VISUAL BASED EYE BLINK TRACKING SYSTEM FOR HUMAN COMPUTER INTERFACE

LAVANYA.K

10PCA10

A Project Report submitted to Avinashilingam Deemed University for Women, Coimbatore in partial fulfillment of the requirements for the Master's Degree in Computer Applications

MAY 2013

VISUAL BASED EYE BLINK TRACKING SYSTEM FOR HUMAN COMPUTER INTERFACE

LAVANYA.K

10PCA10

A Project Report submitted to Avinashilingam Deemed University for Women, Coimbatore in partial fulfillment of the requirements for the Master's Degree in Computer Applications

MAY 2013

SIGNATURE OF THE HEAD OF THE DEPARTMENT

SIGNATURE OF THE SUPERVISOR

SIGNATURE OF THE EXTERNAL EXAMINER

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

I would like to express my sincere thanks to God Almighty, for His constant love and grace that He has showered upon me.

I am very grateful to **Dr.T.S.K.Meenakshi Sundaram, M.A., M.Phil., Ph.D., Chancellor,** Avinashilingam Deemed University, Coimbatore for his support and encouragement during the course of my study.

I heartily thank **Dr. (Mrs). Sheela Ramachandran M.Sc., P.G. Dip., Ph.D., Vice Chancellor,** Avinashilingam Deemed University, Coimbatore for extending all resources that facilitated the conduct of the present study.

I express my humble gratitude to **Dr. (Mrs). Gowri Ramakrishnan M.Sc., M.Phil., Ph.D., Registrar,** Avinashilingam Deemed University, Coimbatore, for providing all facilities necessary for the study.

I am also thankful to **Dr. (Mrs). R. Parvatham M.Sc., Dip.Ed. M.Phil., Ph.D., Dean Faculty of Science,** for granting the facility required.

I wish to place on record my deep sense of gratitude to **Dr.(Mrs.)G.Padmavathi M.Sc., M.Phil., Ph.D.,** Professor and Head, Department of Computer Science, for providing all the facilities to complete the project.

I take this unique opportunity to express my sincere thanks to my project Coordinator and guide **Dr. (Mrs.)V.Srividhya, M.Sc., M.Phil., Ph.D., Assistant Professor,** Department of Computer Science, for her kind advice and knowledgeable suggestion, which helped me to complete my project successfully.

I would extend my hearty thanks to one and all who helped me directly or indirectly for successful completion of my project.

SYNOPSIS

SYNOPSIS

The main objective of the project is to replace the traditional mouse with the human face as a new way to interact with the computer. Facial features (nose tip and eyes) are detected and tracked to use their actions as mouse events. One way to achieve that is to capture the desired feature with a webcam and monitor its action in order to translate it to some events that communicate with the computer.

Eyes were used to simulate mouse clicks, so the user can fire their events as the person blinks. The left/right eye blinks fire left/right mouse click events. The nose tip was selected as the pointing device that acts as the mouse pointer on the users screen.

It helps the people who have hand disabilities that prevent them from using the mouse. The human face that can be captured with the help of night vision camera that detect and tracks the facial features correctly and thereby fast enough to be applied in real-time.

One of the promising fields in artificial intelligence is HCI (Human Computer Interface). To achieve this HCI different human features and monitoring devices were used. This project is interested in involving with the use of facial features (eyes and nose tip) and webcams. Different detection techniques were applied (e.g. feature based, image based) where the goal was to achieve more accurate results with less processing time. Early methods used with simple heuristics are applied to images taken with certain restrictions (e.g. plain background, Frontal view).

Each HCI method had some drawbacks, some methods used expensive equipments, some were not fast enough to achieve real-time execution, and others were not robust and precise enough to replace the mouse. While different devices were used in HCI (e.g. infrared cameras, sensors, microphones) this system used an off-the-shelf webcam that affords a moderate resolution and frame rate as the capturing device in order to make the ability of using the program affordable for all individuals.

CONTENTS

CONTENTS

S.NO	PARTICULARS	PAGE NO
1.	**INTRODUCTION**	
	1.1 PROBLEM DEFINITION	2
	1.2 OVERVIEW OF THE PROJECT	3
2.	**SYSTEM CONFIGURATION**	
	2.1 HARDWARE SPECIFICATION	5
	2.2 SOFTWARE SPECIFICATION	5
	2.3 ABOUT THE SOFTWARE	6
3.	**SYSTEM ANALYSIS**	
	3.1 EXISTING SYSTEM	11
	3.2 PROPOSED SYSTEM	12
4.	**SYSTEM DESIGN**	
	4.1 INPUT DESIGN	13
	4.2 OUTPUT DESIGN	14
5.	**SYSTEM DEVELOPMENT**	
	5.1 LIST OF MODULES	15
	5.2 MODULE DESCRIPTION	15

6. SYSTEM TESTING AND IMPLEMENTATION

6.1 SYSTEM TESTING 26

6.2 SYSTEM IMPLEMENTATION 28

7. CONCLUSION 29

8. SCOPE FOR FUTURE ENHANCEMENT 30

BIBLIOGRAPHY

APPENDIX

SYSTEM FLOW DIAGRAM

SCREEN SHOTS

INTRODUCTION

1. INTRODUCTION

In the past few years high technology has become more progressed, and less expensive. With the availability of high speed processors and inexpensive webcams, more and more people have become interested in real-time applications that involve image processing.

With the growth of attention about computer vision, the interest in HCI has increased proportionally. HCI aims to use human features (e.g. face, hands) to interact with the computer.

Face detection has always been a promising field in the computer vision world, considering that it is the backbone of any application that deals with the human face (e.g. surveillance systems, access control). There are large number of detection methods are available. This method did not spare any effort or imagination in inventing and evolving methods to localize, extract, and verify faces in images.

This project helps the persons who have hand disabilities that prevent them from using the mouse by designing an application. It uses facial features (nose tip and eyes) to interact with the computer. One way to achieve that is to capture the desired feature with a webcam and monitor its action in order to translate it to some events that communicate with the computer.

This method used an off-the-shelf webcam that affords a moderate resolution and frame rate as the capturing device in order to make the ability of using the program affordable for all individuals.

Eyes were used to simulate mouse clicks, so the user can fire their events as the person blinks. The left/right eye blinks fire left/right mouse click events. The nose tip was selected as the pointing device that acts as the mouse pointer on the users screen.

This system detects and tracks the desired facial features precisely and fast enough to be applied in real-time application.

1.1 PROBLEM DESCRIPTION

The objective of the project is to replace the traditional mouse with the human face as a new way to interact with the computer. One of the promising fields in artificial intelligence is HCI. It aims to use human features to interact with the computer.

Facial features (nose tip and eyes) are detected and tracked in real-time to use their actions as mouse events. One way to achieve that is to capture the desired feature with a webcam and monitor its action in order to translate it to some events that communicate with the computer. This external device feeds the program with the video stream.

To control the mouse pointer various points are tracked to acts as the virtual mouse click events. The nose tip is tracked to use its movement and coordinates as the movement and coordinates of the mouse pointer. The eyes are tracked to detect their blinks, where the blink becomes the mouse click.

1.2 OVERVIEW OF THE PROJECT

Facial features (nose tip and eyes) are detected and tracked in real-time to use their actions as mouse events. The first step is to detect the face. Two methods are used to detect the face such as Feature based method and Image based method.

Feature based method is to find the facial features. It is used to verify their authenticity. Image based method is based on scanning the image of interest with a window that looks for faces at all scales and locations.

The next step is to find face candidate. To find SSR (Six Segmented Rectangular) filter and integral image techniques are used.SSR is the sum of pixels in each sector is denoted as S along with the sector number. Intermediate Image representation called the integral image. The integral image at location x,y contains the sum of pixels and to left of the pixel x,y. This project shows an ideal location of SSR filter, where its center is considered as a face candidate.

The next step is to filter out the false positive (other than the face candidate). Using clustering algorithm, eliminate the cluster other than the face candidate. The next stage is to extract the BTE (Between The Eye Template). In order to extract, need to find pupils candidate. If it's found then the lower half of the sector is probably the pupil and the upper half is probably the eyebrow.

The next stage is to verify with SVM (Support Vector Machine). For that, after extracting the template and pass them into SVM in order to classify them as positive and negative classification. Positive classification result means true faces. Negative classification means false faces. The pupil candidates of the template with the positive classification result are set as the detected eyes.

The next step is to find the nose tip. For that two methods are used. The first one is ROI (Region Of Interest). The second one is find the NBP (Nose-Bridge-Point).

After facial features are detected and tracking them in the video stream. The tracking process is based on predicting the place of the feature. To achieve this, Hough Transform, Histogram Matching and Sum of Square Difference are applied.

The next step is to track the nose tip. Tracking the nose tip will be achieved by template matching, which is inside the ROI.

The final stage is to track the eyes. To find the eye's new template in the ROI, two methods are combined: the first one is template matching, the second one is the darkest 5*5 region. Then the mean between the two found coordinates are used to find the eye's new location. Based on the information obtained from the eye region and possible nose tip helps us to moving the cursor on the screen.

SYSTEM CONFIGURATION

2. SYSTEM CONFIGURATION

This section describes the hardware and software specification needed for both development and implementation phases of this project.

2.1 HARDWARE SPECIFICATION

Processor	:	Pentium IV
Speed	:	2.53GHz
Ram	:	2 GB
Hard Disk Drive	:	160 GB SATA
Monitor	:	17” TFT Color Monitor
Keyboard	:	HCL 104 Keys
Mouse	:	Optical
Web Cam	:	Intex 305 Night vision camera.

2.2 SOFTWARE SPECIFICATION

Operating System	:	Windows XP 32bit
Development	:	J2SDK 1.6, JMF 2.1
IDE	:	Net Beans 5.5

2.3 ABOUT THE SOFTWARE

Java was developed at Sun Microsystems. Work on Java originally began with the goal of creating a platform-independent language and operating system for consumer electronics. Java is both a programming language and an environment for executing programs written in the Java language. Unlike traditional compilers, which convert source code into machine-level instructions, the Java compiler translates Java source code into instructions that are interpreted by the runtime Java Virtual Machine. So, unlike languages like C and C++, on which Java is based, Java is an interpreted language. Although most of the current excitement and anticipated use of Java are related to the Internet and the World Wide Web in particular, Java did not begin as an Internet project.

The creators of Java at Sun Microsystems have defined the Java language as "a simple, object-oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, high-performance, multithreaded, and dynamic language.

2.3.1 FEATURES

- **Simple**

The simplicity of Java is enhanced by its similarities to C and C++. Because many of today's current programmers, especially those likely to consider using Java, are experienced in at least C and probably C++, Java is instantly familiar to these programmers.

Java has simplified C++ programming by both adding features beyond those found in C++ and by removing some of the features that make C++ a complicated and difficult language to master. Java is simple because it consists of only three primitive data

types - numbers, Boolean types, and arrays. Everything else in Java is a class. For example, strings are true objects, not just arrays of characters. Similarly, arrays in the Java language are first-class objects, not just memory allocations and runtime representations.

- **Object-Oriented**

Of course, Java is object-oriented. Java classes are comprised of methods and variables. *Class methods* are the functions that an object of the class can respond to. *Class variables* are the data that define the state of an object. In Java, methods and variables can be declared as *private*, *protected*, or *public*. Private methods and variables are not accessible outside of the class. Protected members are accessible to subclasses of the class, but not to other classes. Finally, public methods and variables are accessible to any class.

- **Distributed**

Java facilitates the building of distributed applications by a collection of classes for use in networked applications. By using Java's URL (Uniform Resource Locator) class, an application can easily access a remote server. Classes also are provided for establishing socket-level connections.

- **Robust**

The designers of Java anticipated that it would be used to solve some very complex programming problems. Writing a distributed, multithreaded program that can run on a variety of operating systems with a variety of processors is not a simple task. To do it successfully, you need all the help your programming language can offer you.

- **Secure**

Closely related to Java's robustness is its focus on security. Because Java does not use pointers to directly reference memory locations, as is prevalent in C and C++, Java has a great deal of control over the code that exists within the Java environment.

- **Architecture-Neutral**

It is still not easy to write an application that can be used on Windows NT, UNIX, and a Macintosh. And it's getting more complicated with the move of Windows NT to non-Intel CPU architectures.

A number of commercially available source code libraries (for example, Zinc, Zap, and XVT) attempt to achieve application portability. These libraries attempt this by focusing on either a lowest common denominator among the operating systems or by creating a common core API (Application Programming Interface).

- **Portable**

In addition to being architecture-neutral, Java code is also portable. It was an important design goal of Java that it be portable so that as new architectures (due to hardware, operating system, or both) are developed, the Java environment could be ported to them. In Java, all primitive types (integers, longs, floats, doubles, and so on) are of defined sizes, regardless of the machine or operating system on which the program is run.

- **High-Performance**

For all but the simplest or most infrequently used applications, performance is always a consideration. It is no surprise, then, to discover that achieving high

was one of the initial design goals of the Java developers. A Java application will not achieve the performance of a fully compiled language such as C or C++.

2.3.2 Java Media Framework Details

Java[a] Media Framework (JMF) provides a unified architecture and messaging protocol for managing the acquisition, processing, and delivery of time-based media data. JMF is designed to support most standard media content types, such as MPEG, QuickTime, and WAV. JMF enables the programmers to develop java programs to present time-based media.

JMF 2.0 is designed to:

1. Ease programming.
2. Support capturing media data.
3. Enable the development of media streaming and conferencing applications in Java.
4. Enable advanced developers and technology providers to implement custom solutions based on the existing API and easily integrate new features with the existing framework.
5. Provide access to raw media data.

By exploiting the advantages of the Java platform, JMF delivers the promise of "Write Once, Run Anywhere[a] to developers who want to use media such as audio and video in their Java programs. JMF provides a common cross-platform Java API for accessing underlying media frameworks.

JMF implementations can leverage the capabilities of the underlying operating system, while developers can easily create portable Java programs that feature time-based media by writing to the JMF API. With JMF, one can easily create applets and applications that present, capture, manipulate, and store time-based media.

The framework enables advanced developers and technology providers to perform custom processing of raw media data and seamlessly extend JMF to support additional content types and formats, optimize handling of supported formats, and create new presentation mechanisms.

The RTP APIs in JMF 2.0 support the reception and transmission of RTP streams and address the needs of application developers who want to use RTP to implement media streaming and conferencing applications. Data Sources, Players and Processors are integral parts of JMF's high-level API for managing the capture, presentation, and processing of time-based media.

SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

System analysis is the general term that refers to an orderly and structured procedure for identifying and solving problems. It involves the study of existing system to understand how they function.

3.1 EXISTING SYSTEM

While different devices were used in HCI (e.g. infrared cameras, sensors, microphones) this system used an off-the-shelf webcam that affords a moderate resolution and frame rate as the capturing device in order to make the ability of using the program affordable for all individuals.

Early methods used simple heuristics are applied to images taken with certain restrictions (e.g. plain background, Frontal view).

Each HCI method (detecting facial features) had some drawbacks, some methods used expensive equipments, some are not fast enough to achieve real-time execution, and others are not robust and precise enough to replace the mouse.

DRAWBACKS

- Human Computer Interfacing through Hardware based system is difficult to implement.
- Slow to get the human action.
- Early projects need high resolution webcams and extraction of skin color models is not sufficient.
- Accuracy is low.

3.2 PROPOSED SYSTEM

The objective of the project is to present an application that is able of replacing the traditional mouse with the human face as a new way to interact with the computer. It aims to use human features (nose tip and eyes) to interact with the computer.

To achieve HCI, is to capture the desired feature with a webcam. This device feed the program with the video stream and monitors its action to translate it to some events to communicate with computer.

To control mouse pointer various points are tracked to acts as the virtual mouse click events. The nose tip is tracked to use its movement and coordinates as the movement and coordinates of the mouse pointer. The eyes are tracked to detect their blinks, where the blink becomes the mouse click.

This system proposes an automatic HCI. HCI system aids to support people who have hand disability that prevent them using the mouse. This project detects and tracks the desired facial features precisely and fast enough to be applied in real-time application.

ADVANTAGES

- No loss of accuracy in locating the user's eye and nose tip.
- It is compatible with inexpensive webcams.
- It is used in real-time performance.
- Higher frame rate and finer camera resolutions could lead to more robust.
- Different detection techniques help to achieve more accurate result with less processing time.

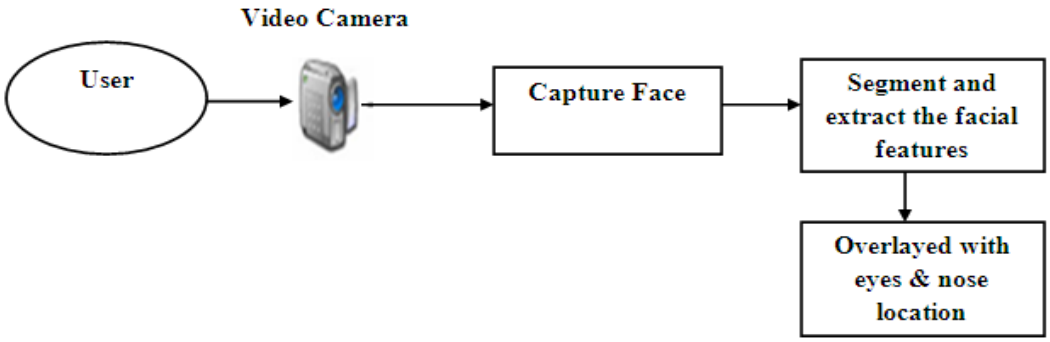
SYSTEM DESIGN

4. SYSTEM DESIGN

System design is the process of planning a new system to complement or altogether replace the old system. The purpose of the design phase is the first step in moving from the problem domain to the solution domain. The design phase translates the logical aspects of the system into physical aspects of the system.

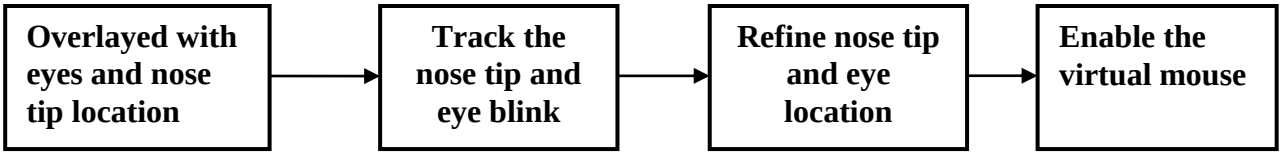
4.1 INPUT DESIGN

Input design is the process of converting user-oriented input to computer-based format. The goal of input data design is to make data entry as easy, logical and free from errors as possible. From the user capture the desired facial features with a webcam in order to translate it to some events to communicate with the computer. Image and Feature based method is used to segment the facial features to find the face candidate. Verify with SVM is used to extract the BTE template. The result of positive classification will be set as the detected eyes. ROI and intensity profile is used to find the nose tip. Finally Video frame overlaid with eyes and nose tip location.



4.2 OUTPUT DESIGN

The output design was done so that results of processing could be communicated to the users. This project found the facial features that are needed to track them in the video stream. To achieve that, Hough Transform, Histogram Matching and Sum of Square Difference are applied. These techniques are used to track the eyes and nose tip. Now the virtual mouse is enabled to move the cursor on the screen.



SYSTEM DEVELOPMENT

5. SYSTEM DEVELOPMENT

5.1 LIST OF MODULES

1. Face Detection
2. Cluster Face Candidate
3. Find Pupil's Candidate
4. Extract BTE (Between The Eye) Template
5. Find Nose Tip
6. Face Tracking

5.2 MODULE DESCRIPTION

5.2.1. FACE DETECTION

Face detection feature based method is used to reduce the area in its face, so this can decrease an execution time. To find face candidates the SSR filter and Integral image are used.

SSR Filter

SSR Filter stands for: Six Segmented Rectangular filter

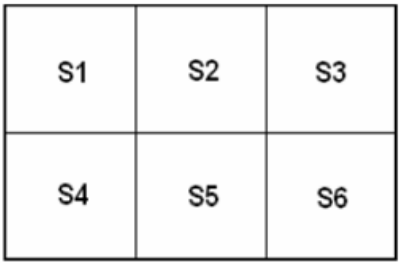


Fig 1: SSR Filter

The sum of pixels in each sector is denoted as S along with the sector number.

Integral Image

In order to facilitate the use of SSR filters an intermediate image representation called integral image will be used. In this representation the integral image at location x, y contains the sum of pixels which are above and to the left of the pixel x, y (see fig.2).

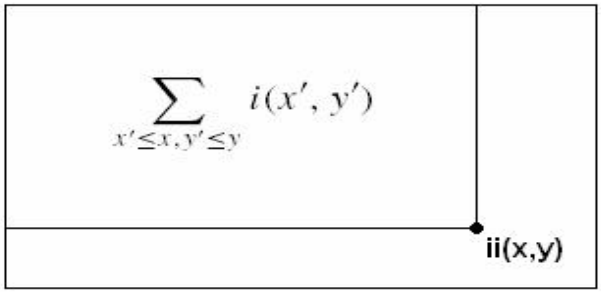


Fig 2: Integral Image

At first this system calculates the integral image by making a one pass over the video frame using these equations

$$s(x, y) = s(x, y-1) + i(x, y) \quad (1)$$

$$ii(x, y) = ii(x-1, y) + s(x, y) \quad (2)$$

where s(x, y) is the cumulative row sum, s(x,-1) = 0, and ii(-1, y) = 0. Fig 3 shows an ideal location of the SSR filter, where its center is considered as a face candidate.

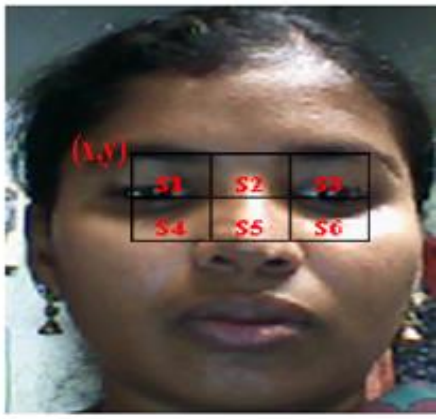


Fig 3: Ideal SSR filter location for a face candidate.

In the fig 3, (x, y) is the location of the filter (upper left corner). The plus sign is the center of the filter which is the face candidate. It can notice that, in this Fig 3 the ideal position of eyes fall in sectors S1 and S3, while the nose falls in sector S5. So in order to find face candidates, place the upper left corner (see fig.3) of the SSR filter on each pixel of the image. For each location (x, y), check equations (3, 4); if the conditions are fulfilled then the center of the filter will be considered as a face candidate. Eventually the candidates will group in clusters (see fig.4). Each dot of each group is a face candidate discovered by the SSR filter.

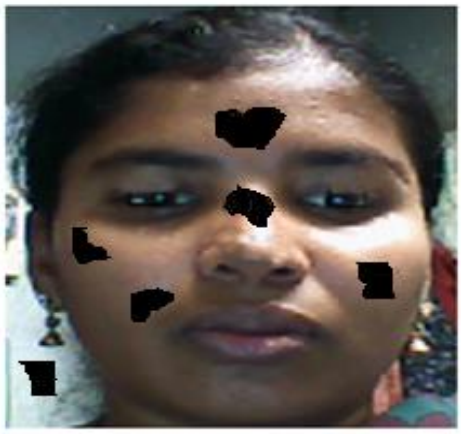


Fig 4: Groups of face candidates.

5.2.2. CLUSTER FACE CANDIDATE

Cluster the face candidate are used to filter the false positive (other than the face candidate). In order to find the clusters of face candidate, consider the center of each cluster as the final candidate. For each face candidate fc (face candidate is labeled as fc):

- If all neighbors are not face candidates assign a new label to fc.
- If one of the neighbors is a face candidate assign its label to fc.
- If several neighbors are face candidates assign the label of one of them to fc and make a note that the labels are equal.

A threshold that is relevant to the size of the currently used SSR filter is used to eliminate the clusters that are small. The center of each cluster that is big enough is set with the following equations:

$$x = [\sum x(i)]/n \qquad (3)$$

$$y = [\sum y(i)]/n \qquad (4)$$

where i – pixel from the cluster, n – the cluster's area. The final results are illustrated in figure 5 shows the centers of the clusters that passed the threshold, are marked as white dots.

Fig 5: Clusters of face candidate and their centers.

5.2.3. FIND PUPIL'S CANDIDATE

In order to extract BTE templates, need to locate pupils' candidates. Left and right pupils' candidates will fall in sectors S1, S3. For each of the sectors, find the pixels that belong to a dark area (in case the pupil). The sector is binarized with a certain threshold. If the threshold produces only one cluster like sector S3 (right eye) in fig 6. Then calculate the area of the part of the cluster on both half of the sector, which half of the sector is larger than a specified threshold. Then the center of that part is considered as the pupil. Otherwise the sector is omitted and no pupil is found.

If there are multiple clusters like sector S1 (left eye) in fig.5, this project need to find the cluster that is largest, darkest and closest to the darkest pixel of the sector. The same way the sector is binarized with certain threshold.

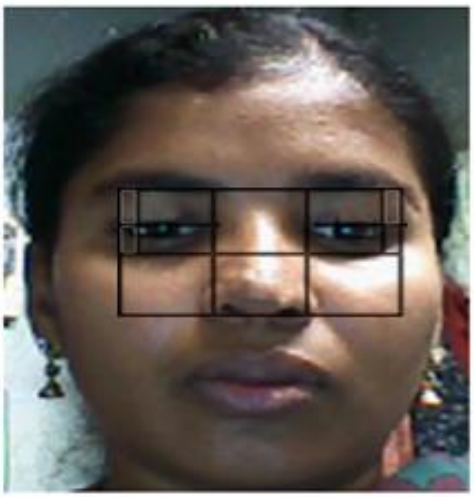


Fig 6:SSR filter centered on the center of the cluster.

5.2.4. EXTRACT BTE TEMPLATE

For each of the clusters (face candidates), extract BTE templates in order to pass them to the support vector machine. To achieve that, train the SVM on templates of size 35 * 21 pixels (see fig.7). In order to find the scale rate, divide the distance between the left and right pupil candidates on 23, where 23 is the distance between the left and right pupils in the training templates (see fig.7and 8). The 35 * 21 pixels from figure 8 have extracting the 35 *21 pixels template (see fig.9).

Fig 7: Illustration of how to extract a training template

Fig 8: Extract training template

Fig 9: the 35 * 21 pixels from figure 8 have formed the 35 *21 pixels template.

After extracting the templates, pass them to the SVM. In order to classify them, took each clusters BTE template and classify it. If the result is positive, multiply the result by the area of that cluster. Positive classification results mean true faces, while negative one are the false faces.

5.2.5. FIND NOSE TIP

The final step is to find the nose tip. To find the nose tip two methods are used. The first one is the ROI. The second one is to locate NBP (Nose-Bridge-Point).

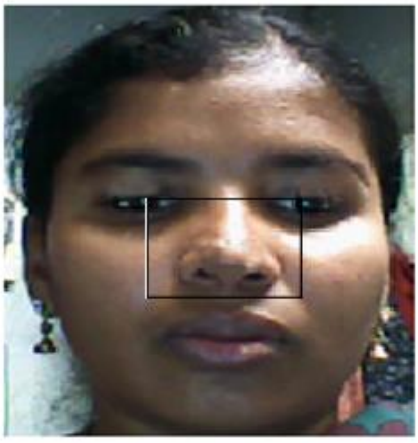
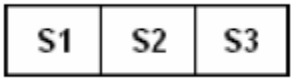


Fig 10: The square that forms the ROI, and the ROI after extraction.

The fig 10 defines a perfect square of the pupils, where the nose tip should fall inside this square, so this square becomes the ROI in finding the nose tip.

The second method is to locate the nose-bridge-point (NBP) on each line of the ROI. SSR filter is used to locate the NBP candidates in each ROI line.



After calculating the integral image of the ROI, each line is scanned with SSR filter. The center of the SSR filter is considered as an NBP candidate if the center sector is brighter than the side sectors. If the below equations are satisfied. Finally the video frame overlayed with eyes and nose tip location (see fig.11).

$$S2 > S1 \quad (5)$$

$$S2 > S3 \quad (6)$$

Fig 11: The final result of the face detection process.

5.2.6. FACE TRACKING

The final result of the face detection process (see fig. 11) is needed to track them in the video stream. The nose tip is tracked to use its movement and coordinates as the movement and coordinates of the mouse pointer. The eyes are tracked to detect their blinks, where the blink becomes the mouse click. The tracking process is based on predicting the place of the feature in the current frame based on its location in previous ones. To track the features, two template matching techniques are applied:

- Histogram Matching
- SSD (Sum of Square Differences)

Histogram Matching

The histogram matching, compare the histograms' shapes of the feature and the candidates. The closest histogram to the one of the feature is considered as the match, and its place is considered as the feature's new place.

Sum of Squared Differences

The next method used in tracking process is (SSD). The SSD between the template and the current window is calculated. The window that has the smallest SSD is chosen as the template's match, and its location is considered as the feature's new location (see fig. 12)

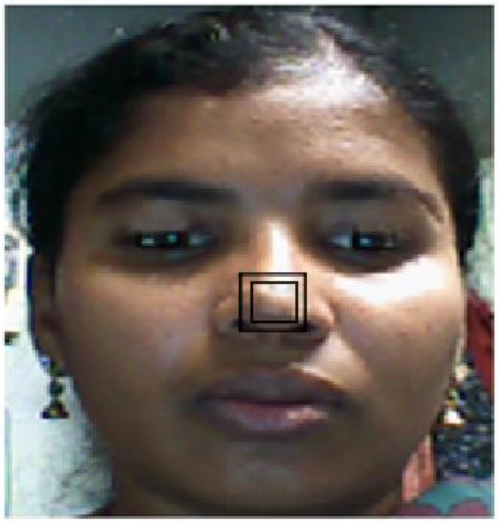


Fig 12: The big square is the nose ROI. The small square is the nose tip new template.

Tracking the Nose Tip

Tracking the nose tip is achieved by template matching, which is inside the ROI, (see fig.12).

Tracking the BTE

Tracking the BTE is achieved by find the 15*15 template which is located between the eyes.

Detecting the Eyebrows

To detect the eyebrow, take a small region above the eye's expected position and threshold it (see fig.13). The threshold should result in points which represent the eyebrow. Next to find the eyebrow line from the set of thresholding points. It is achieved with help of Hough transform technique.

Fig 13: The eyebrow region and thresholding.

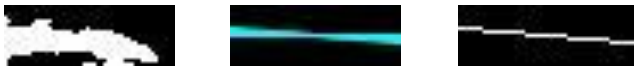


Fig14: Hough transform and the approximation of the result to a single line.

5.2.7. EYES TRACKING

If a left/right blink was detected, the tracking process of the left/right eye is skipped and its location is considered as the same one from the previous frame. To find the eye's new template in the ROI, two methods are combined: the first one is template matching, the second is search in the ROI for the darkest 5*5 region (because the eye pupil is black), then the mean between the two found coordinates are used to find the eye's new location (see fig.15). Finally the video frame overlayed with eyes and nose tip location.

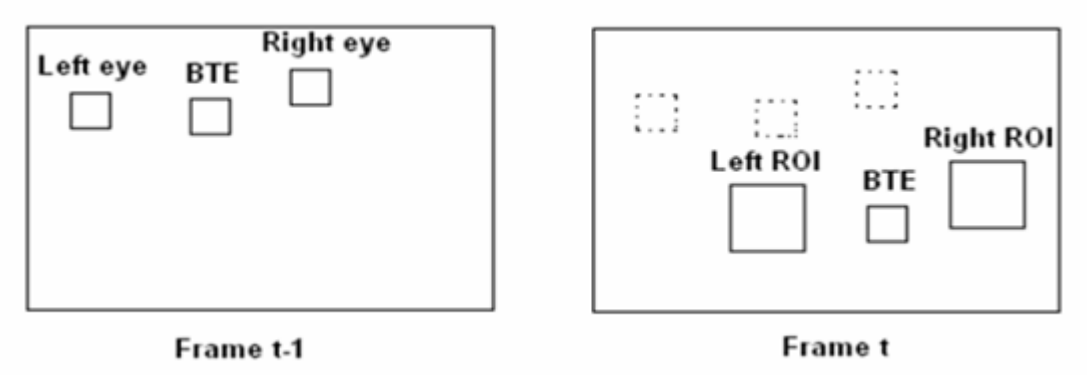


Fig 15: Placing the eye's ROIs relatively to the BTE.

SYSTEM TESTING AND IMPLEMENTATION

6. SYSTEM TESTING AND IMPLEMENTATION

6.1 SYSTEM TESTING

Testing plays vital role in the success of the system. It is an iterative process of both validating functionality and attempting to break the software. It makes a logical assumption that if all the parts of the system are correct. The goal will be successfully achieved once program has been developed, testing begins. It can also be stated as the process of validating and verifying that a software product/application meets the business and technical requirements that guided its design and development, so that it works as expected and can be implemented with the same characteristics. Software testing, depending on the testing method employed, can be implemented at any time in the development process, however the most test effort is employed after the requirements have been defined and coding process has been completed.

The objective of system testing is to find the problems and fix them to improve quality. The testing steps are as follows

1. Unit testing
2. Integration testing
3. Validation Testing
4. Acceptance testing

6.1.1 UNIT TESTING

Unit testing focuses verification effort on the smallest unit of the software. This testing was carried out during programming stage itself.

Each module was tested individually and the corresponding errors generated were reviewed and rectified.In this project, each module is tested separately. The input dataset is given and every module is tested separately to check whether the system provides required output.

6.1.2 INTEGRATION TESTING

Integration testing, also known as integration and testing(I&T), is a software development process which program units are combined and tested as groups in multiple ways. A comprehensive integration testing is carried out using integrated test plans in the design phase of the development as guide to ensure the behavior of functions with datasets.

In this project all modules are integrated and run as a single application. Each module is dependent on one another. The performance of the system will be estimated only after performing all the modules.

6.1.3 VALIDATION TESTING

Validation testing is performed in an approach to verify whether the product functions in a reasonably expected manner by the researchers. After the integration of the modules, the validation test is carried out to the system.

It is a test to determine whether a system fulfills its requirements. In this project, this test involves checking whether the product is being designed in such a way that it performs error free in every phase of the system. It was found that all the modules work well together and meet the overall system function and performance.

6.1.4 ACCEPTANCE TESTING

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at time of developing and making changes wherever required.

6.2 SYSTEM IMPLEMENTATION

Implementation is a process of bringing a developed system into operation and turning it over to the user. Implementation activities extend from the number of plans. The beginning of the phase is to create the schedule and manage the different activities that must be integrated into plan.

Implementation is the stage of the project when the theoretical design is turned in to a working system. The most crucial stage is achieving a successful new system and is giving the user's confidence that the new system will work and be effective in the implementation stage.

In the implementation phase, first, dataset can be selected to begin the processing, as soon as the dataset are selected as input processing in data mining techniques. Thus the implementation phase is completed successfully and dataset are taken as input and the results are displayed.

CONCLUSION

7. CONCLUSION

The main objective of the project is to replace the traditional mouse with the human face as a new way to interact with the computer. This system provides an automatic HCI. HCI system aids to support people who have hand disabilities that prevent them from using the mouse.

The human face of different face is captured with the help of the night vision camera that detect the facial features, in order to communicate with the computer to translate into mouse click events.

This system need an inexpensive webcam, that affords a moderate resolution and frame rate as the capturing device in order to make the ability of using the program affordable for all individuals.

The result shows that it allows users to use their eyes and nose tip to manipulate computer. It detects and tracks the facial features precisely and fast to be applied in real time applications. While these methods have been successful in improving eye detection and tracking, there remains significant potential for further developments.

SCOPE FOR FUTURE ENHANCEMENT

8. SCOPE FOR FUTURE ENHANCEMENT

Feature works may include improving the tracking robustness against lighting conditions. By using more sophisticated and expensive capturing devices such as infrared cameras that can operate in absence of light and give more accurate tracking results.

Other feature work can add some functionality such as adding the double left click and double right click. The double left click is used to detecting the double left eye blink and the double right eye click is used to enabling/disabling the drag mode.

Another opening work can add the voice commands to launch the program. It is used to start the detection process and enabling/disabling the voice command is used to controlling the mouse with the face.

BIBLIOGRAPHY

BIBILIOGRAPHY

JOURNAL REFERENCES

1. Edwards G.J, Cootes T.F, and Taylor C.J, 1998. "Face Recognition Using Active Appearance Models," Proc. Fifth European Conf. Computer Vision, vol. 2, pp. 581-95.

2. Fasel R, Fortenberry B, and Movellan J.R., "A Generative Framework for Real Time Object Detection and Classification," Computer Vision and Image Understanding, Apr. 2005. vol. 98, no. 1, pp. 182-210.

3. Hutchinson T.E, "Human-Computer Interaction Using Eye-Gaze Input," IEEE Tran. Systems, Man, and Cybernetics, Nov. /Dec. 1989. vol. 19, no. 6, pp. 1527-1534.

4. Tomono. A, "Pupil Extraction Processing and Gaze Point Detection System Allowing Head Movement," Trans. Inst. Of Electronics, Information, and Comm. Eng. of Japan, et al 1993, vol. J76-D II, no. 3.

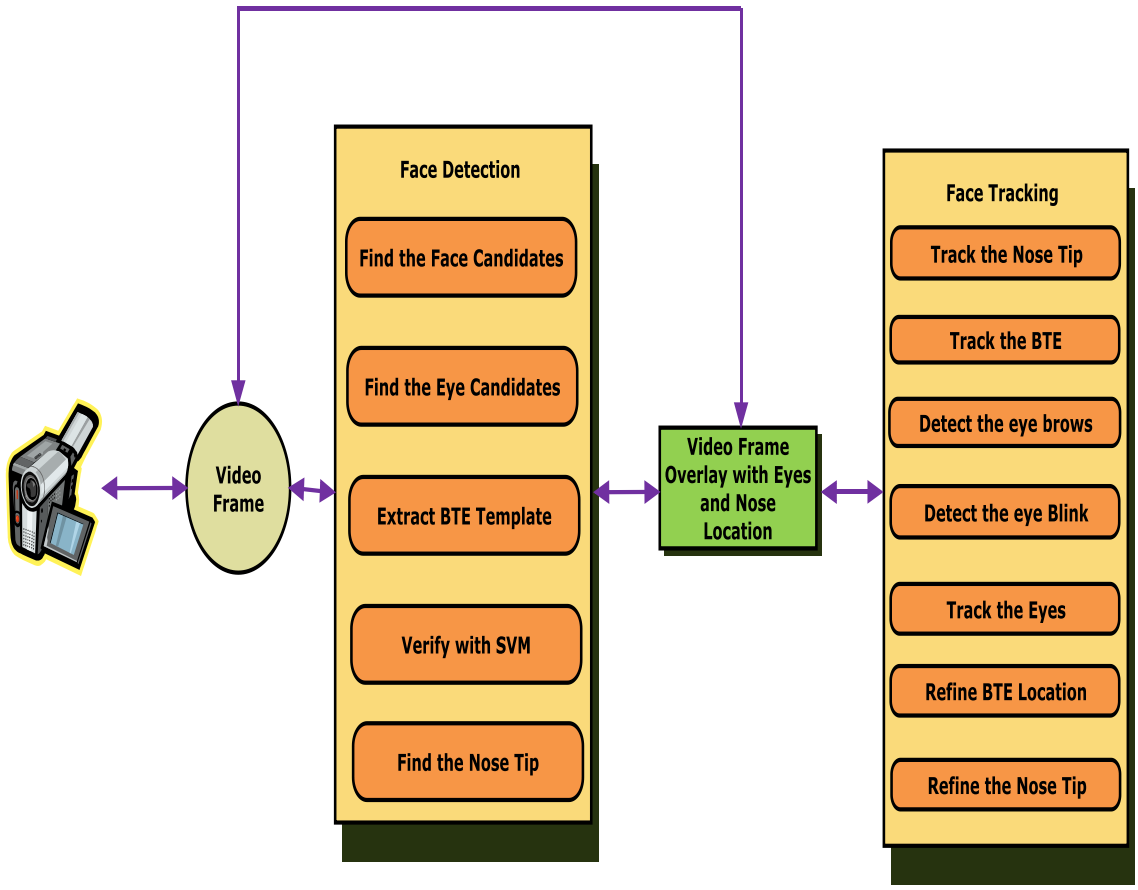
WEBSITE REFERENCES

1. http://arxiv.org/ftp/arxiv/papers/1002/1002.2191.pdf
2. http://www.iu.hio.no/~frodes/unitech10/011-Krolak/
3. https://Eye-blink-detection-system-for-human-computer-interaction.pdf

APPENDIX

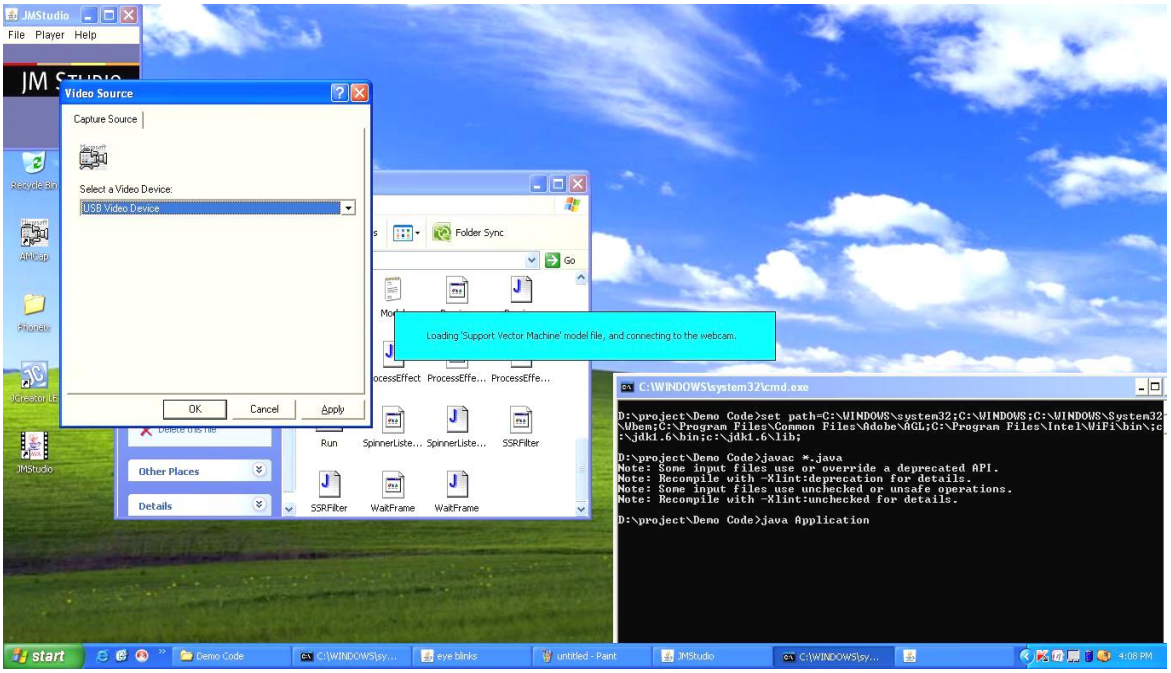
APPENDIX

SYSTEM FLOW DIAGRAM



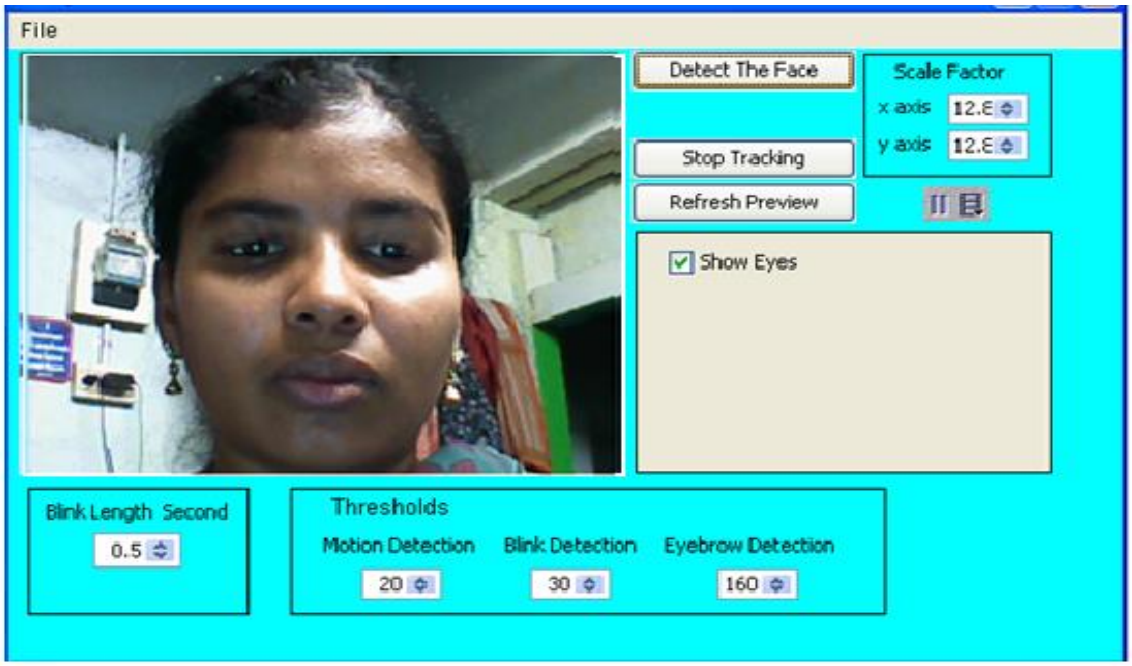
SCREEN SHOTS

Enabling capture device

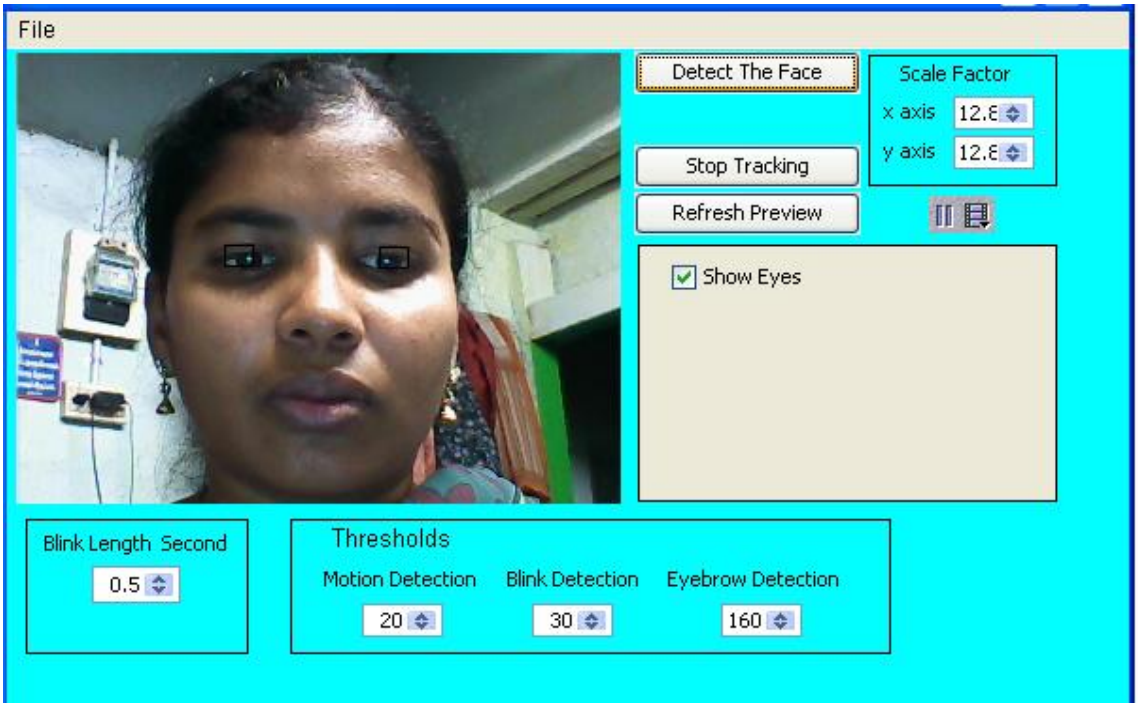


Loading SVM and Web Camera is connected

Video Capturing

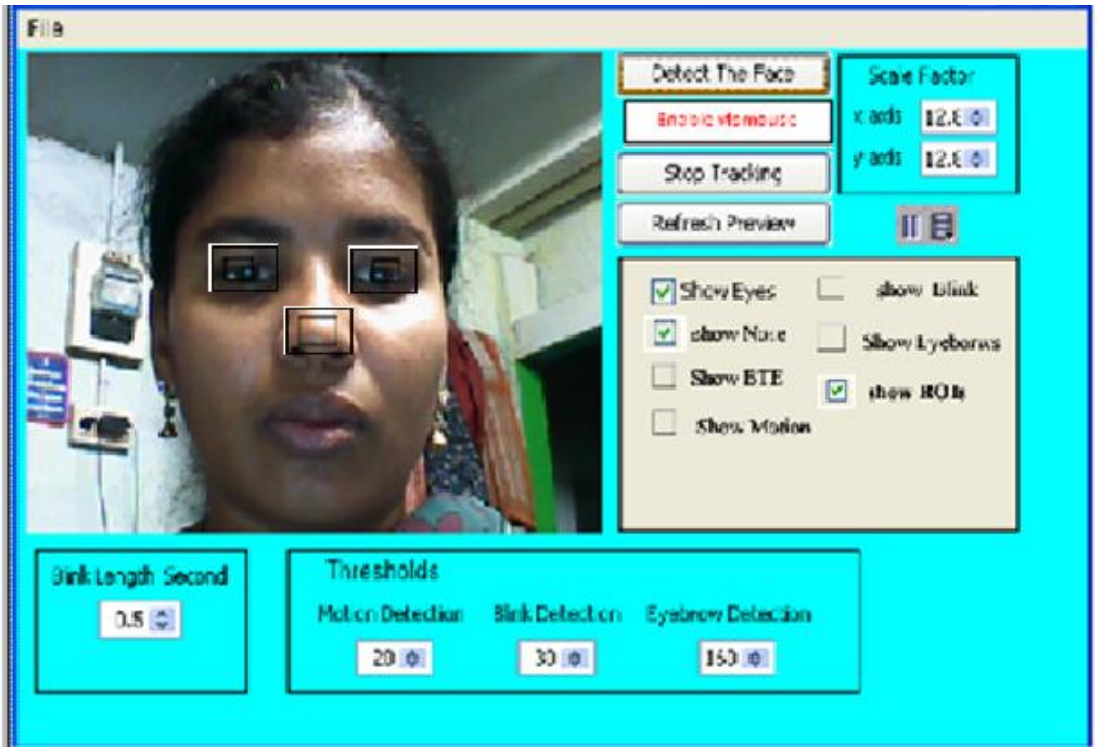


Find Face Candidate and Pupil Candidate



Find The Nose Tip

Show The ROIs



Virtual Mouse Events Enabled