Journal of

Human

Computer

Interaction

Controlling Computer in high accuracy and Fast performance with a standard webcam Without Sensors or Infrared lights

PII: S0047-2727(08)00090-X
DOI: doi: 10.1016/j.jpubeco.2008.05.002
Reference: PUBHCI 2812

To appear in: *Journal of human computer interaction*
Received date: 10 February 2011
Revised date: 21 March 2012
Accepted date: 25 April 2012

Controlling Computer in high accuracy and Fast performance with a standard webcam Without Sensors or Infrared lights

Motaz sabri

Abstract

Through human computer interaction, Machines can help and assist humans with their daily activities in a smoother way; this includes home use, office use for any age level. This project's purpose was to determine an optimized and simplified way to extract eye pupil position in a video frame (Simplified compared to current complex and expensive algorithms) for enhancing human interaction systems by adding vision features. Last year, the Real time auto-focus algorithm has achieved an accuracy of 50% - 75% with high performance cost. Other newly researched algorithms proved close robustness and accuracy with infrared lights and expensive sensors causing a CPU load of 60 -80% (2GHZ Core Duo with 1GB of RAM), a RAM usage of 700MB with 20% tolerance and 700$ for the sensors and IR equipments. This year, the project's purpose was to enhance the performance by replacing the complex hardware equipments and software algorithms with a simple web cam and a set of Fourier filtration steps and AI procedures. A performance evaluation was performed to test the new approach, using a standard webcam of 1.3 megapixels and a 2.2 GHZ CPU core 2 Duo and 1 GB of RAM. The light conditions were changed to be normal, extra light, dim light with a noise –Multiple heads in the background. The results were compared with the latest algorithms that were found during the last year. The CPU load didn't pass 25%, the RAM was used by 300 MB rate and the cost was 5$ - the webcam cost, with high accuracy and tolerance up to 3%. All resources were used in smaller ratio with height accuracy.

Introduction

Since computers are the machines that are widely used among all civilizations and among all ages, their ease of access is one of the most researched fields in human computer interaction field (Michigan State University 2000). Disabled people are 6% of the computer users and they require special interaction mechanism to achieve standard control (Connor et al. 2001). No matter what is the disability level, eyes are the most commonly used control technique that can be used to control computers (Connor et al. 2002). These techniques are based on heavy tools or complex hardware that is expensive and not user friendly enough to be uses for all users (Connor et al. 2005).

Multiple attempts to develop applications, including optical solutions, or controlling the computer by the eye, were performed in the field; most of them share the same properties which are the extra load on computer and being highly affected by the background. Basing the algorithm on extracting the gaze step by step is the major drawback, meaning standard algorithms use complex steps to extract the head then the nose then the eye then the gaze (Benson 2002). Attached sensors to the user's face can be used, achieving higher accuracy but attaching the user with wires is another major drawback. It is important to know that accuracy and robustness are two major roles in generating any human computer interaction (Connor et al. 1998).However, it is not only the algorithm that plays a role in getting an accurate and fast system but artificial intelligence techniques are important factors too (Benson 2002). Those techniques depend on the system that researcher plans to map.

When considering how to achieve robustness and accuracy, it is also important to take into account the type of environment in which the user will be located in, the screen resolutions or the variation of the eye shape (Connor et al. 2006). The algorithm should find something in common for all users, extract the gaze location then map it coordinates- where it is looking- to the screen. Ideally, eye illumination is a good factor to follow.

Eye illumination is the milestone characteristic of the human eye ,or in other word the red eye phenomena that surrounds the eye pupil with red circle due to the fast exposure of light after taking a camera shot.

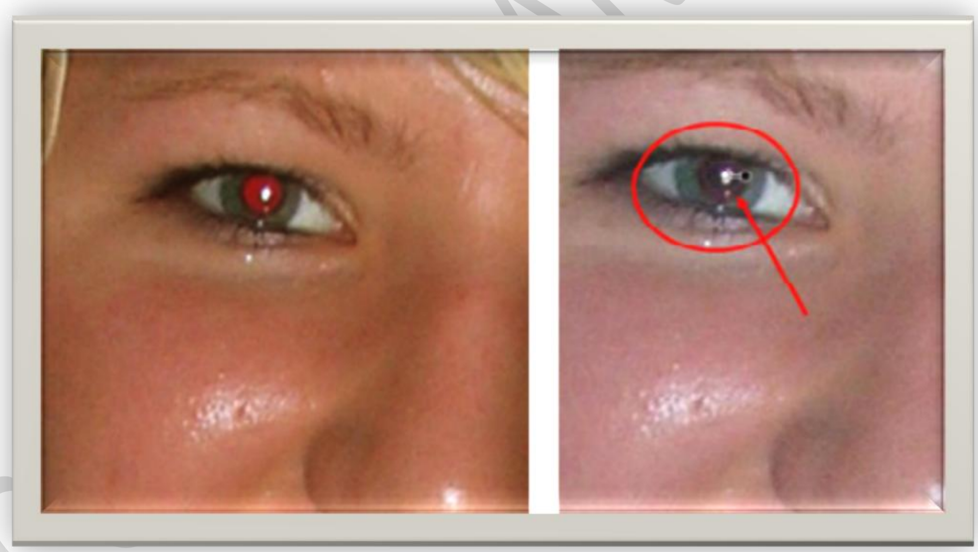


Figure1. Red eye phenomena during image capture (adapted from Jerusalem et al. 1992).

Red Eye is a normal phenomenon and is caused by light from the cameras flash reflecting of the back of the retina and back into the camera lens. The resulting image will show the subject with red eyes because the retina is full of blood vessels and appears to glow red. Red eye is increased when the subjects pupils are wide open (as in a darkened room) which can cause more of the red eye effect.

Knowing that illumination always occurs in all camera frames with no infrared rejection in normal lights condition then in order to invest with this property to distinguish the eye focus we need special image processing filtration. Most of the processing happens in frequency domain, so in order to convert images from spatial domain to frequency domain Fourier transformation is used. Fourier analysis is used in image processing in much the same way as with one-dimensional signals. This will be implemented by 4 steps for gaze tracking and one step for mapping the gaze location to the screen. (Digital Signal Processing Elion 2003)

The first step in preparing the Camera frame to perform Laplace integral transform filter to every captured image, The Laplace integral transformation filter is a set of three by three kernels which approximate the LP integral transform operator, where an LP integral transform operator is defined as the sum of the partial second derivatives in x and y direction simultaneously. The LP integral transform operator is presented as an isotropic edge detector for images with low noise. The LP integral transform of an image highlights regions of rapid intensity change. The input for LP integral transformation filter is list of RGB images to be convolved and the output is a list of images that look like a night vision images, the number of output images must equal the number of input images. If the input image name equals the output image name the convolved image will replace the input image. The LP integral transform L(x, y) of an image with pixel intensity values I(x, y) is given by:

L(x, y) = d2 I/ dx2 + d2I/ d y2.

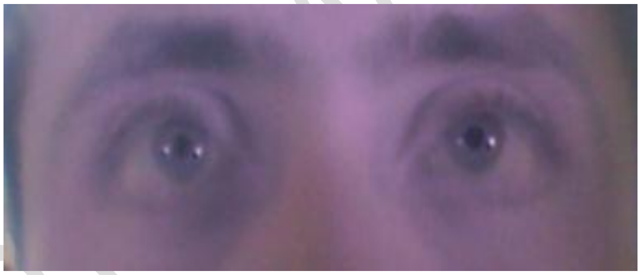


Figure2. Frame received from a 1.3 Mega pixel webcam in standard light more.

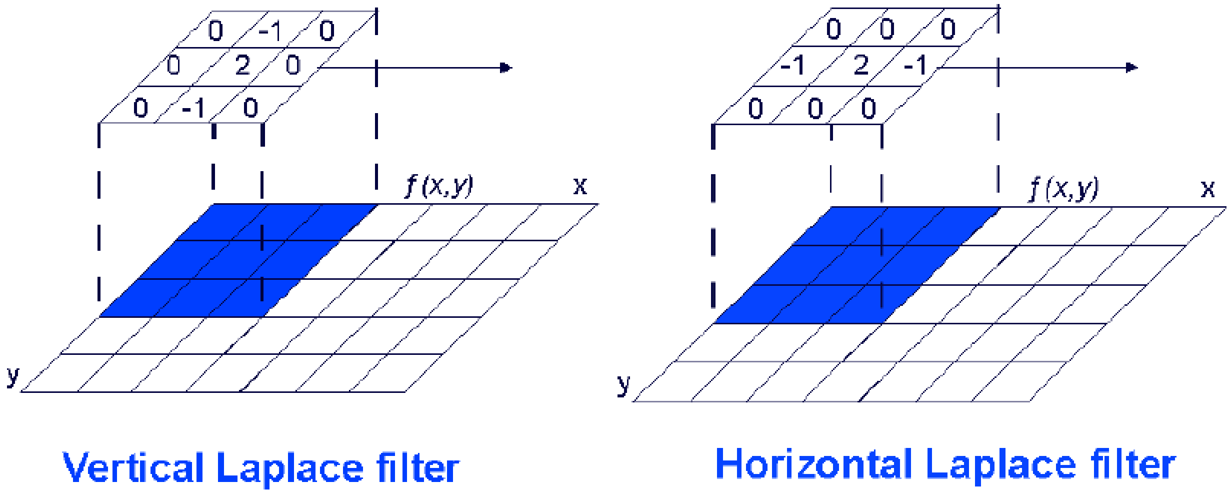


Figure3. Laplace Vertical and Horizontal filtration (adapted from Tortora et al. 1992).

As it is known the input image is represented as a set of pixels "discrete pixels", to find a convolution kernel that will approximate the second derivatives in the definition of the LP integral transform, using one of these kernels, the LP integral transform can be calculated using standard convolution methods. The LP integral transform filters are high-pass filters that acts as a local edge detector as shown in figure 4.

A kernel is a smallish matrix of numbers that is used in image convolutions. Differently sized kernels containing different patterns of numbers give rise to different results under convolution.

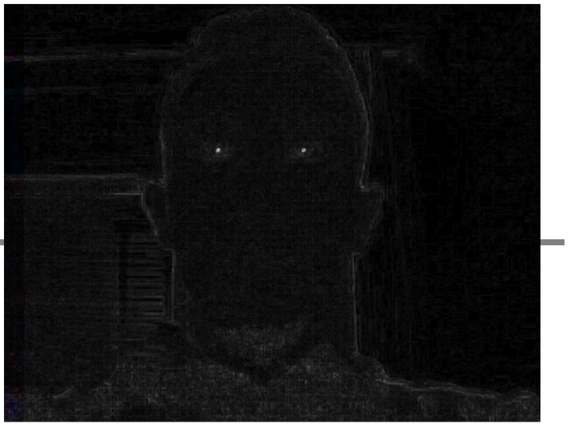
Second step is convolution, which is a mathematical operation that is fundamental to many common image processing operators to produce a third array of numbers of the same dimensionality.

The convolution is performed by sliding the kernel over the image, generally starting at the top left corner, so as to move the kernel through all the positions where the kernel fits entirely within the boundaries of the image. Each kernel position corresponds to a single output pixel, the value of which is calculated by multiplying together the kernel value and the underlying image pixel value for each of the cells in the kernel, and then adding all these numbers together.

O61 = I61* K1 + I62* K2 + I63 *K3 + I70* K4 + I71* K5 + I72* K6 + I79* K7 + I80*K8 + I81* K9

If the image has M rows and N columns, and the kernel has m rows and n columns, then the size of the output image will have "M - m + 1" rows, and "N - n + 1" columns.

Third step is



Monochromatic conversion as shown in figure 5; the image is modified in which all colors are shades of gray. The reason for differentiating the colors in image is that less information needs to be provided for each pixel. In fact a gray color is one in which the red, green and blue components all have equal intensity in RGB space, and so it is only necessary to specify a single intensity value for each pixel, as opposed to the three intensities needed to specify each pixel in a full image. Often, Monochromatic intensity is stored as an 8-bit integer giving possible different shades of gray from black to white. If the levels are evenly spaced then the difference between successive gray levels is significantly better than the gray level resolving power of the human eye. In my project design the Monochromatic stage come directly after performing a LP integral transformation filter on the captured image, we will convert the resulted image from the LP integral transform stage (night vision image) to Monochromatic image were all the colors of pixels in the resulted image will be between the ranges of 0-255. This will help us to remove the UN wanted data from the image and also help us to remove the noise in the images that result from the LP integral transform filter.

Figure4. Output of the Laplace Image conversion.

Figure5. Output of the Convolution and Monochromatic conversion.

Finally the color conversion, the main idea behind this stage is to invert the color in an image. In my project the image will be in Monochromatic, so The Invert stage is used to invert color values in pixels in Monochromatic. Invert simply switches 0 to 255 and 255 to 0 and likewise switches mirror-fashion all values in between. In a simple black and white image, using 0 for black and 255 for white, a near-black pixel value of 5 will be converted to 250, or near-white. The result is a photographic negative image. For RGB color images or any other image with more than one channel this process is applied to each channel. If a pixel has a high blue value and low red and green values, the blue value will end up low and the red and green values will be high to result in a yellow tone, followed by the edging stage which is the last stage of the phase; it will ensure that the accuracy is the highest we can get; the edging stage will distinguish the rapid variation of colors within the frame depending on the threshold value that is set while performing the canny transmission

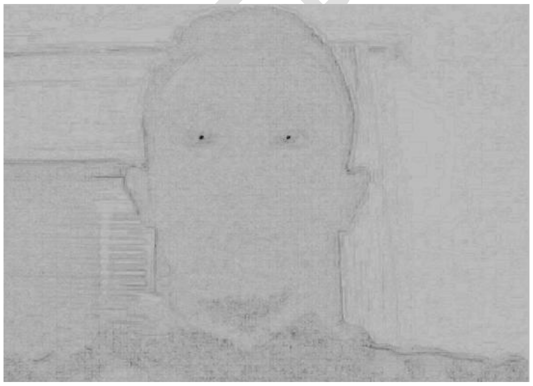


Figure6. Output of the Color conversion.

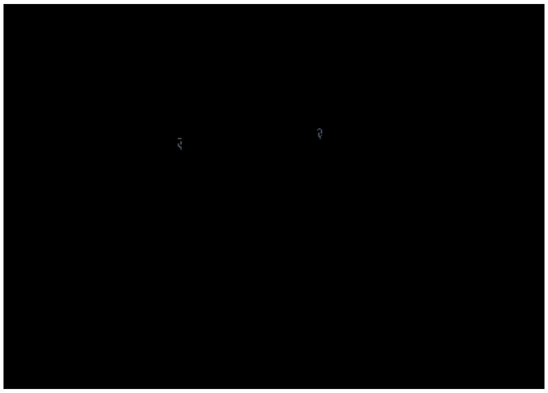


Figure7. Output of the edging conversion

the second phase will be reading the white dots location –In Figure 7- and map them to the screen by finding the frame window size and map it to the screen resolution size by multiplying with the proper factor, the absence of the white pixels for certain amount of time (1 second for clicking and 2 seconds for double clicking) will indicate an operation that will be mapped to the screen, that might be a left click, right click and double click, a hybrid can be obtained by clicking and moving for drag and drop or scrolling.

The objectives of my project were to: 1) Find an accurate algorithm to detect the eye within the camera frame; 2) Determine the best optical filtration sequence to preserve high accuracy and best performance; 3) Determine the optimum Laplace filtration threshold to achieve lowest error rate; 4) Determine the optimum convolution filtration threshold to achieve lowest error rate;5) determine the optimum edging and conversion overlapping ratio and thresholds to achieve lowest error rate; 6) determine the best mapping factor between the user gaze and screen resolution. For my project I used candidates with different eyes sizes and colors in west bank, Palestine. Candidates were from universities, schools and moles.

2 years ago, I was able to complete objectives 1 through 5. Compared to algorithms found through my research, mine showed higher accuracy with the lower price and better performance. The best filtration sequence was Laplace followed by convolution then frame conversion and edging. Laplace filtration threshold were IPL_DEPTH_16S then IPL_DEPTH_8U. The optimum convolution threshold is between 400 and 810 with aperture size 3. Both edging and conversion overlapping should be 100% and their thresh-holdings are from 0 to 255 for both. Since these results showed low CPU and RAM usage with high accuracy for standard light, average dim and extremely bright conditions for limited number of candidates -200 user- , different experiments were carried out for wider range of users, none of them wear glasses because system in this case can determine the eye location but can’t determine where the user is looking due to the high noise results from the light reflection over the glasses.

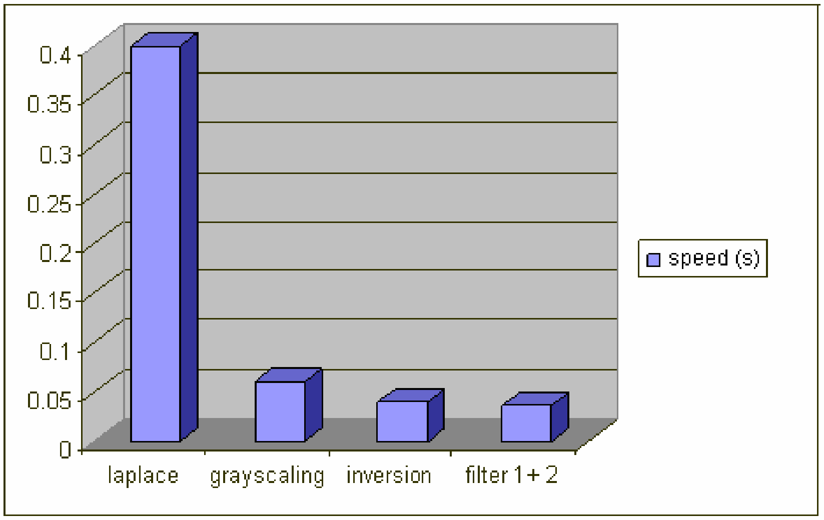


Figure8. Time consumption over system stages.

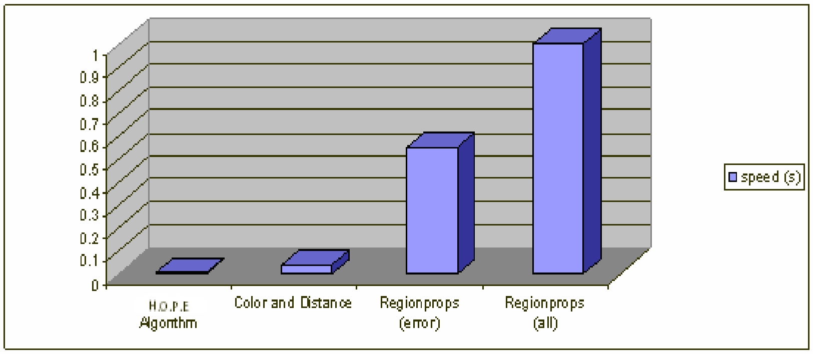


Figure9. Pupil detection algorithm comparison.

Laplace is the most time consuming step among all the others with 0.4 second per frame the rest are extremely acceptable time wise. Time consumption over system stages are shown in Figure 8. The graph shows a usage of 0.4 seconds for the Laplace step relatively longer that the 0.05 seconds for the next slower step (gray scaling) then shows a slight decrease in the processing time for the next steps. Compared to other gaze/Pupil detection techniques as shown in figure 9 H.O.P.E: hyper optical pointer extension is showing the fastest performance, while development being performed in a real time programming language –Open-CV, each frame analysis will take time but through a continues frame processing each step will go through a pipe line treatment and therefore minimized the required time.

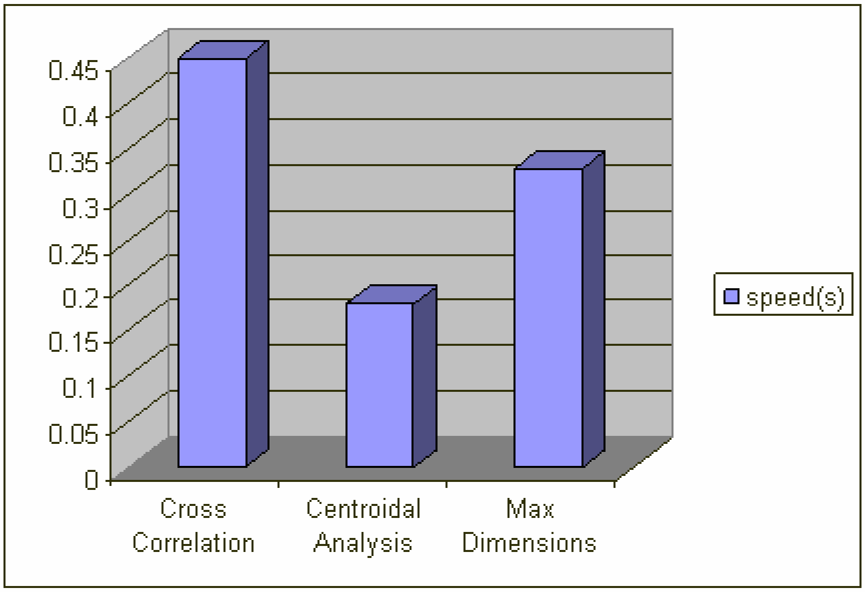


Figure10. X Y calculations algorithms comparison.

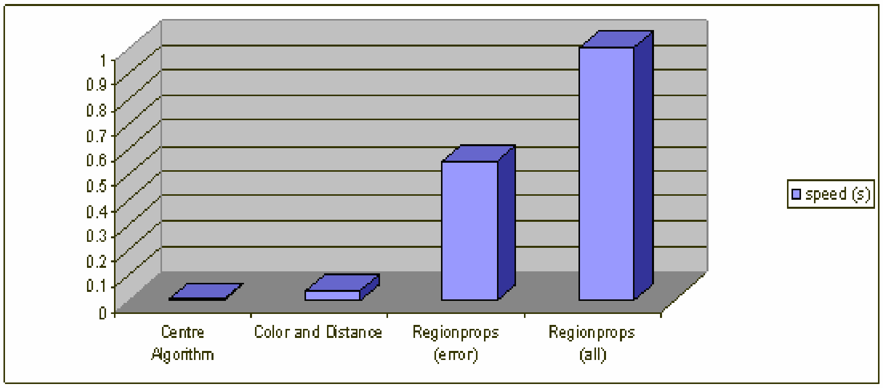


Figure11. Error Detection algorithm comparison.

X, Y calculations algorithms step is embedded at the filtration process of my algorithm, so I had avoided this step and gained extra time, figures 10 shows the advantage of X,Y time cost, for error detection phase it contains filtration from noise that came with the frame form the background; with my project I used color and distance filtrations.

Figure 11 shows the used algorithms on recent researches, numbers shown are obtained from an independent source from the internet, through it we can see that I had developed the optimum solution for each step to ensure the accuracy and the high performance.

For the mapping process I used the ratio/factor multiplication, once an operation is detected (blinking for clicking) the algorithm reads the objects under the mouse and determine which one has the highest possibilities to be a subject for the operation, as an example the mouse is in a region that is near an icon and start menu, and the user sends a double click command -blink for more than 1 second- it is most likely that he wants to click on the icon on the start menu, the same applies for more and more objects within the screen based on a simple judgment algorithm.

The last year, I completed the rest of my objectives, 6. The best mapping ratio is dynamic and was based on the user’s eye width over the screen width which will give the horizontal ratio, on the same hand the height of the user eye over the screen height which will give the vertical ratio. The judgment algorithm will assist easing the operation mapping and supports the user with a faster and friendlier user interface.

Methods and Materials

Gaze extraction by H.O.P.E algorithm

In order to determine the best optical filtration sequence, 2 infrared illuminators were used to ensure a continuous red eye effect in the testing phase only. One camera without infrared rejection feature implemented in its hardware and a standard PC. After installing Open-CV in that PC, the testing process started to extract the threshold values. Initially the noise ratio was extremely high, each frame contained a lot of eye candidates from background and surrounding objects, the optimal thresholds were determined last year, for about 6 months to ensure that all noise factors were eliminated from each frame.

In order to make certain that these noise factors were eliminated or reduced to minimal in all frames, all light conditions were tested: extra bright lighting, dim lighting and standard room lighting conditions. First, I was the only subject to all those tests. The tests started for single images instead of continuative frames from the webcam. Each image was processed individually by filtering its contents –after performing the 4 optical filtration steps -through Photoshop and perform canny filtration to show all similar pixels colors. To minimize the noise – pixels with the same eye colors in the background multiple approaches were evaluated, the condensity and positioning were the approaches with the highest accuracy and reliable output. The occurrence of the eye color indicated an opened eye. if the eye pixels were available, then its location can go to the mapping process otherwise a counter is initiated and if it passes the 1 second time frame it will be considered as an operation(Clicks or double click).

Mapping coordinates and operations to the PC

Mapping is a mathematical relation such that each element of a given set (the domain of the function) is associated with an element of another set (the range of the function).comparing the eye width and height to the screen width and height will result in a big ratio that will cause very low accuracy. In order to ensure high accuracy combined techniques were used, mapping is used and another assisting step is added, which is shifting; so instead of mapping each coordinate separately from the other set of gaze positions, the amount of gaze displacement between the current and the previous positions is used (vertically and horizontally) then this ratio is multiplied by the factor we discussed and the current courser position is shifted according to the resulted value. The accuracy of pointing to components is high that it can point over the on-screen keyboard buttons pretty smoothly. Once a destination region is indentified a collection of all objects that occur to be in this region is gathered, each object has its operation occurrence percentage, like hovering, highlighting, clicking, double-clicking, dragging, scrolling, tool tipping and almost all windows components functions. Any action that the user intends to use will go through the probability tree and artificially determine the component to be the subject of that operation.

So basically the flow chart of H.O.P.E operations and steps is as shown in figure 12:

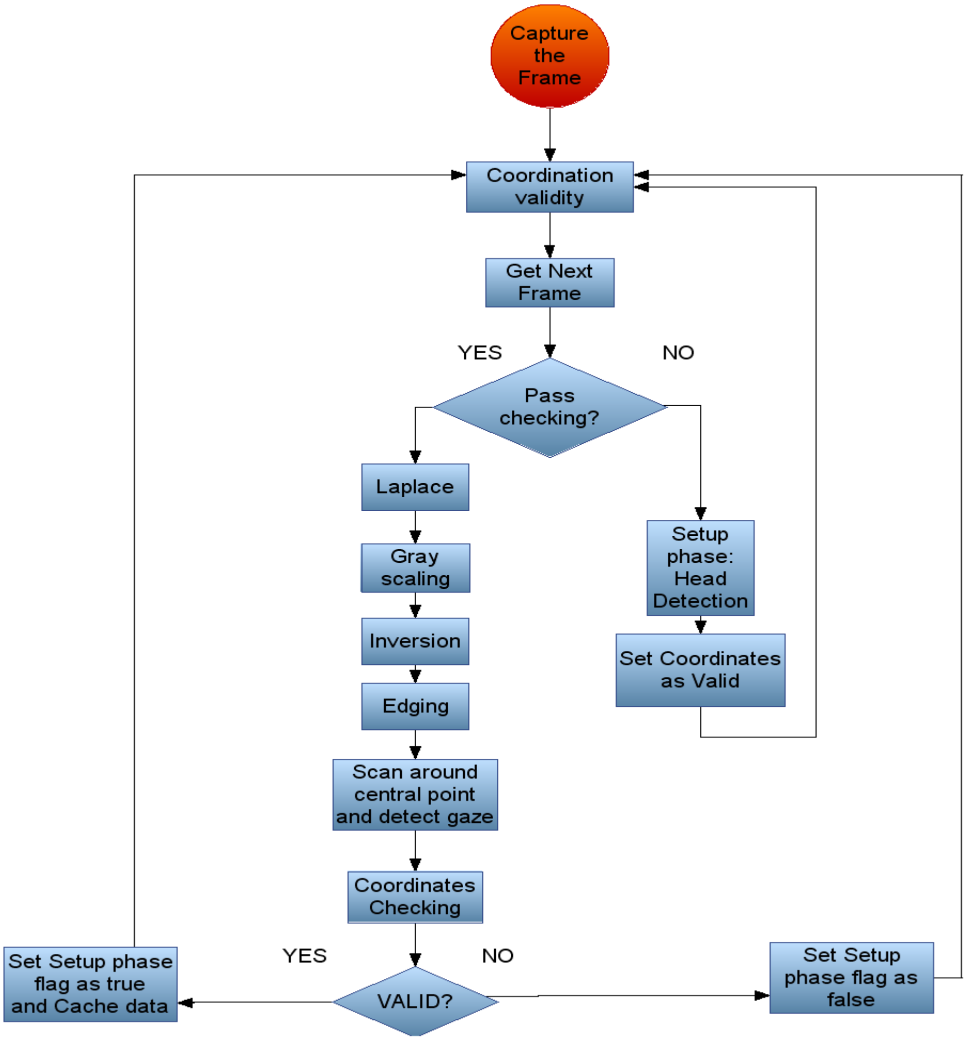


Figure12. Live frame processing stages flow chart.

And below in figure 13 is another chart that shows the extraction and mapping flow charts:

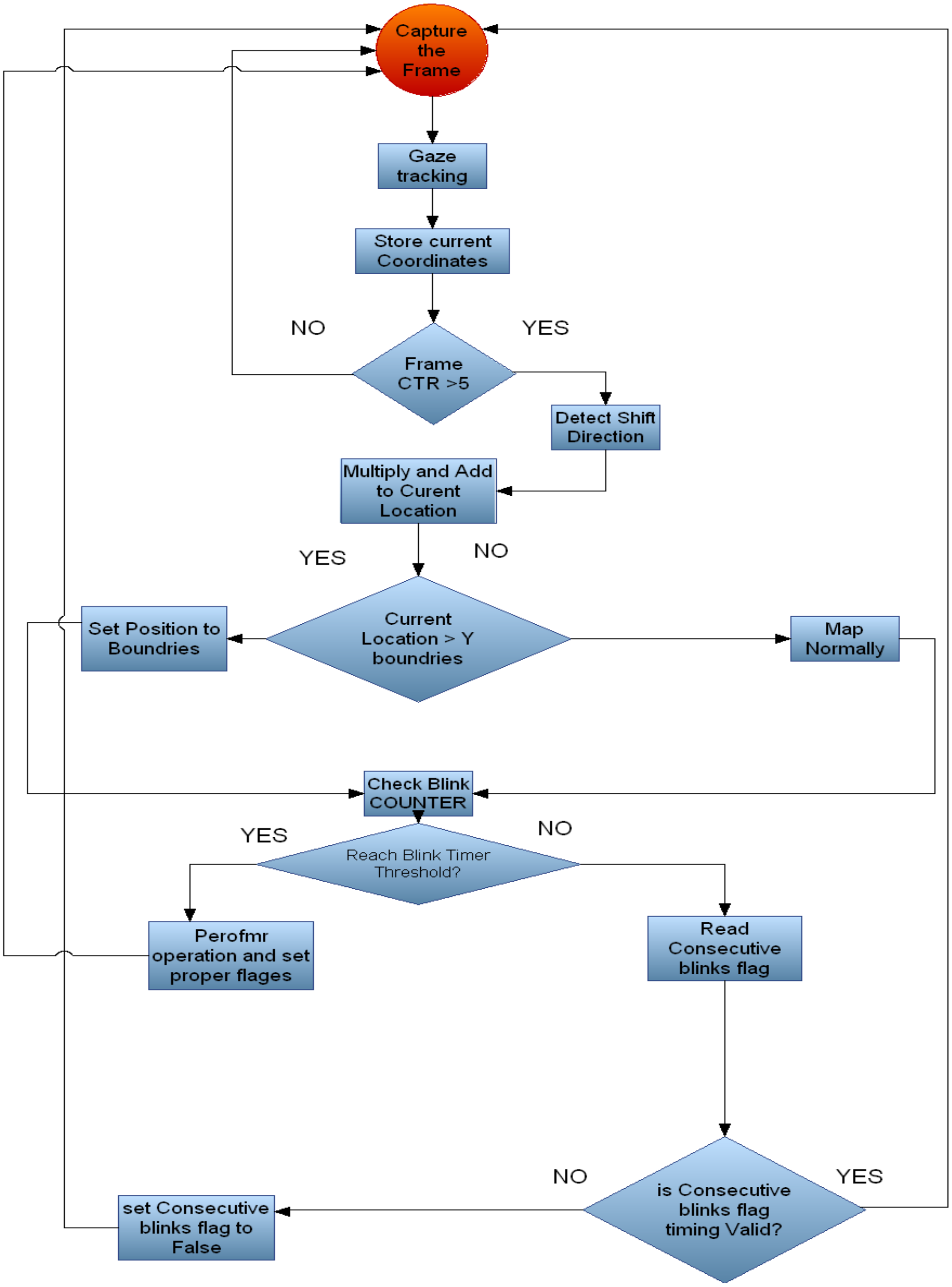

Capture the Frame
Gaze tracking
Store current Coordinates
NO
YES
Frame CTR >5
Detect Shift Direction
Multiply and Add to Curent Location
YES
NO
Current Location > Y boundries
Set Position to Boundries
Map Normally
Check Blink COUNTER
YES
NO
Reach Blink Timer Threshold?
Perofmr operation and set proper flages
Read Consecutive blinks flag
NO
YES
is Consecutive blinks flag timing Valid?
set Consecutive blinks flag to False

Figure13. Flow chart of H.O.P.E.

Implementations from the previous diagrams (Figures 12 and 13) were used to calculate the gaze position in camera frames and map them accurately in the screen using shifting. The performance and accuracy were compared using CPU Benchmark (Windows PC, Core Due 2GHZ and 1GB of ram) and 95% Confidence testing environments with different lighting conditions to determine if they were statistically different.

Results

Hyper optical pointer extension –Gaze extraction using Red eye and optical Fourier filtration

The high speed gaze extraction mechanism was achieved after 6 months of testing. The Fourier filtration tests and combinations indicated that the gaze can be easily distinguished by passing through optical filters with pre defined thresholds because the eye reacts with special illumination when subjected to a proper amount of light. Also, considerable amount of noise in the background appeared and was eliminated by pixels condensity and horizontal alignment checks for the eyes. This meant that any human eye with proper amount of light can be extracted from any camera frame with high speed, and then can be mapped to any other possible human computer interaction based application.

The best filtration sequence was Laplace followed by convolution then frame conversion and edging. Laplace filtration threshold were IPL_DEPTH_16S then IPL_DEPTH_8U. The optimum convolution threshold is between 400 and 810 with aperture size 3. Both edging and conversion overlapping should be 100% and their thresh-holdings are from 0 to 255 for both. Since these results showed low CPU and RAM usage with high accuracy for standard light and extremely dim and extremely bright conditions for limited number of candidates -200 user- , different experiments were carried out for wider range of users, none of them wear glasses.

Discussion

Since the red eye is a phenomenon that occurs in all captured images and cameras, the gaze extraction and location detection can be achieved in lighting conditions (Dim, bright and normal). This supports my hypothesis that the proper Fourier image processing with proper stages and steps with proper thresh-holding would give accurate results after noise elimination to the eye position in any camera frame. If these coordinates were mapped to the proper system, it would be possible that any Human- machine interaction will to be richer and smoother. H.O.P.E is more reliable to detect gaze, eliminate most of background noise, work in almost all lighting conditions with proper extraction speed, meaning it get results in real time , researchers could combine this algorithm with almost any HCI system, finding a balance between cost effectiveness and the time needed to get use focus.

Since the mapping was significantly able to map small region (Eye movement zone) to a very wide and dynamic region (Any monitor), the users will be able to use H.O.P.E at monitors, projectors or wide screen devices. Therefore, when choosing a gaze extraction method, researcher will need to consider the fact that H.O.P.E can support both accuracy and low CPU load with cheaper cost.

Future studies can focus on more applications for human needs –specially disabled people that can't use computers or any machine that has standard input mechanism. Smarter solutions for smart homes control, car accidents protection and adaptive computer gaming can also make good use from H.O.P.E. In addition, lab safety and online human eye signature can be possible candidate for H.O.P.E usage.

Conclusion

Before starting to think in a way to determine programmatically the eye location in an image, one must understand the characteristics of the eye in all of images. It is important that the chosen techniques have characteristics suitable for the possible users. Since the red eye phenomena occur for all humans with proper amount of light, it may be useful to extract gaze location for any 2 eyed human. While the cost of this approach is cheaper than the sensor based methods and the IR methods, H.O.P.E method has the advantage of being able to get information faster. If this algorithm is to be used, researchers will need to determine whether accuracy, speed and user friendliness is more important than complex sensor, IR's and attached devices or not.

Acknowledgements

I would like to thank my mentor, Dr. Maya Nasrawi, for her advice, assistance with lab work, and guidance. Many thanks to: Dr. Rakan Al Zoubi for allowing me to work in his advice and lab work support; Ms Marian George for providing me with users and beta testers for the algorithm; Ms. Nova for her guidance, proofreading, and support; Mr. Rany Zeta for his help with statistical analysis; and Mr. Nadine Olwan for her proofreading and assistance with lab work.

Literature Cited

[1] Eye gaze Eye tracking Systems. What is Eye Tracking good for?
http://world.std.com.
Retrieved September. 2007

[2] History of Eye Tracking. LC Technologies
http://www.eyegaze.com
Retrieved Septemper. 2007

[3] Quick Cam® Orbit AF. Logitech Webcam

http://www.logitech.com
Retrieved October. 2007

[4] A4 technology. Night vision webcam.
http://www.a4tech.com/en/product2.asp?CID=77&SCID=89&MNO=PK-333MB.
Retrieved Sept. 2007.Retrieved Sept. 2007.

[5] Features of C++ Language. The C++ Resources
http://default.co.yu/~xxx/C++/description
Retrieved Septemper. 2007

[6] Programming with OpenCV. Introduction to OpenCV
http://developer.intel.com
Retrieved November. 2007

[7] EYE STRUCTURE AND FUNCTION
http://www.myeyeworld.com/files/eye_structure.htm
Retrieved Septemper. 2007

[8] The Human Visual System
http://www.physics.utoledo.edu/~lsa/_color/17_eye.htm
Retrieved Septemper. 2007

[9] Evolution of the eye
http://en.wikipedia.org/wiki/Evolution_of_the_eye
Retrieved Septemper. 2007

[10] Discrete Fourier Transform
http://local.wasp.uwa.edu.au/~pbourke/other/dft/
Retrieved January. 2008

[11] Mathematical Sciences
http://www.maths.abdn.ac.uk/
Retrieved January. 2008

[12] M. Sonka, V. Hlavac, and R. Boyle, (1999). Image Processing, Analysis, Machine Vision. Brooks/Cole Publishing Company, second

[13] Todd Randall Reed, Boca Raton: CRC Press _ c2005 272 p.: ill. ; 25 cm. Digital Image sequence processing, compression, and analysis

[14] OpenCV
http://www710.univ-lyon1.fr/~bouakaz/OpenCV-0.9.5/
Retrieved October. 2007
[15] Linear Image Processing
http://www.dspguide.com/CH24.PDF

Retrieved October. 2007