A Literature Review of Leveraging the Capabilities of a Multi-touch Device to promote Multi-user Collaboration

Frank J. Garofalo

CGT 513 - Interactive Multimedia Research and Development

Dr. Harris

May 4, 2009

1.1. Introduction

The interest surrounding multi-touch devices and computer applications for such devices increased in 2006 with a presentation at the TED Conference by Jeff Han, a consulting research scientist at New York University Courant Institute of Mathematical Sciences (TED 2006 Conference, 2006). Since then many have conducted research on the various techniques to construct a multi-touch device. Others have conducted studies of the gesture actions users make while interacting with multi-touch devices. The ability for multiple users to interact with a computer simultaneously, such as with a multi-touch device, is referred to as “multi-model” (Natural User Interface Group, 2008a). Several applications have been developed to showcase the capabilities of this “multi-model” form of input to a computer. Research of utilizing a multi-touch device for a group/team collaboration task will also be reviewed.

1.2. Literature Review

1.2.1. Multi-touch Screen Devices

Jeff Han, a consulting research scientist at New York University Courant Institute of Mathematical Sciences, is recognized as one of the leading content experts in the industry regarding multi-touch computing. The multi-touch device created by Han in 2005 takes touch sensing technology which can recognize one point of contact to the next level to recognize multiple points of contact simultaneously. These points of contact can include more than one finger at a time or even an object placed on the device. Han (2006) describes this new model of touch sensing technology as “touch is a very natural and intuitive way for people to interact.” This now presents a new means of interactivity between humans and computers, which Han (2006) describes as “enabling the user to finally interact with both hands at once, as well as employ more complex chording gestures, promising great improvements in usability, intuitiveness, and efficiency."

Now not only can one user be able to use both hands simultaneously and have an application respond accordingly, but more than one user can interact with the same application at the same time. Developers can now create applications with richer experiences for users, which can create an "inviting environment for multiple attendees to be able to walk up to and interact with the display" (Han, 2006). He stated that this new means of interactivity fits naturally with applications for music, games, and entertainment where such applications will be able to "seamlessly accommodate multiple users either collaboratively or competitively" (Han, 2006).

In addition to Han's research, several others in both industry and academia have taken interest in multi-touch devices. An on-line community to promote the collaboration of research related to interactive media and multi-touch technologies was founded in 2006 as the Natural User Interface Group (NUI Group). Today the on-line community has several open-source projects currently being developed with an emphasis on "machine sensing techniques" (Natural User Interface Group, n.d., "About NUI Group"). The goal of the NUI Group is to discover methods to utilize multi-touch applications for the areas of business, education and the arts. By focusing on discovering the most effective, budget-friendly solutions, the NUI Group's community projects are open-source. Two of the NUI Group projects which are used in this study are: T-Beta and Touchlib (Natural User Interface Group, n.d., "About NUI Group"). These two projects will be further discussed.

From the inspiration of Jeff Han's success in addition to the available resources provided by the NUI Group the number of individuals experimenting with multi-touch technologies continues to grow, through this influx of interest several configurations have been developed. There are five primary multi-touch technologies: Frustrated Total Internal Reflection Method (FTIR), Diffused Illumination (DI), Laser Light Plane (LLP), Diffused Surface Illumination (DSI), and Capacitance Testing. The FTIR uses light generated by infrared light emitting diodes (LEDs) through a medium, such as acrylic also referred to as Plexiglas (Han, 2005; Saffer, 2008, pg15). This causes the light to be trapped inside the acrylic by internal reflection. Han (2005) recommends using "high-power infrared LEDs" along all the edges of the medium. Once a finger from a user touches the acrylic surface the light is "frustrated" which results in the light redirected downwards to be sensed by an infrared camera. This technique is currently the most popular, which can be attributed to the devices built by Jeff Han. The FTIR method, as described by Han (2005), is "zero-force and true" meaning that it doesn't rely on pressure, although pressure can affect the accuracy of the contact area of the object. "True" refers to the ability to distinguish the difference between an object hovering over the surface versus an object making contact with the surface (Han, 2005; Natural User Interface Group, 2009a).

Figure 1: Screen capture of a hand from an infrared camera
using the FTIR method. (Natural User Interface Group, 2008b)

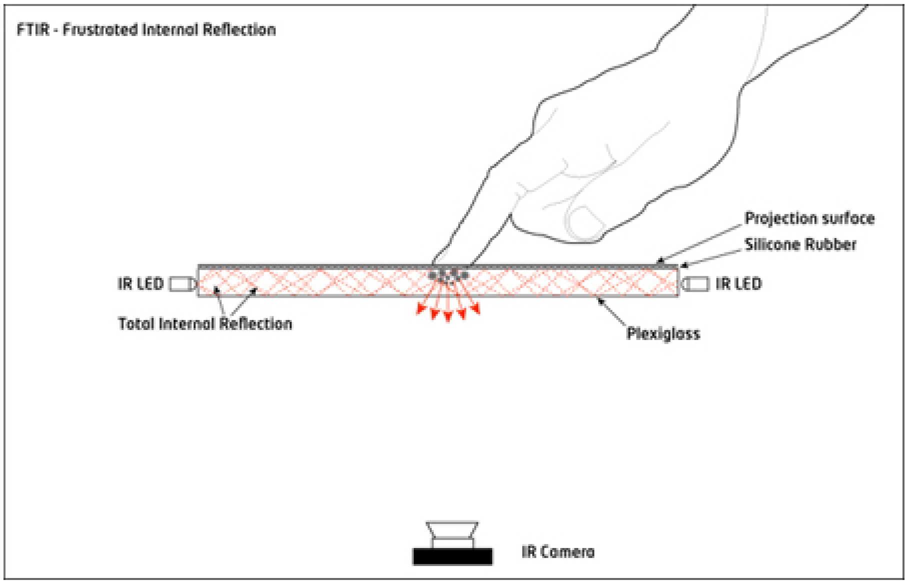

Figure 2: A diagram of the FTIR configuration.
(Natural User Interface Group, 2008b)

As shown in the figure above, there is a silicone rubber and a projection surface needed above the Plexiglas which is referred to as a “compliant surface.” This serves the purpose of enabling the sensor to respond to force rather than just contact (Fantini, 2008; Han, 2005).

Diffused Illumination (DI) is a process in which infrared light is mounted either above or below the surface of the medium, in which either glass or acrylic can be used. The light is directed at the surface of the medium, when an object makes contact with the surface light is reflected which is then detected by the camera. Variations of this method can also sense objects placed on the screen and objects hovering over the screen, depending on the diffuser used (Natural User Interface Group, 2008b, 2009a; Saffer, 2008). The ability to detect objects making contact with the surface of the device provides several potential benefits over the FTIR configuration.

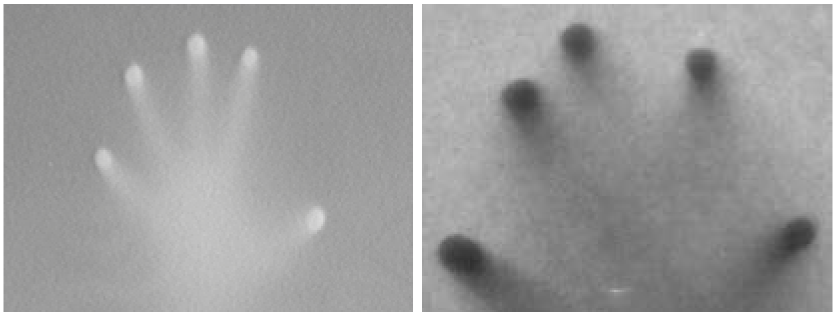

Figure 3: DI Rear Projection, left, and DI Front Project, right.
(Natural User Interface Group, 2008b)

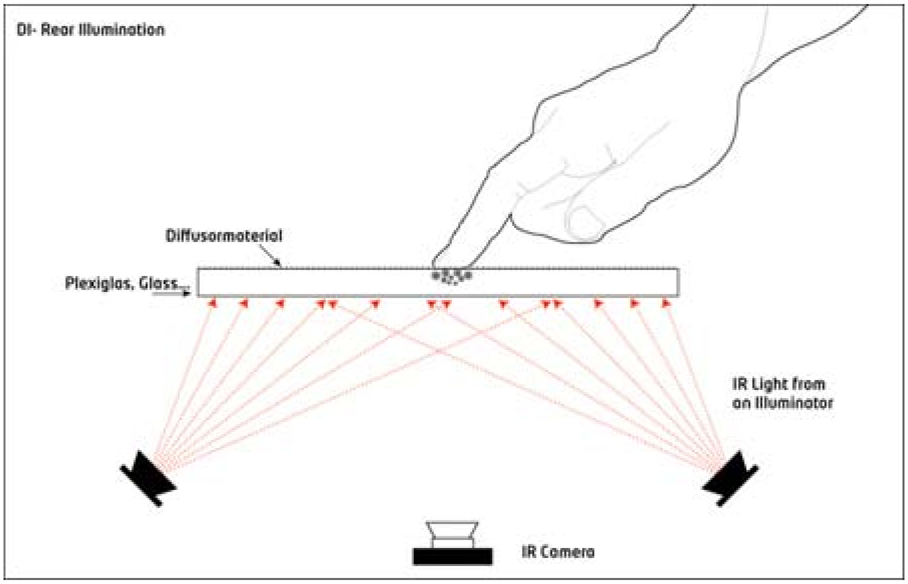

Figure 4: A diagram of the DI Rear Projection configuration.
(Natural User Interface Group, 2008b)

The Microsoft (MS) Surface multi-touch device uses the DI Rear Projection method. As illustrated in a *Popular Mechanics* article by Glenn Derene (2007), the diagram below shows the top of the device with a diffuser which serves as a screen (highlighted as number 1 in the diagram) with the multi-touch capabilities. Since the MS Surface is using DI Rear Projection it can recognize objects placed on the screen of the device based upon their shapes or by recognizing coded tags called “domino” (Derene, 2007). Also labeled in the diagram, number 2 is what Derene (2007) refers to as the Surface’s “machine vision” which is an infrared beam of 850 nanometer-wavelength LED lights which are directed at the screen. Number 3 is the devices computer central processing unit, commonly referred to as a CPU. Finally, number 4 is a projector with a maximum resolution of 1280 x 960 pixels. However what is not labeled in the diagram are the four infrared cameras to capture the infrared light reflected off the screen by fingers of users and objects placed on the screen.

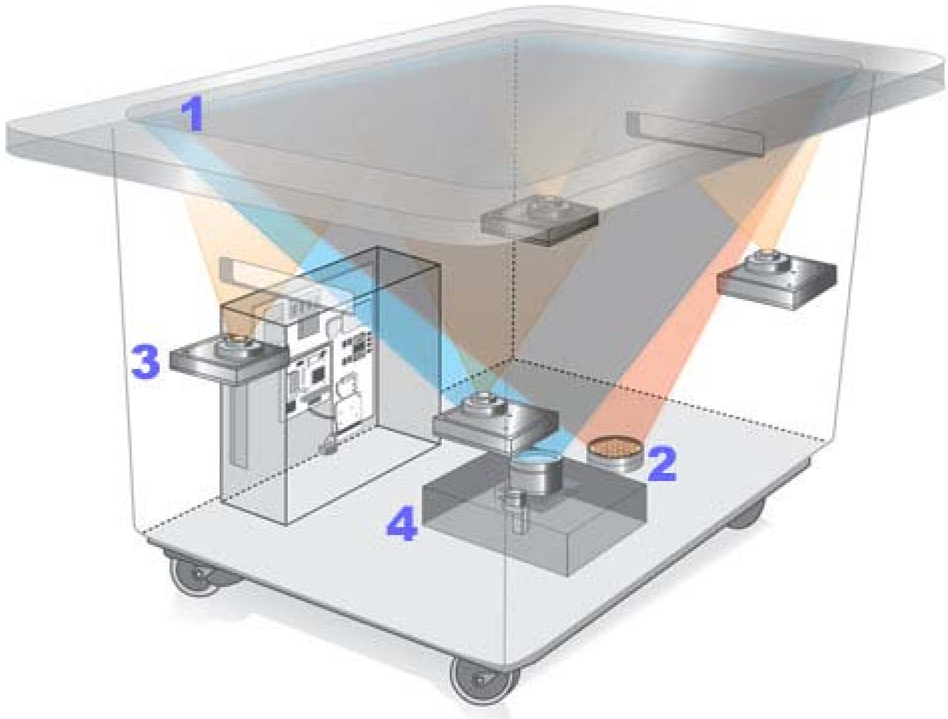

Figure 5: Diagram of a Microsoft Surface device.
Diagram created by Intoaroute (Derene, 2007).

Laser Light Plane (LLP), also referred to as the Gap Method, uses a “wide baffle” located on around the outer edges of the medium, such as glass or acrylic, with space approximately 0.5mm between the baffle and the medium. This space, or gap, allows a small infrared beam to pass over the medium. When an object or user breaks the beam, a camera mounted below the screen senses the infrared light. It is suggested that on all four sides of the medium LEDs should be mounted (Natural User Interface Group, 2008b, 2009a).

Diffused Surface Illumination (DSI) is similar to the FTIR configuration with a series of infrared LEDs around the edges of the acrylic. However the primary difference is with the type of acrylic used called Endlighten Plexiglas. Within this type of Plexiglas are small particles serving as mirrors to bounce the infrared light across the acrylic (Natural User Interface Group, 2009b).

Capacitance Testing method senses an object or user touching the screen using a “complex electronical grid.” This is the technology used for Apple’s iPhone. (Natural User Interface Group, 2008b, 2009a).

Regardless of the technology used for a multi-touch device the robustness of the system is critical. Over time with use the surface of the device in which the user interacts, either with their hands or with objects, can become “contaminated with oils and sweat left behind from users, along with nicks and scratches, creating an increase in background noise against which a true signal must be isolated” (Han, 2005). The background noise created detracts from the sensors, especially for the FTIR and DI methods, used to detect objects making contact with the surface of the device.

Protocols have been developed enable multi-touch device hardware to communicate with software applications. One of the open-source protocols developed by the NUI Group is called Table-Top User Interfaces Objects (TUIO). The communication mechanism is defined by the NUI Group as “a simple yet versatile protocol designed specifically to meet the requirements of table-top tangible user interfaces” (Natural User Interface Group, 2009c). The project web site for TUIO describes the protocol and its application programming interface (API) as having the purpose of transmitting encoded data of objects on the surface of the device from a tracker device, such as an infrared camera, to a software application that is capable of decoding the protocols information packets and responding accordingly (“TUIO.org,” n.d.). In other words, the TUIO protocol has two primary properties: first the detection, recognition, position, and orientation of objects placed on the surface of the screen; second the detection of the users’ gestures on the surface of the screen and assigns unique identification numbers to each object (Kaltenbrunner, Bovermann, Bencina, & Costanza, n.d.; Natural User Interface Group, 2009c). Originally developed by Martin Kaltenbrunner, Ross Bencina, Enrico Costanza, and Till Bovermann, it now has received adoption for a variety of multi-touch and tangible projects including Touchlib and T-Beta, both projects of the NUI Group. The TUIO protocol is based on the OpenSound Control framework which allows for several other programming languages to support the TUIO protocol. Included in the list of these programming languages is ActionScript, Java, C/C++, and C#, just to name a few (Natural User Interface Group, 2009c). OpenSound Control (OSC) was originally developed as “a protocol for the communication between controllers and sound synthesizers” (Natural User Interface Group, 2009d).

Currently there are two protocols extended from the TUIO protocol that helps to configure and calibrate the hardware of a multi-touch device. The first is called Touchlib which is a multi-touch development kit including a library that uses the TUIO protocol. Touchlib communicates with software applications as it tracks blobs on the surface through an infrared camera. Then the information is sent to Touchlib compatible application, which responds to the multi-touch events detected. Applications built on the Adobe Flash platform can interact with Touchlib. At this time, Touchlib is only available on a Microsoft Windows operating system (Natural User Interface Group, n.d., "Touchlib").

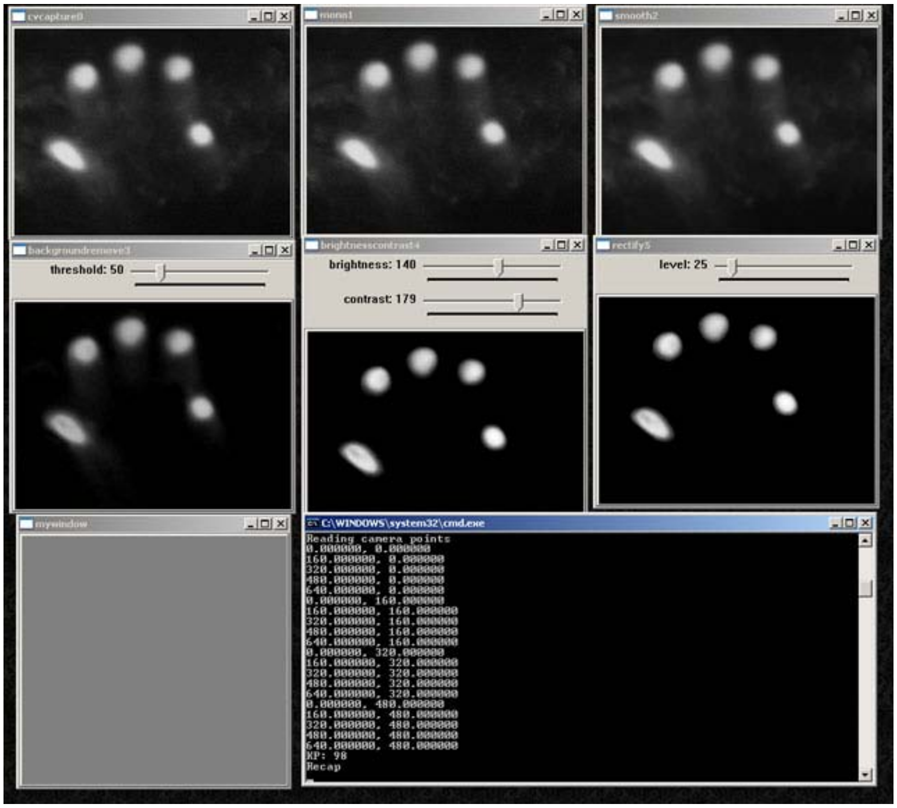

Figure 6: A diagram of the Touchlib interface (Natural User Interface Group, n.d., "Touchlib Screenshots).

The configuration application for the Touchlib development kit, as shown in the figure above, is displayed in a series of windows. Going from left to right, top to bottom the windows are: Capture0, Mono1, Smooth2, BackgroundRemove3, BrightnessContrast4, Rectify5, myWindow, and Microsoft Command-Prompt, a window part of the Microsoft Windows operating system (Natural User Interface Group, 2008c). The Capture0 display provides raw video received by the infrared camera of the multi-touch device. Mono1 ensures the image is in the necessary format by using a filter to convert the raw video input into a grayscale image. Filtering out any background noise that may be captured in the video input is accomplished in the BackgroundRemove3 window. The threshold scale in the BackgroundRemove3 window can amplify the filter's intensity on the video input. The BrightnessContrast4 window, which is also referred to as "High-pass," as the name implies provides controls for the brightness and contrast levels of the video input. Rectify5 is the last filter applied to the video input which prepares the output as a black and white image for detection by the Touchlib library. The level scale controls the intensity of the filter from Rectify5 applied to the video input (Natural User Interface Group, 2008c).

T-Beta (also written as tBeta), an abbreviation for “The Beta,” is another protocol that utilizes TUIO. It is described as an “open source / cross-platform solution for computer vision and multi-touch sensing” (Natural User Interface Group, n.d., "Community Core Vision"). Recently the name of the T-Beta project has been changed to Community Core Vision (CCV). The T-Beta protocol is cross-platform, making it capable of running on the three major operating systems: Microsoft Windows, Apple/Macintosh, and Linux. When using the T-Beta protocol filers, such as “dynamic background subtraction, high-pass, amplify/scaler, [and] threshold,” enable it to be compatible with the following optical multi-touch technologies: FTIR, DI, LLP, & DSI. T-Beta can also take advantage of the computing resources of the GPU on the graphics card of the computer to increase the tracking of contact points (Natural User Interface Group, n.d., "tBeta"). T-Beta can communicate with applications built with ActionScript if the application has been developed to recognize data sent from T-Beta. Compared to Touchlib, T-Beta at times has been found to not be stable. Touchlib is a predecessor of T-Beta, which expanded upon the foundation Touchlib set by adding new features such as an improved configuration interface.

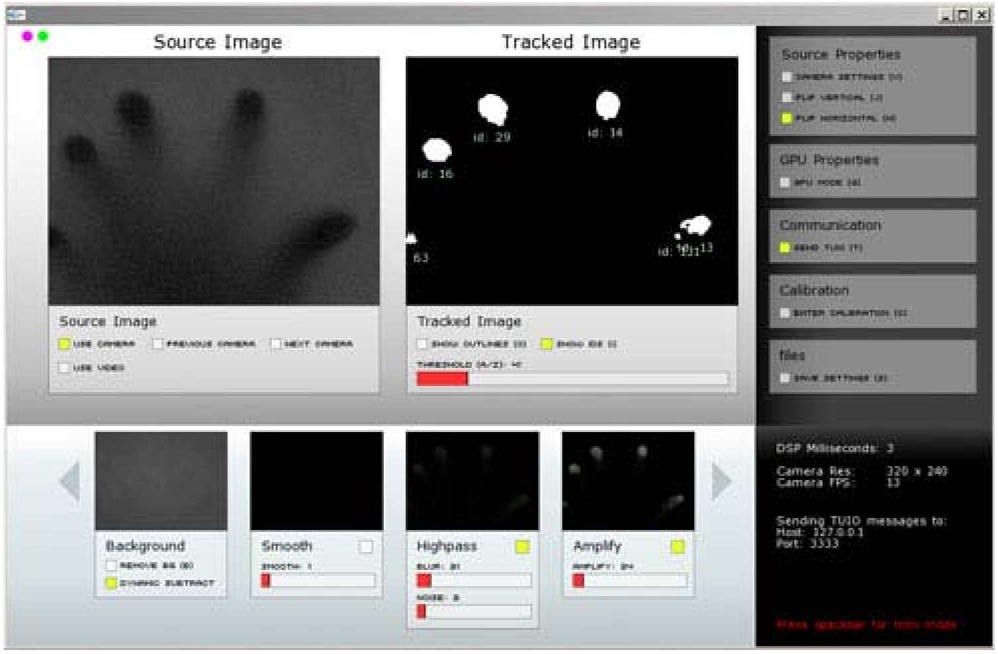

Figure 7: A diagram of the t-Beta interface
(“Multitouch table: finger tracking,” 2009).

In the diagram above the Source Image on the left shows the raw video input captured by the infrared camera of the multi-touch device. On the right is the Tracked Image depicting the processed black and white image by tBeta to track the contact points on the surface of the device. The four smaller images in the lower part of the diagram are similar to Touchlib. Going from left to right, the displays are: Background, Smooth, High-pass, and Amplify. Each of these serve the same control functionality as described for Touchlib. The menu of options on the right side of the diagram provides additional features for configuration. The top section, titled Source Properties, allows the user to control the infrared camera and flip the video input both horizontally and vertically. The next section titled GPU Properties will allow T-Beta to activate a GPU processing capability to use the graphical processor on the computer’s video card. Activating the TUIO data sent by the T-Beta application is controlled in the next section, titled Communication. The Calibration section enables to user to enter into another mode of calibrating the resolution and number of detection points for the T-Beta application to sense contacts make with the surface of the multi-touch device. The last section gives the user the option to save the configuration and calibration settings (Natural User Interface Group, 2008d).

1.2.2. Networking Multi-touch Devices

There has been a challenge in discovering publications related to this topic and is an area that should be experimented and researched further.

1.2.3. Gestural Interfaces

What sets apart a touch device from other means of interacting with a computer, according to William Buxton at the Computer Systems Research Institute at the University of Toronto (Toronto, Ontario, Canada), is “that the user is not required [to] point with some manually held device such as a stylus or puck” (Buxton, Hill, & Rowley, 1985)**.** Buxton and his research team describe the difference from a one-touch device to a device capable of sensing “multiple points of contact” as unlike the one-touch device with a multi-touch device the location of items, such as fingers of users or items placed on the device, making contact with the device are recognized instantaneously. Some devices also have the capability to respond to the pressure the item on the device exerts onto the device. Buxton (1985) highlights the properties of touch-sensitive devices being ideal for hostile environments where input peripherals such as a computer mouse or stylus could become damaged, lost or stolen. He goes on to state that these devices “present no mechanical or kinesthetic restrictions on [a user’s] ability to indicate more than one point at a time,” thus enhancing the experience a user has with an application by allowing a user to have simultaneous points of contact with the device (Buxton et al., 1985). As the device communicates with the application, the data sent back and forth can replicate the interaction that a one-button mouse provides to a user. However, Buxton (1985) points out that the research doesn’t signify that these devices are “equivalent or interchangeable,” rather a touch device should be used where its properties are beneficial for a user over a traditional means of input for a computer (such as a mouse, keyboard, or stylus).

To track the interactions of users with a multi-touch device, the infrared camera of the device will recognize the bright areas of the video input created by objects or fingers of users making contact with the device. This process is referred to as “Blob Detection” (Natural User Interface Group, n.d., "Blog Detection"). The different configurations of multi-touch devices are discussed in Section 1.2.1. Regardless of the configuration the process for each is the same.

Figure 8: Image showing blob detection from an infrared camera on a multi-touch device (Natural User Interface Group, 2009, "Multi-Touch Terminology").

Since video captured by a camera is a series of frames of still images combined together immediately following each other in a chronological order. A method was developed to assign a unique identifier to each blob detected. This method is called “blog tracking” (Natural User Interface Group, n.d., "Blob Tracking"). As each frame in from the video source changes the blobs are compared to the previous frame to determine which blobs on the new frame belong to which unique identifier (Natural User Interface Group, n.d., "Blob Tracking").

Over the past few years the research and development of multi-touch sensing devices has increased in popularity. With these devices comes a new model regarding how a user interacts with computer applications, versus the traditional computer mouse and keyboard. The new model allows the user to use multiple fingers with a series of movements, typically referred to as gestures. Multimedia applications need to be able to recognize this new form of input in order for the application to respond according to the user's interaction. One example of input techniques as used in an application called *RoomPlanner* which was developed to aide people in planning the location of items in a room using a multi-touch device (Wu & Balakrishnan, 2003). The gestures developed for this application are: tap, double tap, flick (tapping and dragging away from the user), catch (tapping and dragging towards the user), flat hand, vertical hand, horizontal hand, tilted hand, two vertical hands, and two corner-shaped hands.

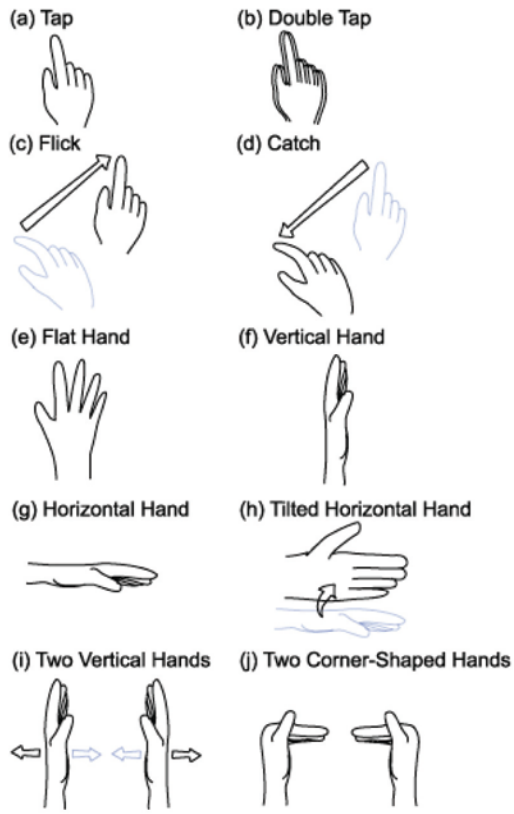

Figure 9: A set of gestures developed for the *RoomPlanner* application
(Wu & Balakrishnan, 2003)

In this specific application, a double tap pops up a pie-shaped menu for the use relative to what the user is doing with options in the menu to control the properties of a specific item. A single tap will select a menu option or select a specific item, depending on the context (Wu & Balakrishnan, 2003).

In *Designing Gestural Interfaces* by Dan Saffer, he describes specific gestures for touch screen devices similar to those used in the *RoomPlanner* application. The first is called Tap which is the action of placing the “tip or pad of the finger” on the surface of the device for less than 100 milliseconds (Saffer, 2008, pp. 181-182). Similarly a Double Tap is as the name implies the act of performing a tap “twice rapidly with a [less than] 75-millisecond pause in between the two contacts” (Saffer, 2008). Saffer recommends the use of these gestures for selecting items or buttons.

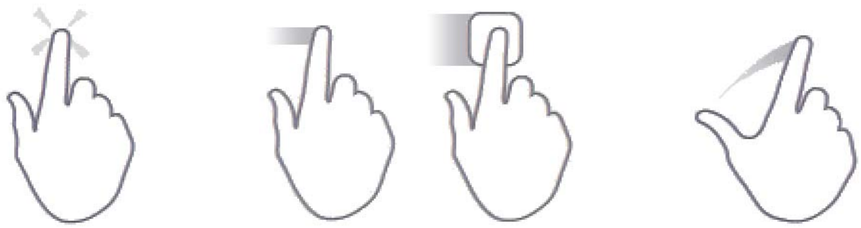

Figure 10: Common gestures used for touchscreen devices, part 1. From left to right: Tap, Drag/Slide, and Flick (“Fling”) (Saffer, 2008, p. 181).

The second gesture is referred to as either Drag or Slide which occurs once the tip or pad of the finger makes contact with the surface and moves without loosing contact. Scrolling and a drag/drop sequence is usually associated to the drag/slide gesture. The next gesture defined by Saffer can be called either Flick or Fling. This action can be performed in one of two ways. The finger starts in a crooked position, then part of the finger makes a brushing motion lasting approximately 75 milliseconds as the finger straightens out. Another method of performing a flick occurs when the finger starts straight and moves in a reversed manner motioning toward the thumb as it makes contact with the surface for the same duration. This action is commonly used for objects to be quickly moved as well as to scroll.

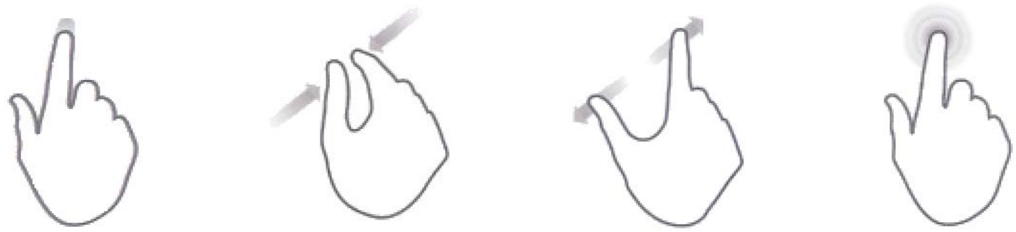

Figure 11: Common gestures used for touchscreen devices, part 2. From left to right: Nudge, Pinch, Spread, and Hold (Saffer, 2008, pp. 181-182).

The fourth gesture is named Nudge and is used to move objects a short distance. This gesture is simply to have the tip or pad of a straight finger make contact with the surface and briefly slides forward lasting no more than two seconds. Pinch is the next gesture described by Saffer, as two fingers, usually the index finger and thumb, move closer together. Scaling elements of an interface typically uses this gesture. The fifth gesture is called Spread which is similar to the Pinch gesture, however the two fingers move away from eachother. This gesture is also used for scaling. Finally, the last gesture defined by Saffer is referred to as Hold or Press. For this action the tip or pad of the finger makes contact with the surface of the device for a longer duration than Tap. Uses for this gesture include selection and scrolling for an extended period of time.

Saffer also describes in his book how direct manipulation of digital objects on a computer, without using the system’s command lines, generates a more interactive experience for users through gestures of what is referred to as a natural user interface (Saffer, 2008, p. 15). He goes on to provide a list of best-practices for gestural interfaces referred to as affordances. Saffer defines affordances as “one or more properties of an object that give some indication of how to interact with that object or a feature on that object” (Saffer, 2008, p. 19). The characteristics are: Discoverable – indicating to the user that the device is touchable and interactive; Trustworthy – the user needs to feel a level of safety and security when using the device; Responsive – the device needs to provide on-going, preferrable in real-time, feedback to the user as an acknowledgement of the user’s actions; Appropriate – the factors of culture, situation as well as context of the situation where the user is engaging with the device should be taken into consideration; Meaningful – the actions and tasks the device asks a user to accomplish should aide the user in reaching the user’s goal for engaging with the device; Smart – the device needs to be able to perform tasks for the user that are usually difficult or challenging for the user; Clever – the interface should be adaptive to the user’s needs and be able to predict what the next needs of the user will be while interacting with the device; Playful – interfaces should be welcoming and promote the exploration of the user to try new features and use variations of gestures; Pleasurable – a positive experience needs to be provided to the user to increase the likelyhood of the user to engage with the device in the future; and Good – the interface should respect all possible users that will engage with the device as to not embarrass or offend any user (Saffer, 2008, pp. 19-22).

1.2.4. Adobe Flash Platform

One of the interface technologies that both Touchlib and T-Beta can communicate with is Adobe Flash. Adobe describes their Flash Platform as “an integrated set of technologies surrounded by an established ecosystem of support programs, business partners, and enthusiastic user communities” (Adobe Systems, n.d., "Adobe Flash Platform"). At the core of the Flash Platform is the Adobe Flash Player which is a free downloadable plug-in for users. According to Adobe the Flash Player is “the de facto standard for rich applications, content and video in the browser… Flash Player is a high-performance, cross-platform client runtime that delivers powerful and consistent user experiences in the browser to more than 99% of Internet users” (Adobe Systems, n.d., "Adobe Flash Platform"). In addition to Flash’s “visual programming interface,” ActionScript is the object-oriented programming language for the Adobe Flash Platform (Adobe Systems, n.d.; Natural User Interface Group, n.d.). Another means of delivering content and interactive applications developed using the Adobe Flash Platform is through the Adobe Integrated Runtime (AIR). Adobe describes AIR as a “runtime” that allows developers familiar with web technologies to create applications that run on the desktop of a computer rather than in an Internet browser (Adobe Systems, n.d., "Adobe AIR"). AIR provides a new channel for businesses to offer their customers engaging experiences across all major operating systems. Developers already familiar with Adobe Flash, Adobe Flex, or ActionScript 3.0 can be utilized to rapidly deploy applications using Adobe AIR.

Developing computer applications using technologies, such as the Adobe Flash Platform is referred to as “rich interactive applications” (Tretola, 2008). Tim O’Reilly of O’Reilly Publishing stated that Macromedia, now owned by Adobe Systems, originally coined the term “to highlight the capabilities of Flash to deliver not just multimedia content but also GUI-style application experiences” (Tretola, 2008). Through an interview with Allen Lewis (2008) of eBay, Inc., who served as the project manager for eBay Desktop, Lewis shared his experience with the selection process his team used for choosing Adobe AIR. He described how his team wanted to offer eBay’s customers a unique experience without reinventing the existing ebay.com web site. Lewis’s experience using AIR improved the production pipeline process of his team by promoting new levels of collaboration between the traditionally silos of designers and developers.

Adobe posted a video in a digital publication, called *Inspire*, produced by the Adobe Experience Design team where Julie Meridian, Senior Experience Designer with Adobe, and Tim Kukulski, Senior Computer Scientist with Adobe, shares how Adobe sees the industry future for multi-touch devices (*Adobe and the future of multitouch*, 2009). Meridian describes multi-touch as “a way to help you get your idea in the form you wanted to faster.” She goes on to state that some current touch technologies provide an indirect means of manipulating objects of an interface, however the best form of multi-touch - as discussed in Section 1.2.3 with Saffer’s description of direct manipulation of interfaces. Meridian references the Steven Spielberg movie *Minority Report* with actor Tom Cruise and the computer system in which Cruise’s character interacts as one of the movies that inspired research in the multi-touch field. She states that Adobe is hardware agnostic with the focus being on delivering tools and services that are cross-device and cross-platform. Kukulski highlights the challenge Adobe has had determining how multi-touch technologies will fit into the Adobe Flash Platform, as well as Adobe’s other tools such as Photoshop and Illustrator. “Content is king” especially when cognitive noise is eliminated to provide a richer experience for users. Meridian states that “touch is encountered in our daily lives when we are doing simple things… multi-touch is just an extension of the touch technology, but it enables much richer interactions.” Adobe foresees multi-touch technologies to enable users to work faster with computers, especially providing more accessibility to users with disabilities through gestural interfaces. Multi-touch is viewed by Adobe as an alliance between “hardware, industrial design, and software” (*Adobe and the future of multitouch*, 2009).

Richard Monson-Haefel, an award winning author as well as a multimedia designer and developer, sheds light on Adobe’s commitment to exploring multi-touch technologies. In his professional option, once Adobe is able to release an ActionScript API for multi-touch devices, it will help Adobe to keep their market share based upon the vast number of developers currently using ActionScript through Adobe Flash and Adobe Flash (Monson-Haefel, 2009).

1.2.5. Affinity Diagram

Six Sigma is a business methodology focused on data-driven decision making processes to strive for a measure of quality that nears perfection. Through this model processes are reviewed to attempt to remove all defects or errors that make occur during a process. The Six Sigma methodology has saved companies hundreds of thousands of dollars by increasing productivity and efficiency (iSixSigma, n.d.). One of the many tools used with the Six Sigma methodology is called an Affinity Diagram which serves as a means of organizing data and ideas. The Affinity Diagram is achieved through a group exercise consisting of five or six people with one individual serving as a moderator. To start the exercise an initial key phrase or topic needs to be defined by the participants that will serve as the overarching theme. Steven Bonacrosi (2008), a Senior Master Black Belt instructor and coach of Six Sigma, recommends that the results produced by the exercise will be more effective if the key phrase is written loosely in broad terms (Arnheiter & Maleyeff, 2005; Bonacorsi, 2008; Mind Tools, n.d.; Quinn, n.d.).

During the exercise there are typically three methods to generate issues, concepts, and/or ideas. The first is to silently have the participants record their ideas onto index cards or Post-it notes. The ideas recorded should include a few characteristics: concise with no more than seven words in length; direct and unambiguous; and limited to one concept per card. A second form of generating ideas is retrieving ideas collected by a qualitative survey with each idea on a separate card, following the same characteristics mentioned above. An additional form of generating ideas is to have the moderator record the ideas that each participant vocalize. While participants are performing one of these methods a few guidelines are recommended: everyone needs to be included; ideas should be recorded exactly as verbalized; criticism or discussion of concepts should not be allowed; and produce as many concepts as quickly as possible (Arnheiter & Maleyeff, 2005; Bonacorsi, 2008; Mind Tools, n.d.; Quinn, n.d.).

Figure 12: Affinity diagramming process in the random ideas stage, using social media as an example. Diagram created by research team.

The next step in the exercise is to arrange all the cards randomly so that all the participants can view them. Simultaneously each of the participants will then silently arrange the cards that have a relation into groups. Forming the groups into vertical columns is the preferred method. As this occurs each card should remain visible and not be covered by other cards. Participants should freely exchange cards as they are clustering, which may involve participants silently disputing where some cards should belong. Some cards may need to belong in more than one group which may create a connection between groups. If repetitive cards exist, the cards may be overlapped but in a fashion where it can still be read. Some cards may not fit into any group as the groups emerge and may need to remain independent. Cards should not be forced into groups. Once the groups and categories are formed, discussion is now permitted with the participants. Through this discussion a card that captures the central idea of a group should be used as the group header and moved to the top as a title. If a central idea doesn't exist on a card, a new card can be created (following the card characteristics already described) with a word or short phrase that highlights the intent of that category. If there are large groups of cards that exist, these groups can be divided into smaller sub-groups each with their own headings. Over all there should not be more than ten groups created (Arnheiter & Maleyeff, 2005; Bonacorsi, 2008; Mind Tools, n.d.; Quinn, n.d.).

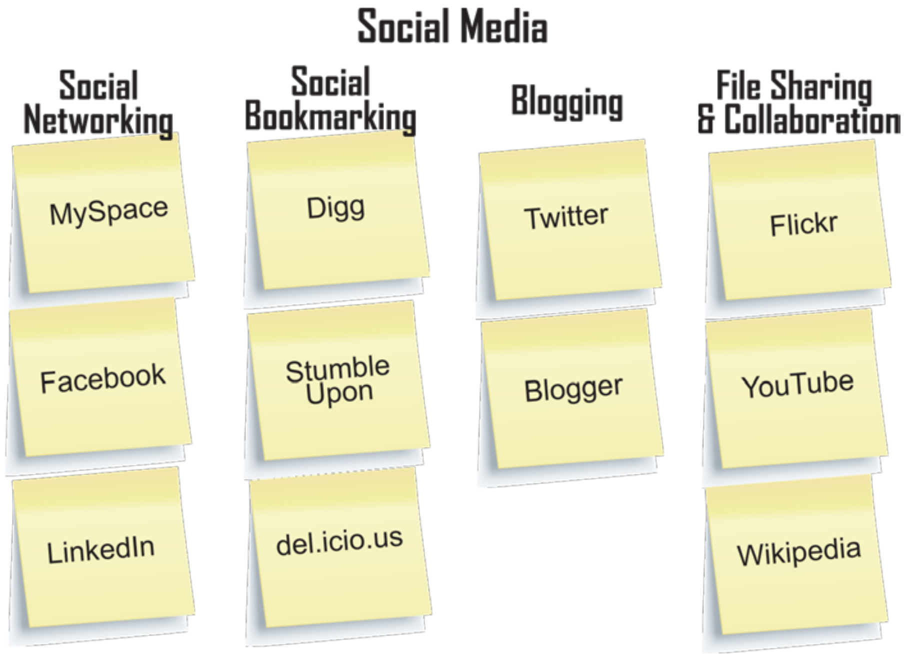

Figure 13: A completed Affinity Diagram, using social media as an example. Diagram created by research team.

By performing this exercise the categories of common themes may help to discover unseen relationships of the ideas. The focus of the exercise is to brainstorm solutions to the root cause of issues (Bonacorsi, 2008; Quinn, n.d.).

1.2.6. Group-based Behavior while Interacting with Computing Devices

The research in this area includes topics such as information sharing, team performance, and computer-supported cooperative work. The primary purpose of a computer-support cooperative work (CSCW) computer application is to “provide users with awareness information” (Gross, Stary, & Totter, 2005, p. 1).

1.2.7. Testing Methodologies

Considered one of the most widely accepted and used techniques to collect subjective qualitative data from users, the National Aeronautics and Space Administration (NASA) Task Load Index (NASA-TLX) is a tool for workload assessment (“NASA TLX: Task

Load Index," n.d.; Proctor & Van Zandt, 2008, p. 255). To determine the workload placed upon a user, six scales have been developed: mental demand, physical demand, temporal demand, performance, effort, and frustration level. Each of these scales has proven through research to "make a relatively unique contribution to the subjective impression of workload" (Proctor & Van Zandt, 2008, p. 255). The NASA-TLX was created by NASA in 1988 to have a multi-dimensional rating scale to assess how "information about the magnitude and sources of size workload-related factors are combined to derive a sensitive and reliable estimate of workload" (Hart & Staveland, 1988, p. 1). The NASA-TLX can be used in a variety of human-machine settings to evaluate workload ("NASA TLX: Task Load Index," n.d.).

1.3. Summary

Currently the two leading expert sources of information regarding multi-touch devices are Jeff Han and the Natural User Interface Group. There are five primary configurations for building a multi-touch device. The two configurations receiving the most attention are Frustrated Total Internal Reflection and Diffused Illumination. The Table-Top User Interfaces Objects (TUIO) is a protocol that allows software applications to communication with an infrared camera sensing objects making contact with the surface of a multi-touch device. Two primary interfaces that aide in calibrating a multi-touch device to work with TUIO are Touchlib and T-Beta. Gestures are the standard method of interacting with applications on a multi-touch device, which include: tap, double tap, drag/slide, flick, nudge, pinch, spread, and hold. The Adobe Flash Platform enables developers to build rich internet applications using the ActionScript programming language. Utilizing either Touchlib or T-Beta applications developed for multi-touch devices with the Adobe Flash Platform can respond to multi-touch gestures. The Affinity Diagramming process is a group exercise that involves multiple people interacting simulataneously, which can take advantage of the multi-user capabilities of a multi-touch device. Information sharing can be increased among teams to increase performance through the use of a computer-supported cooperative work-system. The

NASA-TLX is the most widely accepted assessment for subjective qualitative data for workload analysis of human-machine interactions.

1.4. Bibliography

Adobe and the future of multitouch. (2009). INSPIRE - A publication from the Adobe Experience Design team (Vol. 3). Retrieved May 2, 2009, from https://xd.adobe.com/#/featured/video/160.

Adobe Systems. (n.d.). Adobe Flash Platform. *Application Programming - Adobe Flash Platform*. Retrieved April 13, 2009, from http://www.adobe.com/flashplatform/?promoid=DRHWS.

Adobe Systems. (n.d.). Rich Internet Applications - Adobe AIR. *Adobe AIR*. Retrieved May 4, 2009, from http://www.adobe.com/products/air/.

Arnheiter, E., & Maleyeff, J. (2005). The integration of lean management and Six Sigma. *The TQM Magazine - Emerald Group Publishing Limited*, *17*(1), 5-18. doi: 10.1108/09544780510573020.

Bonacorsi, S. (2008). What is... an Affinity Diagram? *onesixsigma*. Retrieved April 4, 2009, from http://www.onesixsigma.com/article/what-is-an-affinity-diagram.

Buxton, W., Hill, R., & Rowley, P. (1985). Issues and techniques in touch-sensitive tablet input. *ACM SIGGRAPH Computer Graphics '85*, *19*(3), 215-224. doi: 10.1145/325165.325239.

Derene, G. (2007). Microsoft Surface Video - Touchscreen, Multi Touch Coffee Table - Behind the Scenes. *Popular Mechanics*. Retrieved April 30, 2009, from http://www.popularmechanics.com/technology/industry/4217348.html?page=2.

Fantini, G. (2008). Quick empirical comparison between 3 different methods for a compliant surface. *NUI Group - Forum*. Retrieved May 2, 2009, from http://nuigroup.com/forums/viewthread/2911/.

Gross, T., Stary, C., & Totter, A. (2005). User-Centered Awareness in Computer-Supported Cooperative Work-Systems: Structured Embedding of Findings from Social Sciences. *International Journal of Human-Computer Interaction*, *18*(3), 323-360.

Han, J. Y. (2005). Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User interface software and technology* (pp. 115-118). Seattle, WA, USA: ACM. doi: 10.1145/1095034.1095054.

Han, J. Y. (2006). Multi-touch interaction wall. In *ACM SIGGRAPH 2006 Emerging technologies* (p. 25). Boston, Massachusetts: ACM. doi: 10.1145/1179133.1179159.

Hart, S., & Staveland, L. (1988). Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. Aerospace Human Factors Research Division, NASA-Ames Research Center. Retrieved April 29, 2009, from http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20000004342_1999205624.pdf.

iSixSigma. (n.d.). Six Sigma - What is Six Sigma? Retrieved April 29, 2009, from http://www.isixsigma.com/sixsigma/six_sigma.asp.

Kaltenbrunner, M., Bovermann, T., Bencina, R., & Costanza, E. (n.d.). TUIO: A Protocol for Table-Top Tangible User Interfaces. Retrieved March 4, 2009, from http://mtg.upf.edu/files/publications/07a830-GW2005-KaltenBoverBencinaConstanza.pdf.

Lewis, A. (2008). Interview with Product Manager for eBay Desktop.

Mind Tools. (n.d.). Affinity Diagrams - Organizing Ideas Into Common Themes. *Mind Tools*. Retrieved April 16, 2009, from http://www.mindtools.com/pages/article/newTMC_86.htm.

Monson-Haefel, R. (2009, March 12). Adobe on Multitouch. *MultiTouch ++*. Retrieved May 2, 2009, from http://theclevermonkey.blogspot.com/2009/03/adobe-on-multitouch.html.

Multitouch table: finger tracking. (2009). *Modprobe - Blog Archive*. Retrieved May 2, 2009, from http://modprobe.complex.ch/2009/03/23/multitouch-table-finger-tracking/.

NASA TLX: Task Load Index. (n.d.). *NASA*. Retrieved April 29, 2009, from http://humansystems.arc.nasa.gov/groups/TLX/.

Natural User Interface Group. (n.d.). About NUI Group. *NUI Group - Natural User Interface Group*. Retrieved April 13, 2009, from http://nuigroup.com/log/about/.

Natural User Interface Group. (n.d.). Touchlib - A Multi-Touch Development Kit. *NUI Group - Touchlib*. Retrieved April 6, 2009, from http://www.nuigroup.com/touchlib/.

Natural User Interface Group. (n.d.). Community Core Vision. *NUI Group - Code*. Retrieved May 3, 2009, from http://nuicode.com/projects/tbeta.

Natural User Interface Group. (n.d.). T-Beta. *NUI Group – The Beta Release*. Retrieved March 4, 2009, from http://tbeta.nuigroup.com/.

Natural User Interface Group. (n.d.). Blob Detection. *NUI Group - Wiki*. Retrieved February 17, 2009, from http://wiki.nuigroup.com/Blob_Detection.

Natural User Interface Group. (n.d.). Blob Tracking. *NUI Group - Wiki*. Retrieved February 17, 2009, from http://wiki.nuigroup.com/Blob_Tracking.

Natural User Interface Group. (n.d.). Flash. *NUI Group - Wiki*. Retrieved March 3, 2009, from http://wiki.nuigroup.com/Flash.

Natural User Interface Group. (2008a). Multi-model. *NUI Group - Wiki*. Retrieved April 25, 2009, from http://wiki.nuigroup.com/Multi-model.

Natural User Interface Group. (2008b). Getting Started With MultiTouch. *NUI Group - Forum*. Retrieved April 12, 2009, from http://nuigroup.com/forums/viewthread/1982/.

Natural User Interface Group. (2008c). Touchlib Screenshots. *NUI Group – Wiki*. Retrieved March 4, 2009, from http://wiki.nuigroup.com/Touchlib_Screenshots.

Natural User Interface Group. (2008d). Getting Started with tbeta. *NUI Group - Wiki*. Retrieved May 4, 2009, from http://wiki.nuigroup.com/Getting_Started_with_tbeta.

Natural User Interface Group. (2009a). Multitouch Technologies. *NUI Group - Wiki*. Retrieved February 17, 2009, from http://wiki.nuigroup.com/Multitouch_Technologies.

Natural User Interface Group. (2009b). DSI. NUI Group - Wiki, . Retrieved May 4, 2009, from http://wiki.nuigroup.com/DSI.

Natural User Interface Group. (2009c). TUIO. *NUI Group – Wiki*. Retrieved March 4, 2009, from http://wiki.nuigroup.com/TUIO.

Natural User Interface Group. (2009d). OpenSound Control. *NUI Group – Wiki*. Retrieved March 3, 2009, from http://wiki.nuigroup.com/OpenSound_Control/.

Natural User Interface Group. (2009e). Multi-Touch Terminology. *NUI Group - Wiki*. Retrieved May 4, 2009, from http://wiki.nuigroup.com/Multi-Touch_Terminology.

Proctor, R., & Van Zandt, T. (2008). *Human Factors in Simple and Complex Systems* (2nd ed., p. 255). Boca Raton, FL: CRC Press.

Quinn, D. (n.d.). Affinity Diagram. An Introductory Course on Quality Management - Vanderbilt University, . Retrieved April 16, 2009, from http://mot.vuse.vanderbilt.edu/mt322/Affinity.htm.

Saffer, D. (2008). *Designing Gestural Interfaces: Touchscreens and Interactive Devices* (illustrated edition., pp. 4-5, 15, 181-182). O'Reilly Media, Inc.

TED 2006 Conference. (2006). Jeff Han demos his breakthrough touchscreen. *Video on TED.com*. Retrieved April 27, 2009, from http://www.ted.com/index.php/talks/jeff_han_demos_his_breakthrough_touchscreen.html.

Tretola, R. (2008). What is RIA? *O'Reilly - Inside RIA*. Retrieved February 6, 2008, from http://www.insideria.com/2008/01/what-is-ria-1.html.

TUIO.org. (n.d.). *TUIO*. Retrieved May 3, 2009, from http://www.tuio.org/.

Wu, M., & Balakrishnan, R. (2003). Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *Proceedings of the 16th annual ACM symposium on User interface software and technology* (pp. 193-202). Vancouver, Canada: ACM. doi: 10.1145/964696.964718.