International Journal of Information and Computation Technology.
ISSN 0974-2239 Volume 3, Number 4 (2013), pp. 267-278

http://www. irphouse.com /ijict.htm

Blinking Analysis Based on High Frame Rate Video for HCI

R.S. Hemachandran[1], A. Gopinath[2], C.R. Santosh[3] and M. Praveen Kumar[4]

[1]Information Technology, Adhiparasakthi Engineering College, Melmaruvathur-603319, Tamil Nadu, India.
[2]Information Technology, Adhiparasakthi Engineering College, Melmaruvathur-603319, Tamil Nadu, India.
[3]Information Technology, Adhiparasakthi Engineering College, Melmaruvathur-603319, Tamil Nadu, India.
[4]Information Technology, Adhiparasakthi Engineering College, Melmaruvathur-603319, Tamil Nadu, India.

Abstract:

The proposed system enhances the algorithm for blink detection based on visual signs that can be extracted from the analysis of a high frame rate video is presented. A study of different visual features on a consistent database is proposed to evaluate their relevancy to detect blinks by data-mining. Then, an algorithm that merges the most relevant blinking features (duration, percentage of eye closure, frequency of the blinks and amplitude-velocity ratio) using SVM is proposed. The main advantage of this algorithm is that it is independent from the driver and it does not need to be tuned. Moreover, it provides good results with more than 80% of good detections of voluntary blink states which can be implemented for physically challenged peoples for operating the PC using Eyes as a primary input device

We present a system that simultaneously tracks eyes and detects eye blinks. Two interactive particle filters are used for this purpose, one for the closed eyes and the other one for the open eyes. Each particle filter is used to track the eye locations as well as the scales of the eye subjects. The set of particles that gives higher confidence is defined as the primary set and the other one is defined as the secondary set. The eye location is estimated by the primary particle filter, and whether the eye status is open or closed is also decided by the label of the primary particle filter. When a new frame comes, the secondary particle filter is

reinitialized according to the estimates from the primary particle filter. We use auto-regression models for describing the state transition and a classification-based model for measuring the observation. Tensor subspace analysis is used for feature extraction which is followed by a logistic regression model to give the posterior estimation. The performance is carefully evaluated from two aspects: the blink detection rate and the tracking accuracy. The blink detection is achieved using videos from varying scenarios, and the tracking accuracy is given by using the Logitech motion capturing system.

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. One of the promising fields in artificial intelligence is HCI which aims to use human features (e.g. face, hands) 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.

With the growth of attention about computer vision, the interest in HCI has increased proportionally. As we mentioned before different human features and monitoring devices were used to achieve HCI, but during our research we were interested only in works that involved the use of facial features and webcams. We noticed a large diversity of the facial features that were selected, the way they were detected and tracked, and the functionality that they presented for the HCI.

Researchers chose different facial features: eye pupils, eyebrows, nose tip, lips, eye lids' corners, mouth corners for each of which they provided an explanation to choose that particular one. Different detection techniques were applied (e.g. feature based, image based) where the goal was to achieve more accurate results with less processing time.

Introduction to HCI:

HCI (human-computer interaction) is the study of how people interact with computers and to what extent computers are or are not developed for successful interaction with human beings. A significant number of major corporations and academic institutions now study HCI. As its name implies, HCI consists of three parts: the **user**, the **computer** itself, and the ways they work together.

2. Motion Analysis:

During the first stage of processing, the eyes are automatically located by searching temporally for "blink- like" motion. This method analyzes a sequence of the user's

involuntary blinks and exploits the redundancy provided by the fact that humans naturallyblink regularly. The bi-directional difference image

[D]i;j = j([Ft]i;j □ [Ft□1]i;j)j

is formed from previous frame image Ft□1 and current frame image Ft for all pixels (i; j) in order to capture both increasing and decreasing brightness changes. The difference image is threshold to produce a binary image representing regions of significant change, i.e. motion, in the scene. Next the image undergoes erosion with a cross- shaped convolution kernel in order to eliminate spurious pixels generated by phenomena such as flickering lights, high-contrast edges, or arbitrary jitter.

For example, the sharp contrast along the edge between the face and the hair or shadow on the neck permits only a negligible amount of movement to result in a significant brightness change

Background Suppression:
The background suppression module selects foreground points at each time t by computing the distance, in the RGB color space, between the current frame It and the current background model Bt, obtaining the difference image DIt, defined for each image point

Blob Analysis and Mining:
With the help of 8-connectivity, the system detects all the blobs of connected candidate moving points. Blobs with small area are discarded as noise while the rest are validated as actual MVOs (Moving Visual Object).

With every MVO we compute its average speed by means of frame-difference. By using a threshold on *AS* we separate the MVO as a moving MVO and stopped MVO. Background Update:

The background model is computed as a statistical combination of a sequence of previous frames and the previously computed background (adaptability). The statistical function used is the median. In order to improve the background update, the system use selectivity, so the background is updated.

3. Color Analysis Subsystem :

The color analysis subsystem are categorized as follows
Skin Detection:
To reduce the search area for face detection (increasing the processing speed and decreasing the false detection rate), the system used simple rules to verify if a point belonging to MVO has a skin color or not, using the pixels' normalized RG color space information.

Skin's Blob Mining:
A region-based labeling is preformed to compute the connected skin's blobs of skin pixels Blobs with small area are discarded as noise.

Face Analysis Subsystem:

Face Detection:

The implemented detection subsystem detects frontal faces with small in-plane rotations and it is based mainly on radial basis function neural network.

This face detector corresponds to a cascade of filters that discard non-faces and let faces to pass to the next stage of the cascade.

Face detections belonging to consecutive frames were considered to be the same face, by applying the same heuristic used to process overlapping detections

Overlapping Detections Processing:

Face windows obtained in the face detection module are processed and fused for determining the size and position of the final detected faces. Overlapping detections were processed for filtering false detections and for merging correct ones.

Face Identification:

This module was used to filter false detections. This filtering corresponds to an inter-frame operation, while the filtering applied in the Overlapping Detections Processing module.

4. 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. 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 around the eye.

Methodology:

Implementation and Methodology:

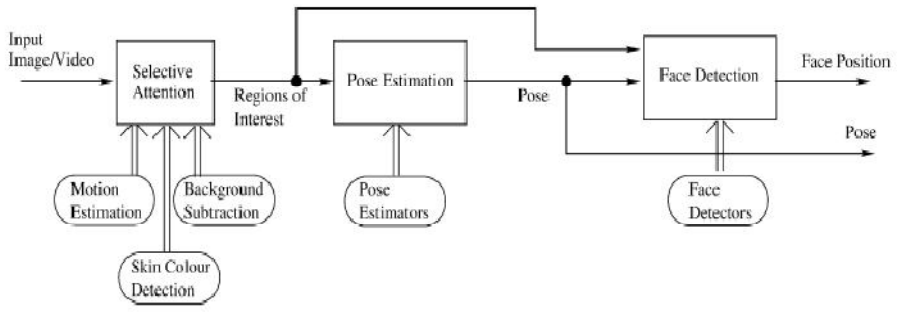

Fig. 1: Implementation of Face Detection.

- Perform motion estimation, skin colour detection or background subtraction on input images or an image sequence to locate ROIs which may contain faces;
- Exhaustively scan these image regions at different scales;
- For each image patch from the scan, estimate the 'pose' (tilt and yaw) using pre-trained pose estimators;
- Choose an appropriate face detector according to the estimated pose to determine if the pattern is a face;

5. Refine the results of detection:

Implementation is the stage in the stage in the project where the theoretical design is turned into a working system. The most critical stage is achieving a successful system and giving confidence on the new system for the users that it will work efficiently. It involves careful planning, investing of the current system, and its constraints on its implementation, design of methods to achieve the change over, and evaluation of the change over method.

The implementation process begins with preparing a plan for the implementation of the system. According to the plan, the activities has to be carried out, discussion has been made regarding the equipment, resources and how to test the activities**.**

The coding step translates a detail design representation into a programming language realization. The coding should have some characteristics:

i. Ease of design to code translation
ii. Code efficiency
iii. Memory efficiency
iv. Maintainability

Modern computers have enjoyed increasing storage capacity, but the mechanisms that harness this storage power haven't improved proportionally. Whether current desktops have scaled to handle the enormous number of files computers must handle compared to just a few years ago is doubtful at best. Scalability includes not only fault tolerance or performance stability of tools for users to harness this power. The lack of appropriate structures and tools for locating, navigating, relating, and sharing bulky file sets is preventing users from harnessing their PCs' full storage power. Powering desktops with metadata, leading to the semantic desktop, is a promising way to realize this potential. These Mouse approach realizes the promising vision of the semantic desktop.

This approach provides seamless integration between file-centered tooling and semantically aware, resource-centered applications. A method for a real-time vision system that automatically detects a user's eye blinks and accurately measures their durations is introduced. 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 location of the eyes is determined automatically through the motion of the user's initial blinks. Subsequently, the eye is tracked by correlation across time, and appearance changes are automatically analyzed in order to classify the eye as either open or closed at each frame. No manual initialization, special lighting, or prior face detection is required. The system has been tested with interactive games and a spelling program. Results demonstrate overall detection accuracy of 95.6% and an average rate of 28 frames per second.

6. Algorithms Used and their Functionality:

Illumination Invariant Motion Detection Algorithm for Moving Object analysis:

Homomorphic filtering models the recorded grey levels g(m, n) as the product of scene illumination i(m, n) and surface reflectance r(m, n). Clearly, structural scene changes are captured by r(m, n). Scene illumination is assumed to vary slowly over the spatial coordinates (m, n), and can hence be suppressed by applying a linear highpass filter to log(g(m, n)). A similar, but slightly more stringent illumination model is used in , where illumination is modelled as a constant factor within small image blocks. We note here that this relationship between illumination and surface reflectance may be altered by potential camera nonlinearities. Commonly, however, the camera gain is described by a so-called -curve, for which it can be shown that the multiplicative relationship is preserved

To decide whether or not a change did occur between the successive frames Gt and Gt−1 at pixel (j, l), we compare the grey levels from Gt and Gt−1 which lie within a small sliding window, which is centered around (j, l). These grey levels are ordered into column vectors x and y, respectively.

If the window contains N pixels, each of these vectors contains N components x(n) and y(n), respectively, where n = 1, 2, . . . ,N. If no scene change occurs within the window, and neglecting the effects of noise, both vectors would be identical if illumination remained constant between times t − 1 and t. Under the same circumstances, a change in illumination would change the norms of these vectors, but not their directions. Consequently, in an ad-hoc test function is employed which detects whether or not x and y are linearly dependent. If they are, a structural change did not occur. Here, we now express this reasoning as a hypothesis test.

Let s = (s(1), . . . , s(N)) denote the noisefree signal in x. Assuming additive white Gaussian camera noise, we observe

$$\mathbf{x} = \mathbf{s} + \epsilon_1 \tag{1}$$

where _1 is a noise vector obeying N(0, _2I). The parameter _2 is the variance of the camera noise, and I the identity matrix. If no scene change occurs, y contains the same signal as x, which may be scaled by a factor k. Hence,

$$\mathbf{y} = k \cdot \mathbf{s} + \epsilon_2 \tag{2}$$

where _2 is another realization of the camera noise which is independent of _1. If the illumination remains constant, we have k = 1, otherwise 0 < k < 1. (This can always be achieved by assigning x and y accordingly to the frames Gt and Gt−1, and implies no loss of generality. When deriving

the distribution of our test statistic, we will at one point approximate the norm of x — but not its direction — by the norm of s. The relative approximation error is smaller when x is chosen as done above.) We call the hypothesis that x and y are given by (1) and (2) the null hypothesis H0.

Given the alternative hypothesis H1, we model the conditional pdf p(d|H1)

$$p(\mathrm{d}|H_1) = \left(\frac{1}{\sqrt{2\pi}\sigma_e}\right)^{N-1} \cdot \exp\left\{-\frac{|\mathrm{d}|^2}{2\cdot\sigma_e^2}\right\} \tag{10}$$

with _2 c _ _2. Furthermore, we model the sought change masks by an MRF such that the detected changed regions tend to be compact and smoothly shaped. From this model, a priori probabilities Prob(c) and Prob(u) for the labels c and u can be obtained. The MAP decision rule then is

$$\frac{p(\mathrm{d}|H_1)}{p(\mathrm{d}|H_0)} \underset{u}{\overset{c}{\gtrless}} \frac{\mathrm{Prob}(u)}{\mathrm{Prob}(c)} \tag{11}$$

This can be manipulated into the context adaptive decision

$$T \underset{u}{\overset{c}{\gtrless}} t + (4-\nu_c)\cdot B \tag{12}$$

where T is the test statistic of (7), and t the threshold according to (8). The parameter _c denotes the number of pixels that carry the label c and lie in the 3 × 3-neighbourhood of the pixel to be processed (Fig. 2). These labels are known for those neighbouring pixels which have already been processed while scanning the image raster (causal neighbourhood). For the pixels which are not yet processed we simply take the labels from the previous change mask (anticausal neighbourhood). Clearly, the adaptive threshold on the right hand side of (12) can only take the nine different values _c = 0, 1, …..8. The parameter B is a positive cost The adaptive threshold hence is the lower, the higher the number _c of adjacent pixels with label c. It is obvious that this behaviour favours the emergence of smoothly shaped changed regions, and discourages noise-like decision errors. The nine different possible values for the adaptive threshold can be precomputed and stored in a look-up table

Fig. 2: 3× 3-neighbourhood of a pixel i, with its causal neighbours shown shaded

7. SVM for Detecting Human in Video Frames:

Two tasks need to be performed for head pose estimation: constructing the pose estimators from face images with known pose information, and applying the estimators to a new face image. We adopt the method of SVM regression to construct two pose estimators, one for tilt (elevation) and the other for yaw (azimuth). The input to the pose estimators is the PCA vectors of face images. The dimensionality of PCA vectors can be reasonably small in our experiments (20, for example). The output is the pose angles in tilt and yaw. The SVM regression problem can be solved by maximizing

$$f(\mathbf{x}) = \sum_{i=1}^{l} (\alpha_i^* - \alpha_i) k(\mathbf{x}, \mathbf{x}_i) + b$$

where x is the PCA feature vector of a face image, k is the kernel function used in the SVM pose estimator, yi is the ground-truth pose angle in yaw or tilt of pattern x; C is the upper bound of the Lagrange multipliers ai and api ; and 1 is the tolerance coefficient. More details about SVM regression can be found. Two pose estimators in the form of Eq. (4), ft for tilt and fy for yaw, are constructed. The support vectors (SVs) and patterns with the largest error from the previous iteration have higher priority for selection. The algorithm is stopped when no significant improvement is achieved.

SVM-based face detector can be constructed by maximizing

$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{st} \sum_{i=1}^{l} \alpha_i y_i = 0$$

$$0 \le \alpha_i \le C, \quad i = 1, 2, \ldots, l$$

where yi is the label of a training example xi which takes value 1 for face and 21 for non-face, k is a kernel function, and C is the upper bound of the Lagrange multiplier ai: For a new pattern x; the trained face detector gives an output

$$f(x) = \sum_{i=1}^{l} y_i \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b$$

where b is the bias.

The Eigenface method actually models the probability of face patterns, while the importance of non-face patterns in this method is not significant except when choosing the threshold. On the other hand, an SVM-based face detector makes use of both face and non-face patterns: it estimates the boundary of positive and negative patterns instead of estimating the probabilities. Generally speaking, the Eigenface method is computationally efficient but less accurate, while the SVM method is more accurate but slow. In order to achieve improved overall performance in terms of both speed and accuracy, a novel approach which combines the Eigenface and the SVM methods is presented.

An SVM-based classifier is trained using the examples in the middle region of Fig. 4. The classifier is only activated when an ambiguous pattern emerges. Usually the SVMbased classifier is computationally more expensive than the Eigenface method, but more accurate. However, since the proportion of the examples in the ambiguous region is relatively small, a significant improvement of the classification speed can be achieved. Furthermore, owing to the fact that the SVM classifier is trained only on the examples in the ambiguous region and not on the whole training set, the SVM classification problem is simplified to some degree.Amore precise and compact set of SVs are obtained.

8. Grey Prediction Algorithm for Eye Blink analysis and Tracking:

The grey system theory takes the random variants as the grey variants varying in a certain scope, the random process as the grey process varying in certain scope and periods. After generating the original data without regularity or with less regularity. It makes them to be the generated data with more regularity for modeling. Therefore the grey model is actually a generated data model instead of original data model obtained by common modeling method.

Suppose $X^{(0)} = (x^{(0)}(1), x^{(0)}(2), \cdots, x^{(0)}(n))$ and $X^{(1)} = (x^{(1)}(1), x^{(1)}(2), \cdots, x^{(1)}(n))$, then the original form of GM (1, 1) model is as follows:

$$x^{(0)}(k) + az^{(1)}(k) = b \quad (1)$$

where

$$z^{(1)}(k) = 0.5x^{(1)}(k) + 0.5x^{(1)}(k-1), \quad k=2,3,\ldots,n. \quad (2)$$

$$x^{(1)}(k) = \sum_{i=1}^{k} x^{(0)}(i), \quad k=1,2,\ldots,n; \quad (3)$$

If $\hat{a} = (a, b)^T$ is parameter vector and

$$Y = \begin{bmatrix} x^0(2) \\ x^0(3) \\ \vdots \\ x^0(n) \end{bmatrix}, \quad B = \begin{bmatrix} -z^{(1)}(2) & 1 \\ -z^{(1)}(3) & 1 \\ \vdots & \vdots \\ -z^{(1)}(n) & 1 \end{bmatrix}. \quad (4)$$

9. System Flow Diagram:

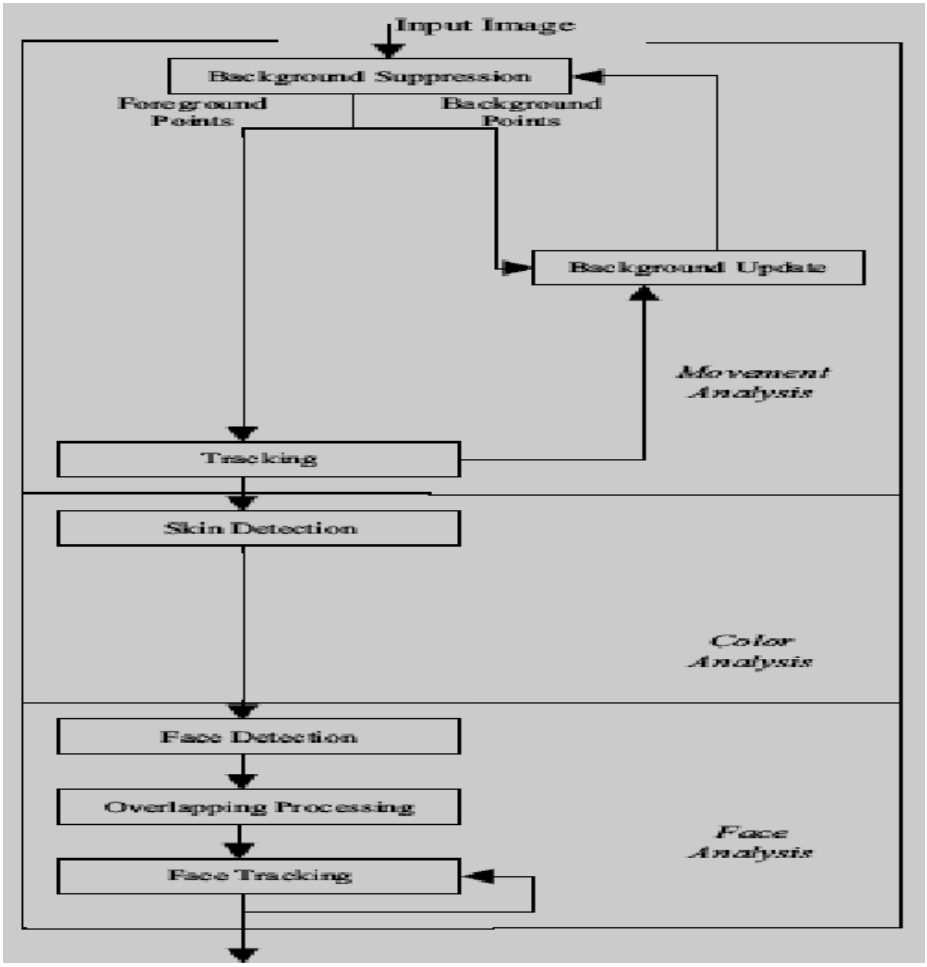

Fig. 3: System Layout.

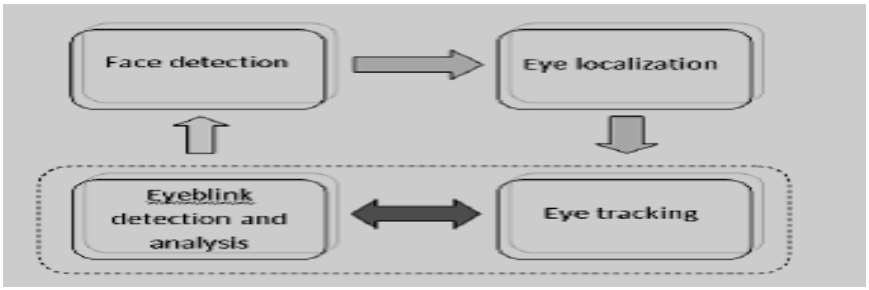

Fig. 4: Overall Diagram.

10. Conclusion:

A simultaneous eye tracking and blink detection system is presented in this paper. We used two Illumination invariant motion detection algorithm for detecting the moving object and the system uses color analysis to detect the human skin color and SVM for detecting the human face and the posture of the human face, which in term works efficiently and automates itself to inactive stage if there is no humans are present before the system . and grey prediction algorithm is used to track the human eye movements and voluntary and involuntary eye blinks. The grey prediction system that gives higher confidence is used to determine the estimated eye location as well as the eye's status (open v.s. closed).

References:

[1] N. Kojima, K. Kozuka, T. Nakano, and S. Yamamoto. Detection of consciousness degradation and concentration of a driver for friendly information service. In Proceedings of the IEEE International Vehicle Electronics Conference 2001, pages 31–36, 2001.

[2] P. Smith, M. Shah, and N. da V. Lobo. Monitoring head/eye motion for driver alertness with one camera. In Proceedings of the Fifteenth IEEE International Conference on Pattern Recognition, September 2000.

[3] K. Grauman, M. Betke, J. Lombardi, J. Gips, and G. Bradski. Communication via eye blinks and eyebrow raises: Video-based human-computer interfaces. Universal Access in the Information Society, 2(4):359–373, November 2003.

[4] K. Grauman, M. Betke, J. Gips, and G. R. Bradski. Communication via eye blinks-detection and duration analysis in real time. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, December 2001.

[5] M. Chau and M. Betke. Real time eye tracking and blink detection with usb cameras. Technical report, Boston University Computer Science, April 2005. No. 2005-12.