Hands-Free Mouse-Pointer Manipulation Using Motion-Tracking and Speech Recognition

Frank Loewenich
Faculty of Information Technology,
Queensland University of Technology
2 George Street, GPO Box 2434,
Brisbane Q 4001, Australia.

f.loewenich@qut.edu.au

Frederic Maire
Faculty of Information Technology,
Queensland University of Technology
2 George Street, GPO Box 2434,
Brisbane Q 4001, Australia.

f.maire@qut.edu.au

ABSTRACT
Technology is advancing at a rapid pace, automating many everyday chores in the process. Information technology (IT) is changing the way we perform work and providing society with a multitude of entertainment options. Unfortunately, in the past designers of many software systems have not considered the disabled as active users of technology, and thus this significant part of the world population has often been neglected. A change in this mindset has been emerging in recent years, however, as private-sector organizations and governments have started to realize that including this user group is not only profitable, but also beneficial to society as a whole. This paper introduces an alternative method to the traditional mouse input device, using a modified Lucas-Kanade optical-flow algorithm for tracking head movements, and speech recognition to activate mouse buttons.

Categories and Subject Descriptors
I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis – *object recognition, tracking, motion*

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *auditory, graphical user interfaces (GUI), input devices and strategies, interaction styles, natural language, voice I/O*

General Terms
Algorithms, Human Factors

Keywords
graphical user interface, auditory interaction, multi-modal interfaces, accessibility, human-computer interaction, audio, rehabilitation engineering, users with special needs, disability

1. INTRODUCTION
It is highly desirable to provide the disabled with an easy way to access a standard PC that does not require specialized equipment. This system should in particular allow the easy navigation of the graphical user interface in a similar fashion experienced by non-disabled users.

In this paper we introduce a graphical user interface navigation utility, similar in functionality to the traditional mouse pointing device for the Microsoft Windows operating system.

Key factors driving development in the area of accessibility for users with disabilities are demographic changes and the ageing population, legislation introduced by government, including the United States of America (USA) and the European Union (EU), as well as a growing awareness in society of the increasing diversity of people requiring access to information technology.

Our implementation of an alternative mouse input device is based on head-tracking technology using a modified Lucas-Kanade optical flow algorithm [12] and speech recognition technology. Using a pyramidal implementation of the Lucas-Kanade algorithm, as described by Vámossy, Tóth and Hirschberg in 2004 [27], allows our system to avoid any specialized, expensive hardware components. Our approach instead relies on a common web camera and microphone, making the technology easily available to all computer users at a low cost.

The remainder of this paper is structured as follows. Section 2 discusses available assistive technologies for users with a disability. Section 3 contains an overview of related work and our proposed system is described in Section 4. Section 5 provides a description of our system implementation. Sections 6 and 7 provide in-depth explanations of the face detection and feature tracking algorithms, respectively, including descriptions of the Haar Classifier Cascade and Lucas-Kanade algorithms. Section 8 contains a description of our approach to using head-tracking to control mouse-pointer movements and Section 9 describes how speech recognition is used to activate mouse buttons. Section 10 contains experimental results and Section 11 touches on future work. Section 12 concludes the paper.

2. ASSISTIVE TECHNOLOGIES
Recognition of the need for technologies and guidelines for providing accessibility to electronic resources for users with disabilities has been increasing in recent years. A majority of persons with disabilities can now lead more independent lives in their communities, attend regular schools, and seek professional

OzCHI 2007 Proceedings, ISBN 978-1-59593-872-5

careers more than ever before in history. Assistive technology providers are changing their focus from people with disabilities as requiring treatment and intervention, to a view of the person with a disability and the minimization of obstacles to living in the community and participating in the workforce. Assistive technologies have been an important key to successful community participation. However, the rate of assistive technology non-use, abandonment and discontinuance remains high - the average being about 1/3 of all devices provided to consumers [20, 21, 22, 23, 24, 30]. According to responses by disabled computer users taking part in a recent study by the Joseph Rowntree Foundation in 2004 [9], who did have aids or equipment adaptations to access a computer at their disposal, almost half were experiencing problems using them. Others did not have the aids available to them, which they felt they needed, and still others did not know about the options available to them to effectively access a computer and make use of the range of services and information accessible through electronic means.

A significant problem faced by developers of assistive technology is the difficulty in adapting their product to every individual user. ABLEDATA (2005), the assistive technology product database sponsored by the Institute on Disability and Rehabilitation Research, U.S. Department of Education [26], lists more than 22,000 current products from over 2,000 different companies for users with disabilities. However, the vast majority (>95%) of products listed are specialized hardware devices aimed at specific disabilities. These are expensive, often hard to handle by the disabled person and only available from specialized vendors. Software solutions comprise less than 5% of total available products and are typically aimed at only a limited number of disabilities, usually screen readers for the blind or visually impaired and learning solutions for intellectually disabled computer users.

The price of assistive technology is a hurdle to many potential users. Ramstein *et al.* [18] reports that assistive technologies enabling human-computer interaction for users with a disability often require some additional hardware, which is either worn or manually operated by the user. This additional requirement adds expense and inconvenience for the user. According to the U.S. Department of Education [26], specially adapted hardware and software for disabled computer users has always been expensive and unfortunately this trend is continuing, if not worsening.

Our goal has been to develop a non-intrusive, reliable and inexpensive system that adds to a disabled person's independence and is inherently adaptable to many different circumstances. To this end we have developed a robust visual tracking system that uses a modified Lucas-Kanade optical flow algorithm to track head movements, without needing to locate any specific features of the face. The proposed system will allow persons, who may have disabilities ranging from not being able to use their hands to severe cases where the person is only able to move their head, to navigate and manipulate the graphical user interface of the Microsoft Windows operating system using head movements and speech.

A head-tracking prototype was outlined in work previously published by the authors in [11]. This prototype system has since undergone significant changes and has now reached a high level of maturity, warranting its implementation as a reliable alternative to traditional mouse pointing devices and introduction to the public domain.

3. Related Work

There are various commercial mouse alternatives available today. NaturalPoint [16] markets several head-tracking based mouse alternatives on their web site. While the benefits are real, these devices still require the user to attach markers either to the head or glasses. Other systems use infrared emitters that are attached to the user's glasses, head-band, or cap. Some systems, for example the Quick Glance system by EyeTech Digital Systems [7], place the transmitter over the monitor and use an infrared-reflector that is attached to the user's forehead or glasses. Mouse clicks are generated with a physical switch or a software interface.

For the severely disabled, the need for markers or reflectors poses the need for a third party to attach these to either a pair of glasses or the head, thus representing an inconvenience and forcing reliance on a third party and taking away from the individual's feeling of independence. Another drawback of commercially available devices is their price, which is usually in the range of ~US$200. Accessories are sold separately and attract an additional expense. Furthermore, few of the commercially available solutions include the mouse-clicking component, forcing the user to buy another component to create a useful interface.

Research is continually improving the options available to computer users with disabilities. Evans *et. al.* [6] recently described a head-mounted infrared-emitting control system that is a 'relative' pointing device and acts like a joystick rather than a mouse. Chen *et. al.* [3] developed a system that contains an infrared transmitter, mounted on to the user's eyeglasses, a set of infrared receiving modules that substitute the keys of a computer keyboard, and a tongue-touch panel to activate the infrared beam. In [1] the authors describe a system for translating a user's motions to mouse movements. Their tracking algorithm relies on detecting specific features such as the eyes or nose to follow head movements across multiple frames.

4. The Proposed System

The tracking of head movements with a web camera provides a simple solution to the previously mentioned problems. The proposed system utilizes off-the-shelf hardware and software components to make the system easily available to the greatest possible number of users. To further reduce implementation costs we have chosen freely available software development components. The .NET Framework was chosen as a development platform, as it offers a good level of interoperability with Microsoft Windows operating system components [13]. We also rely on the OpenCV library from Intel Corporation [8] for feature detection and tracking.

5. System Implementation

Our system uses a static web camera. Identification of the head is achieved using a Haar Classifier Cascade algorithm [19, 20]. This algorithm has the ability to detect faces in an image. Once the user's face has been detected, position and size of a rectangular area encompassing the face is extracted. This is used by the Lucas-Kanade optical-flow algorithm to determine significant features of the face suitable for tracking, which are highlighted in the frame using green dots (see Figure 5). Tracking may be initiated after a calibration routine, which requires the user to look at a point at the center of the computer screen. Once tracking has been initiated (see Figure 6), significant features, as determined by the Lucas-Kanade algorithm, are locked into place (these points are subject to change prior to tracking activation due to subject movements and changes in lighting conditions).

In our prototype system variable points are marked with a green dot, while locked-in points are marked with a red dot. The tracking component compares red-dot coordinates from the first frame to red-dot coordinated in subsequent frames to determine the extent to which head movements are translated to pointer movements on-screen.

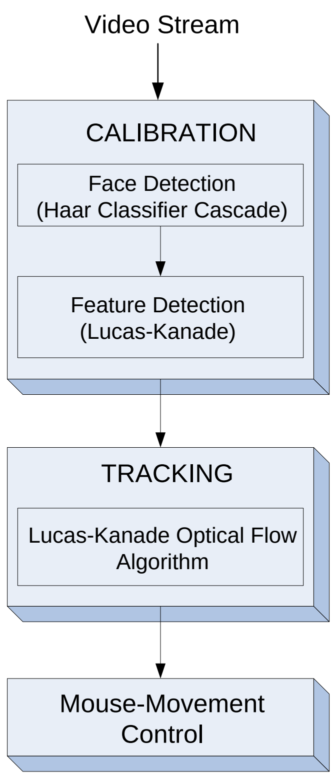

Figure 1. The face-detection and tracking components of the system.

5.1 The .NET Framework

The .NET Framework was chosen for the implementation of our project as it facilitates rapid application development and access to all layers of the operating system and hardware devices.

Developed by Microsoft, the .NET Framework is an add-on run-time environment for Windows operating systems. Recent work by the 'Mono Project' and 'Portable .NET'-project has contributed to making .NET code truly portable. Developers can now develop for Windows, and expect their .NET application to run on Apple and Linux operating systems [5].

The .NET Framework, provides developers with helpful run-time features, such as automatic memory management (garbage collection) and easier access to system services. It also adds many utility features such as easy Internet and database access, and facilitates easier code reuse [13, 17].

A challenge remains crossing the divide between managed (.NET) and unmanaged code. However, facilities do exist and these have been employed extensively in this project.

5.2 The Open Source Computer Vision Library (OpenCV)

The open source computer vision library is a development library written in the C/C++ programming language. It includes more than 300 functions, ranging from basic image processing routines all the way up to state-of-the-art computer vision operations. The library has found use in many application areas, including human-computer interaction, object detection, segmentation and recognition, face recognition, gesture recognition, motion tracking, ego motion, motion understanding, structure from motion and mobile robotics.

A major obstacle encountered during the implementation of our system was the integration of the OpenCV library (unmanaged code) into a managed (.NET) project. Unfortunately, direct integration of the OpenCV library was not possible. OpenCV lacks the object-oriented approach and organization of C# and the .NET Framework. Also, the two feature an incompatible model for exception handling, making a complete integration almost impossible. To address this problem, we created a wrapper for the subset of required OpenCV library functions.

6. Face Detection using a Haar Classifier Cascade

The face detection component is fundamental to the function of our head-tracking system and is based on the Haar-Classifier cascade algorithm. This algorithm was first introduced by Viola and Jones [19, 20] in 2001. It offers a robust framework for rapid visual detection of frontal faces in grey scale images. The algorithm is appearance based and uses a boosted cascade of simple feature classifiers. This section contains the fundamentals of the Viola and Jones algorithm and is included to provide a more thorough understanding of our system.

6.1 Challenges of Face Detection

The process of face detection is complicated by a number of factors. First, variations in pose (frontal, 45-degree, profile) result in the partial or full occlusion of facial features. Beards, moustaches and glasses also hide features. Facial expressions directly affect the appearance of a persons face, and to the computer, the same person first showing a happy face followed by a sad face can result in the classification of this same individual as two different people. Another problem is partial occlusion by other objects. For example, in a group of people some faces may be partially covered by other faces. Finally, the quality of the captured image needs consideration. Face images vary for different rotations about the camera's optical axis and when images are formed, factors such as lighting (intensity, spectra and source distribution) and camera characteristics (sensor and lenses) also affect the appearance of a face.

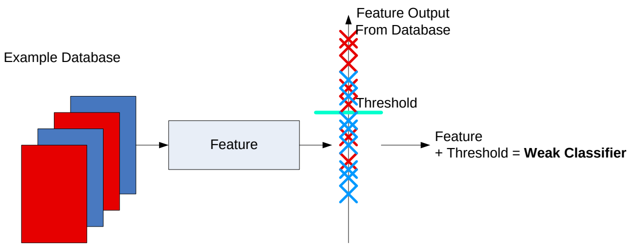

Figure 2. Weak classifier. The weak classifier provides a result only slightly better than a random guess.

6.2 Boosting-based Face Detection

In AdaBoost-based classification, a highly complex, non-linear classifier is constructed as a linear combination of many simpler, easily constructible weak classifiers. Each weak classifier, which on its own delivers a result only slightly better than a random guess, is built by applying a threshold to a scalar feature selected from an over-complete set of Haar wavelet-like features.

6.2.1 Haar-like Features

The four basic types of scalar features proposed by Viola and Jones for the purpose of face detection are shown in Figure 3. Each of these features has a scalar value that can be computed efficiently from the integral image, or summed area table. This set of features has recently been extended to deal with head rotation [10].

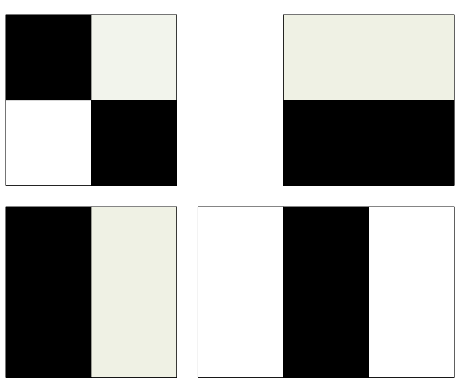

Figure 3. Rectangular Haar wavelet like features. A feature takes a scalar value by summing up the white region and subtracting the black region.

6.2.2 Constructing a Strong Classifier

Boosting is a method to combine a collection of weak classifiers (weak learners) to form a stronger classifier. AdaBoost is an adaptive algorithm to boost a sequence of classifiers, in that the weights are updated dynamically according to the errors in previous learning cycles. The algorithm employed by Viola & Jones [28] has a face detection cascade of 38 stages with 6000 features. According to Viola & Jones, the algorithm nevertheless achieved fast average detection times. On a difficult dataset, which contained 507 faces and 75 million sub-windows, faces are detected using an average of 10 feature evaluations per sub-window. As a comparison, Viola & Jones argue that their system is 15 times faster than a detection system implemented by Rowley et al. [19].

6.2.2.1 Theoretical Foundation

A weak classifier $h_j(x)$, consisting of a simple feature $f_j x$, a threshold θ_j, and a parity p_j indicating the direction of the inequality sign, produces a binary decision:

$$h_j(x) = \begin{cases} 1 \ \ if \ \ p_j f_j(x) < p_j \theta_j \\ 0 \ \ otherwise \end{cases}$$

The final strong classifier is $h(x)$ of the following form:

$$h(x) = \begin{cases} 1 \ \ if \ \ g(x) = \sum_{j=1}^{J} \alpha_j h_j(x) \geq \frac{1}{2} \sum_{j=1}^{J} \alpha_j \\ 0 \ \ otherwise \end{cases}$$

An illustration of a strong classifier, consisting of a cascade of weak classifiers is featured in Figure 4.

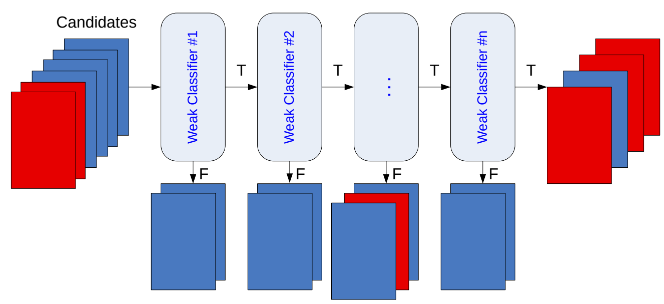

Figure 4. A strong classifier, where weak classifiers are arranged in a cascade.

7. Feature Tracking Algorithm

The optical tracking component uses an implementation of the Lucas-Kanade optical flow algorithm [12], which first identifies and then tracks features in an image. These features are pixels whose spatial gradient matrices have a large enough minimum eigenvalue.

When applied to image registration, the Lucas-Kanade method is usually carried out in a coarse-to-fine iterative manner, in such a way that the spatial derivatives are first computed at a coarse scale in scale-space (or a pyramid), one of the images is warped by the computed deformation, and iterative updates are then computed at successively finer scales.

One of the characteristics of the Lucas-Kanade algorithm, compared to the method proposed by Camus [2], is that it does not yield a very high density of flow vectors. This means that the flow information fades out quickly across motion boundaries. This presents the disadvantage that the inner parts of large homogenous areas show little motion. However, its advantage is the comparative robustness in the presence of noise.

For completeness, the following section explains how the optical flow is derived. It also provides an insight of the potential of the algorithm to serve as a feature tracker for users with disabilities

7.1 Feature Point Detection

The term “feature point” denotes a point in an image that is sufficiently different from its neighbors, such as an L-corner, T-junction, or a white dot on black background. The Lucas-Kanade algorithm detects feature points based on a matrix related to the autocorrelation function. This matrix averages derivatives of the signal in a window around a point (x, y) by capturing the structure of the neighborhood.

If this matrix is of rank two, an interest point or feature, is detected. A matrix of rank one indicates an edge and a matrix of rank zero a homogenous region (see [28] or [29] for a more detailed explanation).

Our algorithm restricts the feature points detected to the face region previously identified. Feature points are marked with a green dot (Figure 5).

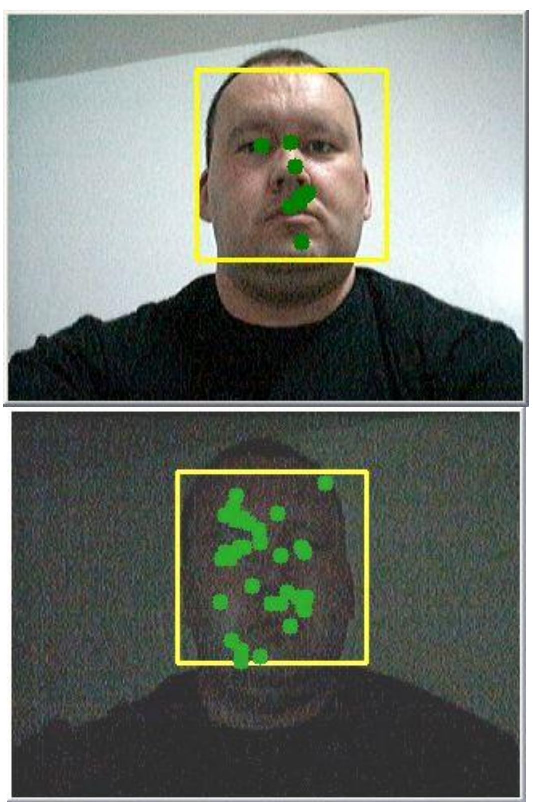

Figure 5. The face is detected using the Haar-Classifier cascade algorithm and marked with a yellow square. Green dots mark significant features identified by the Lucas-Kanade algorithm. The bottom picture demonstrates good performance even in poor lighting conditions.

7.2 Optical Flow

Our system uses the optical flow between two subsequent frames to track the user’s head movement, which is translated to on-screen movement by the mouse pointer. This section provides a brief explanation of the optical flow.

The term ‘optical flow’ refers to a visual phenomenon that is experienced as an individual moves through the world. For example, if one is sitting in a moving vehicle and looking out the window, there are trees, the ground, buildings, etc., that appear to be moving backwards. This motion is optical flow. Optical flow can also be used to indicate the proximity to the different objects in view. Distant objects like clouds and mountains move so slowly they appear still. The objects that are closer, such as buildings and trees, appear to move backwards. The closer the objects, the faster they appear to be moving.

Figure 6. Feature extraction using modified Lucas-Kanade algorithm. The algorithm first identifies regions of interest in the image (subject to constraints) and tracks these regions in subsequent frames

In more technical terms, optical flow is the 2D motion field, for each point (x, y) in an image. It should be noted that that the optical flow does not directly represent the real 3D motion of the object, but only gives the projection of the motion in 2D. Optical flow vectors along with a camera model assumption can be used to estimate the 3D motion and depth of objects, which is also called "Structure from Motion". Given a set of corresponding points in two images, the optical flow at these points is given by the displacement vector between these points (see [12] for a more detailed explanation).

8. Controlling Mouse-Pointer Movements

In our proposed system two operational modes are available for translating head movements to on-screen mouse pointer movements. Selection of the desired mode may be accomplished in real-time by issuing the relevant voice command.

8.1 Relative Mode

This mode simulates the joystick control of a mouse pointer. If the system detects deviation of the tracking points from their original position above a certain threshold, the mouse pointer is moved in the given direction by a single pixel. Movement in this direction continues as long as the deviation of the tracking points is maintained. However, the rate of pixels being moved is steadily increased relative to the amount of time elapsed since the movement was initiated. Should the movement be interrupted, the rate of movement is reset to a single pixel. This mode is especially useful where fine control of the mouse pointer is required, and where navigation of the whole screen is still necessary. This makes the system ideal for disabled users who have to make precise on-screen movement, such as artists or engineers.

8.2 Absolute Mode

This closely resembles mouse pointer control associated with mouse hardware devices. In this mode, the distance of the tracking points from their original location is translated to the location of the mouse pointer from the center of the screen. The distance the mouse pointer will move away from the center of the screen depends on the resolution setting reported by the operating system.

9. Speech Recognition Technology

For humans, speech represents the most natural way to communicate, and human-computer interaction is no exception. If an application can be controlled by voice commands, its features can be opened up to users otherwise unable to use them. Although the idea of using speech recognition as a human-computer interaction method is not new, there is still a distinct lack of speech interaction in today's software market. It can only be of advantage to software vendors to produce more applications that feature a speech interaction component, either as an extension to existing functions, or as an alternative input method.

Speech recognition technology has progressed to a level, where users can reliably control certain actions on the computer using voice commands. The latest release of the Microsoft Windows operating system, Windows Vista, features much improved speech recognition capabilities compared to its predecessor [14]. Using speech recognition in the operating system, the user can dictate emails and documents, and use voice to control programs and navigate Web sites.

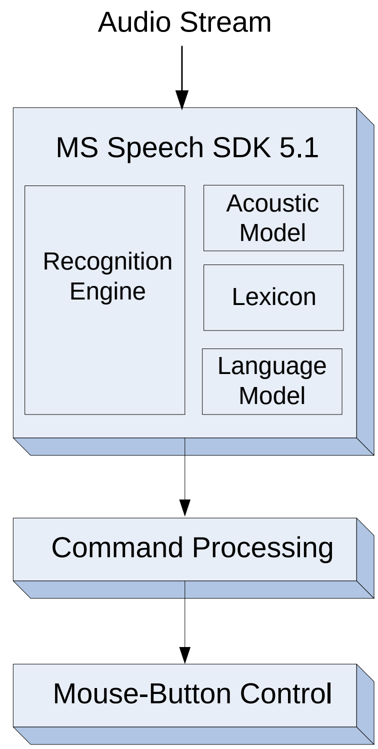

Figure 7. The speech-recognition components of our head-tracking system.

9.1 Using Speech to Activate Mouse Buttons

The proposed system utilizes the Microsoft Speech SDK 5.1, which is available as a free download from the Microsoft website. This SDK enables the addition of speech capabilities to Microsoft Windows-based applications. The Speech SDK can be used in a variety of programming languages, including C#, C++, VB, or any other COM-compliant language. In Figure 7 we provide an overview of the speech-recognition component.

By issuing the relevant voice command, the user may at any time perform common tasks, such as click-drag and drag-drop operations. The need for double-clicking, for example, represents a great challenge for people with reduced dexterity and motor control in their hands. Our system not only enables the user to execute single- or double-clicks, but also more complex operations. For example, a click-drag operation can be initiated with a vocal command or a file may be picked up and dropped on another folder by issuing a simple sequence of two commands (pick-up and release) in combination with head movements (carrying the object to another location).

Voice commands used to activate buttons are:

- “Click” – single left click
- “Double – double left click
- “Right” – single right click
- “Hold” – single left click and hold (click-drag)
- “Drop” – release the left mouse button after performing a click-drag operation

The number of commands is limited, providing good recognition accuracy, and the commands themselves may be changed to suit the individual user. There is scope to extend the set of commands for devices offering more buttons.

9.2 Alternatives to Speech Recognition

For users with speech impairment, the speech component of the system could be disabled. Instead, a suitable switch device could be employed in combination with the head-tracking component.

10. Experimental Results

In preliminary trials, the effectiveness of our system was tested with a group of 10 volunteers. Each of the subjects had previously used a computer, and was thus able to comment on how our system relates to using a traditional mouse pointing device. Each user was given a brief tutorial on how to use the system, and then allowed to practice moving the cursor for one minute. After the practice period, each user was asked to play a video game (Solitaire). The one minute training time was perceived as sufficient to introduce the features of the system. Users preferred to learn on-the-job, while using our mouse-replacement system to play a computer game.

Experiments conducted with the user group provided positive feedback. Of special mention was the simple calibration and setup of our system, which compared to other currently available systems, no longer requires the user to attach or wear markers of any kind on their head of face. This eliminates the need for another person to be present when setting up the system, adding to the autonomy of disabled users.

11. Future Work

A future implementation of our system could further improve the speech recognition component. Allowing the user, for example, to open applications using vocal commands would eliminate much navigating through menu structures.

Although the system has been developed and tested on the Microsoft Windows operating system, exploring implementations for Apple and Linux/Unix operating systems could be an aim for the future. Especially deployment on an open-source platform would open the technology up to a much wider user base.

The system may also prove useful for other applications, for example, where it is necessary to activate controls on a computer interface while at the same time performing precision work using both hands. Another potential area could be computer games. Furthermore, the modular architecture of the system allows for ready integration in any number of software projects requiring a low-cost and reliable head-tracking component.

12. Conclusion

Our implementation for an alternative to the traditional mouse pointer provides a low-cost means of human-computer interaction to users with a disability who are normally inhibited in their use of computers due to their inability to operate a physical mouse device.

The proposed system has proven to be robust, being able to tolerate strong variations in lighting conditions, and a built-in capability to exclude distractions, such as busy backgrounds.

13. REFERENCES

[1] Atyabi, M., Hosseini, M. S. K., & Mokhtari, M. *The Webcam Mouse: Visual 3D Tracking of Body Features to Provide Computer Access for People with Severe Disabilities.* Paper presented at the Annual India Conference, Bangalore, India, 2006.

[2] Camus, T., A.. Real-time optical flow. Technical Report CS-94-36. The University of Rochester. Rochester, New York, 1995.

[3] Chen Y. L., Tang F. T., Chang W. H., Wong M. K., Shih Y. Y., and Kuo T. S., “The new design of an infrared-controlled human-computer interface for the disabled,” *IEEE Trans. Rehab. Eng.*, vol. 7, pp. 474–481, Dec. 1999.

[4] Cole, R., Mariani, J., Uszkoreit, H., Varile, G., B., Zaenen, A., & Zampolli, A.. *Survey of the State of the Art in Human Language Technology*, 1998.

[5] Easton, M. J. & King, J.. *Cross-Platform .NET Development: Using Mono, Portable .NET, and Microsoft .NET*. Apress Publishers, 2004.

[6] Evans, D. G., Drew, R., & Blenkhorn, P. Controlling mouse pointer position using an infrared head-operated joystick. *Rehabilitation Engineering, IEEE Transactions on [see also IEEE Trans. on Neural Systems and Rehabilitation], 8*(1), 107-117, 2000.

[7] EyeTech Digital Systems Inc.. On-line Product Catalog, 2006. Retrieved May 29 from http://www.eitetechds.com/products.htm

[8] Intel Corporation. Open Source Computer Vision Library Reference Manual. Retrieved July 14, 2005, from http://developer.intel.com, 2001.

[9] Joseph Rowntree Foundation. Findings: Does the Internet open up opportunities for disabled people?. Retrieved May 24, 2005, from http://www.jrf.org.uk/knowledge/findings/socialcare/pdf/524.pdf, 2004.

[10] Lienhart, R. & Maydt, J.. *An extended set of Haar-like features for rapid object detection*. Paper presented at the International Conference on Image Processing (ICIP), Rochester, New York, USA, 2002.

[11] Loewenich, F., & Maire, F. A Head-Tracker Based on the Lucas-Kanade Optical Flow Algorithm. In Y. Li, M. Looi & N. Zhong (Eds.), *Advances in Intelligent IT - Active Media Technology 2006* (Vol. 138, pp. 25-30): IOS Press, 2006.

[12] Lucas, B. D., & Kanade, T.. *An Iterative Image Registration Technique with an Application to Stereo Vision.* Paper presented at the Proceedings of the 7th International Joint Conference on Artificial Intelligence, Vancouver, 1981.

[13] Microsoft Corporation. (2005). .Net Framework Developer’s Guide. [Electronic version]. Microsoft Developer Network (MSDN), January 2005.

[14] Microsoft Corporation. (2007). Windows Speech Recognition. Retrieved 20th June, 2007, from http://www.microsoft.com/windows/products/windowsvista/features/details/speechrecognition.mspx

[15] Mynatt, E. D., and Edwards, W. K.. Mapping GUIs to auditory interfaces. [Electronic version]. Proceedings of the 5th annual ACM symposium on User interface software and technology (pp. 61-70). California: Symposium on User Interface Software and Technology, 1992.

[16] NaturalPoint Inc.. On-line Product Catalog, 2006. Retrieved May 29 from http://www.naturalpoint.com/trackir/02-products/product-TrackIR-4-PRO.html

[17] Platt, D. S.. *Introducing Microsoft .NET.* Microsoft Press, 2001.

[18] Ramstein, C., Martial, O., Dufresne, A., Carignan, M., Chasse, P., and Mabilleau, P.. Touching and hearing GUI’s: design issues for the PC-Access system. [Electronic version]. In Proceedings of the second annual ACM conference on Assistive technologies (pp. 2-9). Vancouver: ACM SIGCAPH Conference on Assistive Technologies, 1996.

[19] Rowley, H. A., Baluja, S., & Kanade, T. (1998). Neural network-based face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 20(1)*, 23-38.

[20] Scherer, M. J.. Living in the State of Stuck: How Technology Impacts the Lives of People with Disabilities, Third Edition. Cambridge, MA, 2000: Retrieved June 7, 2005, from http://www.brooklinebooks.com/disabilities/disindex.htm

[21] Scherer, M.J.. Matching Person & Technology Model and Accompanying Assessment Instruments. Webster, NY, 1998: Retrieved June 7, 2005, from http://members.aol.com/impt97/ mpt.html

[22] Scherer, M.J. (Ed.).. Assistive Technology: Matching Device and Consumer for Successful Rehabilitation. Washington, DC, 2002: Retrieved June 7, 2005, from http://www.apa.org/books/ 431667a.html

[23] Stephanidis C.. User interfaces for all—concepts, methods and tools. Mahwah, NJ: Lawrence Erlbaum, 2001.

[24] Taylor and Francis Group.. Special Issue on Assistive Technology: Disability & Rehabilitation, 2002. Retrieved June 7, 2005, from http://www.tandf.co.uk/

[25] Tominaga, H., and Yonekura, T.. A Proposal of an Auditory Interface for the Virtual Three-Dimensional Space. [Electronic version]. Systems and Computers in Japan, 30(11). 77-84, 1999.

[26] U.S. Department of Education. ABLEDATA: Assistive Technology Information: Online Database, 2005. Retrieved May 24, 2005, from http://www.abledata.com

[27] Vámossy, Z., Tóth, A, and Hirschberg, P.. PAL Based Localization Using Pyramidal Lucas-Kanade Feature Tracker. In Proceedings of the Simposium on Intelligent Systems, SISY 2004 (pp. 223-231). Subotica, Serbia and Montenegro, 2004.

[28] Viola, P., and Jones, J. J.. Robust Real-Time Face Detection. IEEE International Conference on Computer Vision, CVPR, 2001.

[29] Viola, P., and Jones, J. J.. Robust Real-Time Face Detection. International Journal of Computer Vision, 57(2). Kluwer Academic Publishers. 137-154, 2003.

[30] World Health Organization.. International Classification of Functioning, Disability and Health. Geneva, Switzerland: Author, 2001. Retrieved June 7, 2005, from http://www.who.int/inf-pr-1999/en/note99-19.html