Exhibit 6

FILED UNDER SEAL

APPENDIX 3

SUBJECT TO PROTECTIVE ORDER
CONTAINS HIGHLY CONFIDENTIAL – ATTORNEYS' EYES ONLY INFORMATION

'915-1
U.S. Patent No. 7,844,915 is Anticipated by the MERL DiamondTouch System[1]

Materials cited:

1. P.H. Dietz and Leigh, D. *DiamondTouch: a multi-user touch technology*. Proc ACM UIST, (ACM 2001) printed in original at pp. 219-226, reprinted as MERL Technical Report No. TR2003-125. ("*MERL-TR2003-125*")
2. Alan Esenther, Cliff Forlines, Kathy Ryall, Sam Shipman. *DiamondTouch SDK: Support for Multi-User, Multi-Touch Applications*, (MERL 2002), printed as MERL Technical Report No. TR2002-48 ("*MERL-TR2002-48"*)
3. Mark S. Hancock, Frederic D. Vernier, Daniel Wigdor, Sheelagh Carpendale, Chia Shen. Rotation and Translation Mechanisms for Tabletop Interaction, printed as MERL Technical Report No. TR2005-118 ("*MERL-TR2005-118*")
4. Edward Tse, Chia Shen, Saul Greenberg, Clifton Forlines: *Enabling interaction with single user applications through speech and gestures on a multi-user tabletop* (AVI 2006), pp. 336-343, reprinted as MERL Technical Report No. TR2005-130 ("*MERL-TR2005-130*").
5. Alan Esenther and Kent Wittenburg, *Multi-User Multi-Touch Games on DiamondTouch with the DTFlash Toolkit* (MERL 2005), printed as MERL Technical Report No. TR2005-105 ("*MERL-TR2005-105*")
6. MERL hard drive production, including the DiamondTouch system, native libraries, executables, and configuration files for same, as part of the DiamondTouch SDK. ("*MERL-Drive*").
7. Oscar de Bruijn, et al., *An Interactive Coffee Table for Opportunistic Browsing*, found at *MERLDrive*/pdh/papers/chi2003/OB_CoffeeTable.pdf, ("*MERL-CoffeeTable paper*")
8. http://www.youtube.com/watch?v=t35HXAjNW6s ("*MERL video*")
9. http://www.youtube.com/watch?v=JKWe9U5PHmQ ("*MERL-Mandelbrot video*")
10. Edward Tse exhibiting Diamond Touch (2006) - http://video.google.com/videoplay?docid=6420668728353654549 ("*Tse video*")
11. DiamondTouch II 88cm Engineering Prototype ("*DiamondTouch Device"*).
12. MERL Mandelbrot source code, found at *MERL-Drive*/diamondtouch/people/forlines/src/com/merl ("*MERL-MandelbrotSource*").
13. Clifton Forlines deposition, March 8, 2012. ("*Forlines Deposition*")

[1] The DiamondTouch system is comprised of the DiamondTouch physical device, an LCD projector, and a computer running Windows XP, Internet Explorer, and the DiamondTouch applications, including, without limitation, Mandelbrot, tablecloth_27.htm DTFlash web page, DTMouse, and Gesture Speech Interface ("GSI") for Google Earth.

SUBJECT TO PROTECTIVE ORDER
CONTAINS HIGHLY CONFIDENTIAL – ATTORNEYS' EYES ONLY INFORMATION

U.S. Patent No. 7,844,915	**MERL DIAMONDTOUCH SYSTEM**
Claim 1	
[1 preamble] A machine implemented method for scrolling on a touch-sensitive display of a device comprising:	The DiamondTouch system discloses a machine implemented method for scrolling on a touch-sensitive display of a device. *See, e.g.,:* DiamondTouch [1] is a multi-touch input technology that supports multiple, simultaneous users; it can distinguish who is touching where. We present the DiamondTouch SDK; it provides support for the development of applications that utilize DiamondTouch's capabilities to implement computer supported collaboration and rich input modalities (such as gestures). Our first demo illustrates the basic utilities and functionality of our system. (*MERL-TR2002-48* at 2) DiamondTouch uses an array of antennas embedded into a surface, with each antenna transmitting a unique signal. Each user has their own receiver, generally attached to their chair. When a person touches the surface, energy from nearby antennas is coupled through the user to their receiver. Using this mechanism, the system determines who is touching where. (*MERL-TR2002-48* at 3) DiamondTouch is a multi-user touch technology for tabletop front-projected displays. (*MERL-TR2003-125*)

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
	 Figure 1: **Prototypical DiamondTouch setup: front-projection onto a tabletop surface.** (*MERL-TR2002-48* at 2)

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
	Our infrastructure is illustrated in Fig. 1. A standard Windows computer drives our infrastructure software, as described below. The table is a 42" MERL Diamond Touch surface [6] with a 4:3 aspect ratio; a digital projector casts a 1280x1024 pixel image on the table's surface. This table is multi-touch sensitive, where contact is presented through the DiamondTouch SDK as an array of horizontal and vertical signals, touch points and bounding boxes (Fig. 1, row 5). The table is also multi-user, as it distinguishes signals from up to four people. While our technology uses the Diamond Touch, the theoretical motivations, strategies developed, and lessons learnt should apply to other touch/vision based surfaces that offer similar multi user capabilities. (*MERL-TR2005-130* at 2)
[1a] receiving a user input, the user input is one or more input points applied to the touch-sensitive display that is integrated with the device;	The DiamondTouch system discloses receiving a user input, the user input is one or more input points applied to the touch-sensitive display that is integrated with the device. *See, e.g., claim 1 preamble. See also,*

SUBJECT TO PROTECTIVE ORDER
CONTAINS HIGHLY CONFIDENTIAL – ATTORNEYS' EYES ONLY INFORMATION

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
	 Figure 2. The table construction and setup. The DiamondTouch surface is embedded within a tabletop surface. The projector is mounted upon a platform supported by a steel rod that protrudes from one of the corners of the table. Underneath the tabletop is a lockable box containing the computer that drives the projector (in this case a laptop computer). This computer has a wireless Ethernet connection to the Local Area Network. Conductive pads are placed on the seats to allow interaction with the table surface (see [4]). *MERL-CoffeeTable paper* at 2, Fig. 2.

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
[1b] creating an event object in response to the user input;	The DiamondTouch system discloses creating an event object in response to the user input. *See e.g.*, The hardware periodically produces frames of data containing scalar values that measure the proximity of the user's finger(s) to each antenna. The DiamondTouch Library (dtlib) reads these data frames from the DiamondTouch device and affords access to the raw data and to various abstractions and interpretations of that data, such as the location of the maximum proximity (the touch point) and the bounding box of the area touched. Other abstractions are possible and are the subject of ongoing research. A weighted interpolation algorithm increases the effective resolution to 2500 x 1500. Median filtering, hysteresis, and adaptive touch thresholding are used to improve robustness in the face of RF interference and other environ- mental variables. The SDK consists of dtlib (ANSI C), jdt (a Java interface layer), merldt (a Windows application providing mouse emulation, projector calibration, and various diagnostic displays), and a simple multi-user application example. *(MERL-TR2002-48* at 1-2) In the DiamondTouch Mandelbrot application, the creation of the event object in response to the user input is shown, for example, in the *MERL-MandelbrotSource* code. *MERL-MandelbrotSource* Q: For the Fractal Zoom application, what programming language is that written in? A: The original version of the Fractal Zoom application was written in Java. Q: Who wrote the Java version of Fractal Zoom? A: I did. (*Forlines Deposition* at 64) Q: And who wrote the code for Mandelbrot? A: I did. Q: And to the best of your recollection, how many versions of Java Mandelbrot were there? A: Two versions.

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
	Q: Do you recall when the first version was complete? MS. MILLER: Objection. Lacks foundation. BY MR. GLASS: Q: Was the first version ever completed? A: Yes. Q: Do you recall when it was completed? A: We sent the initial version of the Java Q: Mandelbrot application to a company named Idelix in late 2004. (*Forlines Deposition* at 109-110) Q: How does the Fractal Zoom application receive information about a user touch at the touch surface? A: Which version of the Fractal Zoom application are we talking about? Q: Let's talk about the Java version. A: So the -- the Java version of Fractal Zoom implements a listener interface to listen to touch events. And it receives DTLib TFrame objects from another class called DTReader. (*Forlines Deposition* at 69) The *DtlibInputTframe* objects are created in response to user input received from the DiamondTouch table: *MERL-MandelbrotSource*/diamondtouch/DtlibInputTFrame.java (Appendix 3.1) The input loop in the DTReader.run() method calls DtlibDevice.read(), which is a native method that accesses the DiamondTouch device driver through the Java Native Interface ("JNI"). The DtlibDevice.read() method returns an array of DtlibInputTFrame objects. *MERL-MandelbrotSource*/diamondtouch/DTReader.java, lines 62-107 (Appendix 3.2) *MERL-MandelbrotSource*/diamondtouch/DtlibDevice.java, lines 111-119 (Appendix 3.3) The native implementation for the DtlibDevice.read() method is found in the Java_com_merl_diamondtouch_DtlibDevice_read function: *MERL-Drive*/diamondtouch/DiamondTouch/SDK/dtsdk2_1_source/dtsdk2/jdt/DiamondTouchJni.c, lines 806-886 (Appendix 3.4)

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
	After obtaining data from the DiamondTouch device driver, the Java_com_merl_diamondtouch_DtlibDevice_read function calls the make_java_object function to create the DtlibInputTframe event object in response to the user input. *MERL-Drive*/diamondtouch/DiamondTouch/SDK/dtsdk2_1_source/dtsdk2/jdt/DiamondTouchJni.c, lines 415-443 (Appendix 3.4)
[1c] determining whether the event object invokes a scroll or gesture operation by distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation;	The DiamondTouch system discloses determining whether the event object invokes a scroll or gesture operation by distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation. *See, e.g.*, The Mandelbrot program distinguishes between a single touch input point interpreted as the scroll operation and two or more touch input points interpreted as the scale operation: **DiamondTouchTM FractalZoom Demo Controls:** **Key** — **Action** 'q' key — quits the demo 'space' key — resets the zoom level Touch the table with 2 fingers and spread them apart to zoom in. Touch the table with 2 fingers and pull them together to zoom out. Touch the table with 1 finger or grab the image and pull to pan. *MERL-Drive*/diamondtouch/people/forlines/Mandelbrot/images/Instructions.JPG

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
	As an example, this functionality can be seen in the FactalZoomApp.touchDetected and FactalZoomApp.allSegmentsAreFingers methods. The touchDetected method determines whether the event object (the dttf variable, a DtlibInputTframe object) invokes the scroll or gesture operation by distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation, and sets the mCurrentMode variable to record the result of such determination. *MERL-MandelbrotSource*/forlines/fractal/FactalZoomApp.java, lines 99-118 and 120-184 (Appendix 3.5) **Table 1. The Speech/Gesture interface to Google Earth** (*MERL-TR2005-130* at Table 1)

Speech commands		Gesture commands	
Fly to *<place name>*	Navigates to location, eg., Boston, Paris	One finger move / flick	Pans map directly / continuously
Places *<place name>*	Flys to custom-created places, e.g., MERL	One finger double tap	Zoom in 2x at tapped location
Navigation panel	Toggles 3D Navigation controls, e.g., rotate	Two fingers, spread apart	Zoom in
Layer *<type>*	Toggles a layer, e.g., bars, banks	Two fingers, spread together	Zoom out
Undo layer	Removes last layer	Above two actions done rapidly	Continuous zoom out / in until release
Reorient	Returns to the default upright orientation	One hand	3D tilt down
Create a path *<points>*Ok	Creates a path that can be travelled in 3D	Five fingers	3D tilt up
Tour last path	Does a 3D flyover of the previously drawn path	Bookmark	Pin + save current location
Create a region *<points>*	Highlight via semi-transparent region	Last bookmark	Fly to last bookmark
Measure Distance	Measures the shortest distances between two	Next bookmark	Fly to previous bookmark

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
	panning by dragging one's finger or hand across the surface is easily understood by others, as is the surface stretching metaphor used in spreading apart or narrowing two fingers to activate discrete or continuous zooming in Google Earth. (*MERL-TR2005-130* at 6) Apple's infringement contentions do not require that an event object actually invoke a scroll or gesture operation. In my experience, event objects do not invoke operations, as would be understood from the plain and ordinary meaning of this phrase by a person of ordinary skill in the art.
[1d] issuing at least one scroll or gesture call based on invoking the scroll or gesture operation;	The DiamondTouch system discloses issuing at least one scroll or gesture call based on invoking the scroll or gesture operation. For example, the Mandelbrot program issues at least one scroll or gesture call based on invoking the scroll or gesture operation in the FactalZoomApp.touchDetected method. The FactalZoomApp.touchDetected method calls repaint(), an asynchronous request to the Java graphics library to update the display. *See* Zukowski, *Java AWT Reference (O'Reilly 1st edition, 1997)*, page 168 ("The repaint() method requests the scheduler to redraw the component as soon as possible. This will result in update() getting called soon thereafter."); *id*. ("The update() method . . . clears graphics context g by drawing a filled rectangle in the background color, resetting the color to the current foreground color, and calling paint()."). *MERL-MandelbrotSource*/forlines/fractal/FactalZoomApp.java, lines 185 (Appendix 3.5)
[1e] responding to at least one scroll call, if issued, by scrolling a window having a view associated with the event object based on an amount of a scroll with the scroll stopped at a predetermined position in relation to the user input; and	The DiamondTouch system discloses responding to at least one scroll call, if issued, by scrolling a window having a view associated with the event object based on an amount of a scroll with the scroll stopped at a predetermined position in relation to the user input. *See, e.g.,* The DiamondTouch system running Mandelbrot responds to at least one scroll call, if issued, by scrolling the view and updating the display in accordance with the mCurrentMode variable set in step [1c]. When the paint() method is called, Mandelbrot responds to at least one scroll call, if issued, by executing code that redraws the screen according to the scroll values set earlier. The paint method calls Java's Graphic.drawImage method to display the scrolled image. *MERL-MandelbrotSource*/forlines/fractal/FactalZoomApp.java, lines 238-244 and 259-265 (Appendix 3.5)

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
	Panning . . . produces an immediate response, as does zooming or issuing a 'Fly to' command in Google Earth. (*MERL-TR2005-130* at 5) *See also MERL-Mandelbrot video* *Tse video* – single-finger scroll and two-finger zoom seen at 0:17-0:29 under the title "Zoom and Pan"; narrator indicates: We've added the ability to do the panning action with one hand and it's also possible to use the zooming action by spreading two fingers apart or moving to fingers together. As shown in the *Tse video*, a scroll gesture stops the scroll at a predetermined position in relation to the user input (specifically, the position at which the user input ends).

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
	Table 1. The Speech/Gesture interface to Google Earth (*MERL-TR2005-130* at Table 1)
[1f] responding to at least one gesture call, if issued, by scaling the view associated with the event object based on receiving the two or more input points in the form of the user input.	The DiamondTouch system discloses responding to at least one gesture call, if issued, by scaling the view associated with the event object based on receiving the two or more input points in the form of the user input. *See, e.g.,* As disclosed by demonstrations of Mandelbrot, the view is associated with the event object, and is scaled based on the input from two or more touch points. The DiamondTouch System running Mandelbrot responds to at least one gesture call, if issued, by scaling the view and updating the display

Speech commands		Gesture commands	
Fly to <*place name*>	Navigates to location, eg., Boston, Paris	One finger move / flick	Pans map directly / continuously
Places <*place name*>	Flys to custom-created places, e.g., MERL	One finger double tap	Zoom in 2x at tapped location
Navigation panel	Toggles 3D Navigation controls, e.g., rotate	Two fingers, spread apart	Zoom in
Layer <*type*>	Toggles a layer, e.g., bars, banks	Two fingers, spread together	Zoom out
Undo layer	Removes last layer	Above two actions done rapidly	Continuous zoom out / in until release
Reorient	Returns to the default upright orientation	One hand	3D tilt down
Create a path <*points*>Ok	Creates a path that can be travelled in 3D	Five fingers	3D tilt up
Tour last path	Does a 3D flyover of the previously drawn path	Bookmark	Pin + save current location
Create a region <*points*>	Highlight via semi-transparent region	Last bookmark	Fly to last bookmark
Measure Distance	Measures the shortest distances between two	Next bookmark	Fly to previous bookmark

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
	in accordance with the mCurrentMode variable set in step [1c]. When the paint() method is called, Mandelbrot responds to at least one gesture call, if issued, by executing code that redraws the screen according to the scale values set earlier. The paint method calls Java's Graphic.drawImage() method to display the scaled image. *MERL-MandelbrotSource*/forlines/fractal/FactalZoomApp.java, lines 227-237 and 259-265 (Appendix 3.5) . . . It is highly interactive, with compelling real time feedback during panning, zooming and 'flying' actions, as well as the ability to tilt and rotate the scene and view 3D terrain or buildings. . . . Table 1 provides a partial list of how we mapped Google Earth onto our multimodal speech and gesture system, while Fig. 2 illustrates Google Earth running on our multimodal, multi user table. (*MERL-TR2005-130* at 4) Because we reserve gestures for spatial manipulations, very little learning is needed: panning by dragging one's finger or hand across the surface is easily understood by others, as is the surface stretching metaphor used in spreading apart or narrowing two fingers to activate discrete or continuous zooming in Google Earth. (*MERL-TR2005-130* at 6.)

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM

Table 1. The Speech/Gesture interface to Google Earth

Speech commands		Gesture commands	
Fly to *<place name>*	Navigates to location, eg., Boston, Paris	One finger move / flick	Pans map directly / continuously
Places *<place name>*	Flys to custom-created places, e.g., MERL	One finger double tap	Zoom in 2x at tapped location
Navigation panel	Toggles 3D Navigation controls, e.g., rotate	Two fingers, spread apart	Zoom in
Layer *<type>*	Toggles a layer, e.g., bars, banks	Two fingers, spread together	Zoom out
Undo layer	Removes last layer	Above two actions done rapidly	Continuous zoom out / in until release
Reorient	Returns to the default upright orientation	One hand	3D tilt down
Create a path *<points>*Ok	Creates a path that can be travelled in 3D	Five fingers	3D tilt up
Tour last path	Does a 3D flyover of the previously drawn path	Bookmark	Pin + save current location
Create a region *<points>*	Highlight via semi-transparent region	Last bookmark	Fly to last bookmark
Measure Distance	Measures the shortest distances between two	Next bookmark	Fly to previous bookmark

(*MERL-TR2005-130* at Table 1)

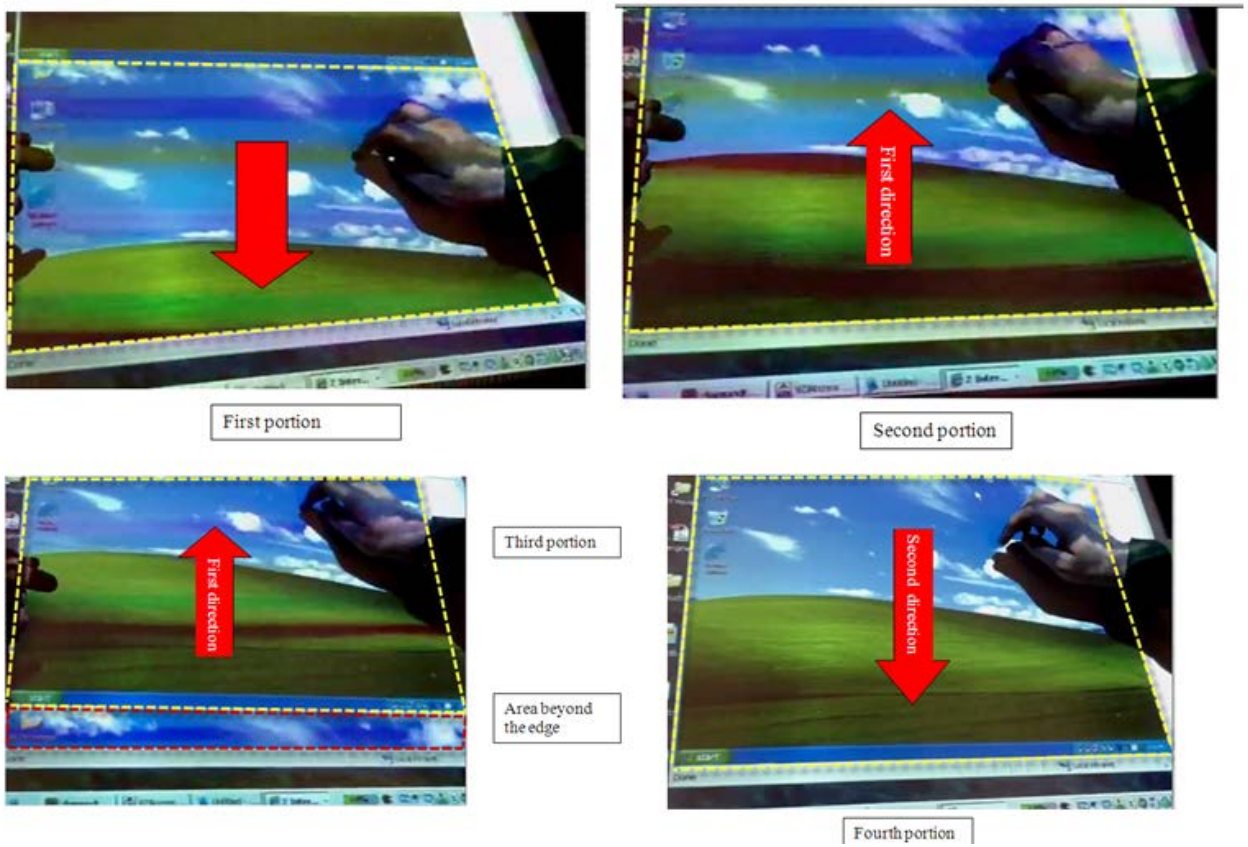
Tse video – single-finger scroll and two-finger zoom seen at 0:17-0:29 under the title "Zoom and Pan"; narrator indicates:

> We've added the ability to do the panning action with one hand and it's also possible to use the zooming action by spreading two fingers apart or moving to fingers together.

As shown in the video, the view of the map is scaled based on receiving two input points in the form of the user input. As the fingers move closer together, the view of the map decreases its size. As the fingers move further apart, the view of the map increases its size.

See also MERL-Mandelbrot video

U.S. Patent No. 7,844,915	**MERL DIAMONDTOUCH SYSTEM**
Claim 2	
[2] The method as in claim 1, further comprising: rubberbanding a scrolling region displayed within the window by a predetermined maximum displacement when the scrolling region exceeds a window edge based on the scroll.	The DiamondTouch system discloses rubberbanding a scrolling region displayed within the window by a predetermined maximum displacement when the scrolling region exceeds a window edge based on the scroll. *See, e.g.*, DTFlash was an application of the MERL DiamondTouch system. Also of note, DTFlash applications can also work as regular web pages, allowing for simple deployment and ushering in a new dimension of multi-user enabled web pages that eliminate the need to take turns with the mouse. Flash is also based on vector graphics and optimized for small downloads, so DTFlash applications have a small memory footprint. But it is the reliance on weak static typing and it's "expressiveness" which make Flash particularly well-suited for exploring drastic changes without breaking existing applications and for facilitating the creation of complex and novel visual interfaces. (*MERL-TR2005-105* at 5-6) Rubberbanding in Claim 2 refers to moving content on a display in a manner that appears elastic, *e.g.*, like a rubber band, so that at the end of a scroll the content slides back making the region outside of the content no longer visible on the display. The DiamondTouch system running the tablecloth_27.htm DTFlash web page implemented a feature that allowed the user to scroll an image inside a window. tablecloth_27 was a single-picture flash-based web page comprising a scrolling region. When the user scrolled up or down past the content edge of the window boundary and then releases the scroll, the image bounces back to its original position. The bounce-back effect, or rubber-banding, simulates a physics-based elastic effect.

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
	Q: Can you describe generally what the TableCloth application was? A: Generally, the TableCloth application was an application that displayed the desktop of an operating system and allowed you to touch and drag that desktop and then release it to allow it to snap back to its original location. (*Forlines Deposition* at 44-45)  *MERL-Drive/*diamondtouch/people/alan/dev/Flash/classes-2005-06-09/tablecloth_27.htm (dated January 11, 2005)

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
Claim 3	
[3] The method as in claim 1, further comprising: attaching scroll indicators to a content edge of the window.	The DiamondTouch system discloses scroll indicators attached to the content edge of the window. *See, e.g., MERL video* at 0:36-0:46 and 0:54-1:04. Our third demo illustrates the mouse emulation capabilities of DiamondTouch. There are two aspects to this demo. First, this capability allows traditional software to be used with DiamondTouch. We currently have several mouse modes (one- touch, two-touch, etc) and are experimenting to determine how best to implement a fully-functioning mouse with DiamondTouch. Our mouse emulator works with traditional software. (*MERL-TR2002-48* at 3) DTMouse was a DiamondTouch program that emulated a mouse via touch input on the DiamondTouch. The MERL DiamondTouch system running DTMouse on Windows XP enabled users to operate programs native to the operating system (e.g., Internet Explorer), comprising scroll indicators attached to a content edge of the window. Q: Are you familiar with what DTMouse refers to in the context of DiamondTouch? A: Yes, I am. Q: What does DTMouse refer to? A: DTMouse is an application that emulates the system's mouse and keyboard via touch input on the DiamondTouch. (*Forlines deposition* at 66-67)
Claim 4	
[4] The method as in claim 1, further comprising: attaching scroll indicators to the window edge.	The DiamondTouch system discloses attaching scroll indicators to the window edge. *See, e.g.,* The MERL DiamondTouch system running DTMouse on Windows XP enabled users to operate programs native to the operating system (e.g., Internet Explorer), comprising scroll indicators attached to the window edge. See also, claim 3.

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
Claim 5	
[5] The method as in claim 1, wherein determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period.	The DiamondTouch system discloses determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period. *See, e.g.,* The DiamondTouch device driver samples data at a set frequency that is dependent on the hardware used. Each frame of touch input consumes a set amount of time. See, for example, the DiamondTouch device driver source code that sets the value of updateperiod, the duration of each frame of DiamondTouch input. A set number of DiamondTouch input frames corresponds to the passage of a set period of time, which is the time period used for receiving a drag user input when determining whether a touch invokes a scroll or a gesture operation. *MERL-Drive*/diamondtouch/DiamondTouch/SDK/dtsdk2_1_source/dtsdk2/dtio/dt_io_win32.c, lines 477-481 and 563-582 (Appendix 3.6) The DiamondTouch system running Mandelbrot determines whether the user input invokes a scroll or gesture operation based on receiving a drag user input for a certain time period, as set by the mForgivingTime value. See, for example, *MERL-MandelbrotSource*/forlines/fractal/FactalZoomApp.java, line 144 (Appendix 3.5) mForgivingTime is set to 200 milliseconds. *MERL-MandelbrotSource*/forlines/fractal/FactalZoomApp.java, line 31 (Appendix 3.5)
Claim 6	
[6] The method as in claim 1, further comprising: responding to at least one gesture call, if issued, by rotating a view associated with the event object based on receiving a plurality of input points in the form of the user input.	The DiamondTouch system discloses responding to at least one gesture call, if issued, by rotating a view associated with the event object based on receiving a plurality of input points in the form of the user input. *See, e.g.,*

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM

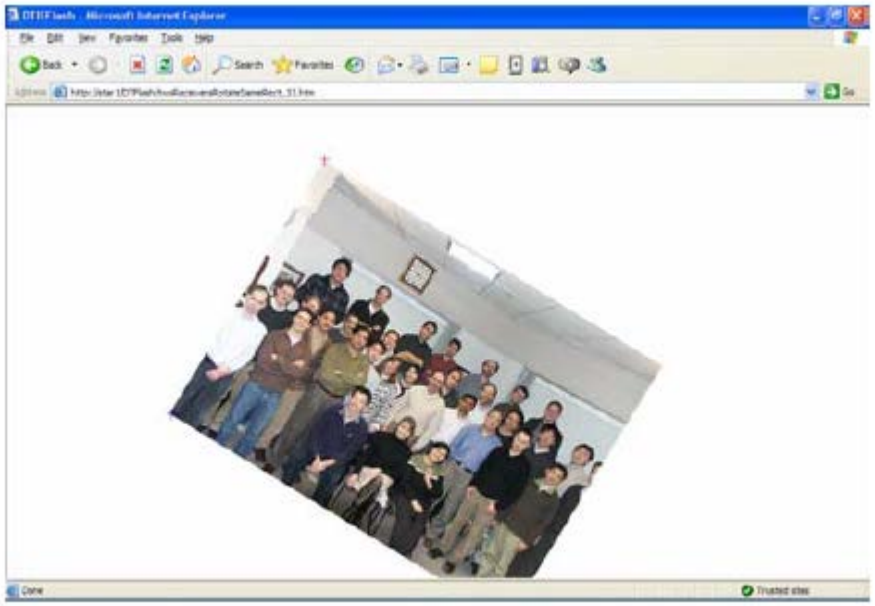
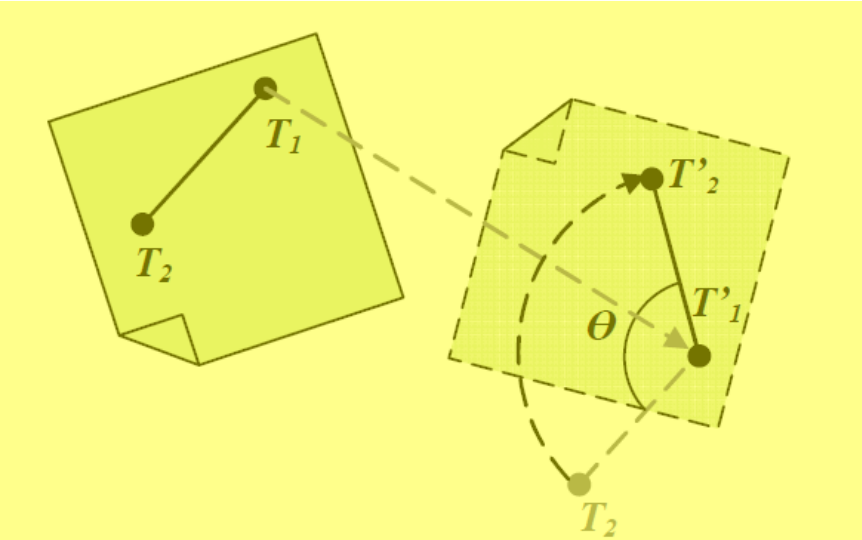


Fig. 4. Collaborative Rotating--players simultaneously interact with the same object.

> The Collaborative Rotating application (Figure 4) lets players simultaneously manipulate the same object. One player's finger determines the location of the upper left corner of the image and the other player's finger determines the location of the lower left corner. The image is rapidly rotated and resized accordingly.

(*MERL-TR2005-105* at 4-5, Fig. 4)

> Google Earth is a free desktop geospatial application that allows one to search, navigate, bookmark, and annotate satellite imagery of the entire planet using a keyboard and mouse. Its database contains detailed satellite imagery with layered geospatial data (e.g., roads, borders, accommodations, etc). It is highly interactive, with compelling real time feedback during panning, zooming and 'flying' actions, *as well as the ability to tilt and rotate the scene and view 3D terrain or buildings*. . . . Table 1 provides a partial list of how we mapped Google Earth onto

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
	our multimodal speech and gesture system, while Fig. 2 illustrates Google Earth running on our multimodal, multi user table." (*MERL-TR2005-130* at 4 (emphasis added))  **Figure 8. In two-point rotation, the first contact point, T_1, is used both as the center of rotation and as a fixed point in translation. The object is rotated based on the angle $\angle T_2 T'_1 T'_2$.** (*MERL-TR2005-118* at 7, Fig. 8) See also, rotation in *Tse video* 1:11-1:16.
Claim 7	
[7] The method as in claim 1, wherein the device is one of: a data processing device, a portable device, a portable data processing	The DiamondTouch system discloses at least a data processing device and a multi touch device. *See, e.g.*, [1 preamble] disclosing a multi touch device and a data processing device.

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
device, a multi touch device, a multi touch portable device, a wireless device, and a cell phone.	
Claim 8	
[8p] A machine readable storage medium storing executable program instructions which when executed cause a data processing system to perform a method comprising:	*See* **[1 preamble]**
[8a] receiving a user input, the user input is one or more input points applied to a touch-sensitive display that is integrated with the data processing system;	*See* **[1a]**
[8b] creating an event object in response to the user input;	*See* **[1b]**
[8c] determining whether the event object invokes a scroll or gesture operation by distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-	*See* **[1c]**

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
sensitive display that are interpreted as the gesture operation;	
[8d] issuing at least one scroll or gesture call based on invoking the scroll or gesture operation;	*See* **[1d]**
[8e] responding to at least one scroll call, if issued, by scrolling a window having a view associated with the event object; and	*See* **[1e]**
[8f] responding to at least one gesture call, if issued, by scaling the view associated with the event object based on receiving the two or more input points in the form of the user input.	*See* **[1f]**
Claim 9	
[9] The medium as in claim 8, further comprising: rubberbanding a scrolling region displayed within the window by a predetermined maximum displacement when the scrolled region exceeds a window edge based on the scroll.	*See* **[2]**
Claim 10	

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
[10] The medium as in claim 8, further comprising: attaching scroll indicators to a content edge of the view.	*See* **[3]**
Claim 11	
[11] The medium as in claim 8, further comprising: attaching scroll indicators to a window edge of the view.	*See* **[4]**
Claim 12	
[12] The medium as in claim 8, wherein determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period.	*See* **[5]**
Claim 13	
[13] The medium as in claim 8, further comprising: responding to at least one gesture call, if issued, by rotating a view associated with the event object based on receiving a plurality of input points in the form of the user input.	*See* **[6]**
Claim 14	

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
[14] The medium as in claim 8, wherein the data processing system is one of: a data processing device, a portable device, a portable data processing device, a multi touch device, a multi touch portable device, a wireless device, and a cell phone.	*See* **[7]**
Claim 15	
[15p] An apparatus, comprising:	*See* **[1 preamble]**
[15a] means for receiving, through a hardware device, a user input on a touch-sensitive display of the apparatus, the user input is one or more input points applied to the touch-sensitive display that is integrated with the apparatus;	*See* **[1a]**
[15b] means for creating an event object in response to the user input;	*See* **[1b]**
[15c] means for determining whether the event object invokes a scroll or gesture operation by distinguishing between a single input point applied	*See* **[1c]**

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation;	
[15d] means for issuing at least one scroll or gesture call based on invoking the scroll or gesture operation;	*See* **[1d]**
[15e] means for responding to at least one scroll call, if issued, by scrolling a window having a view associated with the event object; and	*See* **[1e]**
[15f] means for responding to at least one gesture call, if issued, by scaling the view associated with the event object based on receiving the two or more input points in the form of the user input.	*See* **[1f]**
Claim 16	
[16] The apparatus as in claim 15, further comprising: means for rubberbanding a scrolling region displayed within the	*See* **[2]**

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
window by a predetermined maximum displacement when the scrolling region exceeds a window edge based on the scroll.	
Claim 17	
[17] The apparatus as in claim 15, further comprising: means for attaching scroll indicators to a content edge of the window.	*See* **[3]**
Claim 18	
[18] The apparatus as in claim 15, further comprising: means for attaching scroll indicators to the window edge.	*See* **[4]**
Claim 19	
[19] The apparatus as in claim 15, wherein determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period.	*See* **[5]**
Claim 20	
[20] The apparatus as in claim 15, further	*See* **[6]**

U.S. Patent No. 7,844,915	MERL DIAMONDTOUCH SYSTEM
comprising: means for responding to at least one gesture call, if issued, by rotating a view associated with the event object based on receiving a plurality of input points in the form of the user input.	
Claim 21	
[21] The apparatus as in claim 15, wherein the apparatus is one of: a data processing device, a portable device, a portable data processing device, a multi touch device, a multi touch portable device, a wireless device, and a cell phone.	*See* **[7]**