Second Order Change Detection, and its Application to Blink-Controlled Perceptual Interfaces

Gorodnichy, Dimitry

Proceedings of the International Association of Science and Technology for Development (IASTED) Conference on Visualization, Imaging and Image Processing (VIIP 2003), 2003

National Research Council Canada
Institute for Information Technology

Conseil national de recherches Canada
Institut de technologie de l'information

NRC · CNRC

*Second Order Change Detection, and its Application to Blink-Controlled Perceptual Interfaces **

Gorodnichy, D.
June 2003

* published in the Proceedings of the International Association of Science and Technology for Development (IASTED) Conference on Visualization, Imaging and Image Processing (VIIP 2003). pp. 140-145. Benalmadena, Spain. September 8-10, 2003. NRC 47151.

Canada

SECOND ORDER CHANGE DETECTION, AND ITS APPLICATION TO BLINK-CONTROLLED PERCEPTUAL INTERFACES

Dmitry O. Gorodnichy

Computational Video Group, IIT, NRC-CNRC, Ottawa, Canada K1A 0R6

ABSTRACT

The concept of the second order change detection, which allows one to discriminate the local (most recent) change in image, such as blink of the eyes, from the global (long lasting) change, such as the motion of head, is introduced. This concept sets the base for designing complete face-operated control systems, in which, using the analogy with mouse, "pointing" is performed by nose and "clicking" is performed by double-blinking of the eyes. The implementation of such systems is described.

1. INTRODUCTION

We consider the problem of designing vision-based face-operated control systems. These systems, often referred to as perceptual user interfaces [1], are commonly treated as a hands-free alternative or extension to mouse, joystick, track pad, track ball, and other tangible input devices. The main applications of such systems are seen in human-computer interaction, video-conferencing, immersive and collaborative environments, entertainment, avatar-like computer-generated communication programs, industry for disabled and security.

Figure 1 illustrates a setup of a face-operated interaction system: a user sits in front of a camera or several cameras mounted on top of and/or beneath a computer screen and used to monitor the user's face motion. A visual feedback is provided to the user as means of verifying the successfulness of the received face-operated command.

For these systems to be fully operational, they have to be able to do two tasks. First, they should be able to track a human face both robustly – with respect to head motion, and precisely – with subpixel precision, so that its position can be converted to a position of a cursor or another virtual object in 2D screen. Second, these systems should be able to detect a facial expression event, which a user would use in order to send a binary "on/off" command to the system, analogous to the mouse "click" event.

The recently developed *Nouse "Use your nose as mouse"* tracking technology, which made it possible to track a liberally unconstrained head motion with subpixel precision, offered a solution to the first of these two tasks [2]. This paper proposes a solution to the second task.

We develop a technique, which allows one to detect eye blinks in a moving head. For this purpose we introduce the concept of the second order change, which is the change of a change observed in an image. We show that, similar to the commonly used first order change which is computed by considering two consecutive frames of video, the second order change can be computed using three consecutive frames of video.

The proposed technique not only makes it possible to detect eye blinks in moving faces, which was previously considered difficult, but also significantly enhances the overall perceptual capability of perceptual vision-based systems. Our system operates with off-the-shelf USB cameras, the binaries of it are available from our website [3] for public evaluation.

2. PREVIOUS WORK

Excellent pupil localization and blinking detection performance is reported for systems which use structured infrared light (e.g. see [4]). These systems register the light reflection in the users' eyes in order to locate the pupils. Good results in eye blinking detection are also reported for the systems based on high-resolution videocameras which can capture eye pupils with pixel accuracy [5]. These approaches however are not suitable for applications which use low-cost USB cameras. These systems are also characterized by not using the dynamic component of video, which as discussed in [6] is intensively used by mechanisms of visual attention employed in biological vision systems.

A common approach to detecting moving objects in video is based on detecting the intensity change between two consecutive frames caused by the object motion [7]. The simplest way of detecting such a change, which will refer to as the *first order change*, is to use the binarized thresholded absolute difference between two consecutive video frames. This is what has been used so far by other authors [8, 9, 10, 11] to detect blinks. However as noted by [11], this technique fails to detect eyes when a face moves, because many candidates also appear around the face boundary as

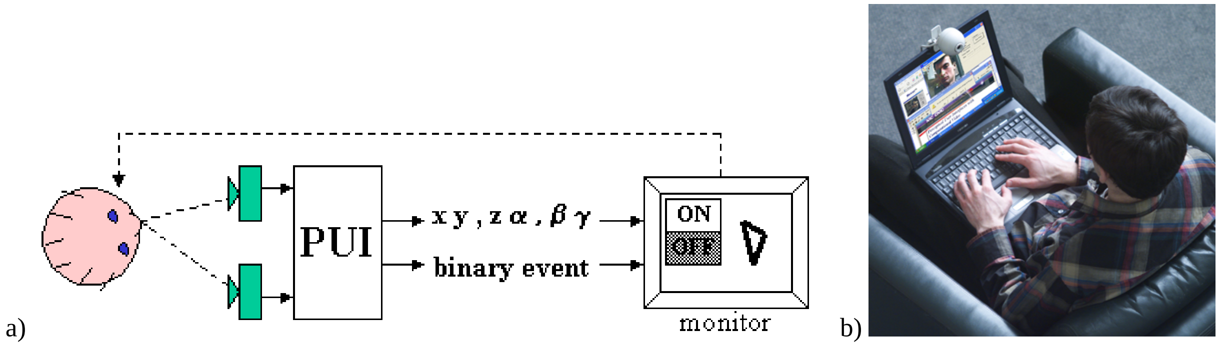

Fig. 1. Face-operated perceptual user interface: the system diagram (a) and the system at work (b). The photo picture shows a user at the moment of selecting a "Yes" button in a pop-up "Continue? Yes || No || Cancel" window, which is performed hands-free by using nose and double blinking.

well as around nose, mouth and other parts of the face.

3. SECOND ORDER CHANGE DETECTION

Using analogy with second order derivatives of functions, we define the *second order change* as a change of a change. We will show now that, similar to the method used in numerical analysis for computing the second order derivative of a function based on considering three neighboring points, the second order change in video can be computed by considering three consecutive frames of the video.

Object motion can be categorized as one of the four types: 1) pure translation in plane parallel to image plane, 2) pure translation in plane other than the one parallel to image plane, 3) rotation around optical axis of the camera, and 4) rotation around other axes.

For now let us consider the motion of the first type, in which an object moves with the velocity $\vec{V}$ in image plane, undertaking a small local change. The goal of the second order change detection is to localize this local change. An example of such combined global (with respect to camera) and local (with respect to the object) motion is shown in Figure 2, in which a floppy disk moves with respect to the camera, with the protective cover sliding rapidly at some point from an open to a close position.

In order to localize the pixels which have changed due the cover motion, one needs to remove out of all pixels detected by the first order change detection, shown in first column in Figure 2, those which have changed because of the floppy disk motion.

Provided that image capture and processing rate is fast, e.g. ten frames per second, which is the case for real-time video systems, the motion of the object during the entire three-frame sequence can be considered unchanged. This makes it possible to find all pixels that have changed because of this motion and to filter them out. The entire procedure is illustrated in Figure 1. The steps of the procedure follow.

Step 1. Compute the first order changes occurred in the current and the last video frames. One way of doing this is to compute the binarized thresholded absolute difference images:

$$D_t = \Theta\big(|I_t - I_{t-1}| - N\big) \text{ and } D_{t-1} = \Theta\big(|I_{t-1} - I_{t-2}| - N\big),$$

where I_t indicates the black-n-white frame at time t, N is a threshold value which takes into account the level of noise, and $\Theta(\cdot)$ is a Heaviside (step) function.

Step 2. Find vector $\vec{V} = (v_i, v_j)$ such that, by shifting image D_{t-1} by (v_i, v_j) number of pixels up/down left/right, it coincides the most with image D_t:

$$\vec{V} = argmin||D_t - D_{t-1}(\vec{V})||,$$

where $||\cdot||$ designates the L_1 norm of a vector.

Step 3. Shift image D_{t-1} by (v_i, v_j) pixels, after which subtract the resulting image $D_{t-1}(\vec{V})$ from image D_t.

Step 4. Filter the obtained image D_t^*, which has already much smaller number of bright (changed) pixels than the first order change image D_t, with a median filter to remove the remaining isolated pixels left on the boundary of the object.

As a result of this four-step procedure, we obtain an image which shows only those pixels which are attributed to the most recent (local) change observed in video.

Figures 2 and 4 show the results of running the described procedure with two objects. In the first case, the procedure is used to detect the closing of the protective cover in a moving floppy disk (Figure 2). In the second case, it is used to detect eye blinks in a moving head (Figure 4).

It can be seen now how the described procedure can be extended into the cases when an object does not move in a plane parallel to the image plane, but moves in a different plane or rotates. For this we need only to see that the goal of Step 3 of the procedure is to remove as many bright pixels

Fig. 2. Second order change detection in a floppy disk moving from left to right. The top row of snapshots shows the second last, last and current frames captured by the camera. The results of the first-order change detection and the second order change detection are shown in the bottom row in the far left and far right images, respectively. The two images in the middle of the row show the pixels which were removed from the left image to obtain the right image by Step 3 (first image) and Step 4 (second image) of the second order change detection procedure described in the paper; the change of these pixels was caused by the global motion of the object.

as possible from where the change due to the global motion of the object happened. The two middle bottom images in Figures 2 and 4 show the pixels which can be removed. In the case of the head motion, whatever it is - rotational or translational, these locations are usually the boundary of the head, brow, mouth, hair and nose locations. The pixels at all of these locations can be filtered away by using the same four-step procedure, with the only modification that image $D_{t-1}(\vec{V})$ to be subtracted from image D_t should be dilated several times before the subtraction to cover not only the changed pixels but also the pixels in the neighborhood of the changed pixels.

3.1. Non-linear change detection

Computing first order change images D_t and D_{t-1} in Step 1 of the procedure by frame subtraction is fast, however it is not very reliable, because of the noise present in image. The amount of noise is especially large when low-quality cameras are used, or when cameras are equipped with mechanisms for automatic brightness adjustment. A frame subtraction based change detection also fails when lighting changes, as when switching the lights on and off.

This is why in order to design a change detector tolerant to noise and illumination changes, we propose to use the algebraic method for detecting change in an image as recently proposed in [7]. According to this method a pixel is considered to have changed only if the area around the pixel, called the support area, has changed non-linearly. One way of detecting such an non-linear change is to compare vectors $\vec{x}_t = \{x_{i,t}\}$ and $\vec{x}_{t+1} = \{x_{i,t+1}\}$, $i = 1...n$, created from the pixel intensities in the support area of pixel x in frames I_t and I_{t+1}, respectively; the size of the support area n is typically chosen to be 3x3. If these vectors are collinear, than the change is linear and there was no change observed in pixel x. Otherwise, pixel x is considered to have changed.

Another, more efficient, way of detecting an non-linear change, is to use the Wronskian determinant, which for the case of two consecutive frames I_t and I_{t+1} is written as

$$W\left(I_{t+1}, I_t\right) = \frac{1}{n}\sum_{i=1}^{n}\frac{x_{i,t}^2}{x_{i,t+1}^2} - \frac{1}{n}\sum_{i=1}^{n}\frac{x_{i,t}}{x_{i,t+1}}. \quad (1)$$

If $|W(I_{t+1}, I_t)| + |W(I_t, I_{t+1})|$ is much larger than zero, then pixel x is considered to have changed. An interesting feature of this approach is that it does not depend much upon the value of the threshold.

It should be mentioned at this point that although the described method for detecting change using linear algebra makes change detection more reliable, it slows down con-

Fig. 4. Detecting eye blinks using the second order change detection. The pixels changed because of head motion removed by the second order change detection procedure are shown at the bottom center images. See also Figure 2.

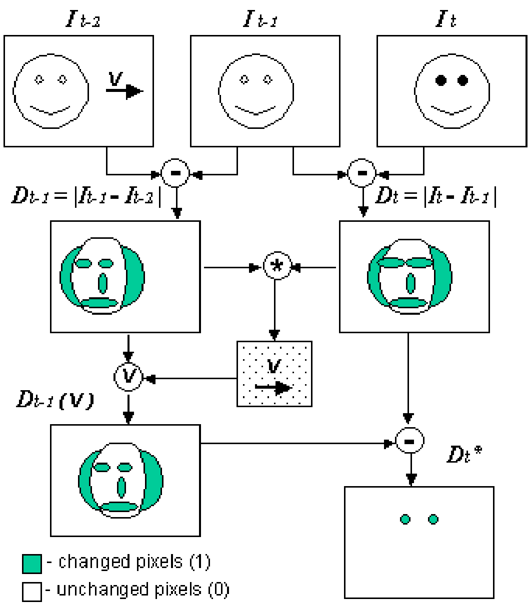

Fig. 3. A flowchart for detection eye blinks using with second order change detection.

siderably the entire change detection procedure, which creates a risk of missing a fast happening local change. Therefore care should be taken when using this method, especially when computer power is not very big..

4. EYE BLINK DETECTION

The best application of the second order change detection, which was also the inspiration for its invention, is detecting of eye blinks in video showing heads. We suggest the following approach to do this.

After the pixels corresponding to the head motion have been filtered out by the second order change detection procedure, the remaining motion pixels are used to decide whether a blink occurred or not. First, the number of the remaining changed pixels is examined and if it is too small, meaning that there is no motion, or if it is too large, meaning that the background has changed, the eye detection technique is skipped. Otherwise, the vector quantization technique is employed in order to classify bright (changed) pixels into two classes, corresponding to the positions of two potential "eyes".

Using the learning vector quantization paradigm [12], we initialize two 2D prototype vectors, corresponding to the image location of potential eyes, with zeros. Then we train them in iterative fashion using the positions of the changed pixels as the training patterns. The winner-take-all update rule is used.

As the result of learning, the prototype vectors move exactly to the locations corresponding to two largest clusters of bright pixels in the difference image (see Figure 4), with the procedure taking only one scan through the image.

The decision on whether a blink occurred or not is based on examining the positions of the converged prototype vectors. Whenever a blink occurs, these positions coincide with eye positions.

4.1. Additional constraints

We have already used one constraint assuming that both eyes blink. While eliminating the case of one eye blinking, which rarely happens involuntary, this constraint proves to be very useful for detecting eyes from blinking.

The following are a few other constraints which are added to speed up and to robustify the procedure. First, geometrical constraints are added such as a constraint on the distance between the eyes, their approximate location and orientation.

Second, we use a multi-channel paradigm for video-based face processing proposed in [6]. According to this paradigm, video information is divided into colour, motion and intensity channels. Each channel is responsible for the specific set of tasks for which the channel is the most suitable. In particular, the colour channel, which converts the captured images to the perceptually uniform colour system (UCS) representation suitable for skin detection, is used to narrow the area of interest to be processed by the motion channel responsible for detecting of the eye blinks.

5. BLINK DETECTION AND THE *NOUSE* TECHNOLOGY

The fact that the second order change detection makes it possible to detect eye blinks in moving heads allows one to incorporate eye blink detection with other vision-based perceptual technologies, one of the most successful of which is called *Nouse – 'Use your nose as a mouse'*.

The main idea behind the *Nouse* perceptual technology consists in tracking the rotation and scale invariant convex-shape nose feature defined as the point on the nose tip surface the closest to the camera. As shown in [2], this technology allows one to track a liberally unconstrained head motion, represented by the position of the nose tip, with sub-pixel accuracy.

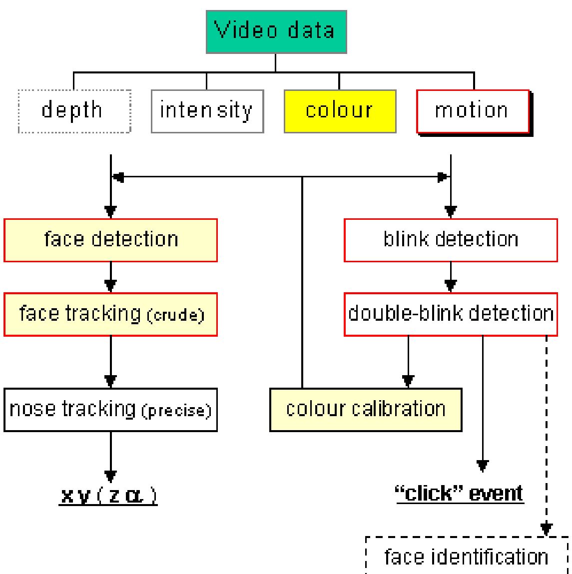

Fig. 5. Hierarchy of tasks in the *Nouse* face-operated system.

Figure 5 shows the architecture of the *Nouse* face-operated system with eye blink detection incorporated into it. As will be seen blink detection allows us to add a few improvements to the system.

First, eye blink detection is used to build the histogram of face skin colour, which is used in face detection and tracking stages preceding the nose detection stage.

Second, eye blink detection provides an excellent way of sending binary commands with a face. In particular, we propose to use the "double-blink" event as a hands-free alternative to a "click" in face-operated interaction systems. The system registers the "double-blink" event if it detects two blinks within 0.5 sec time lapse from each other. Contrary to a single blink which happens involuntary, double-blink is mostly a deliberate action and can be well performed by most people. Another advantage of using double-blinks for control is that it provides an additional spatio-temporal constraint for eye verification — the positions of the eyes detected in two consecutive blinks should coincide within an allowable range. This constraint becomes very valuable for the situations, when false positives are undesirable.

Fig. 6. Using double-blink to switch Nouse on/off.

Figure 6 shows the snapshot of the *Nouse* hands-free interface, in which nose tracking is toggled "on/off" by double-blinking. The picture is captured at the moment of double-clicking. Circles overlaid around the eyes show the detected eyes. The box shows the position of the nose tip. The numbers at top of the image indicate the time (in msec) when the current and last blink happened.

Finally, the ability to retrieve the position of the eyes from blinking allows us to retrieve the entire face in a form convenient for face recognition. This makes face recognition in video more efficient. More details on how this is done are given in [6].

6. CONCLUSIONS

The proposed technique for detecting second order change in video is a powerful tool which has immediate application in detection of the eye blinks in video. When combined with the non-linear change detection described in the paper, this technique allows one to detect local or most recent change reliably with respect to noise, illumination changes and object motion.

While there are other papers on detecting eyes from blinking with the application to designing perceptual interfaces, the present paper seems to be only one which demonstrates good results obtained with low-cost cameras, such as ordinary USB webcams, on video sequences showing moving heads. While the paper shows only the snapshots of our blink detection program, the video sequences of all experimental runs described in the paper are made available at our web-site [3]. The website has also the binary of the program which can be downloaded for testing.

Acknowledgements

The described technology is currently being tested in Access to Communication and Technology (ACT) department at the West Midlands Rehabilitation Center in the UK, where it is used to enable patients with a brain-stem injury a face-to-face communication.

The author thanks Dan Gamache for taking a photo picture used in the paper.

7. REFERENCES

[1] G. Bradski, "Computer vision face tracking for use in a perceptual user interface," in *Intel Technology Journal, No.2*, 1998.

[2] D.O. Gorodnichy, S. Malik, and G. Roth, "Nouse 'Use your nose as a mouse' - a new technology for hands-free games and interfaces," in *Proc. Intern. Conf. on Vision Interface (VI'2002), available online at www.visioninterface.org*, Calgary, May 2002, pp. 354–361.

[3] NRC, "*Nouse* 'Use your nose as mouse' technology website," http://www.cv.iit.nrc.ca/Nouse, 2001.

[4] Z. Zhu, Q. Ji, K. Fujimura, and K. Lee, "Combining kalman filtering and mean shift for real time eye tracking under active ir illumination," in *Proc. of Int. Conf. on Pattern Recognition (ICPR'2002), Vol.IV, pp. 318-321, Quebec*, 2001.

[5] Moriyama T., Kanade T., Cohn J.F., Xiao J., Ambadar Z., J. Gao, and H. Imamura, "Automatic recognition of eye blinking in spontaneously occurring behavior," in *Proc. of Intern. Conf. on Pattern Recognition (ICPR'2002), Vol.IV, pp. 78-81, Quebec*, 2001.

[6] Dmitry O. Gorodnichy, "Face recognition from video," in *Proc. of Int. Conf. on Audio- and Video-based Person Authentication (AVBPA'03), Guilford, UK, May 9-11, Springler-Verlag*, 2003.

[7] E. Durucan and T. Ebrahimi, "Change detection and background extraction by linear algebra," in *Special Issue of IEEE Proc. on Video Communications and Processing for Third Generation Surveillance Systems, 89(10)*, 2001, pp. 1368–1381.

[8] J. L. Crowley and F. Berard, "Multi-modal tracking of faces for video communications," in *Proc. CVPR 97*, 1997, pp. 640–645.

[9] L.-P. Bala, K. Talmi, and J. Liu, "Automatic detection and tracking of faces and facial features in video sequences," in *Picture Coding Symposium, Sept. 1997, Berlin, Germany*, 1997.

[10] K. Grauman, M. Betke, J. Gips, and G. Bradski, "Communication via eye blinks detection and duration analysis in real time," in *Proc. CVPR 01*, 2001.

[11] S. Kawato and N. Tetsutani, "Detection and tracking of eyes for gaze-camera control," in *Proc. Intern. Conf. on Vision Interface (VI'2002)*, Calgary, May 2002, pp. 348–355.

[12] T. Kohonen, *Self-Organization and Associative Memory, 2nd Edition*, Berlin: Springer-Verlag, 1987.