FEATURE EXTRACTION AS A TOOL FOR COMPUTER INPUT

Nimish Mehta*, K.C. Smith and F.E. Holmes

Dept. of Electrical Engineering
University of Toronto
Toronto, Ontario M5S-1A4

ABSTRACT

This paper describes a versatile human-machine interface designed to achieve an important balance between human factors and flexibility. It incorporates attributes of keypads, tablets and 'mice' and possesses freedom of keyshape, size and function. The system uses a TV camera focused on a 'keyboard' area to generate electrical signals in response to optical inputs. Additional optical and electronic feature extraction result in an input device capable of processing discrete and continuous input simultaneously.

INTRODUCTION

In computer systems today, there is a continuing problem of finding a human-machine interface that will allow suitable physical gestures to express logical ideas. Conventional input transducers such as 'mice', joysticks, tablets, pens and keyboards are being used to increased advantage in many computer systems. However, a growing problem, especially for the users of complex graphics systems is the number of these transducers they must operate simultaneously in order to fully utilize the system. The problem is related to the fact that most existing transducers cannot be configured to perform all the necessary continuous and discrete input functions efficiently. This paper describes a human-machine interface that uses a TV camera focused on a 'keyboard' area and appropriate feature extraction to generate electrical outputs in response to optical inputs. The interface simply 'looks' at the keyboard area and activates processes in response to relevant features of the video frame. The result is an input device that can be used, under constraints, as a discrete data entry device and a continuous input device (such as a joystick) simultaneously. Also, it has the potential of being much more general than that-all optical events on the keyboard can be mapped onto an electrical output, with the mapping being totally software alterable.

FUNCTIONAL OVERVIEW

The human-machine interface consists of a 'keyboard' area, a TV camera looking at the area and a high speed image processor to process the video output from the camera. The 'keyboard' area is constructed of a rigid transparent material (glass or plexiglass) with a translucent plastic filter covering it. The camera is focused so that the keyboard maximally fills the viewable field. The plastic surface performs spatial filtering of this field. Consequently, the camera 'sees', as dark shadows and

Fig. 1-Physical Overview of the Interface

*N. Mehta is now with Octek, Inc., Burlington, MA 01803

CH 1746-7/82/0000 - 0818 $ 00.75 © 1982 IEEE

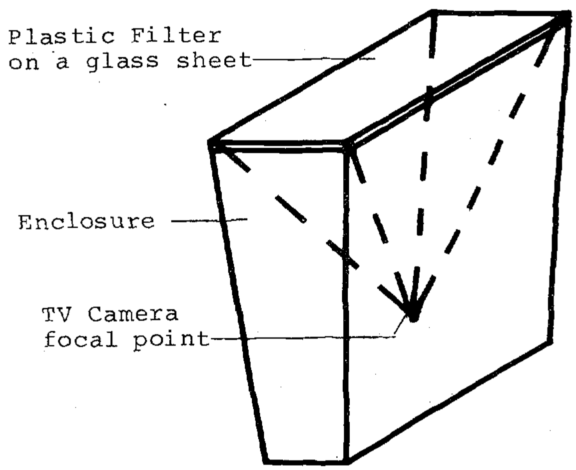

Figure n°1 of

IM3.1 **Feature Extraction as a Tool for Computer Input**

Mehta N., Smith K.C., Holmes F.E., *Department of Electrical Engineering, University of Toronto, Toronto, Ontario M5S 1A4, Canada.*

and lines, all objects placed on or close (1-5 mm.) to the keyboard but a grey blur for all objects beyond that. The image processor interprets this information and outputs appropriate code to an I/O port. Keys on the keyboard are not physical buttons but simply an opaque drawing of their outline on any transparent plastic sheet (called the overlay). To use the

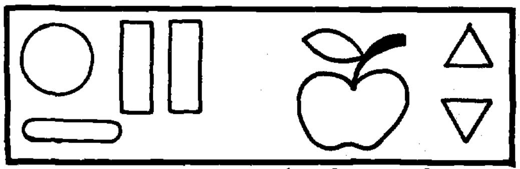
Figure 2 - A typical overlay

interface, a user places an overlay on the keyboard area. The interface enters a "learn" mode, wherein it either requests the nature of each key drawn on the overlay, together with the relevant code to be output, or "recognizes" the overlay and looks for a code-definition file on a mass storage device. Thereafter, it automatically readies itself for use. The user simply places his fingers on any set of keys, and causes the corresponding codes to be output. To change the interface key pattern, only the overlay need be changed. This reconfigurability can be used to great advantage because human short-term memory requirements can be reduced. Alden et al (1) quote studies to show that such memory failures could be a significant factor in keying errors. Further, the user is not constrained to using square or roundkeys: virtually any shape is allowed, the limits being imposed by the implemented software and physical size of the keyboard area. Again, features such as joysticks or variants thereof are both software created and limited. This ability to configure multi-input keypads can be used effectively to create a valuator of the type described by Evans et al (2), but more flexible than envisaged by them since two hand inputs are now allowed. The interface can also be described as a multi-input version of the pressure-sensitive pad described by Sasaki et al (3).

HARDWARE/SOFTWARE DETAILS

The image processor itself is a tightly coupled dual cpu architecture. One of them is a highly parallel pipelined STTL cpu while the other is a slower NMOS microprocessor. The video signal is digitized and a 256x256 pixel map created in real time by the first cpu. It also performs operations on blocks of data such as frame clean or subtract at high speeds. The NMOS cpu performs the feature extraction and processing necessary, as well as external I/O. Periodically, it enters a supervisory mode wherein it performs task division between itself and the high speed cpu based on the overlay details, user generated constraints on the overlay and suitability of each cpu for the task. This architecture has been described more extensively in (4).

The specific attributes of the human-machine interface are determined by the supervisory portion of the implemented software. A specific supervisor which allows only predetermined types of keys on an overlay lends itself to faster execution while a more general one possesses flexibility. However, as an example, a procedure for implementing a discrete "off/on" key would be:

0. Digitize a frame
1. If no fingers are pressed return to supervisor,
2. else find the center of gravity (CG) of the first finger not yet considered(={Xf,Yf}). If all fingers have been considered, this frame has been processed, so go to step #0.
3. Retrieve the closest CG point(={Xk, Yk}) from a file which stores CG's of keys on the overlay. If all CG's have been considered, the finger at {Xf,Yf} is not pressing a key but is placed elsewhere on the overlay. Return to supervisor.
4. Test the segment {(Xf,Yf), (Xk,Yk)}. If it intersects any key outlines on the stored picture of the overlay, go to 3, else the finger at (Xf,Yf) is pressing the key centered at(Xk, Yk). Return to supervisor.
5. Go to step #2.

In general, the supervisor causes a dump of appropriate microcode and data into the high speed cpu and makes macro calls to implement functions such as those mentioned above. A set of routines has been written to form this macro command set. This set is, by no means complete, and additional commands may be necessary as new interface operating modes are envisioned. Examples of some macro commands implemented are: GETPIX, a frame grab routine in which the size of the digitized picture is specifiable.

CENGRAV will find the center of gravity of a blob which is circular or elliptical (a finger) and return its position coordinates with respect to the frame, as well as its height. The latter is useful when implementing a pressure sensitive key. All commands have an inbuilt time-out feature. This is necessary when the maximum response time of a key must be limited.

ONE IMPLEMENTATION

In the implemented version discrete keys can be auto repeat or output code once only. Linear potentiometers can be defined on any area of the keyboard. The number of keys on any overlay is limited only by the outline resolution of the system (3mm.) and human engineering considerations. The maximum number of key rollovers and maximum simultaneously allowed fingers are both arbitrarily limited to 256. With the auto repeat feature selected, response time for any key is 69+/-8 ms. The inter-key dead time is 1 ms. The response time seems adequate if only the eye-hand bandwidth is considered. Card et al (5) report that the tapping rate for step keys in their pointing experiment was around 150 ms./keystroke. They also use a version of Fitt's Law (Welford (6)) to characterize the time to make hand movements when using joysticks. They note that the central processing capacities of the eye-hand guidance system are around 100 ms./bit. Similarly, for inter-key dead time, Alden et al (1) quote studies by Minor on minimum keyboard intercharacter times. A lockout time limit of 50 ms. was hypothesized by Minor. Our 1 ms. inter-key time would obviously seem adequate. However, these figures are inadequate if the human brain cognitive time is considered. Sasaki et al (5) note that for effective percussion-like gestures, response times around 5 ms. are needed. It is estimated that the response time of this interface could be reduced to 6 ms. +/- 2 ms. using the new 240 Hz. field rate CCD cameras and faster integrated circuits now available.

ACKNOWLEDGEMENTS

The authors wish to acknowledge financial assistance from NSERC, Canada.

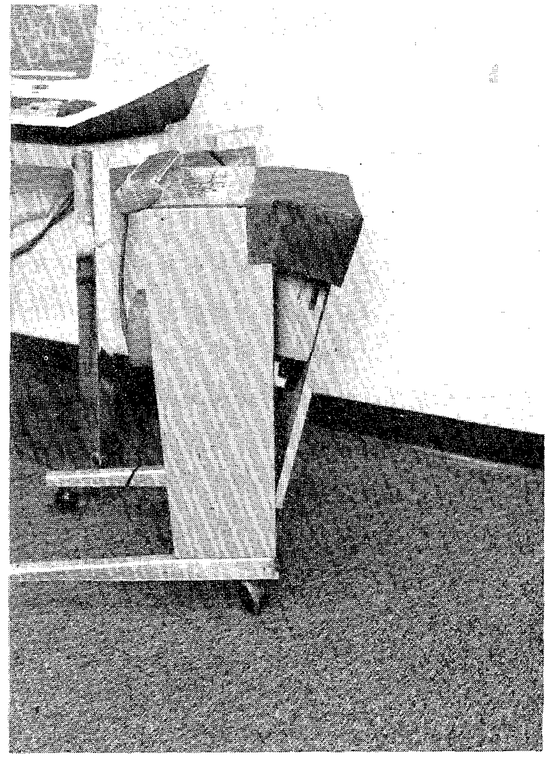

Fig. 3 The Implemented Interface

REFERENCES

(1) D.G. Alden, R.W. Daniels and A.F. Kanarick, "Keyboard Design and Operation: A Review of Major Issues," Human Factors, vol. 14(4), pp. 275-293, 1972

(2) K.B. Evans, P.P. Tanner and M. Wein, "Tablet-Based Valuators that Provide One, Two or Three Degrees of Freedom," Computer Graphics, vol.15, No. 3, August 1981.

(3) L. Sasaki, et al., "A Touch-Sensitive Input Device," to be published, Structured Sound Synthesis Project, CSRG, University of Toronto, Toronto, 1981.

(4) N. Mehta, F.E. Holmes and K.C. Smith, "A Fast Image Processor," Proc. IEEE Conf. & Exp., pp. 242-243, October 1981, Toronto.

(5) S.K. Card, W.K. English and B.J. Burr, "Evaluation of Mouse, Rate-Controlled Isometric Joystick, Step Keys, and Text Keys for Text Selection on a CRT," Ergonomics, vol. 21, No. 8, pp. 601-613, 1978

(6) A.T. Welford, "Fundamentals of Skill" London: Methuen, 1968.