Detection of Eye Blink and Redness by Smart Eye Protector Application

Mr.P.Nagaraj
M.E Dept Of It (Project Guide)

L. R. Siva Shankar (*Author*)
Information Technology
Jei Mathajee College Of Engg (Jmce)
Kanchipuram,Tamilnadu.

D. Venkatesh
Information Technology Jeimathajee College Of Engg (Jmce) Kanchipuram,Tamil Nadu.

***Abstract*--- The main idea of the Project is to preventing users from several common eyes diseases such as shortsightedness, eye blinking and eye redness and so on. An image processing system is combined with a camera service to measure the distance between the screen and user's eyes and record the screen-on time. By using this information obtained from the image processing system, some rules are set to prevent user from using smart phone in improper manner. User will be warned by several formats-screen-block, warning message, and pop-up preview. A usage report is provided to users so that they can understand their habit of using smart phone. This application helps you to keep track of the time of breaks when work with screen. You can comfortably control the timer by application, by widget or by permanent notification. Smart eye protector Android Application is deployed to capture Images through Camera. After capturing images it detects the user eye condition if the eye blinking or redness occurs mean, application automatically closing the application and shutdown the mobile. So user eye will be protected.**

Keywords— MOBILE FRONT CAMERA, ANDROID MOBIlE.

1. INTRODUCTION

In the European Community more than 40% of the today's working population uses mobile phones in their daily work. Computer use is related with static work, constrained sitting and vision problems. For example, approximately 70% of computer workers worldwide are reported to having vision problems leading towards Computer Vision Syndrome. The number of computer-related jobs is expected to increase significantly in the next decade, along with the number of workplace-related illnesses. One of primary causes of vision problems during computer use is insufficient eye movement, caused by long periods of gazing at computer screen. Distance between the screen and the user's head usually doesn't change much and as a result the muscles involved in adaptation of the eye are not exercised for long periods of time, leading to their weakening.

This is usually accompanied by decrease in eye blinking frequency, which leads to excessive dryness of the eye surface (cornea and sclera) and can be harmful to the eye. Chronic dry eyes can eventually lead to scarring of the cornea and sight loss. With preventive measures like regular breaks, eye exercises and relaxation activities most of those disorders can be avoided. Unfortunately, the majority of population is reluctant to change their workplace habits until first signs of health issues appear.

To help users become aware of the problem and assist them in prevention, we propose to use a simple monitor mounted camera (webcam) for capturing video of user at his workplace and estimating his eye blinking patterns. When potentially harmful behavior is detected, the user can be alerted and informed about suitable actions. Remote detection of eye-blinks from video is not as accurate as head-mounted eye trackers, but this is usually compensated by greater ease of use, non-invasiveness and much lower cost. For our purpose, the eye tracking must run in real-time, without any additional hardware (like IR illumination for example), using low quality input video, and be capable of operating under varying indoor conditions (typical office environment). Several eye tracking approaches that address those issues were already published, but few address all of the mentioned constraints and most are not suitable for precise evaluation of eye blinking movements.

Mobile devices such as smart phones or tablet PCs have al- ready become ubiquitous. Recently, it has been increasingly common for people to check e-mail, browse on the Internet, watch movies, and even read books on their portable devices. People are indeed exposed to the Computer Vision Syndrome (CVS) not only in the office but also elsewhere. To protect people who spend much time on mobile devices from this problem, we propose a non-intrusive application that keeps track of the eye blink rate of the mobile device users. If the blink rate is less than desired, the application nudges the user to blink often by vibrating, or modulating the brightness of the screen. To the best of our knowledge, this is the first study that tries to alleviate the CVS in a mobile computing environment. To achieve this goal, a light-weight yet accurate eye detecting and tracking technique specialized for mobile devices are essential. Eye tracking and blink detection algorithms

using a general video camera have been extensively studied in the literature. However, most of them assumed that the user is always gazing at the screen, and the eyes of the user are always in the video camera frames. The existing techniques are inadequate for a mobile device since frequent changes in device position due to the user movements lead to frequent absence or position changes of the eyes in the video frame.

- **Complete:** The designers took a comprehensive approach when they developed the Android Platform.
- **Open:** The Android platform provides open source licensing.
- **Free:** Android applications are free to develop. No required signing or certification fees.

2.PURPOSE

EXISTING SYSTEM:

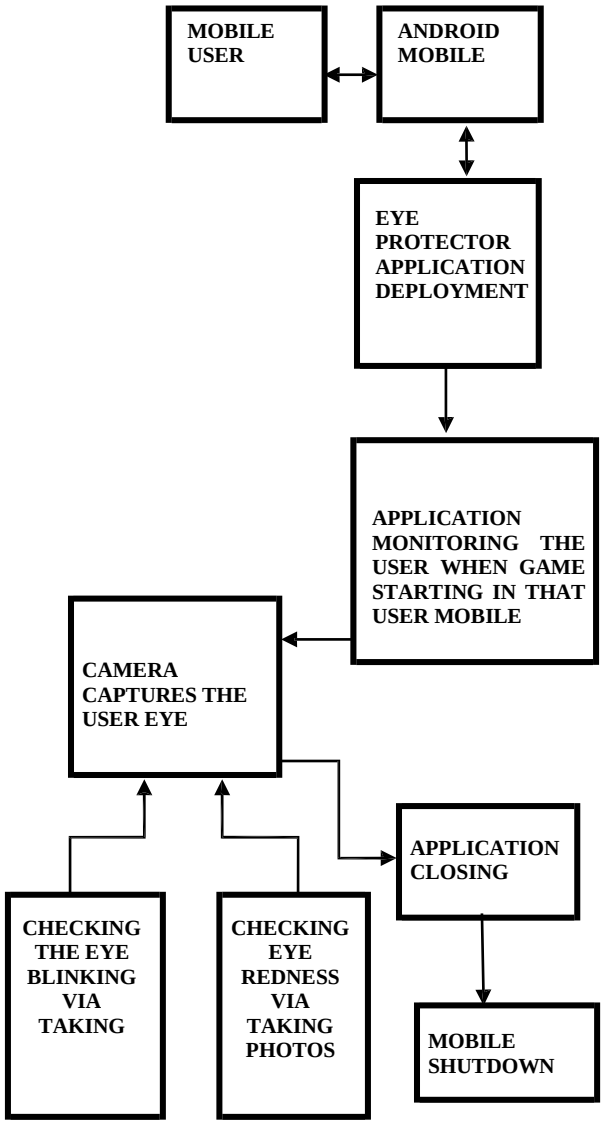
In the **EXISTING SYSTEM,** several common eyes diseases such as shortsightedness, eye blinking, eye redness and Computer Vision Syndrome (CVS) is a common problem in the mobile user and it is becoming more serious as mobile devices (e.g. smart phones and tablet PCs) with small, low-resolution screens are outnumbering the home computers.

However, most people do not realize that they blink less and some do not blink at all in front of the screen. Although exist techniques for detecting the eye blinking is not efficient.

PROPOSED SYSTEM:

In the **PROPOSED SYSTEM,** Smart eye protector Android Application is deployed to capture Images through Camera. After capturing images it detects the user eye condition if the eye blinking or redness occurs means, application automatically shutdown the mobile. So user eye will be protected. This application helps you to keep track of the time of breaks when work with screen.

ARCHITECTURE DIAGRAM:



3. ANALYSIS:

Literature Survey

The currently existing systems for eye monitoring process are as follows:

USB CAMERAS: There are no lighting requirements nor offline templates needed for the proper functioning of the system. The system works with inexpensive USB cameras and runs at a frame rate of 30 frames per second.

VIDEO BASED HUMAN COMPUTER TECHNIQUES : In interaction tools are introduced that can activate a binary switch and issue a selection command. "BlinkLink," as the first tool is called, automatically detects a user's eye blinks and accurately measures their durations. The system is intended to provide an alternate input modality to allow people with severe disabilities to access a computer. Voluntary long blinks trigger mouse clicks, while involuntary short blinks are ignored. The system enables communication using "blink patterns:" sequences of long and short blinks which are interpreted as semiotic messages. The second tool, "EyebrowClicker," automatically detects when a user raises his or her eyebrows and then triggers a mouse click. Both systems can initialize themselves, track the eyes at frame rate, and recover in the event of errors. No special lighting is required. The systems have been tested with interactive games and a spelling program. Results demonstrate overall detection accuracy of 95.6% for BlinkLink and 89.0% for EyebrowClicker.

ROBERT J. K. JACOB proposed that In seeking hitherto-unused methods by which users and computers can communicate, we investigate the usefulness of eye movements as a fast and convenient auxiliary user-to-computer communication mode.

KRISTEN GRAUMAN, MARGRIT BETKE, JAMES GIPS , GARY R. BRADSKI proposed that, a robust, accurate algorithm to detect eye blinks, measure their duration, and interpret them in real time to control a non-intrusive interface for computer users with severe disabilities.

K. GRAUMAN, M. BETKE, J. LOMBARDI, J. GIPS, G.R. BRADSKI proposed that Two video-based human-computer interaction tools are introduced that can activate a binary switch and issue a selection command.

MICHAEL CHAU AND MARGRIT BETKE proposed that a human-computer interface (HCI) system designed for use by people with severe disabilities is presented. People that are severely paralyzed or afflicted with diseases such as ALS (Lou Gehrig's disease) or multiple sclerosis are unable to move or control any parts of their bodies except for their eyes.

IOANA BACIVAROV, MIRCEA IONITA, PETER CORCORAN proposed that, Active Appearance Models offer a form of 2D affine face model which can quickly match the texture and shape of a detected face region. Similar statistical models can be developed for sub-regions of a detected face such as the eyes.

ANTONIO HAROY, MYRON FLICKNERZ, IRFAN ESSAY proposed there is much prior work on finding faces and fa-cial features in complex scenes. Some approaches, like Kothari, et al. find eyes in images without finding faces. They observed that the gradient intersection points were good candidate locations for eyes.

4.FEASIBILTY STUDY

A. Economic Feasibility

Being a software application, it has the development cost to be negligible. The cost per person is very less and can be fitted in the current cost factors of the traditional eye production.

B. Technical Feasibility

The language and the platforms used for the application development are easily available from the organization itself. The team also suffices the requirements of technical IQ required for the proposed system to be completed.

ANDROID EXECUTION ENVIRONMENT

We represent the regular Java and Android execution paths in the figures respectively. It is interesting to note here however is that the Android compilers do not operate on Java Language code. Instead, the Android translators work on the resulting Java byte code emitted from a traditional Java compiler.

As such, it is possible to reuse existing Java libraries, even if the original source code is not available. Such libraries must meet stringent requirements

however, they need to:

1. adhere to the Java SE 5 dialect
2. not use any Java classes or packages found in Java SE 5 not found in the Android platform
3. not use any packages or classes specific to the Sun Microsystems platform

C. Operational Feasibility

If the system is practically implemented, then it will for sure solve the problems' posed in the problem definition and it can be proved with the initial phases of the prototype that well be downloaded. The proposed system is easily scalable and hence can support the additional users that would be added after every year of the organization's admission.

5.METHODOLOGY AND TECHNOLOGY USED

Eye Tracking

Motion analysis alone is not sufficient to give the highly accurate blink information desired. It does not provide precise duration information, and multiple component pair candidates may occur sequentially as the result of a single blink.

Relying on motion would make the system extremely intolerant of extra motion due to facial expressions, head movement, or gestures. The user must be allowed to move his or her head with relative freedom

if necessary.

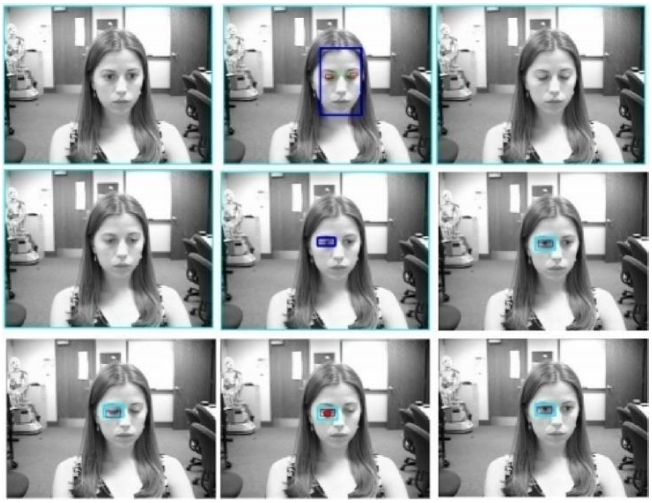


Fig.6. Intermittent frames from a sequence during the motion analysis phase when the template is being found automatically by the user's first several natural blinks. *Rectangles* around the face indicate that blink-like motion was detected. The *small rectangle* that appears around the eye three frames later indicates where the open-eye template is being selected. The *subsequent small rectangles* indicate eye tracking. A *circle* on top of the eye (third row, second column) indicates that a blink is believed to have just ended

Following initial localization, a fast eye tracking procedure maintains exact knowledge about the eye's appearance. Thus, the eye may be evaluated for amount of closure at the next stage. As described, the initial blink detection via motion analysis provides very precise information about the eyes' positions. Consequently, a simple tracking algorithm suffices to update the region of interest centered around the eye. The system utilizes the normalized correlation coefficient.

$R(x,y)$ =

$$\frac{\sum_{y'=0}^{h}\sum_{x'=0}^{w}\mathbf{T}(x',y')\mathbf{I}(x+x',y+y')}{\sqrt{\sum_{y'=0}^{h}\sum_{x'=0}^{w}\mathbf{T}(x',y')^2\sum_{y'=0}^{h}\sum_{x'=0}^{w}\mathbf{I}(x+x',y+y')^2}}$$

Where

$\mathbf{T}(x',y') = T(x',y') - \bar{T}$, $\mathbf{I}(x+x',y+y') = I(x+x',y+y') - \bar{I}(x,y)$, and $T(x,y)$ & $I(x,y)$ are the brightness Of the pixels at (x,y) in the template and source image, respectively, and $\bar{T}$ is the average value of the pixels in the template raster and $\bar{I}(x,y)$ is the average value of the pixels in the current search window of the image. The coefficient $R(x,y)$ is a measure of match between the open-eye template and all points within the small search region surrounding the location of the eye given from the previous frame. In this way, the current eye position is updated nearly 30 times per second and remains accurate barring dramatic, sudden head movements or significant changes in depth. For these events, it is critical that the tracker declare itself lost and reinitialize using blink motion analysis as discussed above. The tracker is believed to be lost if the best match score found using the correlation coefficient falls below a set threshold *F*. The tracker does not get lost during the blink because the closed eye and its closely neighboring pixels bear enough similarity to the open-eye template to pass the

threshold. Figure 6 shows a sequence of frames during the motion analysis phase in which the subjects' eyes are detected and tracked.

Blink Detection And Duration Of Closure Measurement

As the eye closes, it begins to look less and less like an open eye; likewise, it regains its similarity to the open eye slowly as it reopens. This is a simple but powerful observation. During an eye blink, the best correlation scores reported by the tracker can be plotted across time to depict a clear waveform that illustrates the duration of successive blinks (see Fig. 7).

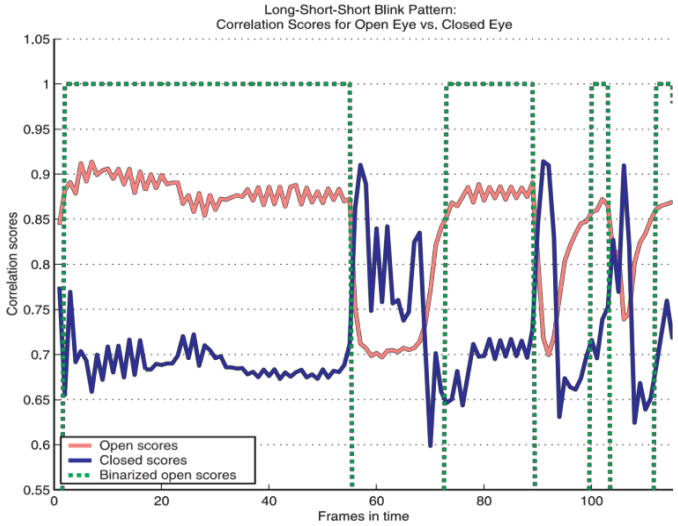


Fig.7. Correlation scores over time comparing the user's eye at each frame to both the open-eye template and the closed-eye template. The open-eye scores present a waveform indicating the captured blink pattern: *long, short, short*. Such samples were collected and used to identify an effective threshold *O* for classifying eyes as opened or closed at each frame

Experiments comparing the correlation scores of the actual eye and its closed template with the scores of the actual eye and its open template confirmed that both methods succeed in identifying blinking. However, the apparent correspondence of the two measures would make it redundant to compute both online, and so only the open eye correlation is used in the current system. Likewise, processing time may be conserved by tracking and computing the correlation for only one eye. The motion analysis above can be used to verify or refute the

correlation score's findings. Since the motion components account for both eyes, correlating for the second eye would be superfluous and is therefore omitted. It is a simple task to specify in the software that a particular eye or both eyes be considered.

the observations from the training data, threshold values O = 0.85 and f = 0.55 were chosen for the BlinkLink interface.

TECHNOLOGY USED

A. Android

Android is a mobile operating system that is based on a modified version of Linux. It was originally developed by a startup of the same name, Android, Inc. In 2005, as part of its strategy to enter the mobile space, Google purchased Android and took over its development work.

B. Development Tools

The Android SDK includes a variety of custom tools that help develop mobile applications on the Android platform. The most important of these are the Android Emulator and the Android Development Tools plug-in for Eclipse, but the SDK also includes a variety of other tools for debugging, packaging, and installing are applications on the emulator.

- *Android Emulator*

A virtual mobile device that runs on computer use the emulator to design, debug, and test r applications in an actual Android run-time environment.

- *Android Development Tools Plug-in for the Eclipse IDE*

The ADT plug-in adds powerful extensions to the Eclipse integrated environment; making creating and debugging are Android applications easier and faster. If use Eclipse, the ADT plug-in gives an incredible boost in developing Android applications:

It gives access to other Android development tools from inside the Eclipse IDE. For example, ADT lets access the many capabilities of the DDMS tool taking screenshots, managing port-forwarding, setting breakpoints, and viewing thread and process information directly from Eclipse.

It provides a New Project Wizard, which helps quickly create and set up all of the basic files will need for a new Android application. It automates and simplifies the process of building are Android application. It provides an Android code editor that helps write valid XML for are Android manifest and resource files.

- *Dalvik Debug Monitor Service (DDMS)*

Integrated with Dalvik, the Android platform's custom VM, this tool lets manage processes on an emulator or device and assists in debugging. Can use it to kill processes, select a specific process to debug, generate trace data, view heap and thread information, take screenshots of the emulator or device, and more.

- *Android Debug Bridge (ADB)*

The ADB tool lets install applications .apk files on an emulator or device and access the emulator or device from a command line. This is also used to link a standard debugger to application code running on an Android emulator or device.

Android Asset Packaging Tool (AAPT)

The AAPT tool lets create .apk files containing the binaries and resources of Android applications.

Android Interface Description Language (AIDL)

AIDL lets to generate code for an interprocess interface, such as what a service might use. Included as a convenience, this tool lets access the SQLite data files created and used by Android applications.

Trace view

This tool produces graphical analysis views of trace log data that can generate from our Android application.

mksdcard

Helps create a disk image that can use with the emulator, to simulate the presence of an external storage card (such as an SD card). The dx tool rewrites .class bytecode into Android bytecode (stored in .dex files.)

activityCreator

A script that generates build files that can use to compile our Android applications. If we are developing on Eclipse with the ADT plug-in, we won't need to use this script.

C. Security and Permissions in Android

Android is a multi-process system, where each application (and parts of the system) runs in its own process. Most security between applications and the system is enforced at the process level through standard

Linux facilities, such as user and group IDs that are assigned to applications. Additional finer-grained security features are provided through a "permission" mechanism that enforces restrictions on the specific operations that a particular process can perform.

Android mobile phone platform is going to be more secure than Apple's iPhone or any other device in the long run. There are several solutions nowadays to protect Google phone from various attacks. One of them is security vendor McAfee, a member of Linux Mobile (LiMo) Foundation. This foundation joins particular companies to develop an open mobile-device software platform. Many of the companies listed in the LiMo Foundation have also become members of the Open Handset Alliance (OHA).

As a result, Linux secure coding practice should successfully be built into the Android development process. However, open platform has its own disadvantages, such as source code vulnerability for black-hat hackers. In parallel with great opportunities for mobile application developers, there is an expectation for exploitation and harm. Stealthy Trojans hidden in animated images, particular viruses passed from friend to friend, used for spying and identity theft, all these threats will be active for a long run.

Another solution for such attacks is SMobile Systems mobile package. Security Shield –an integrated application that includes anti-virus, anti-spam, firewall and other mobile protection is up and ready to run on the Android operating system. Currently, the main problem is availability for viruses to pose as an application and do things like dial phone numbers, send text messages or multi-media messages or make connections to the Internet during normal device use. It is possible for somebody to use the GPS feature to track a person's location without

their knowledge. Hence SMobile Systems is ready to notify and block these secure alerts. But the truth is that it is not possible to secure r mobile device or personal computer completely, as it connects to the internet. And neither the Android phone nor other devices will prove to be the exception.

BACK END

SQLite, the most popular Open Source SQL database management system, is developed, distributed, and supported by SQLite AB. SQLite AB is a commercial company, founded by the SQLite developers, that builds its business by providing services around the SQLite database management system. The SQLite Web site (http://www.SQLite.com/) provides the latest information about SQLite software and SQLite AB. SQLite is a database management system.

7.IMPLEMENTATION

1. ***Database***: the database for our project was created; this will store login, password, account details, stored image, etc.
2. ***Android Application***: the android application which will be available at each portable device is made.

8.ACKNOWLEDGMENT

We would like to take this opportunity to express our sincere gratitude to our proposed system guide, Mr.C.NATARAJAN, Head of the Department, Information Technology and Computer Science and Engineering, and our project guide Mr.P.NAGARAJ M.E… department of IT for their invaluable support and guidance throughout the process in spite of his busy schedule and appreciating our diligent efforts and providing valuable inputs which helped us to mitigate many problems faced in the software development process.

We are grateful to her for her feedback which helped us track and schedule the process effectively. The time, ideas and encouragement that she gave is appreciable.

9.CONCLUSION:

This paper proposes a novel mobile application to alert a user at risk of eye redness, eye blinking and

protects their eyes from possible damages. To accomplish this, eye protector application monitors the user's eye blink frequency via the front camera of the device in a non-intrusive way. Eye protector application determines whether or not it will recommend the users or rest their eyes based on the user's eye blink rate. it efficiently detects the eyes from multiple viewing angles and in various lighting conditions.

10.REFERENCE:

1. Kristen Grauman, Margrit Betke , James Gips , Gary R. Bradski. Communication via Eye Blinks - Detection and Duration Analysis in Real Time. Vision Interface Group Image & Video Computing EagleEyes Visual Interactivity Group MIT AI Lab Boston University Boston College Intel Corporation. Published online: 23 October 2003 – Springer -Verlag 2003.
2. Michael Chau and Margrit Betke. Real Time Eye Tracking and Blink Detection with USB Cameras. Computer Science Department Boston University Boston, MA 02215, USA {mikechau, betke@cs.bu.edu} May 12, 2005.
3. Robert J. K. Jacob(Naval Research Laboratory). The Use of Eye Movements in Human-Computer Interaction Techniques: What You Look At is What You Get. Naval Research Laboratory. ACM Transactions on Information Systems, Vol. 9, No 3, April 1991
4. K. Grauman, M. Betke , J. Lombardi , J. Gips , G.R. Bradski . Communication Via Eye Blinks And Eyebrow Raises: Video-Based Human-Computer Interfaces. Vision Interface Group, AI Laboratory, Massachusetts, Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA; E-mail: kgrauman@ai.mit.edu, http://www.ai.mit.edu/ ~kgrauman .Published online: 23 October 2003 – Springer Verlag 2003.

5. Antonio Haroy, Myron Flicknerz, Irfan Essay. Detecting and Tracking Eyes By Using Their Physiological Properties, Dynamics, and Appearance. yGVU Center / College of Computing zComputer Vision Enhanced User Interfaces Georgia Institute of Technology IBM Almaden Research Center Atlanta, GA 30332-0280 San Jose, CA 95120.
6. Ioana Bacivarov, Mircea Ionita, Peter Corcoran. Statistical Models of Appearance for Eye Tracking and Eye-Blink Detection and Measurement. . Ioana Bacivarov Student *Member*, IEEE, Mircea Ionita, Student *Member*, IEEE and Peter Corcoran, Senior *Member*, IEEE Transactions on Consumer Electronics, Vol. 54, No. 3, AUGUST 2008.
7. M. Betke, J. Gips, and P. Fleming. The camera mouse: Visual tracking of body features to provide computer access for people with severe disabilities. IEEE Trans- actions on Neural Systems and Rehabilitation Engineering, 10:1, pages 1–10, March 2002.
8. M. Betke, W. Mullally, and J. Magee. Active detection of eye sclera's in real time. Proceedings of the IEEE CVPR Workshop on Human Modeling, Analysis and Synthesis (HMAS 2000), Hilton Head Island, SC, June 2000.
9. T.N. Bhaskar, F.T. Keat, S. Ranganath, and Y.V. Venkatesh. Blink detection and eye tracking for eye localization. Proceedings of the Conference on Convergent Technologies for Asia-Pacific Region (TEN- CON 2003),pages 821–824,Bangalore,Inda, October 15- 17 2003.
10. R.L. Cloud, M. Betke, and J. Gips. Experiments with a camera-based human-computer interface system. Proceedings of the 7th ERCIM Workshop, User Interfaces For All (UI4ALL 2002), pages 103–110,Paris, France, October 2002.
11. S. Crampton and M. Betke. Counting fingers in real time: A webcam-based human-computer interface 9 with game applications. Proceedings of the Conference on Universal Access in Human-Computer Inter- action (affiliated with HCI International 2003), pages 1357– 1361, Crete, Greece, June 2003.
12. C. Fagiani, M. Betke, and J. Gips. Evaluation of tracking methods for human-computer interaction. Proceedings of the IEEE Workshop on Applications in Computer Vision (WACV 2002), pages 121–126, Or- lando, Florida, December 2002.
13. D.O. Gorodnichy. On importance of nose for face tracking. Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG 2002), pages 188–196, Washington, D.C., May 20- 21 2002.
14. D.O. Gorodnichy. Second order change detection, and its application to blink-controlled perceptual interfaces. Proceedings of the IASTED Conference on Visualization, Imaging and Image Processing (VIIP 2003), pages 140– 145, Benalmadena, Spain, September 8-10 2003.