62** and bitmap content **64**. The vectorized content and bitmap content are then streamed from the web server to the client in a block **119**, as depicted by a transfer path **67**. Upon arriving at the client, the vectorized content and bitmap content are processed, scaled, and rendered on the client in a block **120**.

A third exemplary system infrastructure **10**C for implementing the invention is shown in FIG. **1**C. In this configuration, the proxy functions are performed at the client. As shown by a block **113** in FIG. **2**C, the process for providing vectorized content to a client in accordance with infrastructure **10**C begins in a block **113**, in which the client sends a content request **37** to a network site, such as web server **26**, via Internet **24**. In response, the network site retrieves the parent HTML document and sends it to the requesting client in a block **115**. In a manner similar to that discussed above with reference to blocks **106**, **108**, and **110** of FIG. **1**A, the client first parses the parent HTML document searching for embedded references to external objects and retrieves these objects, whereupon the embedded reference search is performed on the newly retrieved document until all of the content corresponding to the original content request has been retrieved. This content is depicted by HTML documents **52** and image files **54**, which are sent from the network site to the client via a transfer path **69**. At this point, the client performs translations on the HTML content and the graphic image content that are substantially similar to that performed by the proxy server in FIG. **1**A or at the web site in FIG. **1**B, as provided by blocks **114** and **116**. The vectorized and image content is then processed and scaled by thin client **68** in a block **120**, as depicted by device output **71**.

Attention now is focused on the functionality provided by proxy server **24** in system infrastructure **10**A of FIG. **1**. Fundamentally, the proxy server functions as a proxy. It accepts requests for content from client devices as full URLs using standard HTTP mechanisms carried over a multiplexed TCP connection. Standard HTTP content negotiations features specify the formats in which content is to be delivered (SVF, bitmap, and possibly others, which can be handed off to cooperating client-side display software). As described in further details below, in some embodiments the proxy server appears for the client as a normal proxy (that is, the client knows it is retrieving content via the proxy), while in other embodiments the proxy is transparent to the client.

The proxy server responds to client content requests by delivering content in one of the requested formats, by retrieving the content in an appropriate format from its cache, or from an upstream content source (again using standard HTTP content negotiation features), or by translating upstream content from a supported original format to SVF or the client bitmap format.

Requests from the server installation to its cache and from the cache to upstream content sources are made in HTTP

carried over TCP using simple straightforward Web content requests. For example, requests from clients to the proxy server comprise HTTP proxy requests (e.g., "GET http://www/xyz.com/some_page.html HTTP/1.0 . . . ") carried over TCP or over a lightweight multiplexing protocol over TCP. The multiplexing protocol allows the server to push image thumbnails to the client before the SVF stream is available, as well as offering a channel for control and status information, more simultaneous channels than the client operating system may support, and a mechanism for prioritizing information flow from server to client under loose client control. In addition to HTTP requests, the proxy server architecture supports other user-level protocols, such as FTP and Gopher.

Details of some of the primary components of the proxy server architecture are shown in FIG. **3**. Internally, the proxy server comprises a suite of coordinated processes connecting to upstream content through an HTTP cache **74**. In one embodiment all functions except caching are performed in a single process, wherein multiple threads are used to effect asynchronous I/O. Separate processes communicated via persistent multiplexed connections carried over the most efficient reliable transport available (e.g., Unix sockets over single processor and symmetric multiprocessor (SMP) computers; TCP sockets between separate computers). All processes are capable of servicing multiple requests simultaneously. No process maintains client state outside the context of a single request, so all components can be repeated and load balanced across multiple CPU's of an SMP computer or across separate computers on a LAN.

The various content translators used by the proxy server accept (via HTTP PUT) or request (driven by HTTP proxy GET/POST) content in supported, but client-unsupported, formats; and return (via HTTP PUT or GET/POST response) one or more representations of that content in a client-supported format. In the embodiments illustrated in FIGS. **1**A-C, two translators are used: HTML translator **58** and image translator **60**. Future content types may be accommodated by new translators, by extending existing translators to cover the new content types, or by extending the client's capabilities. Standard HTTP content negotiation mechanisms are used to inform the proxy server of the client's capabilities and expectations on each request.

Managers at the proxy server coordinate the operations of other components. Two managers are presently defined; a client manager **73** that handles client proxy requests, and a request manager **75** that handles unproxied HTTP requests from other services. The managers accept requests, attempt to service them from HTTP cache **74**, and drive HTML translator **58** and image translator **60** when content does not match the clients' requirements. Managers also handle translator requests for inline content (e.g., image dimensions for page layout), and push translated content into HTTP cache **74**. Additionally, the client manager coordinates delivery of primary and inlined content, and provides process and status information to the clients.

As discussed above, HTML translator **58** creates a scalable vector representation of the original HTML content of a requested web page. In order to better explain how translation of HTML content is performed, one embodiment of a translation process is described below as applied to an exemplary web page. In addition, details of conventional web page client and server-side processing are provided so as to clarify how web content is laid out during a pre-rendering process on the client.

FIG. **4** shows a representation of a web page **210** served from an exemplary stock brokerage Internet web site as it would appear when rendered on a modern Internet browser, such as Microsoft's Internet Explorer or Netscape's Navigator. Web page **210** is exemplary of many web pages that implement frames, and includes two adjacent frames **212** and **214**. A logo graphic object **216**A is displayed at the top of frame **212**, which additionally includes a "MARKETS" text header **218**A, an "INVESTMENTS" text header **220**A, and a plurality of links with overlaying graphic objects, including a "DOW" link **222**A, a "NASDAQ" link **224**A, an "OPTIONS" link **226**A, a "CHARTS" link **228**A, a "MUTUAL FUNDS" link **230**A, a "IRA, 401K OPTIONS" link **232**A, and a "TAX INFORMATION" link **234**.

A horizontal group of links **236** is disposed at the top of frame **214**, and includes a "QUOTES" link **238**A, a "HOT PICKS" link **240**A, a "CALENDARS" link **242**A, and a "NEWS" link **244**A. An advertisement banner **246**A is displayed just below the horizontal group of links and just above a "NEWS SPARKS MARKET" headline **248**A. Frame **214** also includes a pair of graphic image objects, including a DOW chart **250**A and a NASDAQ chart **252**A. A set of user input objects is disposed adjacent to DOW chart **250**A within a graphic object **254**A, including an "ACCOUNT #" input box **255**A, an "ACCESS CODE" input box **256**A, and a "LOGIN" button **257**A. In addition to the foregoing objects, frame **214** also includes text objects **258**A and **260**A.

An HTML listing corresponding to web page **210** is presented below as LISTING 1. Note that LISTING 1 sometimes refers to object descriptions and link paths rather than the text or path location of actual objects for simplicity, and that other elements commonly found in HTML pages, such as META entries, are omitted for clarity.

LISTING 1

```
1.  <html>
2.  <head><title>"MARKET HOME"</title></head>
3.
4.  <body bgcolor="#FFFFFF" link="0033CC" vlink="0033CC">
5.
6.  <frameset cols="25%,75% frameborder=0 border=0>
7.  <frame>
8.  <align=left><align=top>
9.  <img src="/directory path/logo.gif" align = left border="0" height="80" width="100">
10. <br><br>
11. <t3>TEXT HEADER #1 align=left</t3><br>
12.
13. <table width="90%" border=0 cellspacing=10 cellpadding=0 bgcolor="#000000"
14. align=center>
15.       <tr>
```

-continued

LISTING 1

```
                        <a href="URL or path for LINK #5" <img src="/directory
                        path/GRAPHIC#2" height="50" width ="150></a>
        <tr>
                        <a href="URL or path for LINK #6" <img src="/directory
                        path/GRAPHIC#3" height="50" width ="150></a>
        <tr>
                        <a href="URL or path for LINK #7" <img src="/directory
                        path/GRAPHIC#4" height="50" width ="150></a>
        <tr>
                        <a href="URL or path for LINK #8" <img src="/directory
                        path/GRAPHIC#5" height="50" width ="150></a>
</table>
        <br>
                        <t3>TEXT HEADER #1 align=left</t3>
        <br>
<table width="90%" border=0 cellspacing=10 cellpadding=0 bgcolor="#000000"
  align=center>
        <tr>
                        <a href="URL or path for LINK #9" <img src="/directory
                        path/GRAPHIC#6" height="50" width ="150></a>
        <tr>
                        <a href="URL or path for LINK #10" <img src="/directory
                        path/GRAPHIC#7" height="50" width ="150></a>
        <tr>
                        <a href="URL or path for LINK #11" <img src="/directory
                        path/GRAPHIC#8" height="50" width ="150></a>

</table>
</frame>

<frame>

<table>
        <tr>
                        <table width="100%" border=0 cellspacing=15 cellpadding=0
                          bgcolor="#000000" align=center>
                <tr>
                        <td><a href="URL or path for link#1"> alt="QUOTES"</a>
                        <td><a href="URL or path for link#2"> alt="HOT PICKS"</a>
                        <td><a href="URL or path for link#3> alt="CALENDERS"</a>
                        <td><a href="URL or path for link#4>alt="NEWS"</a>
</table><br>
          <br>
                        <img src="URL for GRAPHIC #9" align=center
                          border="0" height="80" width="325">
        <br><t1>HEADLINE TEXT>/t1>
        <table>
                  <Colgroup span="2">
                        <Col width = "400" align="center">
                        <Col width = "200" align="center">
                        <tr><td>
                              <img src="/directory path/GRAPHIC #10" align = center
                              border="0" height="180" width="350">
                        <td>
    /* INPUT FOR ACCOUNT NUMBER AND ACCESS CODE */
      <SCRIPT LANGUAGE ="Javascript">
    <!---
                        [Javascript variable declarations]
                        [Javascript functions to enable login] --->
    </SCRIPT>
            <table>
                    <td>
                        <img src="/directory path/GRAPHIC #11" align = center>
                        <table width="150" height="25">
                        <td>
                        <font size=-2 face="arial,helvetica,verdana">Account #</font>
                        <tr><input type=text name="USERID" maxlength=9 size=20>
                        <tr><font size=-2 face="arial, helvetica">Access Code:</font>
                        <tr><input type=password name="PASSWORD" maxlength=10 size=20
                              onKeyDown="SuppressEnterBell(event)"
                              onKeyPress="SuppressEnterBell(event)"
                              onKeyUp="SubmitOnEnter(event)">
                        <br> 
                        <br><input type="button" value="Login"
                        OnClick="ProcessForm( )">  <input type="reset">
```

-continued

LISTING 1

```
91.                         <br> 
92.                         </td>
93.                         </table>
94.                 </table>
95.       <tr>
96.                         <img src="/directory path/GRAPHIC #12" border="0
97.                          height="200" width="350">
98.       <tr>
99.                             <p>TEXT FOR TEXT OBJECT #1</p><br>
100.                            <p>TEXT FOR TEXT OBJECT #2</p>
101.                </table>
102.             </frame>
103.    </frameset>
104.    </html>
```

Web page documents comprise HTML code that is parsed, interpreted, and rendered by a browser. An HTML document comprises a plurality of HTML "markup" elements (tags) with corresponding attributes, that are used to describe the layout and formatting of various objects, including plain text and graphic objects, embedded between tag pairs. Exemplary elements include text tags (e.g., for bolding text), links (e.g.,), formatting (e.g., for creating a new paragraph, graphical (e.g., <img src="name">), wherein "name" defines an absolute or relative location at where an image is stored, tables (e.g., <table></table>) creates a table, and forms (e.g., <form></form> creates all forms).

As of Netscape Navigator 3.0 (and other later browsers), web pages could include frames. When using frames, the display page is divided into multiple framed areas. Framing enables a single display page to include source code from several HTML documents (one for each frame) or optionally, enables a single document to include more complicated grouping of contents whereby different content groups are contained in separate frames. Frames are commonly found on the web pages at sites that display a great deal of text and graphical content, such as MSN.com, ESPN.com, and USA-Today.com.

With reference to the flowchart of FIG. **5**, the process for translating the HTML content into a scalable vector representation proceeds as follows. The process is initiated when the proxy server receives the HTML corresponding to the parent document (and frame documents, if appropriate), whereupon a pre-rendering parsing of the HTML is performed to determine where to place the various objects on the display page in a block **150**. For example, elements such as tables, column definitions, graphic images, paragraphs and line breaks are identified. If frames are included, each frame is examined in the sequential order it appears in the HTML document, or the order in which the HTML documents corresponding to the frames in a frameset are downloaded to the browser. During further processing, the actual objects are rendered in their respective positions. Some of these objects are rendered almost immediately, such as plain text, while other objects, such as graphic objects, must first be retrieved prior to being fully-rendered. With respect to tables, there are some instances in which all of the objects corresponding to the cells in the table must be retrieved prior to rendering any of the table, while a well-designed table can be rendered incrementally. For example, by using Column grouping, the format of the corresponding table can be quickly determined by the browser. In some instances, one or more bitmaps may actually need to be fetched before the page layout can be determined.

Next, in a block **152**, the content is separated into objects based on logical groupings of content portions and a page layout is built using bounding boxes that are produced for each object. As the primary HTML document is parsed, logical groupings of content will emerge. For instance, text content contained within paragraph tags forms a logical grouping of text content. In essence, a logical grouping means the content should appear together as a logical group, such as within a substantially rectangular outline, in the rendered page. Other logical groupings include frames, table content, row content, single line entries such as headlines and headers, and user-interface objects, as well as graphic layout objects, such as separator bars, and graphic images. In addition to logically grouping content into objects, a "bounding box" is defined for each object. In general, the bounding box defines an outlined shape within which the content (text or graphic image) will appear. In most instances, the bounding box will be substantially rectangular in shape. However, bounding boxes comprising more complex shapes may also be produced.

In further detail, the following explains how objects corresponding to graphic images are produced. In HTML, objects comprising graphic content are identified by an <img src="/local directory path/graphic image file" (for a local graphic image) or "URL" (for a remote graphic image)> or <object> or other tags. In the foregoing tag, local graphic images are typically stored on the same server as the web page, or another computer that is local to the site's server, and generally are located through a local directory path (absolute or relative to the location of the present page) that points to the graphic image file. Remote images are those images that are stored on servers at sites that are remote to the web server. For example, with reference to LISTING 1, when the parser encounters line 9, the browser identifies that data comprising a graphic image corresponding to logo graphic object 1 will be arriving (or may have already been received), and the displayed image is to have a height of 80 pixels and a width of 100 pixels. The location of each object on a display page will be dependent on previous HTML layout elements, such as tables, paragraphs, line breaks, and other graphic objects. The size and location of the other graphic objects (i.e., graphic objects #2-12) on the page are determined in a similar manner. The HTML code for these objects are shown in lines 16, 19, 22, 25, 34, 37, 40, 59, 67, 78 and 96, respectively. As identified in the HTML code, data corresponding to graphic objects #9 (advertisement banner **46**A) is forwarded to the browser from an external site (as indicated by the URL to GRAPHIC #9), while graphic objects 1-8 and 10-12 are sent from the web site the parent HTML document is sent from.

In a similar manner, the foregoing technique is applied to the HTML code in the primary document to identify other types of objects as well. In addition to parsing the primary HTML document, similar processing is performed on referenced documents, such as documents that include frame content that is defined and stored separate from the primary HTML document.

A representation of the results of the functions performed in block **152** are shown in FIG. **4**B. In the Figure, objects corresponding to the original content of FIG. **4**A are shown with an appended "B" that is added to each object's root reference number, wherein the root reference number for an object is that same as the logically grouped content in FIG. **4**A that it corresponds to, e.g., an object **248**B is generated for "NEWS SPARKS MARKET" headline **248**A, etc.

Next, in a block **154**, the page layout is defined based on the bounding boxes. In actuality, generation of the page layout information is performed in conjunction with defining the boundary boxes for the objects, wherein the location of a given object is based on the location of other related (e.g., if within a table) or non-related objects corresponding to HTML content that have been previously parsed. For example, the location of a given paragraph will depend on the other content for the page that are listed prior to the definition for the paragraph in the primary HTML document or referenced document, if applicable. As the HTML content of the primary and any referenced HTML documents are parsed, the page layout is generated based on the various HTML tags and the content embedded between tag pairs and/or referenced by a tag pair statement (e.g., graphic images).

As will be recognized by those skilled in the art, the functions performed in blocks **150**, **152**, and **154** are commonly performed by conventional browsers during a pre-rendering process. In some browsers, these functions are performed by the Mozilla rendering engine, which comprises open source software that is readily available for use by developers. At present, the software for the Mozilla rendering engine may be accessed via the Internet at www.mozilla.org. Accordingly, in one embodiment, the present invention uses core functionality provided by the Mozilla rendering engine source code to perform the functions of block **150**, **152**, and **154**.

At this point, the present invention deviates substantially from the prior art by using the various object layout data generated during the pre-rendering process to generate a scalable vector representation of the original page content. First, in a block **156**, a datum point is defined for the page and the bounding box for each object. For example, as shown in FIG. **4**C, a rendered page datum **262** is defined to be coincident with the upper left hand corner of the display frame of the rendered page for the web page. Generally, any point on the page may be used as the page datum—the only requirement is that the page datum that is selected is used consistently throughout the process. The use of the upper left hand corner of the display frame is advantageous since the location of the first object encountered in the HTML code for a page is located relative to this corner.

In general, the datum points for each object may also be located any place on the object, as long as the object datum points are used in a predictable manner. For example, as depicted in FIG. **4**C, various datum points for corresponding objects are defined to be coincident with the upper left hand corner of the bounding box for that object, wherein the object's datum point shares the root reference number of the object with an appended "C."

Once the page's datum point and an object's datum point are known, a vector between these points is generated for each object in a block **158**. With reference to FIG. **4**D, in one embodiment, wherein the page datum point corresponds to the upper left and corner of the display frame and is assigned an XY value 266 of 0,0, the vector for a given object may be stored as the XY value of the datum point of that object relative to 0,0, such as a value of 150, 225 (ref. num. **268**) for a vector **250**D pointing to an object datum **250**C, and a value of 150, 425 (ref. num. **270**) for a vector **252**D pointing to an object datum **252**C. In another embodiment, each vector may be stored as XY data relative to a 0,0 datum point corresponding to the upper left hand corner of the frame the object belongs to. For example, a vector **250**D' from a frame datum **214**D to object datum **250**C is stored as 20, 200 (ref. num. **268**'), while a vector **252**D from frame datum **214**D to object datum **252**C is stored as 20, 425. In this embodiment, offset information for each frame relative to a known datum will also be stored, as depicted by a vector **214**D.

The scalable vector representation is completed in a block **160**, wherein a reference is created for each object that includes or links an object's content and attributes, such as object type (e.g., text, image), object typeface, and boundary box parameters, to the object's vector. For example, object **250**B is a graphic image having a vector **250**D and a bounding box that is 180 pixels high and 350 pixels wide, while object **252**B is a graphic image having a vector **252**D and a bounding box that includes a height of 200 pixels and a width of 350 pixels. This enables client-side operations to be performed that only initially consider the vectors, wherein if it is determined that a vector's endpoint (and/or the bounding box corresponding to the object the vector points to) would appear off of a display, there is no need to retrieve the content and attribute data linked to the vector. This concept is explained in further detail in the following section.

It is noted that a portion of the display content produced on a client device will never contain any rendered content, as this portion is reserved for the browser's user interface. In WINDOWS™ environments, this portion will include the browser's window frame, as well as the pulldown and icon menus provided in the browser's user interface, which are depicted by a box **264** in the Figures herein.

Client-Side Software and Processing

As discussed above, the present invention supports a wide variety of clients, including land-based clients and wireless clients. Each client requires some client-side software that enables the scalable vector content data provided to it to be rendered at a user-selectable scale factor and offset on the client's display, such as a monitor or built-in LCD screen.

By enabling original content from a web site to be displayed in such a resolution-independent manner, users will be able to view content in a manner that did not previously exist, greatly enhancing the user experience. For example, in some implementations the client may be a personal computer (PC). Using a least-common denominator approach, many web pages are designed for a smaller resolution (for example 640×480 pixels, a minimum resolution commonly supported by nearly all PC's, including legacy PC's) than the resolution provided by the video output capabilities available with many of today's PC's, such as 1024×768 pixels, 1280×1024 pixels, and even 1600×1200 pixels. As a result, when these web pages are displayed on a high-resolution display, they occupy only a portion of the display, making portions of the pages, especially those portions containing small text, difficult to read. By enabling users to selectively magnify the entire page, these design flaws are easily overcome. Alternatively, the client may be a small device, such as a hand held computer or a cell phone, which has a smaller display resolution than common Web pages are designed for. As explained below, through use of the invention's scalable vector representation

and client-side processing, users are enabled to view the entire content of billions of existing Web pages using handheld devices in a simple and reasonable way.

In one embodiment, the client software may be a plug-in to a Web browser, such as Netscape Navigator or Microsoft Internet Explorer. Such a plug-in might have the browser download the data and display it in a sub-window of the browser. Alternatively, the client software may be a Java applet running in a browser. As another option, the client software may be a stand-alone program that interfaces with the proxy server or proxy software directly. The client software may bypass the proxy when requesting information that won't be translated to vectors, such as bitmaps.

With reference to FIG. **6**, client-side processing proceeds in the following manner. In a block **160**, the vector representation data (i.e., vectorized HTML content and compressed bitmap content) for the web page is gathered at the client. Typically, this data will be stored in a cache at the client as it is being received, and the client simply retrieved the data from the cache. In a block **162**, a display list of vectors is built. This process is well known in the CAD arts, and is enabling rapid zooming of vector-based objects. In a block **164**, user selectable scale and offset (pan) values are determined. Based on various user interactions with the user-interface of the client, the user is enabled to control the zoom (size) and offset of the rendered page. For example, suppose the user provides zoom and offset inputs to produce a rendered page **210**E, as shown in FIG. **4**E. In this rendered page, the original origin is now off of the screen (the page image is shifted upward and toward the left—see FIG. **4**F), and the view has been scaled approximately 1.3 times.

Next, in a block **166**, the vectors and boundary boxes are processed based on the scale and offset, and a bounding box defining the limits of the display content is determined. The results of this step are shown in FIG. **4**F, while FIG. **4**G shows specific details on how the vectors and bounding boxes corresponding to image objects **250**B and **252**B (now **250**B' and **252**B', respectively) are processed. Logically, there are generally two ways to scale and offset the rendered content. In one embodiment, vectors and bounding boxes are mapped to a virtual display area in memory that has much greater resolution (e.g., 100,000×100,000 pixels) than any real display, and a virtual display limit bounding box is scaled and moved around over the virtual display area. Accordingly, during subsequent processing described below, objects falling within the display bounding box are rendered by reducing the scaling of those objects in the virtual display to how the objects will appear on the client device display relative to the virtual display bounding box. In the alternate, a fixed reference frame corresponding to the display resolution of the client device screen is maintained, wherein all vectors and bounding boxes are scaled and offset relative to the fixed reference frame. Each scheme has its advantages and disadvantages. One advantage of the second method is that the display bounding box is always maintained to have a size that matches the resolution of the content display area on the client device.

As shown in FIG. **4**G, respective offsets in X and Y, (−ΔX and −ΔY in the Figure) are applied to the starting point of each of the vectors. The vectors are then scaled by a scale factor "SF." The results of the new vectors are depicted by vectors **250**D" and **252**D". This produces a new datum for each object's bounding box that is relative to rendered page datum **262**, which remains fixed. As discussed above, only a portion of the display screen will actually be used to display content (as defined by a display limit bounding box **266** in this embodiment), while other portions of the screen, including box **264**, will comprise a generally fixed-size user interface. Accordingly, rendered page datum **262** is not located at the upper left hand corner of the display area, although it possibly could be located at this point when either the current user interface is inactive (i.e., the display portion of the user interface is temporary disabled) or the user interface is contained in other portions of the display.

This foregoing process establishes a starting point (the new datum) for where the content in each object's bounding box will be rendered. At this point, each object's bounding box is then drawn from its new datum using the scaling factor. For example, in the original web page **210**D (FIG. **4**D), bounding box **250**B had an X-axes datum of 150 pixels, a Y-axis datum of 225 pixels, and a height and width of 180×350 pixels. In contrast, after being offset and scaled, bounding box **250**B' has an X-axis datum of 150*SF−ΔX, a Y-axis datum of 225*SF−ΔY, and a height and width of 180*SF×350*SF.

Returning to the flowchart of FIG. **6**, once the vectors and bounding boxes are offset and scaled, content corresponding to objects having at least a portion of their bounding boxes falling within the display limit bounding box is retrieved from the client device's display list in a block **168**. For examples, as shown in FIG. **4**F, content corresponding to all of the objects except for those falling entirely outside of display limit bounding box **266** (objects **216**, **238**, **240**, **242** and **244**) is retrieved from the display list. That content is then scaled in a block **170**. For image content, this comprises decompressing and scaling the compressed bitmaps corresponding to those images. For text content, this comprises scaling the font (i.e., typeface) that the text content portions of the web page are written in the parent HTML document and any referenced documents. There are various techniques for typeface scaling that may be implemented here, depending on the available resources provided by the operating system of the client device. For example, for WINDOWS™ operating systems, many TRUETYPE™ fonts are available, which use a common scalable definition for each font, enabling those fonts to be scaled to just about any size. In other cases, such as current PDA (e.g., Palm Pilots) operating systems, there is no existing feature that supports scaling fonts. As a result, bitmapped fonts of different font sizes and styles may be used. In addition to scaling image and text content, other types of content, such as separator lines and borders may also be scaled by block **170**.

The process is completed in a block **172**, wherein those portions of the scaled content falling within the display limit bounding box are rendered on the client device's display.

As discussed above, it is foreseen that the invention will be used with client devices having small, low resolution displays, such as PDAs and pocket PCs. Examples of various views of an exemplary web pages obtained from the YAHOO™ web site are shown in FIGS. **7**A-B, **8**A-B and **9**A-B. For instance, FIG. **7**A represents how the YAHOO™ home page might appear on a Palm IIIc color PDA.

In addition to directly scaling and offsetting content, the client user-interface software for PDA's provides additional functionality. For instance, a user may select to view a column (results represented in FIG. **7**B by tapping that column with a stylus, a shown in FIG. **7**A. Similarly, the user may select to zoom in on an image by tapping the image with the stylus, as shown in FIGS. **8**A and **8**B, or select to view a paragraph in an article by tapping on the paragraph, as shown in FIGS. **9**A and **9**B. It is noted that in some instances, the display of the paragraph may be reformatted to fit the characteristics of the display, rather than following the original format in the zoom-out view.

It is further noted that that different scaling factors can be applied to the X and Y axis so as to change the aspect ratio of the display. For example, a Web page may be designed to be displayed on a computer having a resolution of 800×600 pixels, or a 4X to 3Y aspect ratio. In this case, the display corresponds to a "landscape" layout, wherein there are more pixels along the X axis than along the Y axis. Conversely, many handheld devices display images having a "portrait" layout, wherein there are more pixels along the Y axis than the X axis. By enabling different scaling factors to be applied to the X and Y axes, the present invention enables the aspect ratio of a rendered display image to be adjusted to better fit the aspect ratio of the client device.

An Exemplary Computer Architecture

An exemplary machine in the form of a computer system **500** in which features of the present invention may be implemented will now be described with reference to FIG. **10**. Computer system **500** may represent a workstation, host, server, print server, or printer controller. Computer system **500** comprises a bus or other communication means **501** for communicating information, and a processing means such as processor **502** coupled with bus **501** for processing information. Computer system **500** further comprises a random access memory (RAM) or other dynamic storage device **504** (referred to as main memory), coupled to bus **501** for storing information and instructions to be executed by processor **502**. Main memory **504** also may be used for storing temporary variables or other intermediate information during execution of instructions by processor **502**. Computer system **500** also comprises a read only memory (ROM) and/or other static storage device **506** coupled to bus **501** for storing static information and instructions for processor **502**.

A data storage device **507** such as a magnetic disk or optical disc and its corresponding drive may also be coupled to bus **501** for storing information and instructions. Computer system **500** can also be coupled via bus **501** to a display device **521**, such as a cathode ray tube (CRT) or Liquid Crystal Display (LCD), for displaying information to an end user. Typically, an alphanumeric input device **522**, including alphanumeric and other keys, may be coupled to bus **501** for communicating information and/or command selections to processor **502**. Another type of user input device is cursor control **523**, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor **502** and for controlling cursor movement on display **521**.

A communication device **525** is also coupled to bus **501**. Depending upon the particular presentation environment implementation, the communication device **525** may include a modem, a network interface card, or other well-known interface devices, such as those used for coupling to Ethernet, token ring, or other types of physical attachment for purposes of providing a communication link to support a local or wide area network, for example. In any event, in this manner, the computer system **500** may be coupled to a number of clients and/or servers via a conventional network infrastructure, such as a company's Intranet and/or the Internet, for example.

Importantly, the present invention is not limited to having all of the routines located on the same computer system. Rather, individual objects, program elements, or portions thereof may be spread over a distributed network of computer systems. Additionally, it is appreciated that a lesser or more equipped computer system than the example described above may be desirable for certain implementations. Therefore, the configuration of computer system **500** will vary from implementation to implementation depending upon numerous factors, such as price constraints, performance requirements, and/or other circumstances. For example, according to one embodiment of the present invention, a cell phone or a hand held computer may comprise only a processor or a micro controller and a memory, such as a micro code ROM or RAM, for storing static or dynamically loaded instructions and/or data.

In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

What is claimed is:

1. A mobile device, comprising:
- a processor,
- a wireless communications device operatively coupled to the processor, to facilitate communication with a network via which Web content may be accessed;
- a touch-sensitive display;
- a memory, operatively coupled to the processor; and
- storage means, operatively coupled to the processor, in which a plurality of instructions are stored that when executed by the processor enable the mobile phone to perform operations including,
  - enabling a user to request access to a Web page comprising HTML-based Web content defining an original page layout, functionality, and design of content on the Web page;
  - retrieving HTML-based Web content associated with the Web page;
  - translating the HTML-based Web content to produce scalable vector-based page layout information;
  - employing the scalable vector-based page layout information and/or data derived therefrom to,
    - render a view of at least a portion of the Web page on the touch-sensitive display using a first scale factor; and
    - re-render the Web page in response to associated user inputs to enable a user to iteratively zoom in and out views of the Web page on the display while preserving the original page layout, functionality, and design of the content on the Web page defined by the HTML-based Web content,
  - wherein preservation of the functionality defined by the HTML-based content includes preservation of hyperlink functionality.

2. The mobile device of claim **1**, wherein the device comprises a mobile phone.

3. The mobile device of claim **1**, wherein the device comprises one of a hand-held device or a palm-held device.

4. The mobile device of claim **1**, wherein execution of the instructions performs further operations comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input made via the touch-sensitive display.

5. The mobile device of claim **1**, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

6. The mobile device of claim **1**, wherein the Web content includes at least one hyperlink, and wherein execution of the instructions performs further operations comprising:
- enabling the user to select the hyperlink via the touch-sensitive display; and, in response thereto,

retrieving and translating HMTL-based Web content associated with the hyperlink to produce additional scalable vector-based page layout information; and

employing the additional scalable vector-based page layout information and/or data derived therefrom to render the Web content associated with the hyperlink on the touch-sensitive display.

7. The mobile device of claim **1**, wherein execution of the instructions performs further operations comprising enabling the Web content to be displayed at different resolutions by scaling the scalable vector-based page layout information to resize a view of the Web page on the display in response to associated user inputs.

8. The mobile device of claim **1**, wherein execution of the instructions performs further operations comprising returning the display of the Web content to a previous view in response to a corresponding user input made via the touch-sensitive display.

9. The mobile device of claim **1**, wherein execution of the instructions performs further operations comprising enabling a user to pan a display of the Web content in response to a corresponding user input made via the touch-sensitive display.

10. The mobile device of claim **9**, wherein execution of the instructions performs further operations comprising enabling the display of the Web content to be panned in real-time.

11. The mobile device of claim **1**, wherein the page layout of the Web page is defined to have an original aspect ratio, and wherein the scalable vector-based page layout information and/or data derived therefrom is scaled to render a display having a different aspect ratio.

12. The mobile device of claim **1**, wherein execution of the instructions performs further operations comprising enabling a user to view a column of the Web content at a higher resolution than a current resolution by tapping on the column via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is displayed to fit across the touch-sensitive display.

13. The mobile device of claim **1**, wherein the Web content includes at least one image, and wherein execution of the instructions performs further operations comprising enabling a user to view an image at a higher resolution than a current resolution by tapping on the image via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that the image is displayed to fit across a width of a display area of the touch-sensitive display.

14. The mobile device of claim **1**, wherein execution of the instructions performs further operations comprising enabling a user to view a paragraph of the Web content at a higher resolution than a current resolution by tapping on the paragraph via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is displayed to fit across a width of a display area of the touch-sensitive display.

15. The mobile device of claim **1**, wherein execution of the instructions performs further operations comprising:

generating a display list derived, at least in part, via use of the vector-based page layout information; and

employing the display list to re-render the display of the Web page.

16. The mobile device of claim **1**, wherein execution of the instructions performs further operations comprising:

parsing HTML-based code corresponding to the received Web content to logically group content into objects, the objects including a plurality of display objects;

defining a primary datum corresponding to a page layout; and,

for each display object,

defining an object datum corresponding to a layout location datum for the object's associated display content;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to its corresponding vector.

17. The mobile device of claim **16**, wherein execution of the instructions performs further operations comprising:

mapping the object vectors to a virtual display area in memory.

18. The mobile device of claim **1**, wherein execution of the instructions performs further operations comprising:

parsing the HTML-based content to logically group content into objects;

generating page layout information including a bounding box for each object, the bounding box defining width and height dimensions for the object; and

storing information that links each object with its corresponding page layout information;

wherein the page layout information further includes information from which a page layout location of each of the bounding boxes can be determined.

19. The mobile device of claim **1**, wherein the scalable vector-based content includes scalable text content, and wherein execution of the instructions performs further operations comprising scaling a scalable font to render the scalable text content.

20. The mobile device of claim **1**, wherein at least a portion of the instructions comprise Java-based instructions configured to be executed on a Java virtual machine.

21. The mobile device of claim **1**, wherein translating the HTML-based Web content to produce scalable vector-based page layout information comprises:

processing the HTML-based Web content with a rendering engine to generate page layout information corresponding to the original page layout as interpreted by the rendering engine; and

employing the page layout information to produce scalable vector-based page layout information.

22. The mobile device of claim **21**, wherein the page layout information defines a layout location for a plurality of objects, including text objects, graphic layout objects, and/or image objects included on the Web page, and wherein producing vector-based page layout information comprises:

defining a primary datum corresponding to a page layout; and,

for each object,

defining an object datum corresponding to the layout location for the object on the page layout;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to its corresponding vector.

23. The mobile device of claim **22**, wherein execution of the instructions performs further operations comprising effecting a zoom operation combined with a pan operation by,

for each of the plurality of display objects to be included in a panned view of the Web page to be rendered on the display,

scaling page layout information associated with the display object using a scale factor corresponding to a zoom level associated with the zoom operation to determine a scaled datum;

determining an offset corresponding to the pan operation and combining the scaled datum with the offset to produce a scaled and offset datum that defines a location where the display object is to be rendered on the panned view of the Web page;

scaling content associated with the display object using the scale factor; and

rendering the scaled content at the location defined by the scaled and offset datum to render the display object on the panned view of the Web page.

24. The mobile device of claim **23**, wherein rendering scaled content associated with a text object comprises:

retrieving presentation attributes for the text object, the presentation attributes including a font typeface, size and color;

employing a scalable font associated with the font typeface to render text associated with the text object in a color associated with the color attribute, wherein the text is rendered relative to a location associated with the scaled and offset datum for the text object, and wherein the scale applied to the scalable font is a function of the scale factor and the font size.

25. The mobile device of claim **21**, wherein the original format of the Web page defines a width for the Web page, as interpreted by the rendering engine, and wherein execution of the instructions performs further operations comprising:

determining an applicable scale factor to fit the width of the Web page across a display area of the touch-sensitive display; and

employing the scale factor that is determined as the first scale factor.

26. The mobile device of claim **1**, wherein zooming operations are effected by applying a mathematical transformation to a plurality of points in a two-dimensional coordinate system comprising X and Y axes, including points comprising datum points having corresponding vectors included in the scalable vector-based page layout information defining page layout locations of corresponding text and image objects mapped to the two-dimensional coordinate system, wherein the mathematical transformation comprises,

$X'=X*SF;$

$Y'=Y*SF;$

wherein X, Y is the location of a point prior to transformation, X', Y' is the location of the point after transformation, and SF is the scale factor.

27. The mobile device of claim **26**, wherein the mathematical transformation is applied to points in a first coordinate system comprising a virtual coordinate system associated with a virtual display area onto which page layout information is mapped to a second coordinate system comprising a device coordinate system corresponding to a pixel resolution of the display of the mobile device, wherein points are mapped from the first coordinate system to the second coordinate system using the mathematical transformation.

28. The mobile device of claim **1**, wherein execution of the instructions performs further operations comprising maintaining at least one instance of the page layout information in a manner that is independent of the zoom levels used to view the web page on the display.

29. The mobile device of claim **1**, wherein the HTML-based Web content includes cascading style sheet content defining layout and presentation attributes for the Web page.

30. A mobile phone, comprising:

a processor,

wireless communications means operatively coupled to the processor, to facilitate communication with a mobile service provider network via which Web content may be accessed;

a touch-sensitive display;

a memory, operatively coupled to the processor; and

storage means, operatively coupled to the processor, in which a plurality of instructions are stored that when executed by the processor enable the mobile phone to perform operations including,

rendering a browser interface via which a user is enabled to request to access to a Web page having an original format comprising HTML-based content defining an original page layout, functionality, and design of content on the Web page;

retrieving HTML-based content associated with the Web page;

translating at least a portion of the HTML-based content from its original format to produce translated content including scalable vector-based content that supports a scalable resolution-independent representation of the HTML-based content that preserves an original page layout, functionality and design of the at least a portion of the HTML-based content when scaled and rendered; and

employing the scalable vector-based content to render a view of at least a portion of the Web page on the display using a first scale factor,

wherein preservation of the functionality defined by the HTML-based content includes preservation of hyperlink functionality.

31. The mobile phone of claim **30**, wherein execution of the instructions performs further operations comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input made via the touch-sensitive display.

32. The mobile phone of claim **31**, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

33. The mobile phone of claim **30**, wherein the Web content includes at least one hyperlink, and wherein execution of the instructions performs further operations comprising:

enabling the user to select the hyperlink via the touch-sensitive display; and, in response thereto,

retrieving and translating the Web content associated with the hyperlink to produce additional scalable vector-based content; and

employing the additional scalable vector-based content to render the Web content associated with the hyperlink on the touch-sensitive display.

34. The mobile phone of claim **30**, wherein execution of the instructions performs further operations comprising:

parsing and processing markup language code associated with the Web page to determine the original page layout of display content within the Web page, wherein the original page layout defines a layout location for a plurality of objects, including text objects, graphic layout objects, and/or graphic image objects included in the Web page;

defining a primary datum corresponding to the original page layout; and,

for each object,

defining an object datum corresponding to the layout location for the object;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to its corresponding vector.

35. The mobile phone of claim **30**, wherein execution of the instructions performs further operations comprising enabling the Web content to be displayed at different resolutions by scaling the scalable vector-based content to re-render the display in response to associated user inputs.

36. The mobile phone of claim **30**, wherein execution of the instructions performs further operations comprising returning the display of the Web content to a previous view in response to a corresponding user input made via the touch-sensitive display.

37. The mobile phone of claim **30**, wherein execution of the instructions performs further operations comprising enabling a user to pan a display of the Web content in response to a corresponding user input made via the touch-sensitive display.

38. The mobile phone of claim **37**, wherein execution of the instructions performs further operations comprising enabling the display of the Web content to be panned in real-time.

39. The mobile phone of claim **30**, wherein the page layout of the Web page is defined to have an original aspect ratio, and wherein the scalable vector-based content is scaled when rendered so as to produce a display having a different aspect ratio.

40. The mobile phone of claim **30**, wherein execution of the instructions performs further operations comprising enabling a user to view a column of the Web content at a higher resolution than a current resolution by tapping on the column via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is displayed to fit across the touch-sensitive display.

41. The mobile phone of claim **30**, wherein the Web content includes at least one image, and wherein execution of the instructions performs further operations comprising enabling a user to view an image at a higher resolution than a current resolution by tapping on the image via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that the image is displayed to fit across at least one of a width and height of a display area of the touch-sensitive display.

42. The mobile phone of claim **30**, wherein execution of the instructions performs further operations comprising enabling a user to view a paragraph of the Web content at a higher resolution than a current resolution by tapping on the paragraph via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is displayed across at least one of a width and height of a display area of the touch-sensitive display.

43. The mobile phone of claim **30**, wherein execution of the instructions performs further operations comprising:

generating a display list associated with the scalable vector-based content; and

employing the display list to re-render the display at different scale factors to enable rapid zooming of the Web page.

44. The mobile phone of claim **30**, wherein execution of the instructions performs further operations comprising:

parsing and processing markup language code corresponding to the received Web content to determine page layout information corresponding to a page layout of the content on the Web page;

logically grouping selected content into objects;

defining a primary datum corresponding to the page layout; and,

for each object,

defining an object datum corresponding to a layout location datum for the object's associated display content;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to its corresponding vector.

45. The mobile phone of claim **44**, wherein execution of the instructions performs further operations comprising:

mapping the object vectors to a virtual display area in memory.

46. The mobile phone of claim **45**, wherein execution of the instructions performs further operations comprising:

determining a first scale factor and offset in response to one or more corresponding user inputs defining a user-selectable zoom level and pan corresponding to a rendered view of the Web content desired by a user;

determining a virtual display limit bounding box for the virtual display area associated with the first scale factor and offset;

identifying objects having at least a portion of their content falling within the virtual display limit bounding box; and,

for each of such objects,

retrieving content associated with that object; and

applying an appropriate scale factor and offset to the content to render the view of the Web content.

47. The mobile phone of claim **30**, wherein execution of the instructions performs further operations comprising:

parsing markup language code corresponding to the received Web content to logically group selected content into objects;

generating page layout information including a bounding box for each object, the bounding box defining width and height dimensions for the object; and

storing information that links each object with its corresponding page layout information;

wherein the page layout information further includes information from which a page layout location of each of the bounding boxes can be determined.

48. The mobile phone of claim **30**, wherein the scalable vector-based content includes scalable text content, and wherein execution of the instructions performs further operations comprising scaling a scalable font to render the scalable text content.

49. The mobile phone of claim **30**, wherein the original format of the Web page defines a width for the Web page, and wherein execution of the instructions performs further operations comprising:

determining an applicable scale factor to fit the width of the Web page across a display area of the touch-sensitive display; and

employing the scale factor that is determined as the first scale factor.

50. The mobile phone of claim **30**, wherein at least a portion of the instructions comprise Java-based instructions configured to be executed on a Java virtual machine.

51. The mobile device of claim **30**, wherein the original format of the Web page comprises HTML-based Web content and the vector-based scalable content comprises scalable vector-based page layout information, and wherein execution of the instructions performs further operations comprising:

processing the HTML-based Web content with a rendering engine to generate page layout information corresponding to the original page layout as interpreted by the rendering engine;

employing the page layout information to generate the scalable vector-based page layout information.

52. A mobile device, comprising:

a processor,

wireless communications means, to facilitate wireless communication with a network via which Web content may be accessed;

a touch-sensitive display;

flash memory, operatively coupled to the processor, in which a plurality of instructions are stored that when executed by the processor enable the mobile device to perform operations including,

rendering a browser interface via which a user is enabled to request access to a Web page comprising HTML-based Web content defining an original page layout, functionality, and design of content on the Web page;

retrieving and processing the HTML-based Web content to produce scalable content; and

employing the scalable content and/or data derived therefrom to,

render a view of the Web page on the touch-sensitive display; and

re-render the Web page in response to associated user inputs to enable the user to iteratively zoom in and out views of the Web page while preserving an original page layout, functionality, and design defined by the HTML-based Web content as interpreted by a rendering engine,

wherein preservation of the functionality defined by the HTML-based Web content includes preservation of hyperlink functionality.

53. The mobile device of claim **52**, wherein the device comprises a mobile phone.

54. The mobile device of claim **52**, wherein the device comprises one of a Personal Digital Assistant (PDA) or pocket PC.

55. The mobile device of claim **52**, wherein execution of the instructions performs further operations comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input made via the touch-sensitive display.

56. The mobile device of claim **55**, wherein the user interface input enables the user to define a window of a current view of the Web page on which to zoom in on.

57. The mobile device of claim **52**, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

58. The mobile device of claim **52**, wherein the Web page includes at least one hyperlink, and wherein execution of the instructions performs further operations comprising:

enabling the user to select the hyperlink via the touch-sensitive display; and, in response thereto,

retrieving and processing HMTL-based Web content associated with the hyperlink to produce additional scalable content; and

employing the additional scalable content and/or data derived therefrom to render the Web content associated with the hyperlink on the touch-sensitive display.

59. The mobile device of claim **52**, wherein at least a portion of the scalable content comprises scalable vector-based content.

60. The mobile device of claim **52**, wherein execution of the instructions performs further operations comprising returning the display of the Web page to a previous view in response to a corresponding user input made via the touch-sensitive display.

61. The mobile device of claim **52**, wherein execution of the instructions performs further operations comprising enabling a user to pan a display of the Web content in response to a corresponding user input made via the touch-sensitive display.

62. The mobile device of claim **61**, wherein execution of the instructions performs further operations comprising enabling the display of the Web content to be panned in real-time.

63. The mobile device of claim **52**, wherein the page layout of the Web page is defined to have an original aspect ratio, and wherein the scalable content and/or data derived therefrom is scaled to render a display having a different aspect ratio.

64. The mobile device of claim **52**, wherein execution of the instructions performs further operations comprising enabling a user to view a column of the Web content at a higher resolution than a current resolution by tapping on the column via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is displayed across the touch-sensitive display.

65. The mobile device of claim **52**, wherein the Web content includes at least one image, and wherein execution of the instructions performs further operations comprising enabling a user to view an image at a higher resolution than a current resolution by tapping on the image via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that the image is displayed to fit a width of a display area of the touch-sensitive display.

66. The mobile device of claim **52**, wherein execution of the instructions performs further operations comprising enabling a user to view a paragraph of the Web content at a higher resolution than a current resolution by tapping on the paragraph via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is displayed across at least one of a width and height of a display area of the touch-sensitive display.

67. The mobile device of claim **66**, wherein the content of the paragraph is reformatted to fit characteristics of the display area when the display is re-rendered.

68. The mobile device of claim **52**, wherein the Web page includes text, layout attributes, and images, and wherein execution of the instructions performs further operations comprising:

receiving content corresponding to the text and layout attributes via a first connection; and

receiving content corresponding to at least one image via a second connection.

69. The mobile device of claim **68**, further comprising dynamic memory having at least a portion employed for rendering purposes, wherein execution of the instructions performs further operations comprising:

mapping the object vectors and associated bounding boxes to a virtual display area in the dynamic memory.

70. The mobile device of claim **69**, wherein execution of the instructions performs further operations comprising:

determining a first scale factor and offset in response to one or more corresponding user inputs defining a user-selectable zoom level and pan corresponding to a rendered display of the Web page desired by a user;

determining a virtual display limit bounding box for the virtual display associated with the first scale factor and offset;

identifying object bounding boxes having at least a portion falling within the virtual display limit bounding box; and,

for each of such object bounding boxes,

retrieving content associated with that object bounding box; and

applying an appropriate scale factor to the content to render the display.

71. The mobile device of claim **52**, further comprising dynamic memory having at least a portion employed for rendering purposes, wherein execution of the instructions performs further operations comprising:

building a display list via use of the scalable content and rendering the display list on a virtual display area in the dynamic memory; and

scaling the display list to re-render the display of the Web page.

72. The mobile device of claim **52**, wherein execution of the instructions performs further operations comprising:

parsing HTML-based code corresponding to the received Web content to identify content on the Web page;

logically grouping selected content into objects;

defining a primary datum corresponding to the original page layout; and,

for each object,

defining an object datum corresponding to a layout location datum for the object's associated display content;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to its corresponding vector.

73. The mobile device of claim **72**, wherein execution of the instructions performs further operations comprising:

generating a bounding box for each object, the bounding box representing a portion of a rendered display page occupied by the object's associated group of content.

74. The mobile device of claim **52**, wherein the scalable content includes scalable text content, and wherein execution of the instructions performs further operations comprising scaling a scalable font to render the scalable text content.

75. The mobile device of claim **52**, wherein the original format of the Web page defines a width for the Web page, and wherein execution of the instructions performs further operations comprising:

determining an applicable scale factor to fit the width of the Web page across a display area of the touch-sensitive display; and

employing the scale factor to render the display area.

76. The mobile device of claim **52**, wherein at least a portion of the instructions comprise Java-based instructions configured to be executed on a Java virtual machine.

77. The mobile device of claim **52**, wherein a portion of the HTML-based Web content comprises XML-based content.

78. The mobile device of claim **52**, wherein a portion of the HTML-based Web content comprises cascading style sheet data.

79. A mobile device, comprising:

processing means;

wireless communications means, to facilitate wireless communication with a network via which Web content may be accessed;

touch-sensitive display means, to facilitate user input and display rendered content;

programmed circuit means; and

storage means, in which a plurality of instructions are stored,

wherein, upon execution of the instructions by at least one of the processing means and programmed circuit means, the mobile device is enabled to perform operations, including,

rendering a browser interface via which a user is enabled to request to access to a Web page comprising HTML-based Web content defining an original page layout, functionality, and design of content on the Web page;

retrieving and processing the HTML-based Web content to produce scalable content; and

employing the scalable content and/or data derived therefrom to,

render a view of the Web page on the touch-sensitive display; and

re-render the Web page in response to associated user inputs made via the touch-sensitive display means to enable the user to iteratively zoom in and out views of the Web page while preserving an original page layout, functionality, and design defined by the HTML-based Web content as interpreted by a rendering engine,

wherein preservation of the functionality defined by the HTML-based Web content includes preservation of hyperlink functionality.

80. The mobile device of claim **79**, wherein the processing means includes a general-purpose processor.

81. The mobile device of claim **79**, wherein at least a portion of the programmed circuit means is embodied as a special-purpose processor.

82. The mobile device of claim **79**, wherein execution of the instructions performs further operations comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input made via the touch-sensitive display.

83. The mobile device of claim **82**, wherein the user interface input enables the user to define a window of a current view of the Web page on which to zoom in on.

84. The mobile device of claim **79**, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

85. The mobile device of claim **79**, wherein execution of the instructions performs further operations comprising enabling a user to pan a display of the Web content in response to a corresponding user input made via the touch-sensitive display.

86. The mobile device of claim **85**, wherein execution of the instructions performs further operations comprising enabling the display of the Web content to be panned in real-time.

87. The mobile device of claim **79**, wherein the Web content includes at least one image, and wherein execution of the instructions performs further operations comprising enabling a user to view an image at a higher resolution than a current resolution by tapping on the image via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that the image is displayed to fit across a width of a display area of the touch-sensitive display.

88. The mobile device of claim **79**, further comprising dynamic memory having at least a portion employed for rendering purposes, wherein execution of the instructions performs further operations comprising:

building a display list of scalable content via use of the scalable content and rendering the display list on a virtual display area in the dynamic memory; and

scaling the scalable content in the display list to re-render the display of the Web page.

* * * * *

Exhibit 3A

TITLE OF INVENTION

Method and system for resolution independent vector display of HTML and XML content.

FIELD OF INVENTION

The present invention relates to conversion of Internet and World Wide Web content to scaleable vector representation and more particularly to systems and methods for retrieval and conversion of HTML, XML, and other Internet content to vector representations of that content.

BACKGROUND OF THE INVENTION

Much of the Internet content has been designed for display on desktop computers with a single target resolution. Even though HTML has the ability to adapt to changes in screen resolution major Internet content providers have chosen to display their Web pages using fixed resolution structures such as tables. This gives them the ability to control the look and feel of their Web sites. This fixed resolution approach has evolved to the point that the fixed resolution layout of Web pages has become the most common method to brand or uniquely identify Web sites.

While this fixed resolution approach is good for marketing and product differentiation it does present a daunting technical problem for technologies like cell phones and hand held computers wanting to display Internet Web page content. The problem is how to display Internet content designed for desktop computers on small screen, low resolution, or different aspect ratio devices. The present invention, method and system for resolution independent vector display of HTML and XML content, addresses and solves this problem.

Embodied in this invention is the solution to displaying Internet content on cell phones, hand held computers, wireless Internet devices, displays with very high resolution, and very large displays. Cell phone users can see their favorite Internet Web sites in the same graphic layout they are use to from their desktop computers. Hand held computers, with different display aspect ratios, display Web pages scaled to fit their displays. Next generation high-resolution displays will also have problems with Internet content. The high resolution will cause Web pages to display very small. Vector based content will scale up display content to readable useful resolutions. Very large displays using vector scaling can display building size images, advertising, educational, and any Internet or Intranet content.

BRIEF SUMMARY OF INVENTION

This invention is a method and system for creating resolution independent vector display of Internet content such as HTML and XML based documents. Application of vector technology allows scaling of Web content to fit any resolution or screen size. The content can be scaled larger and smaller (zoomed) for better viewing.

To accomplish this, HTML and XML data is rendered to a vector format. The client requests Internet content from a proxy. The proxy retrieves the Internet content and converts it to a vector format. The vector-formatted content is then sent back to the client for display.

BRIEF DESCRIPTION OF THE DRAWINGS

The process and advantages of the invention may be better understood by referring to the accompanying drawings in which like reference numbers represent like parts, in which:

FIG. **1** is a schematic representation of the process whereby a client system requests Internet content and receives a vector representation of that content.

FIG. **2** illustrates one embodiment of the invention where various computer devices acquire vector representations of Internet content through a proxy server.

DETAILED DESCRIPTION OF THE INVENTION

One embodiment of the process involved in the invention is illustrated in FIG. **1**. The client **100** is a system such as a computer or phone. The client makes a request **101** for Internet content from a proxy. This content may be an HTML (HyperText Markup Language) or XML (eXtensible Markup Language) document. The proxy may be a server system across a network connection or a software agent present on the client system. The proxy retrieves the Internet content **102**, which may reside anywhere on a network or the proxy itself. The proxy converts the Internet content into scaleable vector information **103**. The proxy then sends this vector information back to the client, where it can be displayed **104**. The client can then zoom and pan on the vector representation to more clearly see an overview or details in the document.

FIG. **2** is an illustration of the preferred embodiment of the invention. A user of a computer device, such as a personal computer **120**, hand held computer **121** or phone **122** requests to view Internet content. As stated above, this content may be in the form of an HTML or XML document. The request is sent across a network, for instance the Internet, using a protocol such as HTTP (HyperText Transfer Protocol) **130, 131** or WAP (Wireless Access Protocol) **132**. The proxy server **125** receives the request and fetches the actual content **135, 136, 137** from the network. The proxy then converts the content into scaleable vector information that is sent back to the client. At this point the user can take advantage of the benefits of vector information to zoom and pan in order to better see the information. The user may select a hyperlink from the content, which starts the process over again.

The client may be a personal computer. Many Web pages are designed for a smaller resolution (for example 640x480 pixels) than typical personal computers display (for example 1024x768 pixels). Sometimes selected text on a Web page is much smaller than other text. Allowing users to magnify the entire page allows them to overcome these design flaws. Alternately, the client may be a small device, such as a hand held computer or a cell phone, which has a smaller display area than common Web pages are designed for. A scaleable vector representation would enable these users to access the information on the billions of existing Web pages in a simple and reasonable way.

The client software may be a plug-in to a World Wide Web browser such as Netscape Navigator or Microsoft Internet Explorer. The browser may be configured to handle the connection to the proxy server. Alternately, the client software may be a Java applet running in a browser. Or the client software may be a stand-alone program that talks to the proxy server or proxy software directly. The client software may bypass the proxy when requesting information that won't be translated to vectors, such as bitmaps.

As mentioned above, the proxy may be a proxy server as understood in the Internet domain. A proxy may also be software that resides on the client system or on the same server as the Internet content.

Internet content is typically represented as HTML. It may consist of bitmaps, animations, audio files or other information referenced in the parent document. Alternately, Internet content may be stored in XML and transformed using something such as XSLT (eXtensible Style Language Transformation) to a rendering format such HTML, XHTML or another XML language, including an XML-based vector language. Internet content may be stored in many other formats, including, but not limited to, GIF, JPG, PDF, SVF, or MP3.

The vector information created by the proxy is understood to be a reasonable representation of the Internet content. It may contain text and layout information. It may contain bitmaps or references to bitmaps. Hyperlinks that point to other Internet content may be present in the vector information. The user may select a hyperlink in order to view other Internet content. Lines and other graphic information may also be present in order to represent the content reasonably.

ABSTRACT OF THE DISCLOSURE

Method and system for creating and accessing a scaleable vector representation of HTML, XML, and other Internet content. A client requests Internet content such as HTML or XML from a proxy. The proxy retrieves the Internet content and converts it to a vector format that is sent back to the client. This invention can be used to view Web pages on devices with limited resolution such as hand held computers.

The following example is submitted to illustrate but not to limit this invention.

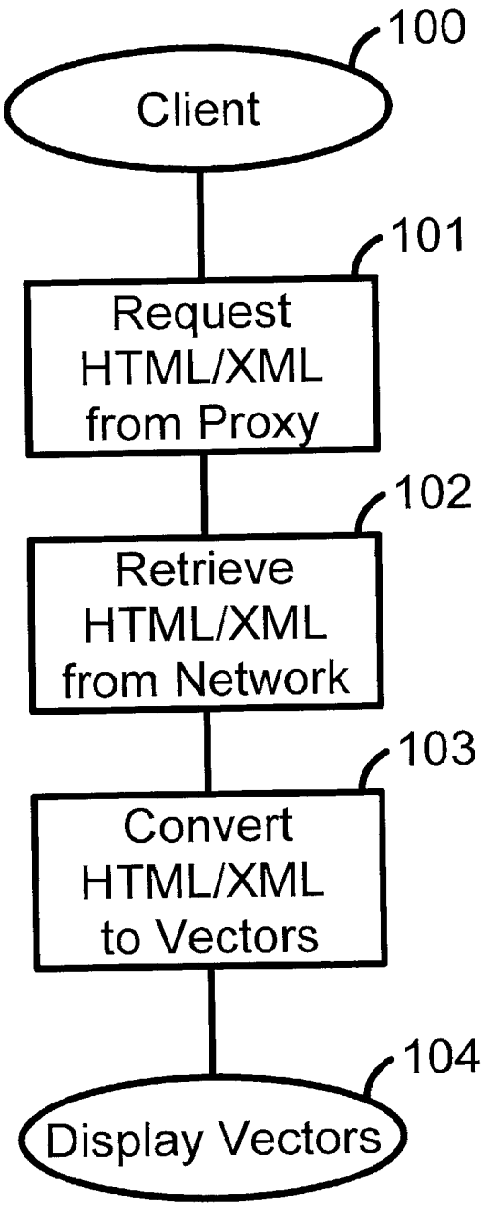


Fig. 1

Exhibit 3B

TITLE OF INVENTION

Method and system for selection, retrieval, and conversion of computer content to vector format for resolution independent display.

FIELD OF INVENTION

The present invention relates to selection and conversion of computer content to vector representation for resolution independent display and more particularly to systems and methods for selection, retrieval, and conversion of CAD drawings, graphic files, spreadsheets, text documents, and other computer content to vector representations for resolution independent zoom and pan enabled display.

COPYRIGHT NOTICE

BACKGROUND OF THE INVENTION

Most computer content is displayed as fixed resolution text and bitmaps. Some programs like CAD systems and bitmap graphic editors provide the ability to zoom displayed data on desktop computers. Plug ins for Internet browsers like Adobe's Acrobat and SoftSource's SVF viewer allow zoom enabled display of PDF and SVF files. Plug ins have been successful in expanding the use of zoom enabled documents on the Internet.

The fixed resolution approach is good solution for displaying most computer content. It does present a daunting technical problem for technologies like cell phones, wireless hand held computers, very large, and high resolution display systems. Resolution independent vector display of computer content solves these problems.

Embodied in this invention is the solution to displaying data in a vector format that allows the user to zoom on the content. Computer content on cell phones, hand held computers, wireless Internet devices, displays with very high resolution, and very large displays. Cell phone users can view documents they are use to from their desktop computers. Hand held computers, with different display aspect ratios, display computer content scaled to fit their display. Next generation high-resolution displays

will also have problems with computer content. The high resolution will cause computer fixed resolution content like bitmaps to display very small. Vector based content will scale up display content to readable useful resolutions. Very large displays using vector scaling can display building size images, advertising, educational, and any content.

BRIEF SUMMARY OF INVENTION

This invention is a method and system for creating resolution independent Internet vector display of computer content such as CAD drawings, graphic files, spreadsheets, and other computer content. Application of vector technology allows scaling of computer content to fit any resolution or screen size. The content can be zoomed (scaled larger and smaller) for better viewing.

To accomplish this, computer content is rendered to a vector format. The client requests computer content from a universal vector translator. The universal vector translator retrieves the computer content and converts it to a vector format. The vector-formatted content is then sent back to the client for display.

BRIEF DESCRIPTION OF THE DRAWINGS

The process and advantages of the invention may be better understood by referring to the accompanying drawings in which like reference numbers represent like parts, in which:

FIG. **1** is a schematic representation of the process whereby a client system requests computer content and receives and displays a vector representation of that content.

FIG. **2** illustrates one embodiment of the invention where various computer devices request and acquire computer content, convert that content to vector format, and display vector representations of computer content on client devices.

FIG. **3** illustrates the client to server and server to client information flow. The client requests computer content and transfers the computer content in vector format for display.

FIG. **4** illustrates the server to computer source, in this case the Internet, and the computer content is its original format transferred to the server. The computer content is then converted to vector format the sent on to the requesting client for display.

DETAILED DESCRIPTION OF THE INVENTION

One embodiment of the process involved in the invention is illustrated in FIG. **1**. The client device **100** is a system such as a computer or phone. The client makes a request **101** for computer content from a server. This content may be a CAD drawing, graphic file, spreadsheet, text based document, or other computer content. The computer content vector converter can work on a server system across a network connection or as a software agent present on the client system. The server retrieves the Internet content **102**, which may reside anywhere on a network, the Internet or on the server itself. The server based universal vector translator converts the Internet content into scaleable vector information **103**. The server then sends this vector information back to the client **104.** The vector content is then displayed on the client device **105**. The client can then zoom and pan on the vector representation to more clearly view details in the document.

FIG. **2** is an illustration of the preferred embodiment of the invention. A user of a computer device, such as a personal computer **120**, wireless hand held computer **121** or cell phone **122** requests to view computer content. As stated above, the computer content may be in the form of a CAD drawing, graphic file, spreadsheet, text based document, or other computer content. The request is sent across a network, for instance the Internet, using a protocol such as HTTP (HyperText Transfer Protocol) **130, 131** or WAP (Wireless Access Protocol) **132**. The server **150** receives the request and fetches the actual content **140** Internet ULR (Uniform Resource Locator), **141** FTP (File Transfer Protocol) using ULI (Uniform Resource Identifier), **142** from the network, local machine, or other source. The server based universal vector translator then converts the content into scaleable vector information that is sent back to the client **160, 161, 162.** The vector representation of the computer content is displayed on the client device. At this point the user can take advantage of the benefits of vector information to zoom and pan in order to better view the information. The user may select a hyperlink from the content, which starts the process over again.

FIG. **3** is an illustration of the preferred embodiment of the client side of the invention. A user of a computer device, such as a personal computer **120**, requests to view computer content. The request is sent across a network, for instance the Internet, using a protocol such as HTTP (HyperText Transfer Protocol) **130**. The converted the content is sent back to the client **160** for display.

FIG. **4** is an illustration of the preferred embodiment of the server side of the invention. The request is sent across a network, for instance the Internet, using a protocol such as HTTP (HyperText Transfer Protocol) **170**. The computer content is then transferred to the server in its original format **171**. The computer content is converted into vector format. The converted computer content is then transferred to the requesting client for display.

The client may be a personal computer or other high resolution display device. Much of the computer content is designed for lower resolution displays(for example 640x480 pixels). Modern computer systems, projector, and wall size displays can have resolutions measured in the millions of pixels. Allowing users to magnify the computer content allows them to overcome the original contents design limitations. Alternately, the client may be a small device, such as a hand held computer or a cell phone, which have a smaller display area. A scaleable vector representation enables access any computer content on any resolution display device.

As mentioned above, the server based universal vector translator may be a server as understood in the network and Internet domain. The server based universal vector translator may also be software that resides on the client system or on the same server as the computer content.

The vector information created by the server based universal vector translator is understood to be a reasonable representation of the computer content. It may contain vector and text and layout information. It may contain bitmaps or references to bitmaps. Hyperlinks that point to other computer content may be present in the vector representation. The user may select a hyperlink in order to view other content. Lines and other graphic information may also be present in order to represent the content reasonably.

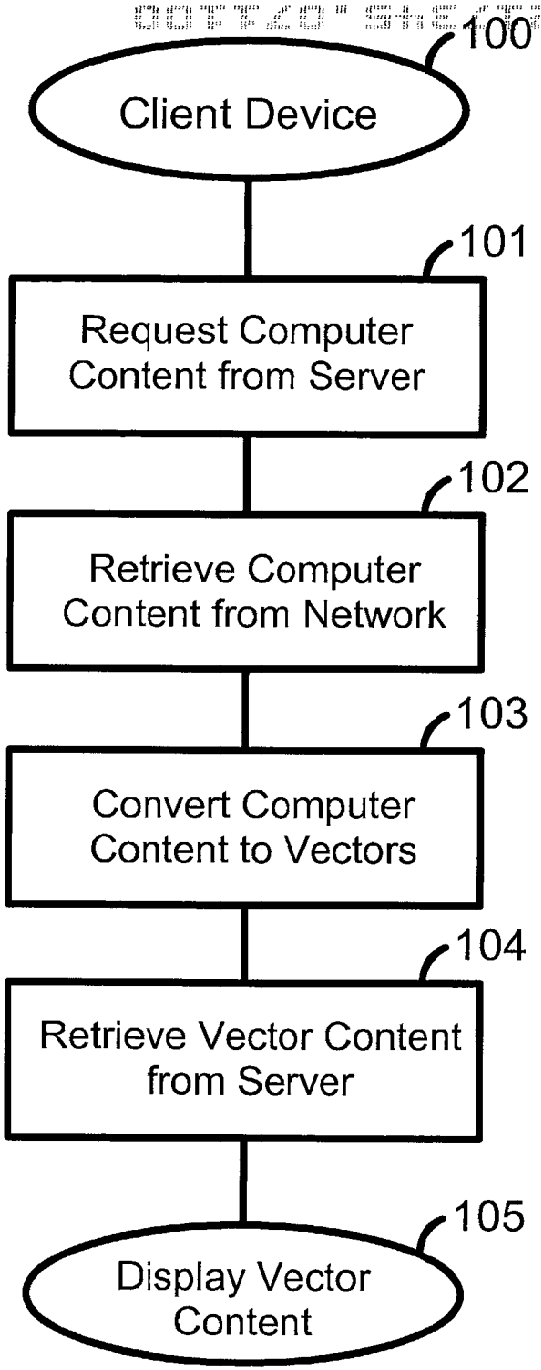


Fig. 1

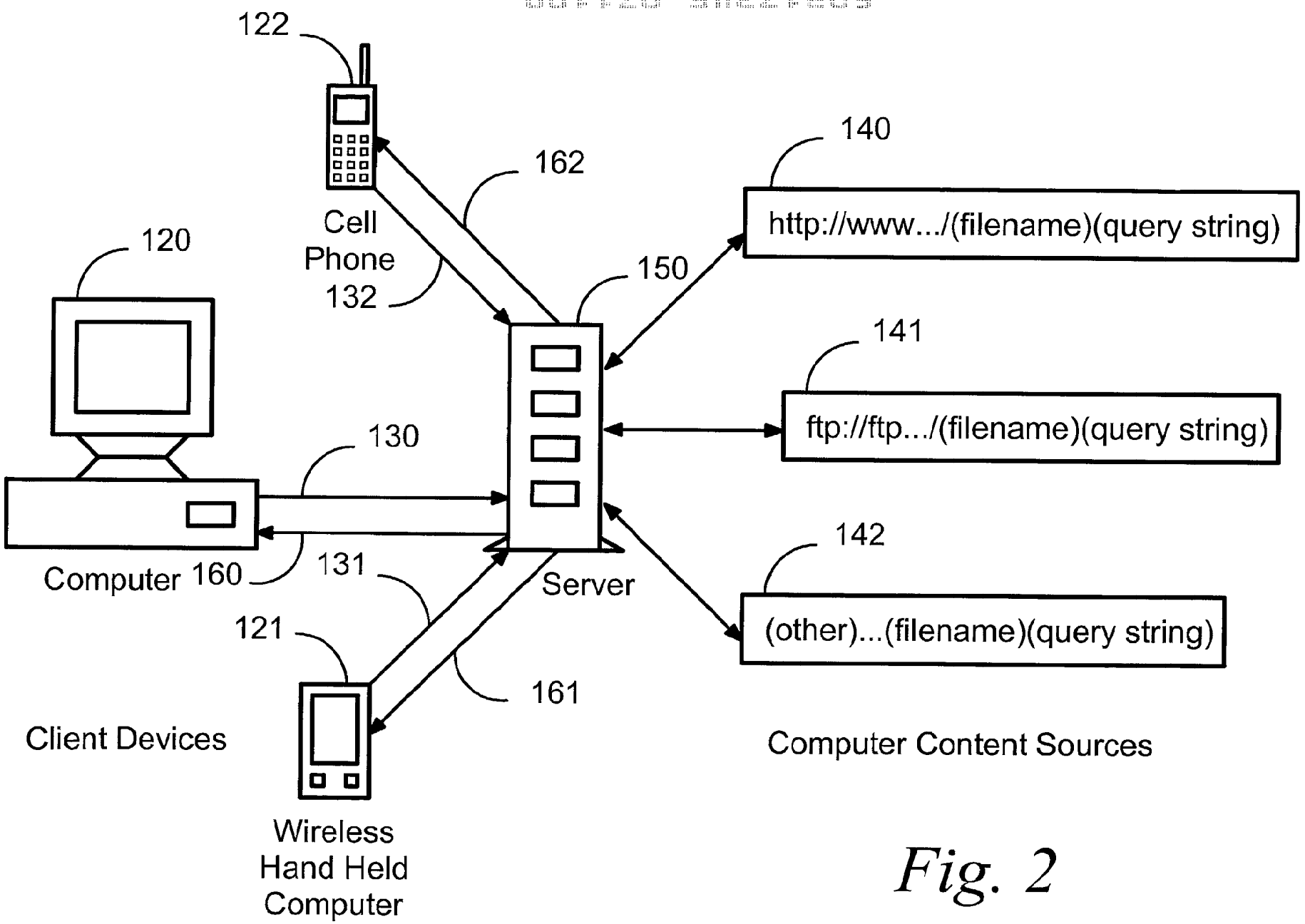


Fig. 2

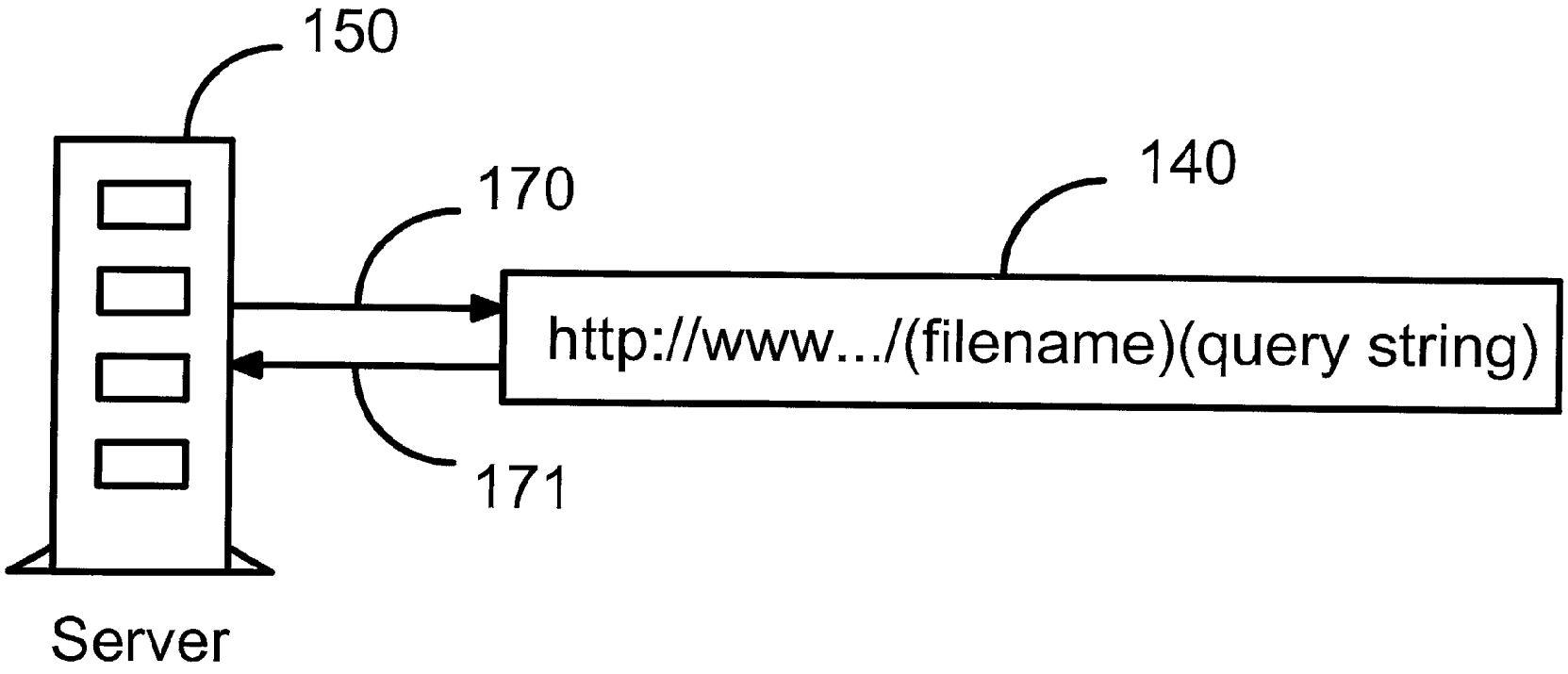


Fig. 4

Exhibit 3C

ABSTRACT OF THE DISCLOSURE

Apparatus and methods are provided for creating resolution independent vector display of Internet content to allow it to be scaled (zoomed) larger and smaller for better viewing or to fit any resolution or screen size. According to one embodiment, novel server processing of Web content is provided that converts Web content requested by a client to a scaleable vector format, such as SVF. The vector format enables the client to substantially retain an original page layout within a set of layouts originally intended to be associated with the requested Web content by including page layout information. Finally, the vector-formatted Web content is provided to the client. A client-side vector viewer receives requested Web content and displays a vector representation of the requested Web content that substantially retains page layout and/or graphics associated with the requested Web content.

RESOLUTION INDEPENDENT VECTOR DISPLAY OF INTERNET CONTENT

[0001] This application claims the benefit of U.S. Provisional Application No. 60/211,019, filed June 12, 2000, which is incorporated herein by reference.

COPYRIGHT NOTICE

BACKGROUND OF THE INVENTION

Field of the Invention

[0003] The invention relates generally to conversion of Internet and World Wide Web content to scaleable vector representation. More particularly, the invention relates to apparatus and methods for zoom enabling the display of content in an Internet information browser by retrieving and converting HyperText Markup Language (HTML), eXtensible Markup Language (XML), and other Internet content to vector representations of that content.

Description of the Related Art

[0004] Text only Internet information browsers began as a project at the CERN, European Organization for Nuclear Research, facility in Geneva Switzerland. From its inception the intent was to provide a mesh or web of access to data with a common user

BRIEF SUMMARY OF THE INVENTION

[0007] A method and apparatus for supporting resolution independent vector display of Internet content is disclosed. According to one embodiment, novel client processing of Web content is provided. The client receives requested Web content and displays a vector representation of the requested Web content that substantially retains page layout and/or graphics associated with the requested Web content.

[0008] According to another embodiment, novel server processing of Web content is provided. First, the server receives a request for Web content from a client. The requested Web content is then converted to a vector format to produce vector-formatted Web content corresponding to the requested Web content. The vector format enables the client to substantially retain an original page layout within a set of layouts originally intended to be associated with the requested Web content by including page layout information. Finally, the vector-formatted Web content is provided to the client.

[0009] Other features of the present invention will be apparent from the accompanying drawings and from the detailed description which follows.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The appended claims set forth the features of the invention with particularity. The invention, together with its advantages, may be best understood from the following detailed description taken in conjunction with the accompanying drawings of which:

[0011] **Figure 1** is a flow diagram illustrating a process whereby a client system requests Internet content and receives a vector representation of that content according to one embodiment of the present invention.

[0012] **Figure 2** illustrates various computer devices acquiring vector representations of Internet content through a proxy server according to one embodiment of the present invention.

[0013] **Figure 3** is a flow diagram illustrating a process whereby layout information is defined and used in combination with a scale factor and offset to zoom enable Internet content according to one embodiment of the present invention.

[0014] **Figure 4** conceptually illustrates the use of layout information in combination with a scale factor and offset to zoom enable Internet content according to one embodiment of the present invention.

[0015] **Figure 5** is an example of a computer system upon which one embodiment of the present invention may be implemented.

[0016] **Figure 6** conceptually illustrates a first architectural model in which content is converted to vector format as a service provided by a web site according to one embodiment of the present invention.

[0017] **Figure 7** conceptually illustrates a second architectural model in which content is converted to vector format as a service provided by a wireless service provider according to one embodiment of the present invention.

[0018] **Figure 8** conceptually illustrates a third architectural model in which content is converted to vector format as a service provided by an Internet Service Provider (ISP) according to one embodiment of the present invention.

application, the map tiles surrounding the viewed map could be downloaded and stored while the user was working with the initial tile, enabling an experience remarkably free from the current frustrations of waiting for a new map to be transferred for even the smallest change in magnification or coverage. If the user knows ahead of time what city they will visit on a business trip, maps and additional travel information in great detail could also be pre-loaded using a high bandwidth connection at home or in the office before heading out to shop or conduct business in the city. Additionally, SVF is a more efficient way to store web content. Resulting content files are reduced in size by anywhere from 20 to 80 percent over their source. SVF is also very compressible. With target file size reduction in the range of 90%, SVF files can take up as little as $1/10^{th}$ the space of the web files in current use. This means that pre-converted content can be moved up to 10 times the rate of current web pages, and as much as 10 times as many pages, maps, stock charts, etc. can be stored for instant retrieval on the hand held platform as can be handled with current web technology.

[0028] When used on content created natively in SVF, additional capability can be extended to the client-side viewer. Because SVF is, at its heart, a database, information in database form can be included, linked to or manipulated through the viewer environment. That means, for example, that data streams for stocks can be transferred to the client and then displayed and manipulated real time on the end device. Imagine checking a quote on a security and instantaneously being able to look at the performance of the stock over any period from minutes to ten years in duration, graph it, and compare the graphs with indexes or stocks within a pre-identified group. Because the data can move as a native database, and not as a set of static graphical images, setting up this impressively powerful world on a device as small as a wireless PDA is easy.

started immediately. In addition to streaming, the server-side content converter may also layer the content by type. This means that text can be put in one layer, links in another, GIF images in another, Javascript in another and so on. Layers can be turned on or off depending upon client capabilities, making files for less capable clients, or for users interested in a reduced functionality, higher transfer performance mode to be handled automatically.

[0033] All operational modes may be controlled through an administrative interface or accessible through a straightforward API.

[0034] The system works with existing firewalls and within standard security protocols. In more secure modes, the server-side content converter and the client-side viewer may operate using Public/Private key authentication and encryption.

[0035] **Figure 1** is a flow diagram illustrating a process whereby a client system requests Internet content and receives a vector representation of that content according to one embodiment of the present invention. In this example, the client **100** is a system such as a computer or phone. The client makes a request at processing block **101** for Internet content from a proxy. This content may be an HTML or XML document. The proxy may be a server system across a network connection or a software agent present on the client system. At processing block **102**, the proxy retrieves the Internet content, which may reside anywhere on a network or the proxy itself. At processing block **103**, the proxy converts the Internet content into scaleable vector information, such as SVF. Then, at processing block **104**, the proxy sends this vector information back to the client, where it can be displayed. The client can then zoom and pan on the vector representation to more clearly see an overview or details in the document.

[0036] **Figure 2** illustrates various computer devices acquiring vector representations of Internet content through a proxy server according to one embodiment of the present invention. In this example, a user of a computer device, such as a personal computer **120**, hand held computer **121** or phone **122** requests to view Internet content. As stated above, this content may be in the form of an HTML or XML document. The request is sent across a network, for instance the Internet, using a protocol, such as HyperText Transfer Protocol (HTTP) **130**, **131** or Wireless Access Protocol (WAP) **132**. The proxy server 125 receives the request and fetches the actual content **135**, **136**, **137** from the network. The proxy then converts the content into scaleable vector information that is sent back to the client. At this point, the user can take advantage of the benefits of vector information to zoom and pan in order to better see the information. The user may select a hyperlink from the content, which starts the process over again.

[0037] The client may be a personal computer. Many Web pages are design for a smaller resolution (for example 640x480 pixels) than typical personal computers display (for example 1024x768 pixels). Sometimes selected text on a Web page is much smaller than other text. Allowing users to magnify the entire page allows them to overcome these design flaws. Alternatively, the client may be a small device, such as a hand held computer or a cell phone, which has a smaller display area than common Web pages are designed for. A scaleable vector representation would enable these users to access the information on the billions of existing Web pages in a simple and reasonable way.

[0038] The client software may be a plug-in to a Web browser, such as Netscape Navigator or Microsoft Internet Explorer. The browser may be configured to handle the connection to the proxy server. Alternatively, the client software may be a Java applet running in a browser. Or the client software may be a stand-alone program that interfaces

CLAIMS

What is claimed is:

1. A method of presenting Web content on a client, the method comprising displaying a vector representation of requested Web content that substantially retains page layout and/or graphics associated with the requested Web content.

2. The method of claim 1, wherein prior to said displaying the method further comprising converting the requested Web content to the vector representation, the vector representation including information regarding an intended original page layout for the Web content.

3. The method of claim 2, wherein the converting is performed prior to receipt of the Web content by the client.

4. The method of claim 2, wherein the converting is performed local to the client.

5. The method of claim 1, wherein the client comprises a wireless Internet device, a cellular phone, a handheld computer, a computer with a high resolution display, a computer with a very large display, an electronic billboard, or a device having a display with an unconventional aspect ratio.

6. A method comprising:

receiving a request for Web content from a client; and

converting the requested Web content to a vector format and producing vector-formatted Web content corresponding to the requested Web content, the vector format enabling the client to substantially retain an original page

layout within a set of layouts originally intended to be associated with the requested Web content by including page layout information; and

providing the vector-formatted Web content to the client.

7. A method of displaying Web content on a client, the method comprising:

responsive to a request for Web content by the client, receiving, at the client, vector-formatted Web content representing the requested Web content converted to a vector format; and

displaying a vector representation of the requested Web content on the client by rendering the vector-formatted Web content.

8. The method of claim 7, wherein the vector representation substantially retains page layout and/or graphics associated with the requested Web content.

9. The method of claim 7, wherein the vector format includes layout information associated with the requested Web content, and wherein the method further comprises displaying the vector representation of the requested Web content in accordance with the layout information.

10. The method of claim 7, wherein the vector-formatted Web content is represented in the form of a database of mathematical elements rather than a set of static graphical images.

11. The method of claim 7, wherein the vector format comprises Standard Vector Format (SVF).

12. A method comprising:

a client requesting Web content from a server;

responsive to the client request, the server converting the requested Web content to a vector format and producing vector-formatted Web content corresponding to the requested Web content, the vector format enabling the client to substantially retain an original page layout within a set of layouts originally intended to be associated with the requested Web content by including page layout information;

the server transferring the vector-formatted Web content to the client;

the client receiving the vector-formatted Web content from the server; and

the client displaying a vector representation of the requested Web content on the client while substantially retaining the original page layout associated with the requested Web content by rendering the vector-formatted Web content in accordance with the page layout information.

13. The method of claim 12, further comprising manipulating the vector representation of the requested Web content and redisplaying the resulting vector representation without the need for further requests to the server.

14. The method of claim 13, wherein the manipulating the vector representation comprises panning on the vector representation.

15. The method of claim 13, wherein the manipulating the vector representation comprises zooming on the vector representation.

16. A system for transferring and displaying Web content comprising:

a server to deliver Web content in response to content requests and to convert the Web content to a vector format that retains an original page layout within a

set of layouts originally intended to be associated with the Web content by including page layout information; and

a client communicatively coupled to the server to receive requested Web content and to display a vector representation of the requested Web content on the client while substantially retaining the original page layout associated with the requested Web content by rendering the vector-formatted Web content in accordance with the page layout information.

17. The system of claim 16, wherein the server is associated with a web site and conversion of Web content to the vector format is a service provided by the web site.

18. The system of claim 16, wherein the server is associated with a wireless service provider and conversion of Web content to the vector format is a service provided by the wireless service provider.

19. The system of claim 16, wherein the server is associated with an Internet Service Provider (ISP) and conversion of Web content to the vector format is a service provided by the ISP.

20. The system of claim 16, wherein the client comprises a wireless Internet device.

21. The system of claim 16, wherein the client comprises a cellular phone.

22. The system of claim 16, wherein the client comprises a handheld computer.

23. The system of claim 16, wherein the client comprises a computer with a high resolution display.

24. The system of claim 16, wherein the client comprises a computer with a very large display.

25. The system of claim 16, wherein the client comprises an electronic billboard.

26. The system of claim 16, wherein the client comprises a device having a display with an unconventional aspect ratio.

27. A system for transferring and displaying Web content comprising:
a content delivery means for delivering Web content in response to content requests and for converting the Web content to a vector format that retains an original page layout within a set of layouts originally intended to be associated with the Web content by including page layout information; and
a content display means, communicatively coupled to the content delivery means, for receiving requested Web content and for displaying a vector representation of the requested Web content on the client while substantially retaining the original page layout associated with the requested Web content by rendering the vector-formatted Web content in accordance with the page layout information.

28. A machine-readable medium having stored thereon data representing sequences of instructions, the sequences of instructions which, when executed by a processor, cause the processor to:
receive a request for Web content from a client; and
produce vector-formatted Web content corresponding to the requested Web content by converting the requested Web content to a vector format, the

vector format enabling the client to substantially retain an original page layout within a set of layouts originally intended to be associated with the requested Web content by including page layout information; and

provide the vector-formatted Web content to the client.

29. A machine-readable medium having stored thereon data representing sequences of instructions, the sequences of instructions which, when executed by a processor, of a client cause the processor to:

receive, at the client, vector-formatted Web content representing requested Web content converted to a vector format; and

display a vector representation of the requested Web content on the client by rendering the vector-formatted Web content.

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No.
PCT/US01/40920

V. Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement

1. statement

Novelty (N)	Claims	1-38	YES
	Claims	NONE	NO
Inventive Step (IS)	Claims	1-38	YES
	Claims	NONE	NO
Industrial Applicability (IA)	Claims	1-38	YES
	Claims	NONE	NO

2. citations and explanations (Rule 70.7)

Claims 1-38 meet the criteria set out in PCT Article 33(2)-(4), because the prior art does not teach or fairly suggest a method comprises step of *translating Web content from its original format into a scalable vector representation of the Web content, wherein the scalable vector representation of the Web content provides a scalable resolution-independent display of the content that substantially retains the original page layout and attributes of the content defined by its original format when rendered.*

NEW CITATIONS

US 6,057,854 A (DAVIS, JR. et al.) 02 MAY 2000, see col. 2, lines 5-35.
US 6,011,905 A (HUTTENLOCHER et al.) 04 JANUARY 2000, see abstract and fig. 10.

878097
828511

Form PCT/IPEA/409 (Box V) (July 1998)*

Exhibit 3D

BRIEF SUMMARY OF THE INVENTION

[0007] A method and apparatus for supporting resolution independent vector display of Internet content is disclosed. According to one embodiment, novel client processing of Web content is provided. The client receives requested Web content and displays a vector representation of the requested Web content that substantially retains page layout and/or graphics associated with the requested Web content.

[0008] According to another embodiment, novel server processing of Web content is provided. First, the server receives a request for Web content from a client. The requested Web content is then translated into a scalable vector format to produce vector-formatted Web content corresponding to the requested Web content. The vector format enables the client to substantially retain an original page layout within a set of layouts originally intended to be associated with the requested Web content by including page layout information in a vector database. Finally, the vector-formatted Web content is provided to the client.

[0009] Other features of the present invention will be apparent from the accompanying drawings and from the detailed description that follows.

CLAIMS

What is claimed is:

1. A method comprising:

retrieving Web content having an original format defining an original page layout and attributes of the Web content from an Internet site in response to a request of the Web content from a client; and

translating the Web content from its original format into a scalable vector representation of the Web content, wherein the scalable vector representation of the Web content provides a scalable resolution-independent display of the content that substantially retains the original page layout and attributes of the content defined by its original format when rendered.

2. The method of claim 1, wherein the Web content comprises a Web page that is stored on the Internet site in one or more markup language documents that include markup language code defining the original page layout and attributes of the Web page.

3. The method of claim 1, wherein the Web content is translated from its original format into the scalable vector representation on the client.

4. The method of claim 1, wherein the content is translated from it original format into the scalable vector representation on a computing device that is remote from the client, further comprising sending the scalable vector representation of the Web content to the client.

5. The method of claim 4, wherein the computing device that is remote from the client is operated by the Internet site that receives a request for the Web content from the client via the Internet, retrieves the Web content in its original format and translates it into the scalable vector representation of the Web content, and returns the scalable vector representation of the Web content via the Internet to the client.

6. The method of claim 4, wherein the computing device that is remote from the client comprises a proxy server that receives the request of the Web content from the client via the Internet, retrieves the Web content in its original format from the Internet site, translates the content from its original format into the scalable vector representation of the Web content, and sends the scalable vector representation of the Web content back to the client via the Internet.

7. The method of claim 6, wherein the proxy server appears transparent to the client.

8. The method of claim 1, wherein the client comprises one of a wireless Internet device, a cellular phone, a handheld computer, a desktop computer or workstation, a laptop computer, an electronic billboard, or a device having a display with an unconventional aspect ratio.

9. A method comprising:

enabling a client to request Web content comprising a Web page that is stored on an Internet site in an original format that include markup language code defining an original page layout and attributes of objects included in the Web page,

routing the request through a proxy server that retrieves the Web content in its original format from the Internet site and translates it into a scalable vector representation of the Web content, wherein the scalable vector representation of the Web content provides a scalable resolution-independent display of the content that substantially retains the original page layout and attributes of the Web page's objects defined by the original format of the Web page when rendered; and

sending the scalable vector representation of the Web content from the proxy server back to the client.

10. The method of claim 9, wherein the client enables a user to request the Web page in a conventional manner by entering a URL (uniform resource locator) for the Web page via a browser hosted by the client, and a software component running on the client intercepts the request for the Web page and routes the request to the proxy server rather than an Internet site corresponding to the URL such that the proxy server appears transparent to the client.

11. The method of claim 9, wherein translating the Web content from its original format into the scalable vector representation of the Web content comprises:

parsing the markup language code to determine the original page layout of the Web page, wherein the original page layout defines a layout location for a plurality of text objects and/or graphic image objects included in the Web page;

defining a primary datum corresponding to the original page layout;

defining an object datum corresponding to the layout location for each of said plurality of text objects, graphic layout objects, and/or graphic image objects;

generating a vector from the primary datum to the object datum for each of said plurality of text objects, graphic layout objects, and/or graphic image objects; and

creating a reference that links each of said plurality of text objects, graphic layout objects, and/or graphic image objects to its corresponding vector.

12. The method of claim 11, further comprising translating graphic image objects from an original format into a scalable bitmap format.

13. The method of claim 11, further comprising storing attribute information pertaining to each text object, said attribute information including a color and typefont for each text object.

14. The method of claim 11, further comprising generating a bounding box for each object that substantially circumscribes a page layout area occupied by that object in correspondence with the original page layout.

15. A method of displaying Web content on a client, comprising:

responsive to a request for Web content by the client, said Web content comprising a Web page that is defined to be rendered at a predetermined resolution,

receiving, at the client, vector-formatted Web content comprising a scalable vector representation of the Web content that provides a scalable resolution-independent display of the Web content that substantially retains an original page layout and attributes of the Web content corresponding to an appearance of the Web page when it is rendered at its predetermined resolution; and

rendering the vector-formatted Web content on the client such that it is displayed to have a different resolution than the predetermined resolution.

16. The method of claim 15, wherein the Web content is displaced at a different resolution by scaling the vector-formatted Web content.

17. The method of claim 16, further comprising enabling a user of the client to select a zoom factor by which the vector-formatted Web content is scaled.

18. The method of claim 16, further comprising enabling a user to cause a display of the Web content to be both scaled and offset.

19. The method of claim 15, wherein the page layout of the Web page is defined to have an original aspect ratio, and wherein the vector-formatted Web content is scaled so as to produce a display having a different aspect ratio.

20. The method of claim 15, wherein the scalable vector representation of the Web content comprises Simple Vector Format (SVF).

21. A computer system comprising:

a processor,

a communications device coupled to the processor, to enable the computer system to be linked via the Internet to a client and an Internet site; and

a memory, coupled to the processor, in which a plurality of machine-executable instructions are stored that when executed by the processor enable the computer system to perform the operations of:

retrieving Web content having an original format defining an original page layout and attributes of the Web content from the Internet site in response to a request of the Web content from the client; and

translating the Web content from its original format into a scalable vector representation of the Web content, wherein the scalable vector representation of the Web content provides a scalable resolution-independent display of the content that substantially retains the original page layout and attributes of the content defined by its original format when rendered.

22. The computer system of claim 21, wherein execution of the plurality of the machine instructions translates the Web content from its original format into the scalable vector representation of the Web content by performing the operation of:

parsing the markup language code to determine the original page layout of the Web page, wherein the original page layout defines a layout location for a plurality of text objects and/or graphic image objects included in the Web page;

defining a primary datum corresponding to the original page layout;

defining an object datum corresponding to the layout location for each of said plurality of text objects, graphic layout objects, and/or graphic image objects;

generating a vector from the primary datum to the object datum for each of said plurality of text objects, graphic layout objects, and/or graphic image objects; and

creating a reference that links each of said plurality of text objects, graphic layout objects, and/or graphic image objects to its corresponding vector.

23. The computer system of claim 22, wherein execution of the plurality of the machine instructions further performs the operation of translating graphic image objects from an original format into a scalable bitmap format.

24. The computer system of claim 22, wherein execution of the plurality of the machine instructions further performs the operation storing attribute information pertaining to each text object, said attribute information including a color and typefont for each text object.

25. A computer system comprising:

a processor,

a communications device coupled to the processor, to enable the computer system to be linked via the Internet an Internet site; and

a memory, coupled to the processor, in which a plurality of machine-executable instructions are stored that when executed by the processor enable the computer system to perform the operations of:

receiving vector-formatted Web content comprising a scalable vector representation of a Web page that is originally defined to be rendered at a predetermined resolution, said vector-formatted Web content providing a scalable resolution-independent display of the Web page that substantially retains an original page layout and attributes of the Web page corresponding to an appearance of the Web page when it is rendered at its predetermined resolution; and

rendering the vector-formatted Web content on the client such that it is displayed to have a different resolution than the predetermined resolution.

26. The computer system of claim 25, wherein the Web content is displayed at a different resolution by scaling the vector-formatted Web content.

27. The computer system of claim 25, further comprising enabling a user of the client to select a zoom factor by which the vector-formatted Web content is scaled.

28. The computer system of claim 25, wherein execution of the plurality of machine instructions by the processor further performs the operation of enabling a user to cause a display of the Web content to be both scaled and offset.

29. The computer system of claim 25, wherein the page layout of the Web page is defined to have an original aspect ratio, and wherein the vector-formatted Web content is scaled so as to produce a display having a different aspect ratio.

30. A machine-readable medium having stored thereon a plurality of machine-executable instructions that when executed by a machine performs the operations of:

retrieving Web content having an original format defining an original page layout and attributes of the Web content from an Internet site in response to a request of the Web content from the client;

translating the Web content from its original format into a scalable vector representation of the Web content, wherein the scalable vector representation of the Web content provides a scalable resolution-independent display of the content that substantially retains the original page layout and attributes of the content defined by its original format when rendered.

31. The machine-readable medium of claim 30, wherein execution of the plurality of the machine instructions translates the Web content from its original format into the scalable vector representation of the Web content by performing the operation of:

parsing the markup language code to determine the original page layout of the Web page, wherein the original page layout defines a layout location for a plurality of text objects and/or graphic image objects included in the Web page;

defining a primary datum corresponding to the original page layout;

defining an object datum corresponding to the layout location for each of said plurality of text objects, graphic layout objects, and/or graphic image objects;

generating a vector from the primary datum to the object datum for each of said plurality of text objects, graphic layout objects, and/or graphic image objects; and

creating a reference that links each of said plurality of text objects, graphic layout objects, and/or graphic image objects to its corresponding vector.

32. The machine-readable medium of claim 31, wherein execution of the plurality of the machine instructions further performs the operation of translating graphic image objects from an original format into a scalable bitmap format.

33. The machine-readable medium of claim 31, wherein execution of the plurality of the machine instructions further performs the operation storing attribute information pertaining to each text object, said attribute information including a color and typefont for each text object.

34. A machine-readable medium having stored thereon a plurality of machine-executable instructions that when executed by a machine performs the operations of:

receiving vector-formatted Web content comprising a scalable vector representation of a Web page that is originally defined to be rendered at a predetermined

resolution, said vector-formatted Web content providing a scalable resolution-independent display of the Web page that substantially retains an original page layout and attributes of the Web page corresponding to an appearance of the Web page when it is rendered at its predetermined resolution; and

rendering the vector-formatted Web content on the client such that it is displayed to have a different resolution than the predetermined resolution.

35. The machine-readable medium of claim 34, wherein the Web content is displayed at a different resolution by scaling the vector-formatted Web content.

36. The machine-readable medium of claim 34, further comprising enabling a user of the client to select a zoom factor by which the vector-formatted Web content is scaled.

37. The machine-readable medium of claim 34, wherein execution of the plurality of machine instructions further performs the operation of enabling a user to cause a display of the Web content to be both scaled and offset.

38. The machine-readable medium 34, wherein the original page layout of the Web page is defined to have an original aspect ratio, and wherein the vector-formatted Web content is scaled so as to produce a display having a different aspect ratio.

ABSTRACT OF THE DISCLOSURE

Apparatus and methods are provided for creating resolution-independent vector display of Internet content to allow it to be scaled (zoomed) larger and smaller for better viewing or to fit any resolution or screen size. According to one embodiment, novel server processing of Web content is provided that converts Web content requested by a client to a scalable vector format, such as Simple Vector Format. The vector format enables the Web content to be rendered by the client such that the rendered display substantially retains an original page layout defined in mark-up language document(s) in which the Web content is stored. In one embodiment, a proxy server receives Web content requests from a client and translates the content from an original format into a scalable vector representation. The scalable vector representation is then sent to client, where it is scaled and/or offset, enabling users to zoom and/or pan the Web content.

AMENDMENT

In the Specification

Please replace paragraph [0007] with the following amended paragraph:

[0007] In accordance with aspects of the invention, ~~A method and apparatus~~ methods, apparatus, and software for supporting resolution independent vector display of Internet (i.e., Web) content [[is]] are disclosed. According to one ~~embodiment~~ aspect, novel client processing of vector-formatted Web content is provided to enable Web content to be scaled and panned on a client while supporting traditional Web browser functionality. ~~The client receives requested Web content and displays a vector representation of the requested Web content that substantially retains page layout and/or graphics associated with the requested Web content~~ The client processing enables the vector-formatted web content to be rendered at various scales and offsets, enabling users to iteratively zoom and pan display of web content. Moreover, support for traditional Web page functionality, such as hyperlinks, is provided, enabling Web content to be accessed in an enhanced manner on devices having various screen resolutions, including mobile devices such as cellular phones, PDAs, and handheld devices, as well as desktop computers and laptops.

Please remove paragraph [0008]

Please replace paragraph [0053] with the following amended paragraph:

[0053] In addition to HTML and other markup and scripting language content, it is very common for web pages to include graphical content. In general, graphical content is usually stored in an image file or files that are external from the parent HTML document for the web page. For example, the parent HTML document may contain one or more embedded image tags that reference the location where those images are stored. As before, the graphic images may be stored locally, or may be stored on

Exhibit 3E

RESOLUTION INDEPENDENT VECTOR DISPLAY OF INTERNET CONTENT

RELATED APPLICATIONS

[0001] The present application is a Divisional Application of U.S. Application No. 09/878,097, filed June 8, 2001, which is a Continuation-in-Part of U.S. Non-Provisional Application No. 09/828,511, filed April 7, 2001, entitled "RESOLUTION INDEPENDENT VECTOR DISPLAY OF INTERNET CONTENT," the benefit of the filing date of which is claimed under 35 U.S.C. § 120. This application further claims the benefit of the filing dates of U.S. Provisional Application No. 60/211,019, filed June 12, 2000, entitled "METHOD AND SYSTEM FOR RESOLUTION INDEPENDENT DISPLAY OF HTML AND XML CONTENT" and U.S. Provisional Application No. 60/217,345, filed July 11, 2000, entitled "METHOD AND SYSTEM FOR SELECTION, RETRIEVAL, AND CONVERSION OF COMPUTER CONTENT TO VECTOR FORMAT FOR RESOLUTION INDEPENDENT DISPLAY," under 35 U.S.C. § 119(e).

COPYRIGHT NOTICE

BACKGROUND OF THE INVENTION

Field of the Invention

[0003] The invention relates generally to translation of Internet and World Wide Web content to scalable vector representation. More particularly, the invention relates to apparatus and methods for zoom enabling the display of content in an Internet information browser by

retrieving and translating HyperText Markup Language (HTML), eXtensible Markup Language (XML), and other Internet content to vector representations of that content.

Description of the Related Art

[0004] Text only Internet information browsers began as a project at the CERN, European Organization for Nuclear Research, facility in Geneva Switzerland. From its inception the intent was to provide a mesh or web of access to data with a common user interface. Browsers moved from the academic environment when NCSA, the National Center for Supercomputing Applications at the University of Illinois in Urbana-Champaign developed Mosaic, an Internet information browser and World Wide Web client.

[0005] Internet content is stored in multiple file formats. These formats include HTML (Hyper Text Markup Language) and XML (eXtended Markup Language) as well as graphic file format GIF (Graphics Interchange Format) and JPEG (Joint Photographic Experts Group). These four file formats constitute the majority of Internet content. Font size and resizing display area for content can alter the size of the display of Internet content in existing browsers. The majority of Internet content displays as a flat single resolution with no browser support for zoom.

[0006] Much of the Internet content has been designed for display on desktop computers with a single target resolution. Even though HTML has the ability to adapt to changes in screen resolution, major Internet content providers have chosen to create their Web pages using fixed resolution structures, such as tables. This gives them the ability to control the look and feel of their Web sites. This fixed resolution approach has evolved to the point that the fixed resolution layout of Web pages has become the most common method to brand or uniquely identify Web sites. While this fixed resolution approach is good for site branding and product differentiation it does present a daunting technical problem for display of Internet content (designed for desktop computers) on small screen, low resolution, or different aspect ratio devices, such as cell phones and hand held computers.

BRIEF SUMMARY OF THE INVENTION

[0007] A method and apparatus for supporting resolution independent vector display of Internet content is disclosed. According to one embodiment, novel client processing of Web content is provided. The client receives requested Web content and displays a vector representation of the requested Web content that substantially retains page layout and/or graphics associated with the requested Web content.

[0008] According to another embodiment, novel server processing of Web content is provided. First, the server receives a request for Web content from a client. The requested Web content is then translated into a scalable vector format to produce vector-formatted Web content corresponding to the requested Web content. The vector format enables the client to substantially retain an original page layout within a set of layouts originally intended to be associated with the requested Web content by including page layout information in a vector database. Finally, the vector-formatted Web content is provided to the client.

[0009] Other features of the present invention will be apparent from the accompanying drawings and from the detailed description that follows.

translated into a scalable vector representation (i.e., SVF, also referred to herein as "vectorized content") through use of a proxy server 32 and sent to the requesting client. Upon being received by the client, the vectorized content is processed and rendered using a thin client to enable a user to view the content on the client device.

[0049] With reference to the flowchart of FIGURE 2A, the foregoing process is initiated by a client in a block 100, wherein the client submits a request to proxy server 32 to retrieve and convert selected content. As depicted by a transfer path 34, this comprises sending data 36, which includes content network location indicia from which the content can be retrieved and proxy server network location information by which the content request may be delivered to over Internet 24 to proxy server 32. Typically, it will be desired to retrieve a particular web page. Accordingly, the content network location indicia will comprise a URL (uniform resource locator) for the web page. Similarly, the proxy server network location information may also comprise a URL corresponding to a network access point for the proxy server. Optionally, the location information may comprise a network IP address for one or both of the content location and the proxy server location. If the content is to be retrieved from an Internet resource, the request will typically be sent using the HyperText Transfer Protocol (HTTP) over the TCP/IP transport.

[0050] Next, in a block 102, the request is received by the proxy server and the proxy server checks its cache to see if it already has the request content in its cache. If it does, it sends this cached content back to the client. If it does not have the requested content cached, the proxy server sends out a request to retrieve the content from the network resource. For illustrative purposes, it will be assumed for the present example that the desired content comprises a web page that is stored on web server 26. Typically, when the requested content comprises a web page, the content may be retrieved using conventional web content retrieval techniques, such as that employed by various modern browser clients, including Netscape Navigator and Internet Explorer. This generally comprises providing routing information, such as the URL for the web page (URL 38) to routing services provided by Internet 24,

corresponding to the embedded references will usually require additional transfer time. Furthermore, graphic content oftentimes comprises significantly larger file sizes than HTML content, leading to significant transfer times in some instances. For simplicity, the transfer of the various HTML documents and graphic files for the content request are depicted by HTML documents 52 and graphic documents 54, which are transferred over a transfer path 56.

[0056] When the HTML documents and graphic content are received by proxy server 32, a scalable vector representation of the web page is generated in a block 114 by an HTML translator 58. In brief, HTML translator 58 translates HTML, XML, and cascaded style sheet (CSS) layout content into a scalable vector representation, such as SVF. Details of the HTML translation process are contained below. In addition, the graphic images are converted into a compressed bitmap format in a block 116 by a graphics translator 60. The vectorized content 62 and compressed bitmaps 64 are then streamed back to the client (i.e., computer 18) in a block 118, as depicted by a transfer path 66. In one embodiment, the content portions are sent in separate streams using multiple connections. In another embodiment, the content portions are sent via a multiplexed stream using a single connection. As the vectorized content and compressed bitmap data are received by the client device, they are processed by a thin client 68 running on the client device, whereby a representation of the original web page content may be rendered on the client device's display screen at various user-selectable scaled resolutions and pan offsets in a block 120, thereby enabling a user to more clearly see an overview or details in the web page. Further details of the client side processing are provided below.

[0057] As discussed above, wireless clients may also access the vectorized network (e.g., web site) content provided via proxy server 24. The majority of this process is identical to that described above for land-line clients (e.g., computers 18, 20, and 22), except for provisions required for sending data to and receiving data from wireless devices. In general, most wireless devices will access the Internet via a wireless service provider (i.e., a wireless

At present, the software for the Mozilla rendering engine may be accessed via the Internet at www.mozilla.org. Accordingly, in one embodiment, the present invention uses core functionality provided by the Mozilla rendering engine source code to perform the functions of block 150, 152, and 154.

[0083] At this point, the present invention deviates substantially from the prior art by using the various object layout data generated during the pre-rendering process to generate a scalable vector representation of the original page content. First, in a block 156, a datum point is defined for the page and the bounding box for each object. For example, as shown in FIGURE 4C, a rendered page datum 262 is defined to be coincident with the upper left hand corner of the display frame of the rendered page for the web page. Generally, any point on the page may be used as the page datum – the only requirement is that the page datum that is selected is used consistently throughout the process. The use of the upper left hand corner of the display frame is advantageous since the location of the first object encountered in the HTML code for a page is located relative to this corner.

[0084] In general, the datum points for each object may also be located any place on the object, as long as the object datum points are used in a predictable manner. For example, as depicted in FIGURE 4C, various datum points for corresponding objects are defined to be coincident with the upper left hand corner of the bounding box for that object, wherein the object's datum point shares the root reference number of the object with an appended "C."

[0085] Once the page's datum point and an object's datum point are known, a vector between these points is generated for each object in a block 158. With reference to FIGURE 4D, in one embodiment, wherein the page datum point corresponds to the upper left and corner of the display frame and is assigned an XY value 266 of 0,0, the vector for a given object may be stored as the XY value of the datum point of that object relative to 0,0, such as a value of 150, 225 (ref. num. 268) for a vector 250D pointing to an object datum 250C, and a value of 150, 425 (ref. num. 270) for a vector 252D pointing to an object datum 252C. In another embodiment, each vector may be stored as XY data relative to a 0,0 datum point

corresponding to the upper left hand corner of the frame the object belongs to. For example, a vector 250D' from a frame datum 214D to object datum 250C is stored as 20, 200 (ref. num. 268'), while a vector 252D from frame datum 214D to object datum 252C is stored as 20, 425. In this embodiment, offset information for each frame relative to a known datum will also be stored, as depicted by a vector 214D.

[0086] The scalable vector representation is completed in a block 160, wherein a reference is created for each object that includes or links an object's content and attributes, such as object type (e.g., text, image), object typeface, and boundary box parameters, to the object's vector. For example, object 250B is a graphic image having a vector 250D and a bounding box that is 180 pixels high and 350 pixels wide, while object 252B is a graphic image having a vector 252D and a bounding box that includes a height of 200 pixels and a width of 350 pixels. . This enables client-side operations to be performed that only initially consider the vectors, wherein if it is determined that a vector's endpoint (and/or the bounding box corresponding to the object the vector points to) would appear off of a display, there is no need to retrieve the content and attribute data linked to the vector. This concept is explained in further detail in the following section.

[0087] It is noted that a portion of the display content produced on a client device will never contain any rendered content, as this portion is reserved for the browser's user interface. In WINDOWS™ environments, this portion will include the browser's window frame, as well as the pulldown and icon menus provided in the browser's user interface, which are depicted by a box 264 in the Figures herein.

[0088] Client-Side Software and Processing

[0089] As discussed above, the present invention supports a wide variety of clients, including land-based clients and wireless clients. Each client requires some client-side software that enables the scalable vector content data provided to it to be rendered at a user-

CLAIMS

What is claimed is:

1. A method comprising:

retrieving Web content having an original format defining an original page layout and attributes of the Web content from an Internet site in response to a request of the Web content from a client; and

translating the Web content from its original format into a scalable vector representation of the Web content, wherein the scalable vector representation of the Web content provides a scalable resolution-independent display of the content that substantially retains the original page layout and attributes of the content defined by its original format when rendered.

2. The method of claim 1, wherein the Web content comprises a Web page that is stored on the Internet site in one or more markup language documents that include markup language code defining the original page layout and attributes of the Web page.

3. The method of claim 1, wherein the Web content is translated from its original format into the scalable vector representation on the client.

4. The method of claim 1, wherein the content is translated from it original format into the scalable vector representation on a computing device that is remote from the client, further comprising sending the scalable vector representation of the Web content to the client.

5. The method of claim 4, wherein the computing device that is remote from the client is operated by the Internet site that receives a request for the Web content from the client via the Internet, retrieves the Web content in its original format and translates it into the scalable

vector representation of the Web content, and returns the scalable vector representation of the Web content via the Internet to the client.

6. The method of claim 4, wherein the computing device that is remote from the client comprises a proxy server that receives the request of the Web content from the client via the Internet, retrieves the Web content in its original format from the Internet site, translates the content from its original format into the scalable vector representation of the Web content, and sends the scalable vector representation of the Web content back to the client via the Internet.

7. The method of claim 6, wherein the proxy server appears transparent to the client.

8. The method of claim 1, wherein the client comprises one of a wireless Internet device, a cellular phone, a handheld computer, a desktop computer or workstation, a laptop computer, an electronic billboard, or a device having a display with an unconventional aspect ratio.

9. A computer system comprising:

a processor,

a communications device coupled to the processor, to enable the computer system to be linked via the Internet to a client and an Internet site; and

a memory, coupled to the processor, in which a plurality of machine-executable instructions are stored that when executed by the processor enable the computer system to perform the operations of:

retrieving Web content having an original format defining an original page layout and attributes of the Web content from the Internet site in response to a request of the Web content from the client; and

translating the Web content from its original format into a scalable vector representation of the Web content, wherein the scalable vector representation of the Web content provides a scalable resolution-independent display of the content that substantially retains the original page layout and attributes of the content defined by its original format when rendered.

10. The computer system of claim 9, wherein execution of the plurality of the machine instructions translates the Web content from its original format into the scalable vector representation of the Web content by performing the operation of:

parsing the markup language code to determine the original page layout of the Web page, wherein the original page layout defines a layout location for a plurality of text objects and/or graphic image objects included in the Web page;

defining a primary datum corresponding to the original page layout;

defining an object datum corresponding to the layout location for each of said plurality of text objects, graphic layout objects, and/or graphic image objects;

generating a vector from the primary datum to the object datum for each of said plurality of text objects, graphic layout objects, and/or graphic image objects; and

creating a reference that links each of said plurality of text objects, graphic layout objects, and/or graphic image objects to its corresponding vector.

11. The computer system of claim 9, wherein execution of the plurality of the machine instructions further performs the operation of translating graphic image objects from an original format into a scalable bitmap format.

12. The computer system of claim 9, wherein execution of the plurality of the machine instructions further performs the operation storing attribute information pertaining to each text object, said attribute information including a color and typefont for each text object.

13. A machine-readable medium having stored thereon a plurality of machine-executable instructions that when executed by a machine performs the operations of:

retrieving Web content having an original format defining an original page layout and attributes of the Web content from an Internet site in response to a request of the Web content from the client;

translating the Web content from its original format into a scalable vector representation of the Web content, wherein the scalable vector representation of the Web content provides a scalable resolution-independent display of the content that substantially retains the original page layout and attributes of the content defined by its original format when rendered.

14. The machine-readable medium of claim 13, wherein execution of the plurality of the machine instructions translates the Web content from its original format into the scalable vector representation of the Web content by performing the operation of:

parsing the markup language code to determine the original page layout of the Web page, wherein the original page layout defines a layout location for a plurality of text objects and/or graphic image objects included in the Web page;

defining a primary datum corresponding to the original page layout;

defining an object datum corresponding to the layout location for each of said plurality of text objects, graphic layout objects, and/or graphic image objects;

generating a vector from the primary datum to the object datum for each of said plurality of text objects, graphic layout objects, and/or graphic image objects; and

creating a reference that links each of said plurality of text objects, graphic layout objects, and/or graphic image objects to its corresponding vector.

15. The machine-readable medium of claim 13, wherein execution of the plurality of the machine instructions further performs the operation of translating graphic image objects from an original format into a scalable bitmap format.

16. The machine-readable medium of claim 13, wherein execution of the plurality of the machine instructions further performs the operation storing attribute information pertaining to each text object, said attribute information including a color and typefont for each text object.

ABSTRACT OF THE DISCLOSURE

Apparatus and methods are provided for creating resolution-independent vector display of Internet content to allow it to be scaled (zoomed) larger and smaller for better viewing or to fit any resolution or screen size. According to one embodiment, novel server processing of Web content is provided that converts Web content requested by a client to a scalable vector format, such as Simple Vector Format. The vector format enables the Web content to be rendered by the client such that the rendered display substantially retains an original page layout defined in mark-up language document(s) in which the Web content is stored. In one embodiment, a proxy server receives Web content requests from a client and translates the content from an original format into a scalable vector representation. The scalable vector representation is then sent to client, where it is scaled and/or offset, enabling users to zoom and/or pan the Web content.

Please replace paragraph [0065] with the following amended paragraph:

[0049] The proxy server responds to client content requests by delivering content in one of the requested formats, by retrieving the content in an appropriate format from its cache, or from an upstream content source (again using standard HTTP content negotiation features), or by translating upstream content from a supported original format to SVF or the client bitmap format.

Please replace paragraph [0068] with the following amended paragraph:

[0068] The various content translators used by the proxy server accept (via HTTP PUT) or request (driven by HTTP proxy GET/POST) content in supported, but client-unsupported, formats; and return (via HTTP PUT or GET/POST response) one or more representations of that content in a client-supported format. In the embodiments illustrated in FIGURE 1A-C, two translators are used: HTML translator 58 and image translator 60. Future content types may be accommodated by new translators, by extending existing translators to cover the new content types, or by extending the ~~clients~~ client's capabilities. Standard HTTP content negotiation mechanisms are used to inform the proxy server of the ~~clients~~ client's capabilities and expectations on each request.

Please replace paragraph [0076] with the following amended paragraph:

[0076] With reference to the flowchart of FIGURE 5, the process for translating the HTML content into a scalable vector representation proceeds as follows. The process is initiated when the proxy server receives the HTML corresponding to the parent document (and frame documents, if appropriate), whereupon a pre-rendering parsing of the HTML is performed to determine where to place the various objects on the display page in a block 150. For example, elements such as tables, column

definitions, graphic images, paragraphs and line breaks are identified. If frames are included, each frame is examined in the sequential order it appears in the HTML document, or the order in which the HTML documents corresponding to the frames in a frameset are downloaded to the browser. During further processing, the actual objects are rendered in their respective positions. Some of these objects are rendered almost immediately, such as plain text, while other objects, such as graphic objects, must first be retrieved prior to being fully-rendered. With respect to tables, there are some instances in which [[the]] all of the objects corresponding to the cells in the table must be retrieved prior to rendering any of the table, while a well-designed table can be rendered incrementally. For example, by using Column grouping, the format of the corresponding table can be quickly determined by the browser. In some instances, one or more bitmaps may actually need to be fetched before the page layout can be determined.

Please replace paragraph [0086] with the following amended paragraph:

[0086] The scalable vector representation is completed in a block 160, wherein a reference is created for each object that includes or links an object's content and attributes, such as object type (e.g., text, image), object typeface, and boundary box parameters, to the object's vector. For example, object 250B is a graphic image having a vector 250D and a bounding box that is 180 pixels high and 350 pixels wide, while object 252B is a graphic image having a vector 252D and a bounding box that includes a height of 200 pixels and a width of 350 pixels. [[.]] This enables client-side operations to be performed that only initially consider the vectors, wherein if it is determined that a vector's endpoint (and/or the bounding box corresponding to the object the vector points to) would appear off of a display, there is no need to retrieve the content and attribute data linked to the vector. This concept is explained in further detail in the following section.

17. (New) The method of claim 1, wherein the browser is configured to run on at least one of a desktop computer, workstation, and laptop computer.

18. (New) The method of claim 1, wherein the browser is configured to run on a device having a display with an unconventional aspect ratio.

19. (New) The method of claim 1, further comprising:

parsing markup language code to determine the original page layout of the Web page, wherein the original page layout defines a layout location for a plurality of objects, including text objects, graphic layout objects, and/or graphic image objects included in the Web page;

defining a primary datum corresponding to the original page layout; and,

for each object,

defining an object datum corresponding to the layout location for the object;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to its corresponding vector.

20. (New) The method of claim 19, further comprising translating graphic image objects from an original format into a scalable format.

21. (New) The method of claim 19, further comprising storing attribute information pertaining to each text object, said attribute information including a color and typefont for each text object.

22. (New) The method of claim 6, wherein the requester is an Internet site.

Please replace the BRIEF SUMMARY OF THE INVENTION section with the following paragraphs:

[0007] In accordance with aspects of the invention, mobile devices enabled to support resolution-independent scalable display of Internet (Web) content to allow Web pages to be scaled (zoomed) and panned for better viewing on smaller screen sizes are disclosed. The mobile devices employ novel processing of original Web content, including HTML-based content, XML, cascade style sheets, etc. to generate scalable content. The scalable content and/or data derived therefrom are then employed to enable the Web content to be rapidly rendered, zoomed, and panned. Moreover, the rendered displays provide substantially the same or identical layout as the original Web page, enabling users to easily navigate to selected content and features on familiar Web pages. Display lists may also be employed to provide further enhancements in rendering speed. Additionally, hardware-based programmed logic may also be employed to facilitate various operations.

[0008] According to further aspects, some mobile devices may employ touch-sensitive display screens that enable users to provide various inputs to control display of content within Web pages. Exemplary user inputs include tap-based inputs to selectively zoom in on columns, images, and paragraphs. Users can also define a window to zoom in on via the touch-sensitive display.

[0009] Other features of the present invention will be apparent from the accompanying drawings and from the detailed description that follows.

Abstract

Please replace the Abstract with the following:

Mobile devices enabled to support resolution-independent scalable display of Internet (Web) content to allow Web pages to be scaled (zoomed) and panned for better viewing on smaller screen sizes. The mobile devices employ software-based processing of original Web content, including HTML-based content, XML, cascade style sheets, etc. to generate scalable content. The scalable content and/or data derived therefrom are then employed to enable the Web content to be rapidly rendered, zoomed, and panned. Moreover, the rendered displays provide substantially the same or identical layout as the original Web page, enabling users to easily navigate to selected content and features on familiar Web pages. Display lists may also be employed to provide further enhancements in rendering speed. Additionally, hardware-based programmed logic may be employed to facilitate various operations.

In the Claims

Please cancel claims 1-39 without prejudice.

Please add new claims 40 – 141 as follows.

40. (New) A mobile device, comprising:

a processor,

a wireless communications device operatively coupled to the processor, to facilitate communication with a network via which Web content may be accessed;

a display;

a memory, operatively coupled to the processor; and

storage means, operatively coupled to the processor, in which a plurality of instructions are stored that when executed by the processor enable the mobile device to perform operations including,

enabling a user to request access to a Web page having an original format comprising HTML-based Web content defining an original page layout of content on the Web page;

retrieving at least a portion of the HTML-based Web content associated with the Web page;

translating the at least a portion of the HTML-based Web content to produce scalable page layout information; and

employing the scalable page layout information and/or data derived therefrom to,

render at least a portion of the Web page on the display using a first scale factor; and

re-render the Web page in response to associated user inputs to enable a user to zoom in and out a display of the Web page.

41. (New) The mobile device of claim 40, wherein the device comprises a mobile phone.

42. (New) The mobile device of claim 40, wherein the device comprises one of a Personal Digital Assistant (PDA), handheld computer, notebook computer, or laptop computer.

43. (New) The mobile device of claim 40, wherein execution of the instructions performs further operations comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a corresponding user interface input.

44. (New) The mobile device of claim 43, wherein the display of the Web page is re-rendered substantially in real-time to effect zooming operations.

45. (New) The mobile device of claim 40, wherein the Web content includes at least one hyperlink, and wherein execution of the instructions performs further operations comprising:

enabling the user to select the hyperlink; and, in response thereto,

retrieving and translating HMTL-based Web content associated with the hyperlink to produce additional scalable page layout information; and

employing the additional scalable page layout information and/or data derived therefrom to render the Web content associated with the hyperlink on the display.

46. (New) The mobile device of claim 40, wherein execution of the instructions performs further operations comprising:

parsing HTML-based code to determine the original page layout of display content within the Web page, wherein the original page layout defines a layout location

for a plurality of objects, including text objects, graphic layout objects, and/or graphic image objects included in the Web page;

defining a primary datum corresponding to the original page layout; and,

for each object,

defining an object datum corresponding to the layout location for the object;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to its corresponding vector.

47. (New) The mobile device of claim 40, wherein execution of the instructions performs further operations comprising enabling the Web content to be displayed at different resolutions by,

generating scalable content via use of the scalable page layout information; and

scaling the scalable content to re-render the display in response to associated user inputs.

48. (New) The mobile device of claim 40, wherein execution of the instructions performs further operations comprising returning the display of the Web content to a previous view in response to a corresponding user input made.

49. (New) The mobile device of claim 40, wherein execution of the instructions performs further operations comprising enabling a user to pan a display of the Web content in response to a corresponding user input.

50. (New) The mobile device of claim 49, wherein execution of the instructions performs further operations comprising enabling the display of the Web content to be panned substantially in real-time.

51. (New) The mobile device of claim 40, wherein the page layout of the Web page is defined to have an original aspect ratio, and wherein the scalable page layout information and/or data derived therefrom is scaled to render a display having a different aspect ratio.

52. (New) The mobile device of claim 40, wherein execution of the instructions performs further operations comprising enabling a user to view a column of the Web content at a higher resolution than a current resolution via a corresponding user interface input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is displayed substantially across the display.

53. (New) The mobile device of claim 52, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

54. (New) The mobile device of claim 52, wherein the display comprises a touch-sensitive display, and wherein the corresponding user interface input comprises tapping on the column.

55. (New) The mobile device of claim 40, wherein the Web content includes at least one image, and wherein execution of the instructions performs further operations comprising enabling a user to view an image at a higher resolution than a current resolution via a corresponding user interface input, wherein in response thereto, the display is re-rendered such that the image is displayed substantially across at least one of a width and height of a display area of the display.

56. (New) The mobile device of claim 52, wherein the display comprises a touch-sensitive display, and wherein the corresponding user interface input comprises tapping on the image.

57. (New) The mobile device of claim 40, wherein execution of the instructions performs further operations comprising enabling a user to view a paragraph of the Web content at a higher resolution than a current resolution via a corresponding user interface input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is displayed substantially across at least one of a width and height of a display area of the display.

58. (New) The mobile device of claim 57, wherein the content of the paragraph is reformatted to fit characteristics of the display area when the display is re-rendered.

59. (New) The mobile device of claim 40, wherein the Web page includes text, layout attributes, and images, and wherein execution of the instructions performs further operations comprising:

receiving content corresponding to the text and layout attributes via a first connection; and

receiving content corresponding to at least one image via a second connection.

60. (New) The mobile device of claim 40, wherein execution of the instructions performs further operations comprising:

generating a display list of vectors derived, at least in part, via use of the scalable page layout information; and

employing the display list to re-render the display of the Web page.

61. (New) The mobile device of claim 40, wherein execution of the instructions performs further operations comprising:

parsing HTML-based code corresponding to the received Web content to determine the original page layout of the content on the Web page;

logically grouping selected content into objects;

defining a primary datum corresponding to the original page layout; and,

retrieving content associated with that object bounding box;

clipping portions of the content that are outside the virtual display bounding box; and

applying an appropriate scale factor to the content within the virtual display bounding box to render the content on the display.

65. (New) The mobile device of claim 40, wherein the Web content includes text content, and wherein execution of the instructions performs further operations comprising scaling a scalable font to render the text content at different scale factors.

66. (New) The mobile device of claim 40, wherein the original format of the Web page defines a height and width for the Web page, and wherein execution of the instructions performs further operations comprising:

determining an applicable scale factor to display at least one of the width and height of the Web page substantially across a display area of the display; and

employing the scale factor that is determined as the first scale factor.

67. (New) The mobile device of claim 40, wherein at least a portion of the instructions comprise Java-based instructions configured to be executed on a Java virtual machine.

68. (New) The mobile device of claim 40, wherein the Web content includes cascaded style sheets.

69. (New) The mobile device of claim 40, wherein the scalable page layout information is vector-based.

70. (New) The method of claim 40, wherein the network comprises a mobile service provider network.

71. (New) A wireless device, comprising:

processing means,

wireless communications means, to facilitate wireless communication with a network via which Web content may be accessed;

a display;

memory; and

storage means, in which a plurality of instructions are stored that when executed by the processing means enable the wireless device to perform operations including,

rendering a browser interface via which a user is enabled to request access to a Web page, the Web page including associated Web content having an original format defining an original page layout and attributes of content on the Web page;

retrieving and translating at least a portion of the Web content from its original format into scalable content that supports a scalable resolution-independent display of the content that substantially retains the original page layout and attributes of the content defined by its original format when rendered; and

employing the scalable content to render at least a portion of the Web page on the display using a first scale factor.

72. (New) The wireless device of claim 71, wherein execution of the instructions performs further operations comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a corresponding user interface input.

73. (New) The wireless device of claim 72, wherein the display of the Web page is re-rendered substantially in real-time to effect zooming operations.

77. (New) The wireless device of claim 71, wherein execution of the instructions performs further operations comprising returning the display of the Web content to a previous view in response to a corresponding user input.

78. (New) The wireless device of claim 71, wherein execution of the instructions performs further operations comprising enabling a user to pan a display of the Web content in response to a corresponding user input.

79. (New) The wireless device of claim 78, wherein execution of the instructions performs further operations comprising enabling the display of the Web content to be panned substantially in real-time.

80. (New) The wireless device of claim 71, wherein the page layout of the Web page is defined to have an original aspect ratio, and wherein the scalable content is scaled when rendered so as to produce a display having a different aspect ratio.

81. (New) The wireless device of claim 71, wherein the display comprises a touch-sensitive display, and wherein execution of the instructions performs further operations comprising enabling a user to view a column of the Web content at a higher resolution than a current resolution by tapping on the column via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is displayed substantially across the touch-sensitive display.

82. (New) The wireless device of claim 81, wherein the content of the column is reformatted to fit characteristics of the touch-sensitive display when the display is re-rendered.

83. (New) The wireless device of claim 71, wherein the display comprises a touch-sensitive display, and wherein the Web content includes at least one image, and wherein execution of the instructions performs further operations comprising enabling a

user to view an image at a higher resolution than a current resolution by tapping on the image via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that the image is displayed substantially across at least one of a width and height of a display area of the touch-sensitive display.

84. (New) The wireless device of claim 71, wherein the display comprises a touch-sensitive display, and wherein execution of the instructions performs further operations comprising enabling a user to view a paragraph of the Web content at a higher resolution than a current resolution by tapping on the paragraph via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is displayed substantially across at least one of a width and height of a display area of the touch-sensitive display.

85. (New) The wireless device of claim 84, wherein the content of the paragraph is reformatted to fit characteristics of the display area when the display is re-rendered.

86. (New) The wireless device of claim 71, wherein the Web page includes text, layout attributes, and images, and wherein execution of the instructions performs further operations comprising:

receiving content corresponding to the text and layout attributes via a first connection; and

receiving content corresponding to at least one image via a second connection.

87. (New) The wireless device of claim 71, wherein execution of the instructions performs further operations comprising:

generating a vector-based display list associated with the scalable content; and

employing the display list to re-render the display at different scale factors to enable rapid zooming of the Web page.

to a rendered display of the Web content desired by a user;

determining a virtual display bounding box for the virtual display associated with the first scale factor and offset;

identifying object bounding boxes having at least a portion falling within the virtual display bounding box; and,

for each of such object bounding boxes,

retrieving content associated with that object bounding box;

clipping portions of the content that are outside the virtual display bounding box; and

applying an appropriate scale factor to the content within the virtual display bounding box to render the content on the display.

92. (New) The wireless device of claim 71, wherein the scalable content includes scalable text content, and wherein execution of the instructions performs further operations comprising scaling a scalable font to render the scalable text content.

93. (New) The wireless device of claim 71, wherein the original format of the Web page defines a height and width for the Web page, and wherein execution of the instructions performs further operations comprising:

determining an applicable scale factor to display at least one of the width and height of the Web page substantially across a display area of the display; and

employing the scale factor that is determined as the first scale factor.

94. (New) The wireless device of claim 71, wherein at least a portion of the instructions comprise Java-based instructions configured to be executed on a Java virtual machine.

95. (New) The wireless device of claim 71, wherein the device comprises a mobile phone.

phone.

101. (New) The mobile device of claim 99, wherein the device comprises one of a Personal Digital Assistant (PDA), handheld computer, notebook computer, or laptop computer.

102. (New) The mobile device of claim 99, wherein execution of the instructions performs further operations comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input made via the touch-sensitive display.

103. (New) The mobile device of claim 102, wherein the user interface input enables the user to define a window of a current view of the Web page on which to zoom in on.

104. (New) The mobile device of claim 99, wherein the display of the Web page is re-rendered substantially in real-time to effect zooming operations.

105. (New) The mobile device of claim 99, wherein the Web page includes at least one hyperlink, and wherein execution of the instructions performs further operations comprising:

enabling the user to select the hyperlink via the touch-sensitive display; and, in response thereto,

retrieving and processing HMTL-based Web content associated with the hyperlink to produce additional scalable content; and

employing the additional scalable content and/or data derived therefrom to render the Web content associated with the hyperlink on the touch-sensitive display.

106. (New) The mobile device of claim 99, wherein at least a portion of the scalable content comprises scalable vector-based content.

107. (New) The mobile device of claim 99, wherein execution of the instructions performs further operations comprising returning the display of the Web page to a previous view in response to a corresponding user input made via the touch-sensitive display.

108. (New) The mobile device of claim 99, wherein execution of the instructions performs further operations comprising enabling a user to pan a display of the Web content in response to a corresponding user input made via the touch-sensitive display.

109. (New) The mobile device of claim 108, wherein execution of the instructions performs further operations comprising enabling the display of the Web content to be panned substantially in real-time.

110. (New) The mobile device of claim 99, wherein the page layout of the Web page is defined to have an original aspect ratio, and wherein the scalable content and/or data derived therefrom is scaled to render a display having a different aspect ratio.

111. (New) The mobile device of claim 99, wherein execution of the instructions performs further operations comprising enabling a user to view a column of the Web content at a higher resolution than a current resolution by tapping on the column via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is displayed substantially across the touch-sensitive display.

112. (New) The mobile device of claim 111, wherein the content of the column is reformatted to fit characteristics of the touch-sensitive display when the display is re-rendered.

113. (New) The mobile device of claim 99, wherein the Web content includes at least one image, and wherein execution of the instructions performs further operations

comprising enabling a user to view an image at a higher resolution than a current resolution by tapping on the image via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that the image is displayed substantially across at least one of a width and height of a display area of the touch-sensitive display.

114. (New) The mobile device of claim 99, wherein execution of the instructions performs further operations comprising enabling a user to view a paragraph of the Web content at a higher resolution than a current resolution by tapping on the paragraph via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is displayed substantially across at least one of a width and height of a display area of the touch-sensitive display.

115. (New) The mobile device of claim 114, wherein the content of the paragraph is reformatted to fit characteristics of the display area when the display is re-rendered.

116. (New) The mobile device of claim 99, wherein the Web page includes text, layout attributes, and images, and wherein execution of the instructions performs further operations comprising:

receiving content corresponding to the text and layout attributes via a first connection; and

receiving content corresponding to at least one image via a second connection.

117. (New) The mobile device of claim 99, further comprising dynamic memory having at least a portion employed for rendering purposes, wherein execution of the instructions performs further operations comprising:

building a display list via use of the scalable content and rendering display list content on a virtual display in the dynamic memory; and

scaling the display list content to re-render the display of the Web page.

determining a first scale factor and offset in response to one or more corresponding user inputs defining a user-selectable zoom level and pan corresponding to a rendered display of the Web page desired by a user;

determining a virtual display bounding box for the virtual display associated with the first scale factor and offset;

identifying object bounding boxes having at least a portion falling within the virtual display bounding box; and,

for each of such object bounding boxes,

retrieving content associated with that object bounding box;

clipping portions of the content that are outside the virtual display bounding box; and

applying an appropriate scale factor to the content within the virtual display bounding box to render the content on the display.

122. (New) The mobile device of claim 99, wherein the scalable content includes scalable text content, and wherein execution of the instructions performs further operations comprising scaling a scalable font to render the scalable text content.

123. (New) The mobile device of claim 99, wherein the original format of the Web page defines a height and width for the Web page, and wherein execution of the instructions performs further operations comprising:

determining an applicable scale factor to display at least one of the width and height of the Web page substantially across a display area of the touch-sensitive display; and

employing the scale factor to render the display area.

124. (New) The mobile device of claim 99, wherein at least a portion of the instructions comprise Java-based instructions configured to be executed on a Java

made via the touch-sensitive display means to enable the user to zoom in and out a display of the Web page.

129. (New) The mobile device of claim 128, wherein the processing means includes a general-purpose processor.

130. (New) The mobile device of claim 128, wherein at least a portion of the programmed circuit means is embodied as a special-purpose processor.

131. (New) The mobile device of claim 128, wherein execution of the instructions performs further operations comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input made via the touch-sensitive display.

132. (New) The mobile device of claim 131, wherein the user interface input enables the user to define a window of a current view of the Web page on which to zoom in on.

133. (New) The mobile device of claim 128, wherein the display of the Web page is re-rendered substantially in real-time to effect zooming operations.

134. (New) The mobile device of claim 128, wherein execution of the instructions performs further operations comprising enabling a user to pan a display of the Web content in response to a corresponding user input made via the touch-sensitive display.

135. (New) The mobile device of claim 134, wherein execution of the instructions performs further operations comprising enabling the display of the Web content to be panned substantially in real-time.

136. (New) The mobile device of claim 128, wherein the Web content includes at least one image, and wherein execution of the instructions performs further operations comprising enabling a user to view an image at a higher resolution than a current

resolution by tapping on the image via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that the image is displayed substantially across at least one of a width and height of a display area of the touch-sensitive display.

137. (New) The mobile device of claim 128, further comprising dynamic memory having at least a portion employed for rendering purposes, wherein execution of the instructions performs further operations comprising:

building a display list via use of the scalable content and rendering display list objects on a virtual display in the dynamic memory; and

scaling display list objects to re-render the display of the Web page.

138. (New) The mobile device of claim 128, wherein the network comprises a mobile service provider network.

139. (New) The wireless device of claim 128, wherein the device comprises a mobile phone.

140. (New) The wireless device of claim 128, wherein the device comprises one of a Personal Digital Assistant (PDA), handheld computer, or handheld device.

141. (New) The wireless device of claim 128, wherein a portion of the scalable content comprises vector-based content.

	instructions are stored that when executed by the processor enable the device to perform operations including, (see claim 25), and The device of claim 25, wherein the device comprises a wireless hand-held device (see claim 43)
f) scalinq the scalable content to render the Web page on the display such that the original width of the Web page is rendered to fit substantially across the display.	scalable vector representation of at least a portion of the Web page that supports a scalable resolution-independent display of the Web content that substantially retains the original page layout and attributes of the Web page when it is rendered on the client device (see claim 15), and rendering the vector-formatted Web content on the client device using a first scaling factor to generate a first display of the Web content on the client device (see claim 15).

In addition, it is obvious that the claimed limitation cites above, (b), (d), and (f) the original width and height of the Web page is rendered to fit substantially across the display is equivalent to and equated to receiving, at the client device, vector-formatted Web content comprising a scalable vector representation of at least a portion of the Web page that supports a scalable resolution-independent display of the Web content that substantially retains *the original page layout and attributes of the Web page* when it is rendered on the client device (see claim 15), and rendering the vector-formatted Web content on the client device using a first scaling factor to generate a first display of the Web content on the client device (see claim 15), to produce a method of Resolution independent vector displaying on Internet content of wireless hand-help devices of copending Application No. 11/045,757.

Allowable Subject Matter

Claims 88-91, 118-121, 125-126,158-159, objected to as being dependent upon a rejected base claims, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims and rewritten to overcome 35 USC 101, and 35 USC 112, and Terminal Disclaimer.

Claims Rejection – 35 U.S.C. 103

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

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

Claims 71-87, 92, 95-117, 122-123, 127-157, and 160-179 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chithambaram et al. US006674445B1 -Provisional No.60/159,069 filed 10/12/1999 [hereinafter "Chithambaram"], in view of Roy et al. US006642925B2 -Continuation of No.08/757,706 filed 10/30/1996 [hereinafter "Roy"].

Independent claim 71, Chithambaram teaches:

A wireless device, a display, a memory, and storage means,

(See Chithambaram Column 3, Lines 65-67, discloses a personal digital assistance (PDA).)

Comprising: wireless communications means, to facilitate wireless communication with a network via which Web content may be accessed; a display; memory; and storage means, in which a plurality of instructions are stored that when executed by the processing means enable the wireless device to perform operations including, rendering a browser interface via which a user is enabled to request access to a Web page,

(See Chithambaram Fig. 1 Column 5, Lines 25-30, discloses a hardware and software environment for the architecture uses a network/Internet 118 to connect technicians utilizing clients such as a thin client 102 (e.g. a PDA, WINCE, or PALM device) or a thick client 104 (e.g., a computer system running a browser) to server computers 106.)

retrieving, via the wireless communication means, and translating at least a portion of the HTML-based Web content from its original format into scalable content that supports a scalable resolution-independent display of the content Web page that substantially retains the original page layout and attributes of the content defined by its original format when rendered; and, scaling the scalable

content to render the Web page on the display such that the original width of the Web page is rendered to fit substantially across the display.

(See Chithambaram Fig. 1 Column 5, Lines 25-30, discloses a hardware and software environment for the architecture uses a network/Internet 118 to connect technicians utilizing clients such as a thin client 102 (e.g. a PDA, WINCE, or PALM device) or a thick client 104 (e.g., a computer system running a browser) to server computers 106.

Also see Chithambaram Column 4 Line 60 → Column 5, Line 10, teaching the indexing raster (i.e. Raster maps provide multiple zoom levels with each zoom level, by scaling existing raster tiles) and vector based for interacting and highlighting with user objects.

Also see Chithambaram Column 6 Lines 55-65, teaching the SVG (Scalable Vector Graphics) allows vector graphic shapes (e.g., paths consisting of straight lines and curves), images, and text. Graphical objects can be grouped, styled, transformed, and composite into previously rendered objects.

Also, see Chithambaram Column 8 Lines 30-65, teaching the offset for location of the object is obtained and the offset is encoded using bounding box to zoom in and filtering out the unwanted object for display on the PDA.)

In addition, Chithambaram does not expressly teach, but Roy teaches:

the Web page including associated HTML-based Web content having an original format defining an original width and height of the Web page and an original page layout and attributes of content on the Web page;

(See Roy Column 10, Lines 1-15, discloses an HTML document using specify the width and height of the map in pixels with the WIDTH=NNN and HEIGHT=NNN parameters. For example: <EMBED SRC="http://www.mapguide.com/map pictures/usa.mwf" WIDTH=300HEIGHT=200>. This entry displays a map of the US in the current document. The map picture is 300x200 pixels in size.)

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified the teaching Chithambaram, to include the Web page including associated HTML-based Web content having an original format defining an original width and height of the Web page and an original page layout and attributes of content on the Web page as taught by Roy to produce a predictable result, as evidence, using Roy's specify the width and height of the map in pixels with the WIDTH=NNN and HEIGHT=NNN parameters with Chithambaram's SVG (Scalable Vector Graphics) and the offset for location of the object is obtained and the offset is encoded using bounding box of to zoom in and filtering out the unwanted object for display on the PDA (see Chithambaram Column 6 Lines 55-65, and also see Chithambaram Column 8 Lines 30-65).

Independent claim 99,

the rejection of claim 71 is fully incorporated, which cites above, and is similarly rejected under the same rationale. In addition, Chithambaram teaches:

employing the scalable content and/or data derived therefrom to, render the Web page on the touch-sensitive display;

(See Chithambaram Column 6 Lines 55-65, teaching the SVG (Scalable Vector Graphics) allows vector graphic shapes (e.g., paths consisting of straight lines and curves), images, and text. Graphical objects can be grouped, styled, transformed, and composite into previously rendered objects.

Also, see Chithambaram Column 8 Lines 30-65, teaching the offset for location of the object is obtained and the offset is encoded using bounding box to zoom in and filtering out the unwanted object for display on the PDA (i.e. It is noted PDA is inherently includes the tough sensitive display as claimed)

and re-render the Web page in response to associated user inputs to enable the user to zoom in and out a display of the Web page;

(See Chithambaram Column 8 Lines 30-65, teaching the offset for location of the object is obtained and the offset is encoded using bounding box to zoom in and filtering out the unwanted object for display on the PDA.)

Independent claim 128,

the rejection of claims 71 and 99 are fully incorporated, which cite above, and are similarly rejected under the same rationale.

Independent claim 143,

the rejection of claims 71 and 99 are fully incorporated, which cite above, and are similarly rejected under the same rationale.

Independent claim 174,

the rejection of claims 71 and 99 are fully incorporated, which cite above, and are similarly rejected under the same rationale. In addition, Chithambaram teaches:

translating at least a portion of the HTML-based Web content from its original format into scalable content that supports a scalable resolution-independent display of the Web page that substantially retains the original page layout and attributes of the content defined by its original format when rendered;

(See Chithambaram Fig. 1 Column 5, Lines 25-30, discloses a hardware and software environment for the architecture uses a network/Internet 118 to connect technicians utilizing clients such as a thin client 102 (e.g. a PDA, WINCE, or PALM device) or a thick client 104 (e.g., a computer system running a browser) to server computers 106.

Also see Chithambaram Column 4 Line 60 → Column 5, Line 10, teaching the indexing raster (i.e. Raster maps provide multiple zoom levels with each zoom level, by scaling existing raster tiles) and vector based for interacting and highlighting with user objects.

Also see Chithambaram Column 6 Lines 55-65, teaching the SVG (Scalable Vector Graphics) allows vector graphic shapes (e.g., paths consisting of straight lines

and curves), images, and text. Graphical objects can be grouped, styled, transformed, and composite into previously rendered objects.

Also, see Chithambaram Column 8 Lines 30-65, teaching the offset for location of the object is obtained and the offset is encoded using bounding box to zoom in and filtering out the unwanted object for display on the PDA.)

and employing the scalable content to render the Web page on the display using a first scale factor; and enabling the Web page to be displayed at a different resolution by scaling the scalable content using a second scale factor to re-render the display;

(See Chithambaram Column 6 Lines 55-65, teaching the SVG (Scalable Vector Graphics) allows vector graphic shapes (e.g., paths consisting of straight lines and curves), images, and text. Graphical objects can be grouped, styled, transformed, and composite into previously rendered objects.

Also, see Chithambaram Column 8 Lines 30-65, teaching the offset for location of the object is obtained and the offset is encoded using bounding box to zoom in and filtering out the unwanted object for display on the PDA.)

Claim 72, Chithambaram teaches:

enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a corresponding user interface input.

(See Chithambaram Column 7 Lines 4-6 and Column 8 Lines 30-65, teaching using bounding box to zoom in and filtering out the unwanted object for display on the PDA.)

Claim 73, Chithambaram teaches:

wherein the display of the Web page is re-rendered to effect zooming operations.

(See Chithambaram Column 7 Lines 4-6 and Column 8 Lines 30-65, teaching using bounding box to zoom in and filtering out the unwanted object for display on the PDA.)

In addition, Chithambaram does not expressly teach, but Roy teaches:

substantially in real-time,

(See Roy Column 3, Lines 40-45, discloses real-time access to dynamic map pictures and associated map data through a Web browser interface suitable for a wide range of users.

See also Roy at Column 2, Lines 20-31, teaching the vector-based data includes Zoom in feature.

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified the teaching Chithambaram, to include the Web page is re-rendered to effect zooming operations in real time as taught by Roy to produce a predictable result, as evidence, using Roy's re-rendered to effect zooming

operations in real time and the offset for location of the object is obtained and the offset is encoded using bounding box of to zoom in and filtering out the unwanted object for display on the PDA (see Chithambaram Column 6 Lines 55-65, and also see Chithambaram Column 8 Lines 30-65).

Claim 74, Chithambaram teaches:

wherein the Web content page includes at least one hyperlink,

(See Chithambaram Column 10 Lines 20-25, teaching attributes stored in records 532-540 may include a URL, an id, and other relevant information.)

and wherein execution of the instructions performs further operations comprising: enabling the user to select the hyperlink; and, in response thereto, retrieving and translating Web content associated with the hyperlink.

(See Chithambaram Column 10 Lines 20-25, teaching attributes stored in records 532-540 may include a URL, an id, and other relevant information.)

to produce additional scalable content; and employing the additional scalable content to render the Web content associated with the hyperlink on the display.

(See Chithambaram Fig. 1 Column 5, Lines 25-30, discloses a hardware and software environment for the architecture uses a network/Internet 118 to connect technicians