GAMING CONCEPTS IN ACCESSIBLE HCI FOR BARE-HAND COMPUTER INTERACTION

Kristopher Reese, Ahmed Salem, and George Dimitoglou
Department of Computer Science
Hood College
Frederick, MD, USA
{kwr2, salem, dimitoglou}@hood.edu

KEYWORDS
Accessibility, Human-Computer Interaction, Human Factors, Computer Vision, Bare-Hand Controls, Object Detection, Future Interfaces

ABSTRACT

Hand-Computer interaction is a frequently researched topic in the field of computer gaming. However, very few of the devices or techniques used for Hand interaction with are accessible within many smaller groups, especially those with disabilities. For this paper, we have developed a tool to show several of the techniques used in gaming research and describe how these techniques could be implemented in Human-Computing Interaction with a focus on accessibility for the physically challenged. A Bare-Hand tracking technique is used to track the location of the Hand in relation to a single web camera. This proof of concept interface offers a look at how future devices could be both fun for typical users without sacrificing on accessibility.

1. INTRODUCTION

The recent decade has brought growth and innovation to the computer gaming industry. Much of the innovation has been focused on human-computer interaction and techniques to enhance how users interface with video games. New interfaces, such as the Nintendo Wii-mote and the Sony Eye-Toy, are the results of more than 20 years of research in the fields of Virtual Reality and Spatial 3D Interactions (LaViola 2008). These technologies now dominate gaming but their potential applications to other areas have hardly been met. In particular, applying these technologies to provide enhanced computing accessibility to those with physical disabilities. For example, replacing computer mice on computer systems or implementing virtual input devices replacing keyboards for those with motor skill disabilities that are unable to use physical input devices.

Such technologies often remain inaccessible to those with some forms of disabilities. In 2004, the United Nations released a report with statistics about disabilities at an international scale, stating that in most countries at least one in ten people are disabled by physical, mental, or sensory impairment and that at least 25 percent of any population is adversely affected by the presence of disabilities (United Nations, 2004). Population aging also augments these numbers by introducing computer users with degraded body functions, including vision, memory, and motor skills. According to the World Health Organization, there were 600 million people above the age of 60 in 2000, and it is expected that the number will double by 2025 (World Health Organization, 2004).

It is clear that new and innovative forms of interfacing with computers are becoming increasingly necessary both to those with disabilities and to an aging but computer savvy population. While current technologies have a lot of potential providing a wide variety of solutions, we focus on techniques that allow users to interact with computers strictly by motion.

The following section provides an overview of related work in the areas of assistive technologies for game interfaces and later covers the background work specific to object and gesture detection. The next section describes a new method for hand detection that allows human-computer interactions using a web camera. Subsequently, the architecture of the Motion Control Application (Mocoa) is described, along with the algorithm that detects a user's hand and how the overall system operates. Recognizing the importance of performance and how it may affect the effectiveness of an interfacing application a brief performances analysis section illustrates key areas that can enhance the utilization of the application. The last section includes the conclusion and discussed future work.

2. RELATED WORK

As many of these devices begin their development in the gaming industry, looking at the development and accessibility becomes a necessity. By looking at Gaming Accessibility we can observe the development of these devices beyond the typical user interactions that they were developed. However, Grammenos et al. make clear observations that relatively few efforts have been devoted to game accessibility and those with the primary concern of accessibility in gaming are organized groups of disabled people (Grammenos et al 2009).

Grammenos et al. later states that there are often two main approaches to addressing the issue of game accessibility, which often applies to fields outside of gaming as well. Grammenos et al. states that inaccessible games are often made accessible through third-party assistive technologies or are developed from scratch, often with a specific disability in mind (Grammenos et al. 2009). There are drawbacks to both approaches; however, these drawback would be limited if new forms of input devices are developed that would be easy to use and developed for users with both disabilities and for those without disabilities. The Nintendo Wii-mote and the Sony Eye-toy are a step in the right direction, but more work needs to be done to make them more usable for a larger percentage of the disabled population.

LaViola's article explores the history of Video Game interaction and discusses some of the future developments of spatial 3D Interactions. LaViola recognizes that though these devices work well enough for current gaming, future developments in these technologies must occur in order to keep up with computing and become useful in fields outside of gaming. (LaViola 2008) There are limitations in the current technologies that prevent the use of these technologies outside of gaming. The Sony Eye-Toy, for example, was built to detect hand gestures and does not measure a third dimensional space. We see similar limitations in the Nintendo Wii-mote, which will detect six degrees of freedom with conventional methodologies and handle only exaggerated gestures reliably.

Unlike commercial devices, the web camera offers a cheaper and often just as effective device for hand-computer interactions in gaming. The web camera is also a common device used in the research for bare-hand computer interaction. Because of this, we must look at research related to bare-hand interaction in relation to web cameras.

One often employed method of using the webcam as an input interface is through the use of estimation of motion vectors. This technique is proposed through the use of a block matching method. (Jain and Jain, 1981; Sun and Cheng, 2007) We also see similar techniques used in a three-step search (Li et al., 1994), four-step search (Po and Ma 1996), and a diamond search (Zhu and Ma, 1999). These techniques offer an effective method for tracking objects that move in front of the web camera, but have proven ineffective in many other situations. In these methods, a simple change in lighting or an object moving into the plane may cause miscalculations and ultimately lead to false positive motion tracking.

Another frequently used method for detecting hands and gestures is through the use of a form of glove with varying colors, a method introduced by Frederickson (Frederickson et. al, 2008). This technique creates hotspots on the hand by giving each finger a specified color. It also adds a marker to allow the computer to determine the x-, y-, and z- coordinates. This practice allows a user to use many different hand positions and gestures. The glove is also low cost as there is no special hardware inside of each glove. However, the use of this technique does not allow a user who is not able to put on a glove the use of this technique. For this reason, a bare-hand control interface would be best suited to allow accessible usage for the disabled to computing and gaming

There has been significant progress in bare-hand control interface research in the last few years. Lee and Lee as well as Hardenberg and Bérard have built bare-hand detection algorithms that can be used for tracking the motion of a users hand. Lee and Lee have built an algorithm that detects the hand and head using a color difference technique (Lee and Lee, 2008a; Lee and Lee 2008b). For this technique to work, Lee and Lee have assumed human skin color is always of a red tone. This, however, is not always true since human skin colors vary in tone and hues. This assumption will work for the majority of skin types but not with all skin types.

Hardenberg and Bérard propose a different method for the detection of a hand. They developed an algorithm to detect user fingers using the basic shape of the tip of a finger (Hardenberg and Bérard, 2001). They then use a motion vector tracking algorithm to determine which objects, in this case fingers, have moved and to what locations. This algorithm, however, is dependent on a user's ability to move fingers. Because of this constraint, the algorithm is not suited for all cases of accessibility, especially for those physically disabled with quadriplegia.

Due to lack of a full interface that put accessibility above other goals, it became necessary to create a new framework that met specific requirements for accessibility purposes. This framework would need to meet several basic requirements that would allow a physically disabled user the ability to use this interface. However, meeting these requirements would not make the interface usable by users. For this, several advanced requirements would need to be addressed in order to

3. FRAMEWORK ARCHITECTURE

The Mocoa (Motion Control Application) project is developed as an accessibility framework to enable the integration of user gesture and motion accessible applications. Mocoa is structured so that any application could be easily integrated into the system as a software module. Figure 1 shows the top-level architecture of the project.

The software contains three primary components, the web camera controller, the object detector component, and the user interface. The web camera controller takes

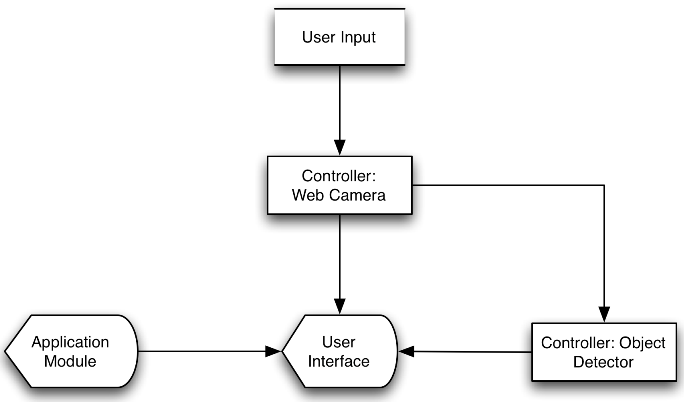


Figure 1: Basic Structure of Mocoa

in a captured frame as user input. This captured frame is then sent to both the User Interface, for display to the user, and the object detector component where the primary detection of the hand is completed. The User Interface then receives the information from the Object Detector. This information is sent between an application module and the User Interface. In order to ensure that the system is usable and accessible to users of all physical capabilities, the interface must meet several basic requirements, as well as suitably meet several advanced framework requirements.

3.1. Basic Framework Requirements

The basic framework requirements serve as the minimal requirements for making a system, which offers bare-hand interaction with a computer. Hardenberg and Bérard identify several different restraints for a vision-based human computer interaction system. These requirements consist of: Detection, Identification, and Tracking (Hardenberg and Bérard, 2001). Mocoa offers a unique method for object Detection and Tracking using a web camera as the input device.

3.1.1 Detection

Detection is necessary to determine the presence of an object within the frame of the captured image. Mocoa uses the Viola-Jones algorithm (Viola and Jones, 2001) for detecting the presence of various objects. This object detection algorithm is flexible enough to enable changes between control objects such as a head or a fist. This flexibility that requires only a brief calibration exercise offers great versatility and allows use by a diverse pool of users with different types of disabilities,

3.1.2 Identification

Object identification is an important factor for a hand motion controller. Without some form of identification, determining which object is the controlling object becomes difficult. This difficulty becomes more obvious as more objects are introduced into the field of view. There are other types of identification that are necessary for determining an object. In Mocoa, the following types are necessary:

- Hand posture Identification. Mocoa uses the hand posture of a sign language letter “A” to identify as an object within the frame. However, other object identifiers, such as a user's head or various other body parts, can serve as an identifier.
- The 2-dimensional location of the object. Mocoa uses this information to move a cursor or other virtual object based on the *x*- and *y*-coordinates of the real object.
- A third dimensional space. This offers a more Realistic experience with the computer by allowing the user to interact with a virtual object as they would with a realistic object.

Mocoa achieves identification through the use of the classifiers to differentiate between various objects in the image. Since a user’s hand is a 3-dimensional object, for this project we have limited the identification of the object to a sign-language letter "A" gesture. After this object is identified, it returns the central location of the detected object as *x*- and *y*- coordinates.

3.1.3. Tracking

Once the object is detected and identified, it has to be tracked to allow the “registering” of the user’s motion as a controlling motion in order to control various aspects of computing. Mocoa achieves a form of tracking by restarting the object detection and identification process during each captured frame. This helps to ensure that the stability of the input remains consistent throughout the remainder of the user’s input. This may have caused some issues related to performance, but the benefits of tracking an object in this way far exceeded the loss of performance.

3.2. Advanced Framework Requirements

The basic framework requirements ensure that the framework is able to accommodate basic vision-based human computer functionality. However, to make an application useable in a practical way, there are advanced requirements that need to be met. These advanced requirements consist of input stability, spatial resolution, and input latency.

3.2.1. Input Stability

Input instability manifests when ambient lighting, motion of other objects, or electrical noise become present in the user’s environment and interferes with Mocoa’s components. The result of any of these problems may cause erratic behavior of the controlled component (e.g. mouse pointer) or false positives in hand detection. If the input mechanism does not remain stable, oscillations could occur making the interface difficult to control. However, even if they occur, Mocoa is resilient enough to allow the instability to last only a few frames, except in extreme instances where an object may have a similar shape as the object or where lighting is not great enough to detect the hand.

3.2.2. Spatial Resolution

Spatial Resolution becomes necessary when the resolution of the captured frame is much smaller than resolution of the device that is being controlled. Hardenberg and Bérard suggest that for a point-and-click system, the smallest possible pointer movement must be no larger than the smallest selectable object on the screen. Mocoa handles spatial resolution through the use of its calibration tool. The calibration tool allows a user access to set up a maximum and minimum coordinate for movement. This information then translates movements into percentages, which can then used to control an application with variable pixel sizes.

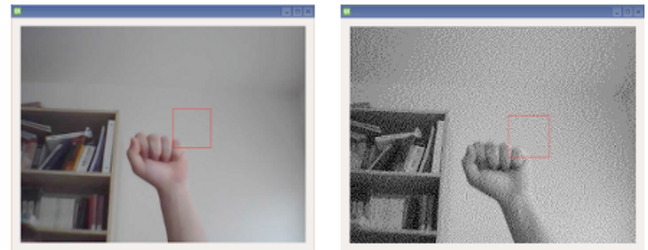
3.2.3. Input Latency

Input Latency is one of the most significant advanced framework requirements that have an impact on overall usability of the interface. High latency times from the input to the running module would significantly degrade the user's experience of the interface. For any form of input device, especially in gaming, low latency inputs offer players the ability to control the system with a more natural feel. Mocoa achieves low latency times through the use of the detection algorithm. The algorithm used achieves a high rate of success in a short amount of time. This, coupled with the stability of the tracking system, creates a low latency interface, which creates a very natural feel when controlling virtual objects.

4. HAND DETECTION

The algorithm used by Mocoa for hand detection and identification is the Viola-Jones algorithm (Viola and Jones, 2001). The Viola and Jones algorithm was designed primarily for face detection, but has proven to be just as effective in a search for other objects when the haar classifier files are changed. Unlike other algorithms (Lee and Lee, 2008; Lee and Lee, 2008b), the use of an object detection algorithm eliminates the reliance on the color or hue of the skin.

The Viola-Jones algorithm is an implementation of the AdaBoost machine-learning algorithm. The Viola Jones Algorithm works by taking areas of the image and scanning the image for various sum calculations based on an integral-image cascade classifier file. In our implementation, these classifier files are an XML based



(a) *(b)*

Figure 2: Hand Detection in both a color image (a) and a black and white monochrome image (b)

file that allows the Viola-Jones algorithm to determine if the object we are searching for is located within the portion of the image that is being scanned. This is repeated for all portions of the image.

The implementation of the Viola-Jones used by Mocoa does not force users into a specific hand gesture or require users to be physically able to use a specific body part by allowing a user to easily change the haar classifier file. Through the use of the Viola-Jones algorithm we can now also reliably detect an object under various lighting instances without the requirement of a color image. Figure 2 shows that by using the Viola-Jones algorithm, Mocoa is able to detect a hand in both a color and a black and white frame. The algorithm, also addresses several other issues that had been identified by Lee and Lee such as the issue of overlapping body parts; for example, when a users hand overlaps with their face in the Lee and Lee algorithm, the face is fallaciously identified as the controlling object along with the users hand

One of the primary issues with Lee and Lee's detection algorithm is that overlapping body parts (with the same skin color) will prevent a hand from being detected. The Viola-Jones algorithm implemented in Mocoa allows for the collision of various body parts without degrading the recognition of the object in the frame. This is achieved by using the Viola-Jones detection algorithm and the lack of need for identification using skin color. Figure 3 shows the detection algorithm when the hand and a user's face are close or overlapping.

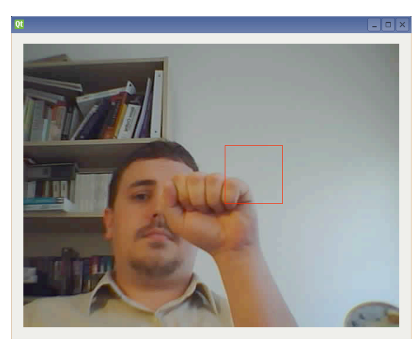
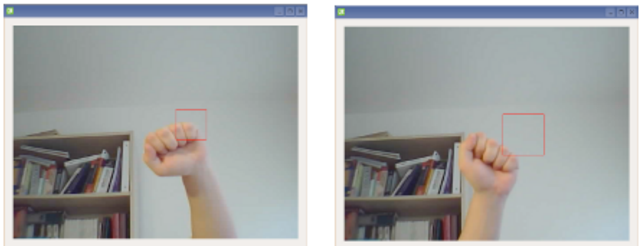


Figure 3: Overlapping Face and Hand, an issue seen in (Lee and Lee 2008a; Lee and Lee, 2008b), is resolved.



(a) *(b)*

Figure 4: Hand Detection with 45 degree turn to the left (a) and to right (b).

The Viola-Jones algorithm is able to quickly identify objects in the frame; however, simply using the Viola-Jones algorithm will not detect objects that have been rotated. Lienhardt and Maydt extended the Viola-Jones algorithm to allow the algorithm to detect objects that have been rotated up to 45 degrees from standard orientation (Lienhardt and Maydt, 2002). The Viola-Jones algorithm implemented by Mocoa uses the extension to the algorithm made by Lienhardt and Maydt. Figure 4 shows the object detected at approximately 45 degrees in both directions. By implementing this extension, we allow users the ability to naturally turn their hand as they move without causing loss of object detection.

The detection algorithm is able to determine the 2-dimensional location of the hand. Mocoa uses the central location of the hand to control aspects of computer. The detection algorithm used has been fitted to calculate the *x* and *y* coordinates of the bottom left corner of the box that wraps the hand when detection is successful as well as the height and width of the object. In using this point, as well as the height and the width of the box, we can calculate the *x*- and *y*- coordinates for the center of the box. Equations 1a and 1b show the central *x* and *y* coordinate.

$$f(x) = \frac{rect_x + rect_w}{2} \qquad f(y) = \frac{rect_y + rect_h}{2}$$

(a) *(b)*

Equation 1: Mathematical equation for calculating the central x (a) and central y (b) coordinates which are used for moving a virtual object.

Calculating a third dimensional space, which could be beneficial inside many applications, becomes a bigger challenge. However, simple mathematics can be used to determine a *z*-coordinate based on the size of the object detected with relation to a calibrated 0 point. Equation 2 is a mathematical representation of the algorithm used to calculate the *z*- coordinate in Mocoa, where h_c is the maximum calibrated height in Mocoa and h_0 is the current height of the detected object within the frame. "*sf*" signifies that the results of the equation should be floored to the number of significant figures given in the equation.

$$f(z) = \begin{cases} +z & if \left\lfloor \frac{h_c}{h_0} \right\rfloor_{sf=1} > 1 \\ 0 & if \left\lfloor \frac{h_c}{h_0} \right\rfloor_{sf=1} = 1 \\ -z*10 & if \left\lfloor \frac{h_c}{h_0} \right\rfloor_{sf=1} < 1 \end{cases}$$

$$where \quad z = \left\lfloor \frac{h_c}{h_o} \right\rfloor_{sf=1}$$

Equation 2: Mathematical representation of the z-coordinate algorithm used for calculating 3-dimensional space on the 2-dimensional framed based on relative sizes.

The information received through the calculations in three-dimensional spaces is then transmitted to requesting modules. However, the resolution of a web camera may be very small in comparison to the resolution of a module; for example, a web camera may only capture a 600 pixel by 600 pixel frame while the resolution of the operating system is likely to greatly exceed the frame resolution of the web camera, potentially 1600 pixel by 1200 pixel, or greater. Because of this, the coordinate information received from the object detector component is checked against the calibration tool to calculate a percentage value. This is then used to move a virtual object with relation to the maximum and minimum *x*- and *y*- coordinates.

5. PERFORMANCE ANALYSIS

Performance of an interface plays a major role in the usability and accessibility of an application. Poor performance may lead to higher latency times and a degradation of the overall stability of the system. Mocoa was initially designed as a single-threaded application. During development, Mocoa began to show signs of high computational needs, which ultimately caused issues with latency times and the overall usability of the interface. Because of this, Mocoa was later made multi-threaded to take advantage of the multi-core processors that are becoming more prevalent in home computers today. This performance analysis was conducted to determine the most suitable environment for this interface to exist.

To conduct this analysis, it was necessary to integrate a module that could be controlled using the Mocoa framework. For this, a simple version of one player Pong was integrated into the system. This module took advantage of each of the components inside of Mocoa so that the entire system could be analyzed for performance related reasons.

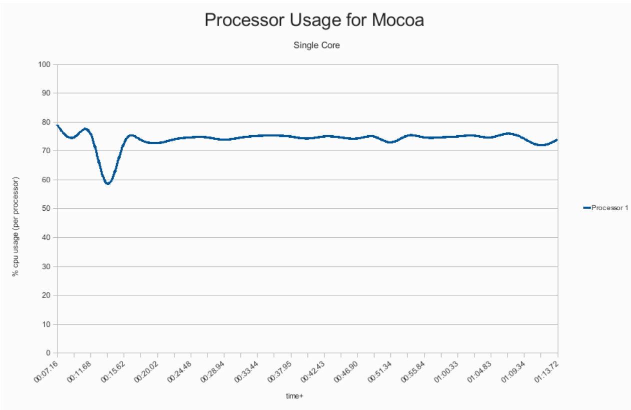


Figure 7: Single Core Processor Usage Performance Analysis

This performance analysis was conducted on a Linux machine running a standard installation of Ubuntu 8.10 64-bit version using a "Creative Live! Notebook Pro" web camera with the V4L2 drivers for Linux webcams. We run the analysis on a single-, dual-, and quad- core processor running at the same clock speed and cache per processor. Table 1 shows a more detailed description of the systems used for each of the system performance analyses.

	Computer 1	**Computer 2**	**Computer 3**
OS	Ubuntu 8.10	Ubuntu 8.10	Ubuntu 8.10
Clock Speed	2.66 GHz	2.66 GHz	2.66 GHz
Cache	2 MB	4 MB	8 MB
FSB	1066 MHz	1066 MHz	1066 MHz
Cores	1	2	4
RAM	4GB	4GB	4GB

Table 1: The hardware specifications used for the performance analysis of Mocoa

We began this analysis by measuring the average CPU usage over a minute of typical usage of the operating system. In all cases, the operating system ran at approximately 20% of the processing with a standard deviation of 1-2%. Knowing the processor overhead, we can begin to analyze all of the information received from our processing dumps for each core of the processor. For the analysis, the pong module was played for 1 minute while the processor usage of the application was analyzed.

Our first analysis was conducted on a single core system. Figure 5 shows a graphical representation of the results of this analysis on a single core processor. The user saw significant latency issues with both the game and user input as the processor attempted to calculate the user's input on a processor running at nearly its full capacity. We see that the average processing power necessary on a single core processor is approximately 74%. With the average Operating System overhead, we can conclude that the latency issues within the application were likely due to the processor running at nearly 100%.

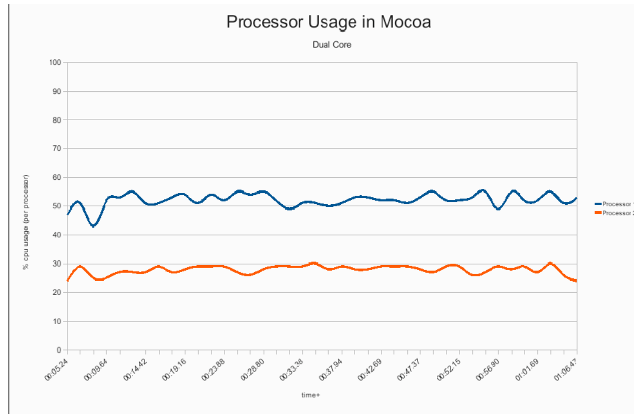


Figure 5: Dual Core Processor Usage Performance Analysis

After Multi-threading the interface, we see significant improvements in the latency times between hand movement and virtual interaction. While running on multiple cores, we see significant improvements in the processing needs for the application. Figure 6 graphs the processor usage of Mocoa on a dual core processor. We can see that on a dual core processor, Mocoa uses just over 50% of the first processor while secondary necessities require 27% of the second processor. This results in a much lower latency time, allowing the user to more naturally play a game or control other aspects of computing.

The quad core analysis shows insignificant changes in the processing needs of the application. The first core remains unchanged from the dual Core analysis, running at just over 50% of the processing needs. Each subsequent core ran at 19% of the processor cores. No latency changes were noticed between runs on a quad core and a dual core.

Through these analyses, we can assert that it is becoming more feasible to implement a Bare-Hand computer input interface, such as Mocoa, which requires high processing power, in many aspects of computing, including home computing and gaming.

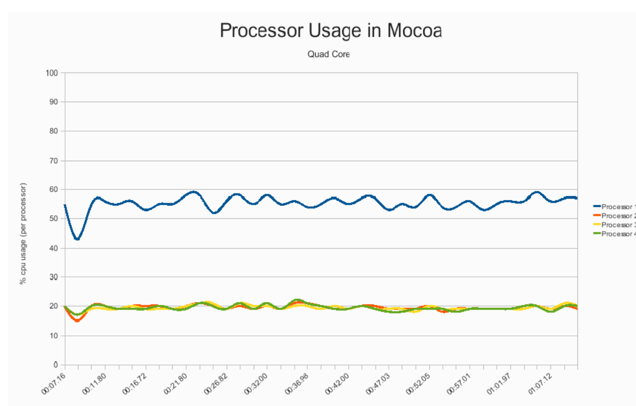


Figure 6: Quad Core Processor Usage Performance Analysis

6. CONCLUSION & FUTURE WORK

Mocoa serves, as a tool that shows that computing today is capable for handling an interface that can be used by a variety of users, including those with physical disabilities. One idea for future development of Mocoa is to integrate this technology in specialty devices that are used by patients with certain disabilities such as quadriplegia.

In certain forms of quadriplegia, the disabled have limited control over the movement of their hand, but little to no control over the movement of fingers or the strength at which they hit the various input devices. Because of this, keyboards and other input devices that are frequently touched need to be replaced often.

Mocoa offers a unique solution for hand-computer interaction controls. It resolves many of the issues that are found in various other algorithms and successfully addresses issues that were present with different systems related to user characteristics such as skin color or saturation. Still, Mocoa has its own flaw: the problem of false positives, due to surrounding image noise, which remains unresolved. However, we are optimistic that existing techniques can be used to reduce the noise and address this current shortcoming.

With the success of net-book computers and with mobile devices, such as cell phones, becoming more able to handle complex computing, it would also be beneficial to determine where improvements in the implementation of Mocoa and its object detection algorithm so that this interface might be used with these types of low resource devices.

Overall, Mocoa shows promise in the deployment of hand-based computer interaction, but more work remains to be done to take what currently is a useable proof-of-concept to a more comprehensive and feature-full framework.

ACKNOWLEDGEMENTS

We are grateful to Mr. James Devilbiss for several of his ideas with respect to accessible computing applications for this project.

REFERENCES

Fredriksson, J., Ryen, S. B., and Fjeld, M. 2008. Real-time 3D hand-computer interaction: optimization and complexity reduction. In Proceedings of the 5th Nordic Conference on Human-Computer interaction: Building Bridges (Lund, Sweden, October 20 - 22, 2008). NordiCHI '08, vol. 358. ACM, New York, NY, 133-141.

Grammenos, D., Savidis, A., and Stephanidis, C. 2009. “Designing universally accessible games.” Comput. Entertain. 7, 1 (Feb. 2009), 1-29.

Hardenberg, C. von and Bérard, F. 2001. “Bare-hand human-computer interaction.” In Proceedings of the 2001 Workshop on Perceptive User interfaces (Orlando, Florida, November 15 - 16, 2001). PUI '01, vol. 15. ACM, New York, NY, 1-8.

Jain, J.; Jain, A.; 1981. “Displacement Measurement and Its Application in Interframe Image Coding” In Transactions on Communications, IEEE (Dec. 1981): 1799-1808

LaViola, J.J., 2008. “Bringing VR and Spatial 3D Interaction to the Masses through Video Games.” Computer Graphics and Applications, IEEE. Vol.28, no. 5 (Sept.-Oct.): 10-15.

Lee, Dae Ho and Young Jae Lee; 2008a. “Sensing and Motion Control of Virtual Objects for Web Camera-Based Game.” In Proceedings of the Second Conference on Future Generation Communication and Networking. (Sanya, China, Dec. 13-15) FGCN: 28-33.

Lee, Young Jae and Dae Ho Lee, 2008b. “Research on Detecting Face and Hands for Motion-Based Game Using Web Camera.” In Proceedings of the 2008 International Conference on Security Technology. (Sanya, China, Dec. 13-15). SECTECH: 7-12.

Li, R, B. Zeng, and M.L. Liou, 1994. “A new three-step search algorithm for block motion estimation.” In Transactions on Circuits and Systems for Video Technology, IEEE (4), 438-442.

Lienhart, R.; Maydt, J., 2002 “An extended set of Haar-like features for rapid object detection.” In Proceedings of the 2002 International Conference on Image Processing. (Rochester, NY, Sept. 22-25). IEEE: I-900-I903.

Po, L.M; Ma, W.C. “A novel four-step search algorithm for fast block motion estimation” In Transactions on Circuits and Systems for Video Technology, IEEE (June 1996): 313-317.

Sun, Huey-Min and Wen-Lin Cheng; 2007. “The Input-Interface of Webcam Applied in the Digital Content Interaction” In Proceedings of the Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing. (Splendor Kaohsiung, Taiwan, Nov. 26-28). IIH-MSP: 2007.

United Nations, 2004. World programme of action concerning disabled persons: Current Situation. <http://www.un.org/esa/socdev/enable/diswpa04.htm>

Viola, P.; Jones, M., 2001. “Rapid object detection using a boosted cascade of simple features” In Proceedings of the 2001 Conference on Computer Vision and Pattern Recognition. (Kauai, Hawaii, Dec. 9-14). CVPR: I-511-I-518.

World Health Organization, 2004. Department of Chronic Diseases and Health Promotion (CHP): About aging and life course. <http://www.who.int/ageing/en/>

Zhu, Shan; Ma K.K., 2000 “A new diamond search algorithm for fast block-matching motion estimation.” In Transactions on Image Processing, IEEE (Feb 2000): 287-290