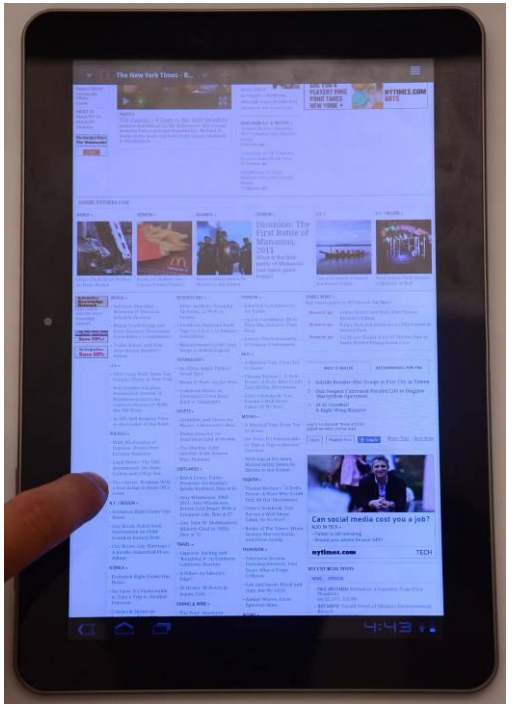
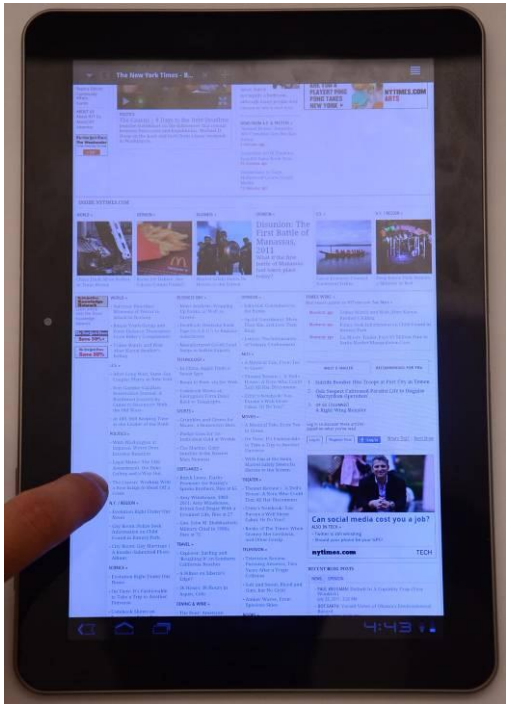
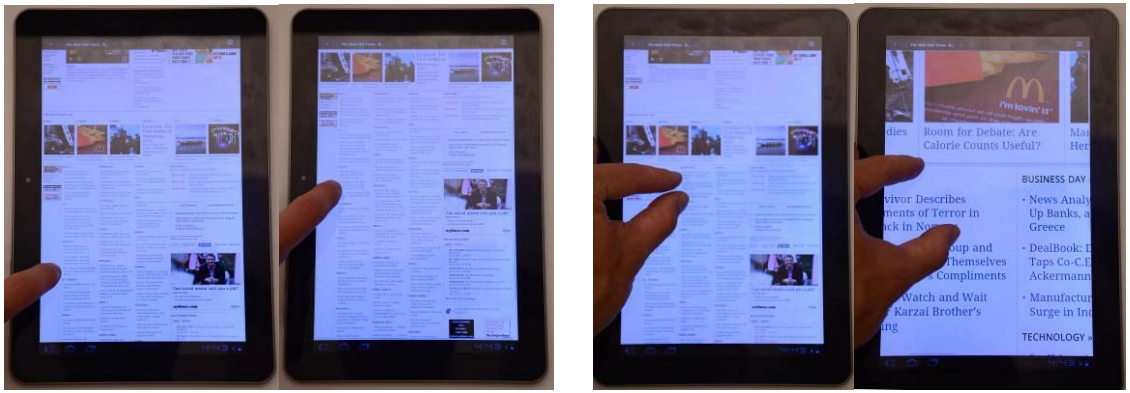
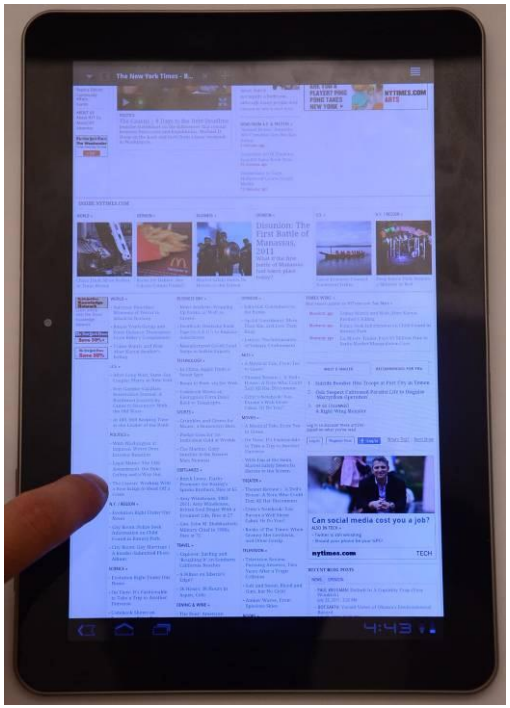
Infringement Claim Chart for U.S. Patent No. 7,844,915 against the Galaxy Tab 10.1 Tablet

U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
Claim 1	
A machine implemented method for scrolling on a touch-sensitive display of a device comprising:	The Samsung device, which includes a touch-sensitive display, performs a machine implemented method for scrolling on the touch-sensitive display.  (Screenshot of the Samsung Galaxy Tab 10.1 scrolling an image.)

U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
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 Samsung device receives a user input. The user input includes one or more input points (one or more fingers) applied to the touch-sensitive display that is integrated with the Samsung device.  (Screenshot of the Samsung Galaxy Tab 10.1 receiving user input.)

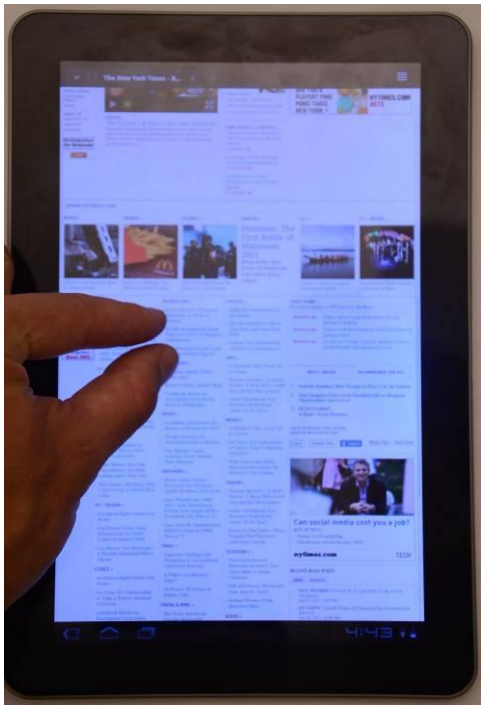
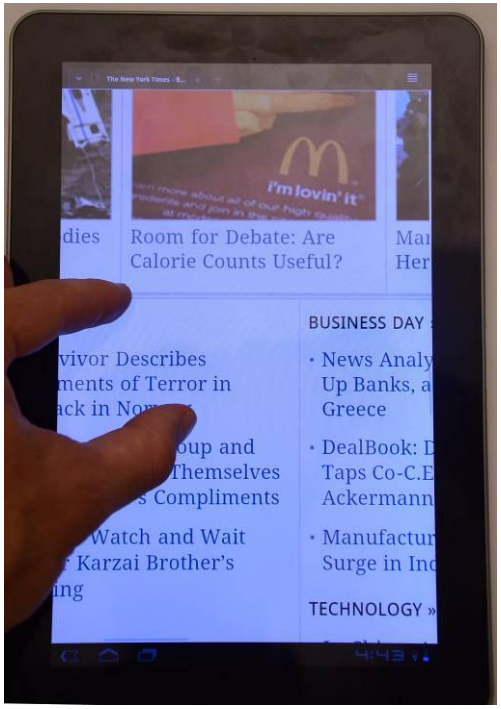
U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
creating an event object in response to the user input; determining whether the event object invokes a scroll or gesture operation by distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation;	The Samsung device, via the Android platform on which the device operates, creates an event object in response to the user input and determines whether the event object invokes a scroll or gesture operation by distinguishing between a single input point (single finger) applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points (two or more fingers) applied to the touch-sensitive display that are interpreted as the gesture operation. As an example, under the Android platform, a MotionEvent object is created in response to a touch on the touchscreen. (Android Developers Site at Class MotionEvent)(Available at http://developer.android.com/reference/android/view/MotionEvent.html.)  (Screenshots of the Samsung Galaxy Tab 10.1 scrolling in response to a single input point applied to the touch-sensitive display and scaling in response to two or more input points applied to the touch-sensitive display.) • **The Samsung Galaxy Tab 10.1 has source code that enables it to "creating an event object in response to the user input."** On the Galaxy Tab 10.1 tablet, user input is processed by the device driver, which passes the input into user space and parses it into an event object referred to as the "MotionEvent" object. This object is an event object created by the method

U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
	InputConsumer::populateMotionEvent(). (*See* frameworks/base/libs/ui/inputTransport.cpp:683-712 [SAMNDCA-C000002822]; *see also* frameworks/base/libs/ui/input.cpp:351-382 [SAMNDCA-C000002830 to -C000002831] (MotionEvent::initialize() method)) • **The Samsung Galaxy Tab 10.1 has source code that enables it to "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."** • On the Galaxy Tab 10.1 tablet, the WebView class's handleQueuedMotionEvent() method interprets the input points associated with the MotionEvent object it processes. The handleQueueMotionEvent() method distinguishes between a single input point (ev.getPointerCount == 1) and two or more input points (ev.getPointerCount > 1). (*See* WebView.java:10281-10314 [SAMDNCA-C000002857].) If one input point is detected, the contact is interpreted as a scroll operation in handleTouchEventCommon(). (*See* WebView.java:10312 [SAMNDCA-C000002857].) If two or more input points are detected, the contact is interpreted as a gesture operation via a call to handleMultiTouchInWebView(). (*See* WebView.java:10302 [SAMNDCA-C000002857]; WebView.java:7887-7944 [SAMNDCA-C000002858].)

U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
issuing at least one scroll or gesture call based on invoking the scroll or gesture operation; 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 Samsung device issues at least one scroll or gesture call based on invoking the scroll or gesture operation. The Samsung device responds 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 (related to the distance the finger is moved) with the scroll stopped at a predetermined position in relation to the user input.   (Screenshot of the Samsung Galaxy Tab 10.1 scrolling an image.) • **The Samsung Galaxy Tab 10.1 has source code that enables it to issue "at least one scroll or gesture call based on invoking the scroll or gesture operation."** • On the Galaxy 10.1 tablet, if one input point is detected,

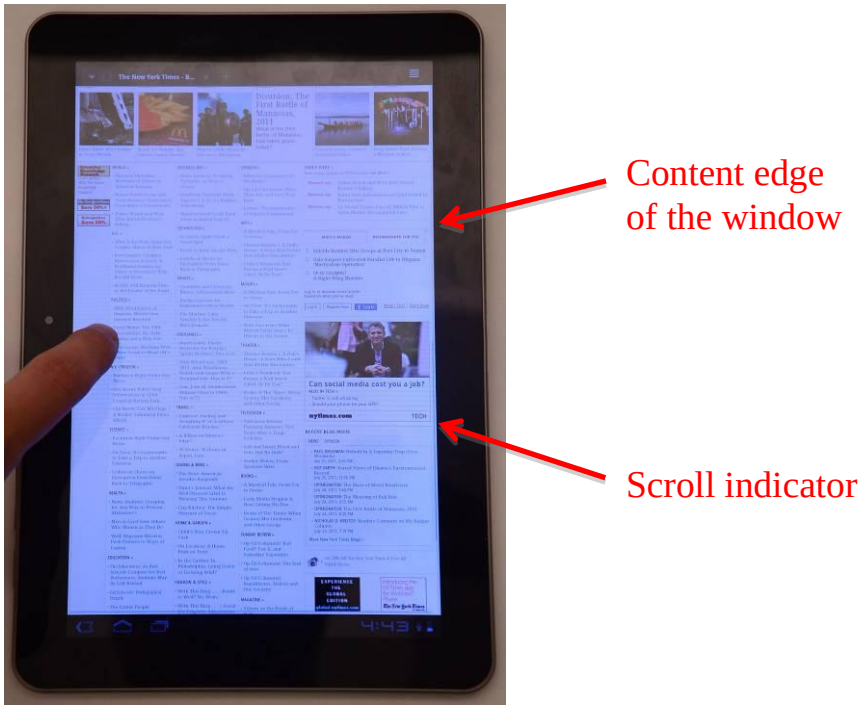
U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
	handleQueuedMotionEvent() will call handleTouchEventCommon() (WebView.java:10312 [SAMNDCA-C000002926]), which issue a scroll call to doDrag() or doFling(). (WebView.java:7617, 7772 [SAMNDCA-C000002926, -C000002930]) If two or more input points are detected, the contact is interpreted as a gesture operation and issue a call to handleMultiTouchInWebView(). (*See* WebView.java:10302 [SAMNDCA-C000002857]; WebView.java:7887-7944 [SAMNDCA-C000002858].) • **The Samsung Galaxy Tab 10.1 has source code that enables it to respond "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"** • On the Galaxy 10.1 tablet, the handleTouchEventCommon() method calls doFling() for a scroll operation. (*See* WebView.java:7272-7821 [SAMNDCA-C000002919 to –C000002931] (call done at 7772).) doFling() then calls the Overscroller.fling() method. (*See* WebView.java:9236-9376 [SAMNDCA-C000002932 to –C000002935].) Overscroller.fling() itself calls two instances of the SplineOverScroller class, each of which is responsible for scrolling in one axis (i.e., one scrolls horizontally and the other scrolls vertically). (*See* OverScroller.java:406-448 [SAMNDCA-C000002945].) The SplineOverScroller class thus maintains state information for the fling. (*See id.*) • The SplineOverScroller class tracks the start points, start time, duration, total distance, and the final position for the scroll at the end of the fling operation. (OverScroller.java:748-782 [SAMNDCA-C000002952 to –C000002953].) The SplineOverScroller.fling() function thus determines the final position of the fling before beginning the fling operation. • The actual rendering of the fling occurs subsequently as part of the drawing cycle. At the end of an event processing cycle, the method computeScroll() is called to compute which part of the view should be rendered to the user. (*See* WebView.java:3568-3654 [SAMNDCA-C000002958 to –C000002959]. The computeScroll() method uses the SplineOverScroller class to extract the state information for the fling. (*See id.*) Afterwards, it calls

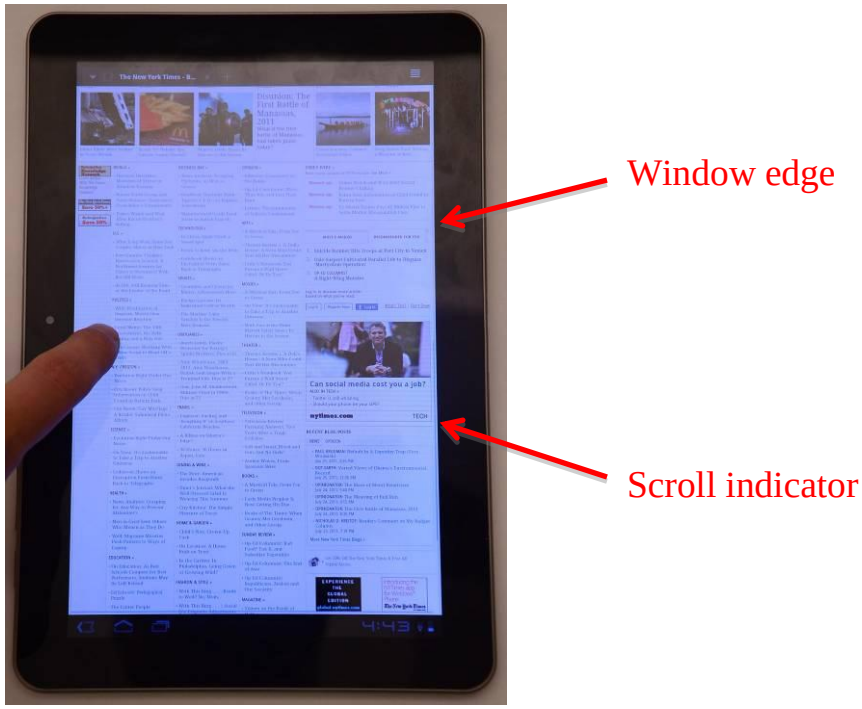
U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
	WebView.overScrollBy() to scroll the content—this method calculates maximums for the distance the user can scroll beyond the edge of the content and whether content should be fixed to a particular axis. (*See id.*; *see also* View.java:11663-11715 [SAMNDCA-C000002960 to –C000002961] (WebView.overScrollBy()).) onOverScrollBy() itself calls onOverScroller() to ensure the intended scroll coordinates are valid and then calls View.scrollTo() to invoke the scroll operation. (*See* View.java:11663-11715 [SAMNDCA-C000002960 to –C000002961]; WebView.java:3130-3162 [SAMDNCA-2962].) View.scrollTo() scrolls the window (setting mScrollX and mScrollY) based on the amount of a scroll with the scroll stopped at a "predetermined position in relation to the user input." (*See* WebView.java:3130-3162 [SAMDNCA-2962].) • Alternatively, the scroll stops at a "predetermined position in relation to the user input" because after the mScrollX and mScrollY fields are set (or determined), the WebView.onDraw() method is subsequently called to translate and draw the view shown to the user. (*See* WebView.java:4261-4418 [SAMNDCA-C000002965 to –C000002968] (with call to trackFPS() at 4416); WebView.java:8757-8791 [SAMNDCA-C000002964] (trackFPS() translates based on mScrollX and mScrollY then draws).)

U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
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 Samsung device responds 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 (two or more fingers) in the form of the user input.   (Screenshot of the Samsung Galaxy Tab 10.1 scaling an image.) • On the Galaxy 10.1 tablet, the handleMultiTouchInWebView() method calls the WebViewScaleGestureDecetor.onTouchEvent() method to perform the scaling (zoom) operation using the MotionEvent object information, which includes the two or more input points touching the screen. (*See* WebViewScaleGestureDetector.java:189 [SAMNDCA-C000002905].) onTouchEvent() calls setContext(), which records information about the

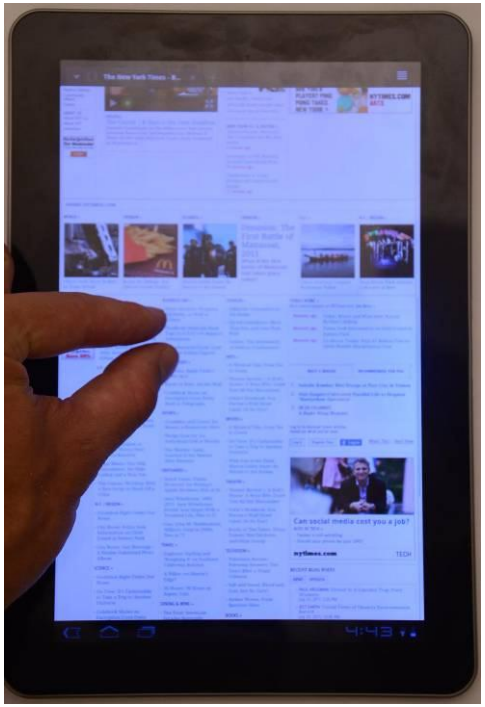
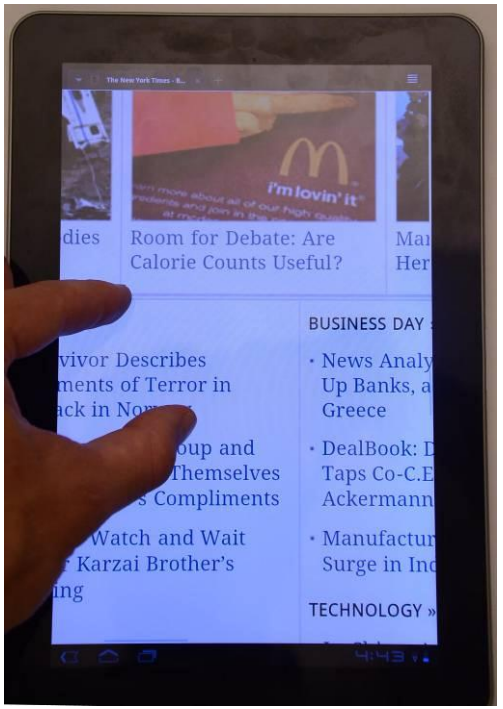
U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
	position of the two input points corresponding, for example, to the user's fingers on the screen (WebviewScaleGestureDetector.java:581-630 [SAMNDCA-C000002524 to -C000002525]). As the user moves his fingers relative to one another—as in, for example, a pinching or de-pinching gesture—the handleScale() method of the ZoomManager class calls the WebviewScaleGestureDetector's getScaleFactor() method to calculate the scale factor based on the ratio of the current distance between the fingers and the previous distance between them (as of the last time the touch screen was polled for input). (ZoomManager.java:1323 [SAMNDCA-C000002410]; WebScaleGestureDetector.java:763-768 [SAMNDCA-C000002528].) handleScale() then calls setZoomScale(), which uses the calculated scale factor to scale the WebView and all of its child views. ZoomManager.java:1372 [SAMNDCA-C000002411]; ZoomManager.java:851-949 [SAMNDCA-C000002399 to -C000002402].)

U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
Claim 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 Samsung device rubberbands a scrolling region displayed within the window by a predetermined maximum displacement when the scrolling region exceeds a window edge based on the scroll.  (Screenshots of the Samsung Galaxy Tab 10.1 rubberbanding an image.) • The predetermined maximum displacement is defined in the Galaxy Tab 10.1 tablet source code to be 1/6 the height and 1/6 the width of the screen for a fling (i.e., a quick, flicking motion of the user's finger on the screen that causes the view to scroll a predetermined distance without further user input). The handleTouchEventCommon() method calls doFling(). (*See* WebView.java:7272-7821 [SAMNDCA-C000002919 to -C000002931] (call done at 7772).) In the doFling() method, if the isElasticEffectEnabled() method

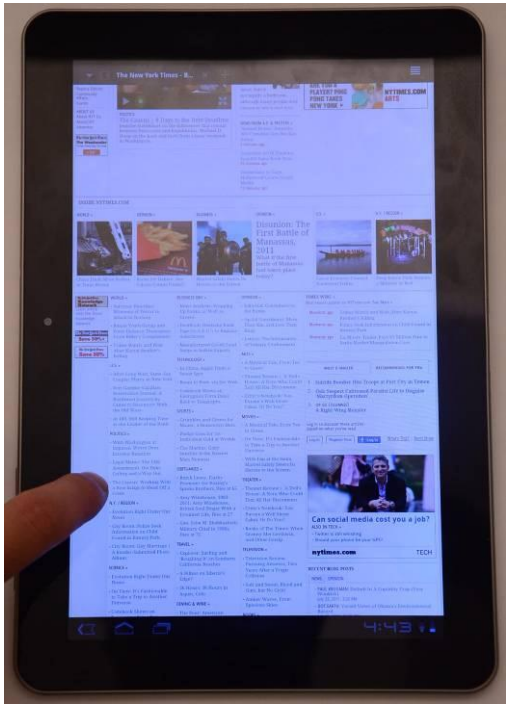
U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
	returns a true value (i.e., if the device is configured to "rubberband") the variables "overX" and "overY" are set to 1/6 the screen width and 1/6 the screen height, respectively. (*See* WebView.java:9236-9376 [SAMNDCA-C000002932-2935] (particularly lines 9350-9361).) The overX and overY variables are then passed to the Overscroller.fling() method, and they set the maximum amount for rubberbanding displacement. (*See id.*)
Claim 3	
The method as in claim **1**, further comprising: attaching scroll indicators to a content edge of the window.	The Samsung device attaches scroll indicators to the window edge.  (Screenshot of the Samsung Galaxy Tab 10.1 attaching a scroll indicator to a content edge of the window.)

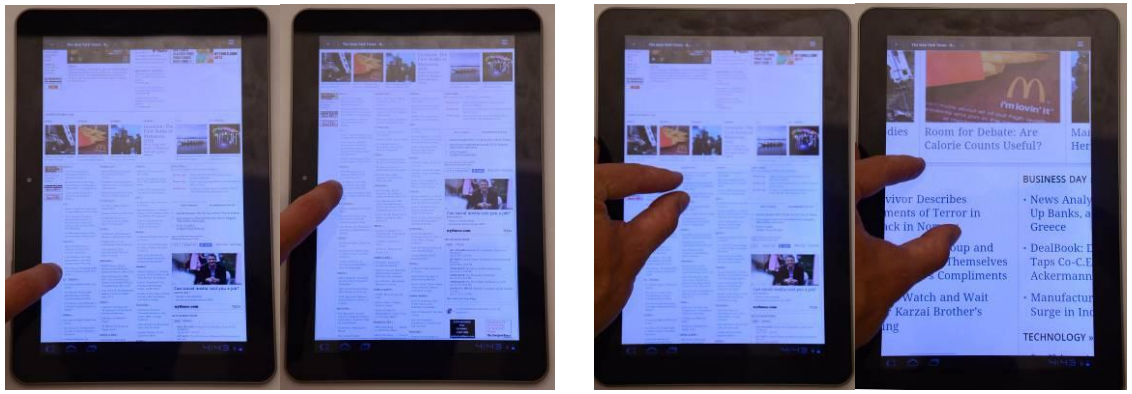
U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
Claim 4	
The method as in claim **1**, further comprising: attaching scroll indicators to the window edge.	The Samsung device attaches scroll indicators to the window edge.  (Screenshot of the Samsung Galaxy Tab 10.1 attaching a scroll indicator to the window edge.)

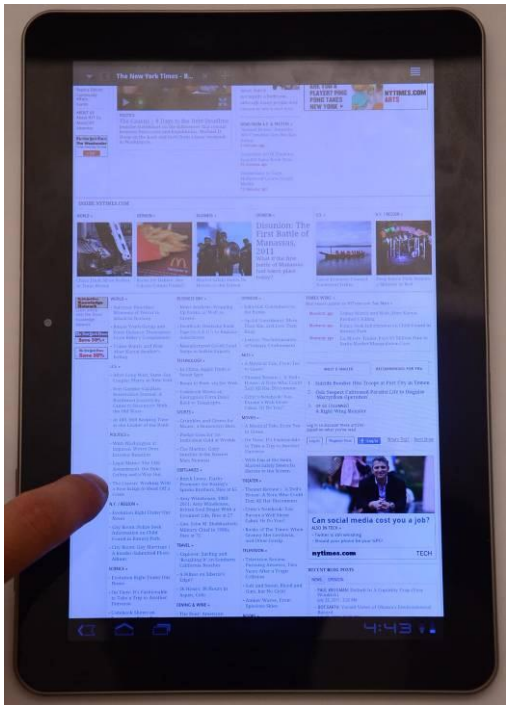
U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
Claim 5	
The method as in claim **1**, wherein determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period.	The Samsung device determines whether the event object invokes a scroll or gesture operation based on receiving a drag user input for a certain time period. • The Galaxy Tab 10.1 tablet determines whether the event object invokes the scroll operation based on receiving a drag user input for a certain time period. The handleTouchEventCommon() invokes the fling operation based on the user scrolling within a certain period of time. (*See* WebView.java:7758 [SAMDNCA00002919 to –C000002931].)
Claim 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 Samsung device responds 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 (plurality of fingers) in the form of the user input.   (Screenshot of the Samsung Galaxy Tab 10.1 rotating an image.)

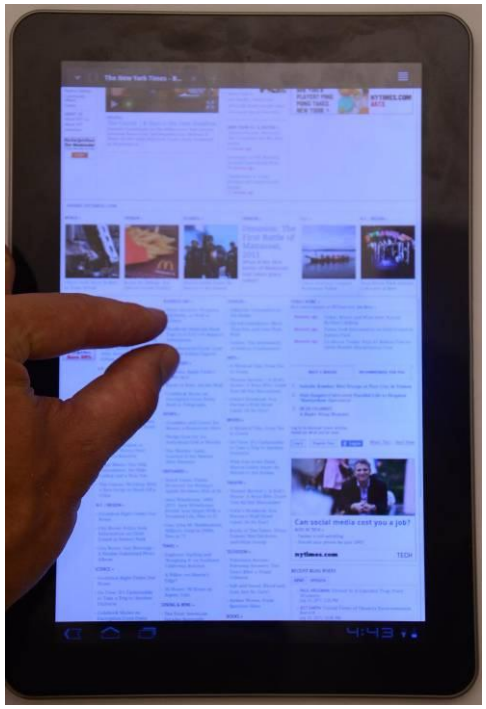
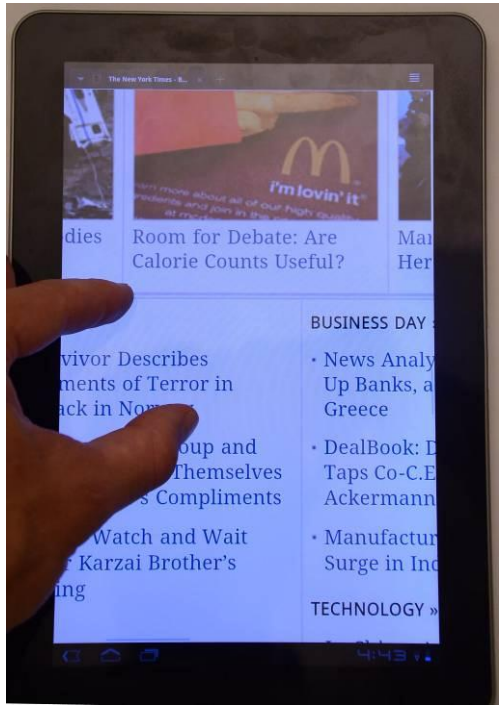
U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
Claim 7	
The method as in claim **1**, wherein the device is one of: a data processing device, a portable device, a portable data processing device, a multi touch device, a multi touch portable device, a wireless device, and a cell phone.	The Samsung device is a multi touch portable device.   (Screenshot of the Samsung Galaxy Tab 10.1 receiving multiple input points.)

U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
Claim 8	
A machine readable storage medium storing executable program instructions which when executed cause a data processing system to perform a method comprising:	The Samsung device includes a computer readable storage medium storing executable program instructions. The executable program instructions, when executed, cause the Samsung device to perform a method. 

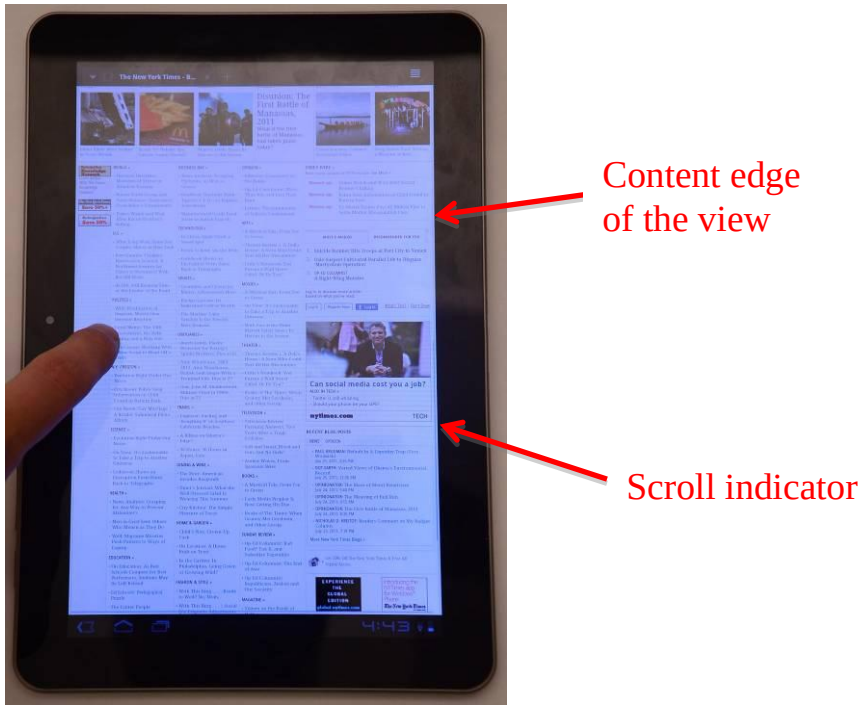
U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
receiving a user input, the user input is one or more input points applied to a touch-sensitive display that is integrated with the data processing system;	The instructions, when executed, cause the Samsung device to receive a user input. The user input includes one or more input points (one or more fingers) applied to the touch-sensitive display that is integrated with the Samsung device.  (Screenshot of the Samsung Galaxy Tab 10.1 receiving user input.)

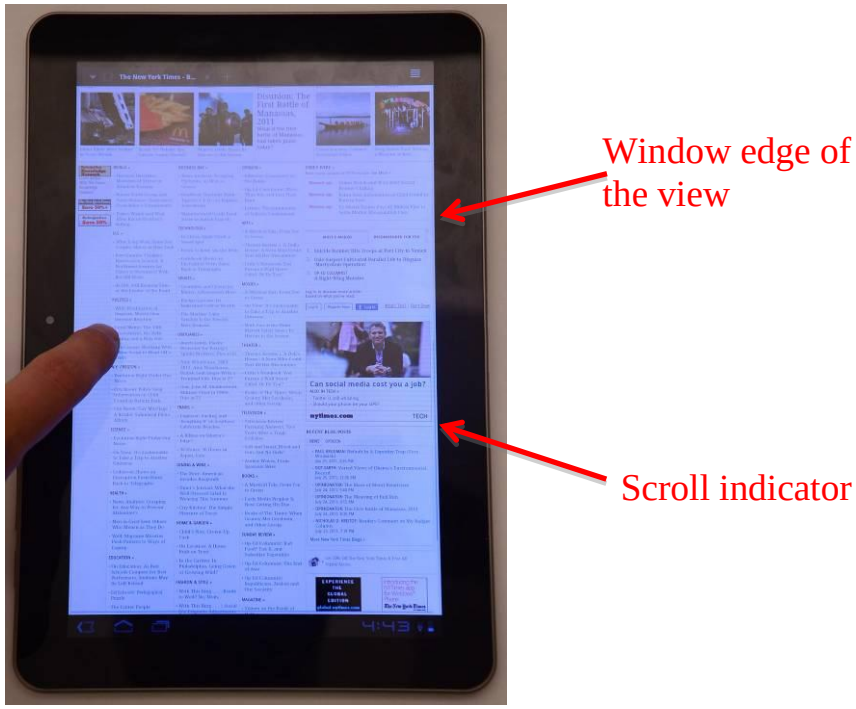
U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
creating an event object in response to the user input; determining whether the event object invokes a scroll or gesture operation by distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation;	The instructions, when executed, cause the Samsung device, via the Android platform on which the device operates, to create an event object in response to the user input. The instructions, when executed, also cause the Samsung device to determine whether the event object invokes a scroll or gesture operation by distinguishing between a single input point (single finger) applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points (two or more fingers) applied to the touch-sensitive display that are interpreted as the gesture operation. As an example, under the Android platform, a MotionEvent object is created in response to a touch on the touchscreen. (Android Developers Site at Class MotionEvent) (Available at http://developer.android.com/reference/android/view/MotionEvent.html.)  (Screenshots of the Samsung Galaxy Tab 10.1 scrolling in response to a single input point applied to the touch-sensitive display and scaling in response to two or more input points applied to the touch-sensitive display.)

U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
issuing at least one scroll or gesture call based on invoking the scroll or gesture operation; responding to at least one scroll call, if issued, by scrolling a window having a view associated with the event object; and	The instructions, when executed, cause the Samsung device to issue at least one scroll or gesture call based on invoking the scroll or gesture operation. The instructions, when executed, also cause the Samsung device to respond to at least one scroll call, if issued, by scrolling a window having a view associated with the event object.   (Screenshot of the Samsung Galaxy Tab 10.1 scrolling an image.)

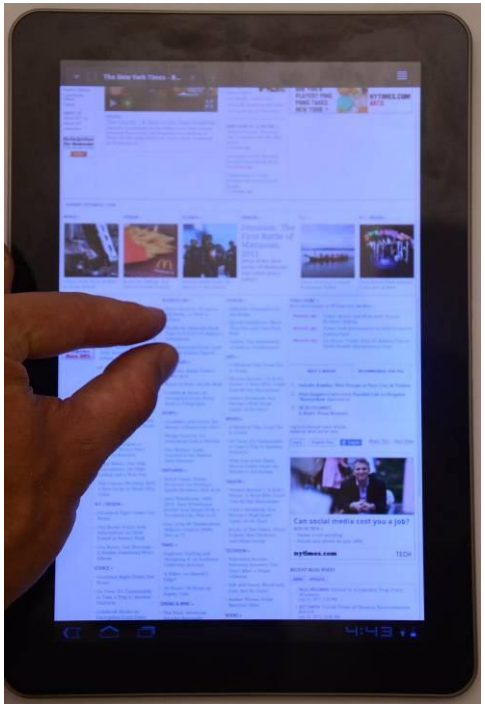
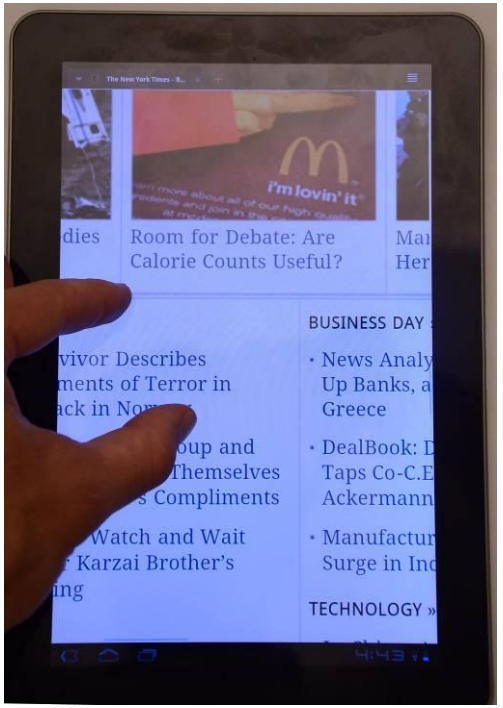
U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
responding to at least one gesture call, if issued, by scaling the view associated with the event object based on receiving the two or more input points in the form of the user input.	The instructions, when executed, cause the Samsung device to at least one gesture call, if issued, by scaling the view associated with the event object based on receiving the two or more input points (two or more fingers) in the form of the user input.   (Screenshot of the Samsung Galaxy Tab 10.1 scaling an image.)

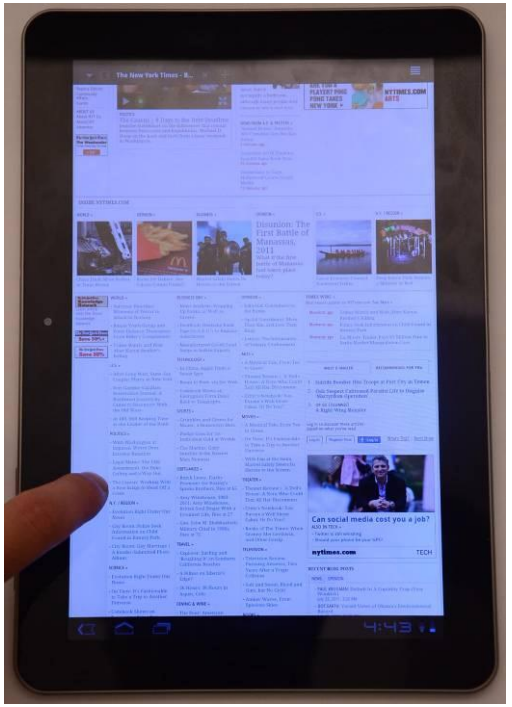
U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
Claim 9	
The medium as in claim **8**, further comprising: rubberbanding a scrolling region displayed within the window by a predetermined maximum displacement when the scrolled region exceeds a window edge based on the scroll.	The instructions, when executed, cause the Samsung device to rubberband a scrolling region displayed within the window by a predetermined maximum displacement when the scrolled region exceeds a window edge based on the scroll.  (Screenshots of the Samsung Galaxy Tab 10.1 rubberbanding an image.)

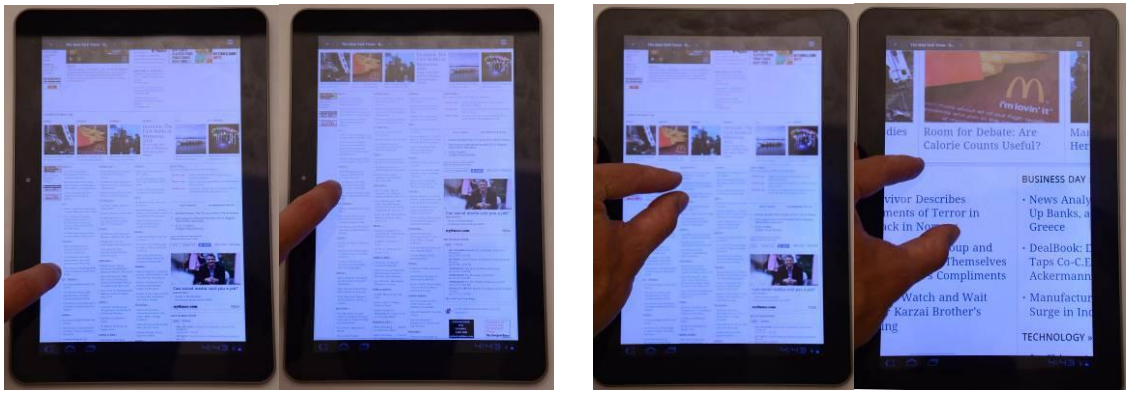
U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
Claim 10	
The medium as in claim **8**, further comprising: attaching scroll indicators to a content edge of the view.	The instructions, when executed, cause the Samsung device to attach scroll indicators to a content edge of the view.  (Screenshot of the Samsung Galaxy Tab 10.1 attaching a scroll indicator to a content edge of the view.)

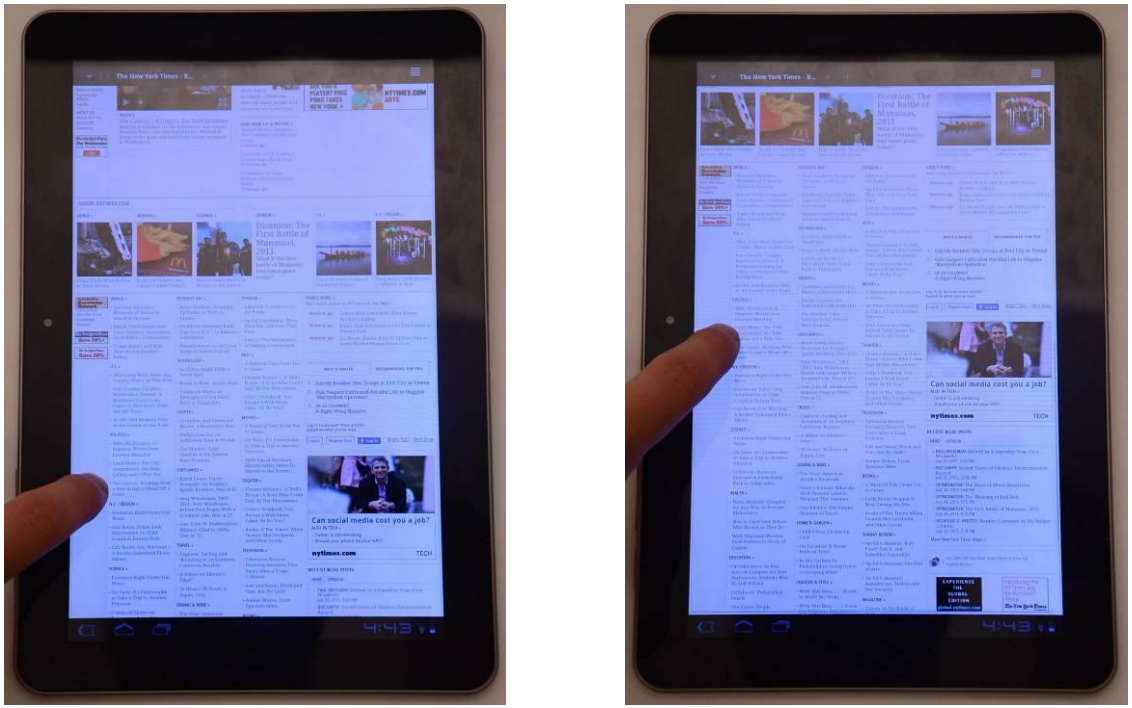
U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
Claim 11	
The medium as in claim **8**, further comprising: attaching scroll indicators to a window edge of the view.	The instructions, when executed, cause the Samsung device to attach scroll indicators to a window edge of the view.  (Screenshot of the Samsung Galaxy Tab 10.1 attaching a scroll indicator to a window edge of the view.)

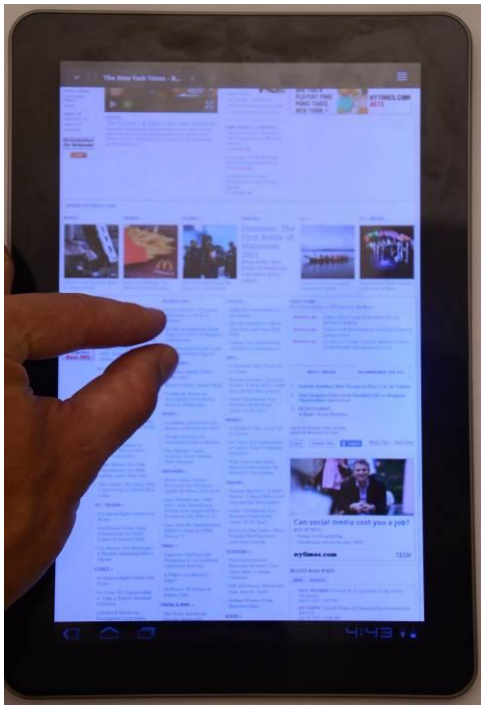
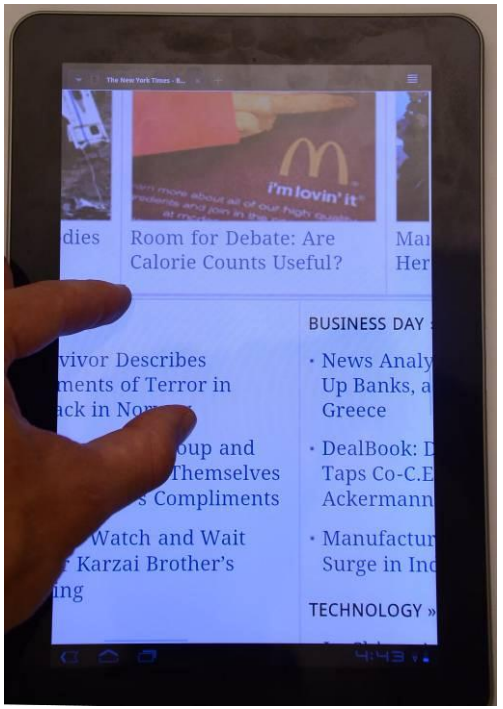
U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
Claim 12	
The medium as in claim **8**, wherein determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period.	The instructions, when executed, cause the Samsung device to determine whether the event object invokes a scroll or gesture operation based on receiving a drag user input for a certain time period.
Claim 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.	The Samsung device responds 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 (plurality of fingers) in the form of the user input.   (Screenshot of the Samsung Galaxy Tab 10.1 rotating an image.)

U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
Claim 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.	The Samsung device is a multi touch portable device.   (Screenshot of the Samsung Galaxy Tab 10.1 receiving multiple input points.)

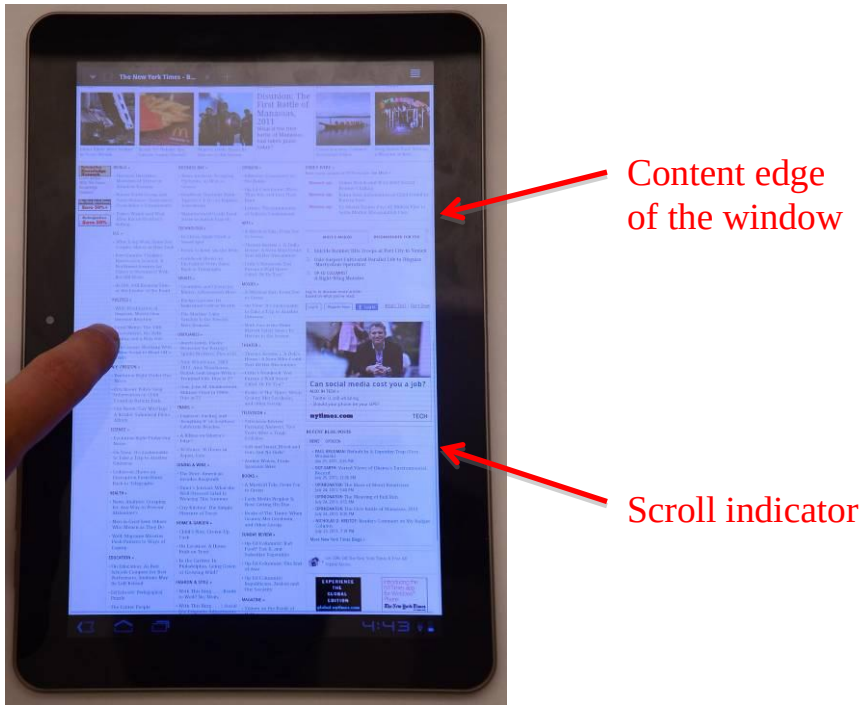
U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
Claim 15	
An apparatus, comprising: means for receiving, through a hardware device, a user input on a touch-sensitive display of the apparatus, the user input is one or more input points applied to the touch-sensitive display that is integrated with the apparatus;	The Samsung device includes a processor executing computer instructions 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 (one or more fingers) applied to the touch-sensitive display that is integrated with the Samsung device.  (Screenshot of the Samsung Galaxy Tab 10.1 receiving user input.)

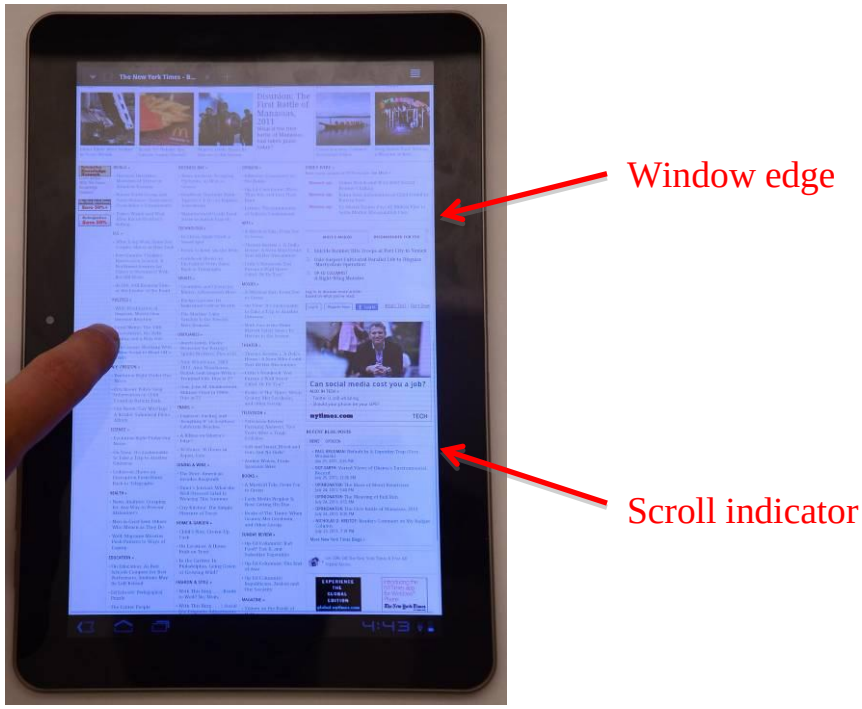
U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
means for creating an event object in response to the user input; means for determining whether the event object invokes a scroll or gesture operation by distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation;	The Samsung device, via the Android platform on which the device operates, includes a processor executing computer instructions for creating an event object in response to the user input and determining whether the event object invokes a scroll or gesture operation by distinguishing between a single input point (single finger) applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points (two or more fingers) applied to the touch-sensitive display that are interpreted as the gesture operation. As an example, under the Android platform, a MotionEvent object is created in response to a touch on the touchscreen. (Android Developers Site at Class MotionEvent) (Available at http://developer.android.com/reference/android/view/MotionEvent.html.)  (Screenshots of the Samsung Galaxy Tab 10.1 scrolling in response to a single input point applied to the touch-sensitive display and scaling in response to two or more input points applied to the touch-sensitive display.)

U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
means for issuing at least one scroll or gesture call based on invoking the scroll or gesture operation; means for responding to at least one scroll call, if issued, by scrolling a window having a view associated with the event object; and	The Samsung device includes a processor executing computer instructions for issuing at least one scroll or gesture call based on invoking the scroll or gesture operation. The processor also executing computer instructions for responding to at least one scroll call, if issued, by scrolling a window having a view associated with the event object.  (Screenshot of the Samsung Galaxy Tab 10.1 scrolling an image.)

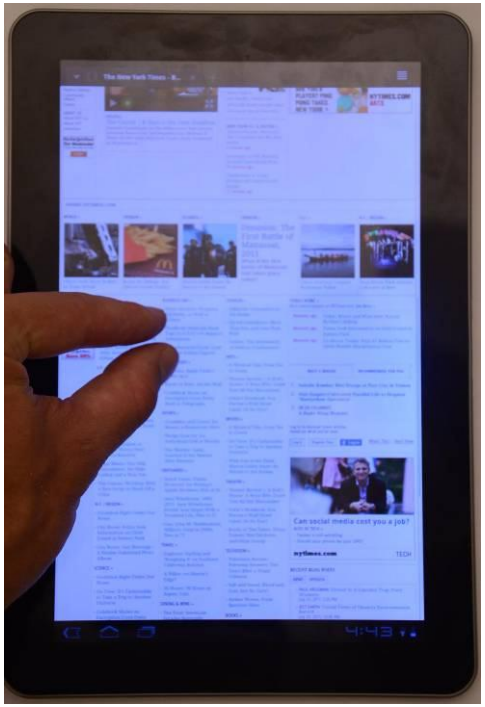
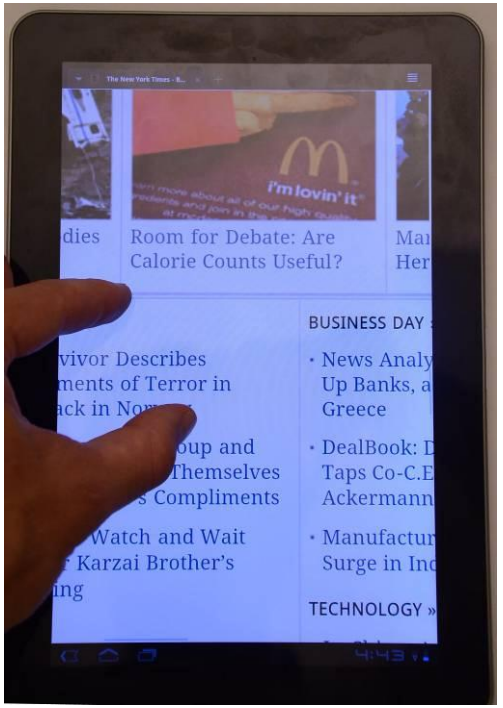
U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
means for responding to at least one gesture call, if issued, by scaling the view associated with the event object based on receiving the two or more input points in the form of the user input.	The Samsung device includes a processor executing computer instructions 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 (two or more fingers) in the form of the user input.   (Screenshot of the Samsung Galaxy Tab 10.1 scaling an image.)

U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
Claim 16	
The apparatus as in claim **15**, further comprising: means for 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 Samsung device includes a processor executing computer instructions for 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.  (Screenshots of the Samsung Galaxy Tab 10.1 rubberbanding an image.)

U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
Claim 17	
The apparatus as in claim **15**, further comprising: means for attaching scroll indicators to a content edge of the window.	The Samsung device includes a processor executing computer instructions for attaching scroll indicators to a content edge of the window.  (Screenshot of the Samsung Galaxy Tab 10.1 attaching a scroll indicator to a content edge of the window.)

U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
Claim 18	
The apparatus as in claim **15**, further comprising: means for attaching scroll indicators to the window edge.	The Samsung device includes a processor executing computer instructions for attaching scroll indicators to the window edge.  (Screenshot of the Samsung Galaxy Tab 10.1 attaching a scroll indicator to the window edge.)
Claim 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.	The Samsung device includes a processor executing computer instructions for determining whether the event object invokes a scroll or gesture operation based on receiving a drag user input for a certain time period.

U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
Claim 20	
The apparatus as in claim **15**, further 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.	The Samsung device responds 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 (plurality of fingers) in the form of the user input.   (Screenshot of the Samsung Galaxy Tab 10.1 rotating an image.)

U.S. Patent No. 7,844,915	Samsung Galaxy Tab 10.1
Claim 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.	The Samsung device is a multi touch portable device.   (Screenshot of the Samsung Galaxy Tab 10.1 receiving multiple input points.)