EXHIBIT 4.04

Since the spatial resolution in the local interpolation scheme is limited by the number of bits available from the intensities of an array of 3 by 3 sensors, other scheme was considered. In this scheme, all the points from a complete scan of a tablet are interpolated allowing the potential resolution to be almost infinite. However this process simply emulates a projective device and accordingly reports only single point, which is interpolated from all the points on the tablet. However with this scheme, there are a great many ways of pointing to a specific location on a display screen, a feature with some intriguing application possibilities.

7.3 Response Time Delay

The response time delay is the time delay from the beginning of a touch to an output received either by local terminal or by an output device attached to the host computer. For multiple touches, this delay will increase with the number of touches. The prototype used with a 9600 baud-rate terminal to measure time delays. Actual response times were measured several times and averaged for various cases and are tabulated in Table 1.

Case	best	typical	worst
(a) pts/sec	17.6	15.2	12.8
msec/pt	56.8	65.8	78.1
(b) pts/sec	19.2	17.2	16.0
msec/pt	52.1	58.1	62.5
(c) pts/sec	24.0	22.0	18.6
msec/pt	41.6	45.5	53.2

TABLE 1. Actual Response Time Delays

The cases in Table one are to be interpreted as follows:

a one sensor touched continuously

b two sensors touched at the same time continuously

c four sensors touched at the same time continuously

8. CONCLUSIONS

A prototype of a fast-scanning multiple-touch-sensitive input tablet having both the adaptability and flexibility for a broad range of applications has been designed and implemented. Capacitance measurement of individual sensor(s) which can be uniquely addressed using two diodes per sensor, makes it possible to sense both the positions and intensities of one or more simultaneous touches without ambiguity. The sensor matrix is controlled by University of Toronto 6809 board whose serial port is connected to one of the I/O ports of a host computer. Software that utilizes the recursive subdivision algorithm for fast scanning an array of 64 by 32 sensors on the tablet, and that communicates with the host computer, has been implemented and tested.

9. ACKNOWLEDGEMENTS

The research described in this paper has been funded by the Natural Sciences and Engineering Research Council of Canada. This support is gratefully acknowledged.

10. REFERENCES

Brown, E., Buxton, W. & Murtagh, K. (1985). Windows on Tablets as a Means of Achieving Virtual Input Devices. Computer Systems Research Institute, University of Toronto.

Buxton, W. (1982). Lexical and Pragmatic Considerations of Input Structures, *Computer Graphics*, 17 (1), 31 - 37.

Buxton, W., Hill, R. & Rowley, P. (1985). Issues and Techniques in Touch-Sensitive Tablet Input, Computer Systems Research Institute, University of Toronto.

Hillis, W.D. (1982). A High Resolution Imaging Touch Sensor, *International Journal of Robotics Research*, 1 (2), 33 - 44.

Hurst, G. (1974). Electrographic Sensor for Determining Planar Coordinates, United State Patent 3,798,370, March 19, 1974, Elographics, Incorporated.

JSR (1981), Pressure-Sensitive Conductive Rubber Data Sheet, Japan Synthetic Rubber Co., New Product Development Department, JSR Building, 2-11-24 Tsukiji, Chuo-Ku, Tokyo 104, Japan.

Lee, S. (1984), *A Fast Multiple-Touch-Sensitive Input Device*, M.A.Sc. Thesis, Department of Electrical Engineering, University of Toronto.

Metha, N. (1982), *A Flexible Machine Interface*, M.A.Sc. Thesis, Department of Electrical Engineering, University of Toronto.

Sasaki, L., Fedorkow, G., Buxton, W., Retterath, C., & Smith, K.C. (1981). A Touch-Sensitive Input Device. *Proceedings of the Fifth International Conference on Computer Music*, North Texas State University, Denton, Texas, November, 1981.

TASA (1980), Model: x-y 3600 and x-y controller, Model: FR-105 Data Sheet, Touch Activated Switch Arrays Inc., 1270 Lawrence Station Road., Suite G., Sunnyvale, CA 94089.

TSD (1982), Touch Screen Digitizer Data Sheet, TSD Display Products Inc., 35 Orville Drive, Bohemia, NY 11716.

11. APPENDIX A: TOUCH TABLET SOURCES

Big Briar: 3 by 3 inch continuous pressure sensing touch tablet

Big Briar, Inc.
Leicester, NC
28748

Chalk Board Inc.: "Power Pad", large touch table for microcomputers

Chalk Board Inc.
3772 Pleasantdale Rd.,
Atlanta, GA 30340

Elographics: various sizes of touch tablets, including pressure sensing

Elographics, Inc.
1976 Oak Ridge Turnpike
Oak Ridge, Tennessee
37830

KoalaPad Technologies: Approx. 5 by 7 inch touch tablet for microcomputers

Koala Technologies
3100 Patrick Henry Drive
Santa Clara, California
95050

Spiral Systems: Trazor Touch Panel, 3 by 3 inch touch tablet

Spiral System Instruments, Inc.
4853 Cordell Avenue, Suite A-10
Bethesda, Maryland
20814

TASA: 4 by 4 inch touch tablet (relative sensing only)

Touch Activated Switch Arrays Inc.
1270 Lawrence Stn. Road, Suite G
Sunnyvale, California
94089

Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review

Vladimir I. Pavlovic, *Student Member, IEEE*,
Rajeev Sharma, *Member, IEEE*,
and Thomas S. Huang, *Fellow, IEEE*

Abstract—The use of hand gestures provides an attractive alternative to cumbersome interface devices for human-computer interaction (HCI). In particular, visual interpretation of hand gestures can help in achieving the ease and naturalness desired for HCI. This has motivated a very active research area concerned with computer vision-based analysis and interpretation of hand gestures. We survey the literature on visual interpretation of hand gestures in the context of its role in HCI. This discussion is organized on the basis of the method used for modeling, analyzing, and recognizing gestures. Important differences in the gesture interpretation approaches arise depending on whether a *3D model* of the human hand or an image *appearance model* of the human hand is used. 3D hand models offer a way of more elaborate modeling of hand gestures but lead to computational hurdles that have not been overcome given the real-time requirements of HCI. Appearance-based models lead to computationally efficient "purposive" approaches that work well under constrained situations but seem to lack the generality desirable for HCI. We also discuss implemented gestural systems as well as other potential applications of vision-based gesture recognition. Although the current progress is encouraging, further theoretical as well as computational advances are needed before gestures can be widely used for HCI. We discuss directions of future research in gesture recognition, including its integration with other natural modes of human-computer interaction.

Index Terms—Vision-based gesture recognition, gesture analysis, hand tracking, nonrigid motion analysis, human-computer interaction.

1 Introduction

With the massive influx of computers in society, *human-computer interaction*, or HCI, has become an increasingly important part of our daily lives. It is widely believed that as the computing, communication, and display technologies progress even further, the existing HCI techniques may become a bottleneck in the effective utilization of the available information flow. For example, the most popular mode of HCI is based on simple mechanical devices—keyboards and mice. These devices have grown to be familiar but inherently limit the speed and naturalness with which we can interact with the computer. This limitation has become even more apparent with the emergence of novel display technology such as virtual reality [2], [78], [41]. Thus in recent years there has been a tremendous push in research toward novel devices and techniques that will address this HCI bottleneck.

One long-term attempt in HCI has been to migrate the "natural" means that humans employ to communicate with each other into HCI. With this motivation automatic speech recognition has been a topic of research for decades. Tremendous progress has been made in speech recognition, and several commercially successful speech interfaces have been deployed [75]. However, it has only been in recent years that there has been an increased interest in trying to introduce other human-to-human communication modalities into HCI. This includes a class of techniques based on the movement of the human arm and hand, or *hand gestures*. Human hand gestures are a means of non-verbal interaction among people. They range from simple actions of using our hand to point at and move objects around to the more complex ones that express our feelings and allow us to communicate with others.

To exploit the use of gestures in HCI it is necessary to provide the means by which they can be interpreted by computers. The HCI interpretation of gestures requires that dynamic and/or static configurations of the human hand, arm, and even other parts of the human body, be measurable by the machine. First attempts to solve this problem resulted in mechanical devices that directly measure hand and/or arm joint angles and spatial position. This group is best represented by the so-called *glove-based devices* [9], [32], [88], [70], [101]. Glove-based gestural interfaces require the user to wear a cumbersome device, and generally carry a load of cables that connect the device to a computer. This hinders the ease and naturalness with which the user can interact with the computer controlled environment. Even though the use of such specific devices may be justified by a highly specialized application domain, for example simulation of surgery in a virtual reality environment, the "everyday" user will certainly be deterred by such cumbersome interface tools. This has spawned active research toward more "natural" HCI techniques.

• *V. Pavlovic and T.S. Huang are with The Beckman Institute and Department of Electrical and Computer Engineering, University of Illinois, Urbana, IL 61801. E-mail: vladimir@ifp.uiuc.edu.*
• *R. Sharma is with the Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802. E-mail: rsharma@cse.psu.edu.*

Manuscript received 17 Nov. 1995; revised 5 May 1997. Recommended for acceptance by R. Kasturi.
For information on obtaining reprints of this article, please send e-mail to: transpami@computer.org, and reference IEEECS Log Number 105027.

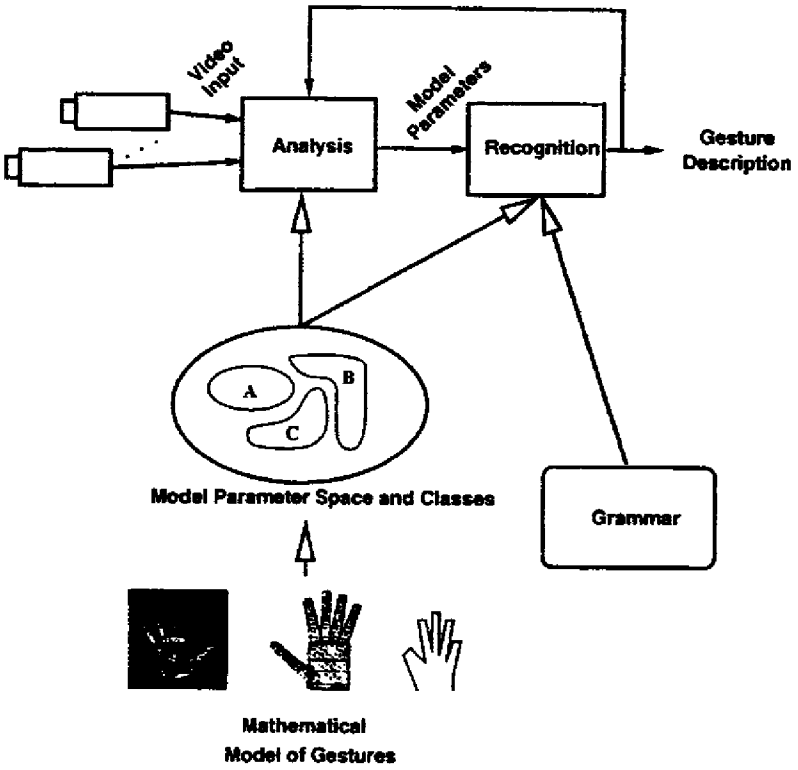


Fig. 1. Vision-based gesture interpretation system. Visual images of gesturers are acquired by one or more video cameras. They are processed in the analysis stage where the gesture model parameters are estimated. Using the estimated parameters and some higher level knowledge, the observed gestures are inferred in the recognition stage.

Potentially, any awkwardness in using gloves and other devices can be overcome by using video-based *noncontact* interaction techniques. This approach suggests using a set of video cameras and computer vision techniques to interpret gestures. The nonobstructiveness of the resulting vision-based interface has resulted in a burst of recent activity in this area. Other factors that may have contributed to this increased interest include the availability of fast computing that makes real-time vision processing feasible, and recent advances in computer vision techniques. Numerous approaches have been applied to the problem of visual interpretation of gestures for HCI, as will be seen in the following sections. Many of those approaches have been chosen and implemented so that they focus on one particular aspect of gestures, such as, hand tracking, hand posture estimation, or hand pose classification. Many studies have been undertaken within the context of a particular application, such as using a finger as a pointer to control a TV, or interpretation of American Sign Language.

Until recently, most of the work on vision-based gestural HCI has been focused on the recognition of static hand gestures or *postures*. A variety of models, most of them taken directly from general object recognition approaches, have been utilized for that purpose. Images of hands, geometric moments, contours, silhouettes, and 3D hand skeleton models are a few examples. In recent year, however, there has been an interest in incorporating the dynamic characteristics of gestures. The rationale is that hand gestures are dynamic actions and the motion of the hands conveys as much meaning as their posture does. Numerous approaches, ranging from global hand motion analysis to independent fingertip motion analysis, have been proposed for gesture analysis. There has thus been rapid growth of various studies related to vision-based gesture analysis fueled by a need to develop more natural and efficient human-computer interfaces. These studies are reported in disparate literature and are sometimes confusing in their claims and their scope. Thus there is a growing need to survey the state-of-the-art in vision-based gesture recognition and to systematically analyze the progress toward vision-based gestural human-computer interface. This paper attempts to bring together the recent progress in visual gesture interpretation within the context of its role in HCI.

We organize the survey by breaking the discussion into the following main components based on the general view of a gesture recognition system as shown in Fig. 1:

- *Gesture Modeling* (Section 2)
- *Gesture Analysis* (Section 3)
- *Gesture Recognition* (Section 4)
- *Gesture-Based Systems and Applications* (Section 5)

The first phase of a recognition task (whether considered explicitly or implicitly in a particular study) is choosing a model of the gesture. The mathematical model may consider both the spatial and temporal characteristic of the hand and hand gestures. We devote Section 2 to an in-depth discussion of gesture modeling issues. The approach used for modeling plays a pivotal role in the nature and performance of gesture interpretation.

Once the model is decided upon, an analysis stage is used to compute the model parameters from the image features that are extracted from single or multiple video

input streams. These parameters constitute some description of the hand pose or trajectory and depend on the modeling approach used. Among the important problems involved in the analysis are that of hand localization, hand tracking, and selection of suitable image features. We discuss these and other issues of gesture analysis in Section 3.

The computation of model parameters is followed by gesture recognition. Here, the parameters are classified and interpreted in the light of the accepted model and perhaps the rules imposed by some grammar. The grammar could reflect not only the internal syntax of gestural commands but also the possibility of interaction of gestures with other communication modes like speech, gaze, or facial expressions. Evaluation of a particular gesture recognition approach encompasses both accuracy, robustness, and speed, as well as the variability in the number of different classes of hand/arm movements it covers. We survey the various gesture recognition approaches in Section 4.

A major motivation for the reported studies on gesture recognition is the potential to use hand gestures in various applications aiming at a natural interaction between the human and various computer-controlled displays. Some of these applications have been used as a basis for defining gesture recognition, using a "purposive" formulation of the underlying computer vision problem. In Section 5 we survey the reported as well as other potential applications of visual interpretation of hand gestures.

Although the current progress in gesture recognition is encouraging, further theoretical as well as computational advances are needed before gestures can be widely used for HCI. We discuss some of the directions of research for gesture recognition, including its integration with other natural modes of human-computer interaction in Section 6. This is followed by concluding remarks in Section 7.

2 Gesture Modeling

In order to systematically discuss the literature on gesture interpretation, it is important to first consider what model the authors have used for the hand gesture. In fact, the scope of a gestural interface for HCI is directly related to the proper modeling of hand gestures. How to model hand gestures depends primarily on the intended application within the HCI context. For a given application, a very coarse and simple model may be sufficient. However, if the purpose is a natural-like interaction, a model has to be established that allows many if not all natural gestures to be interpreted by the computer. The following discussion addresses the question of modeling of hand gestures for HCI.

2.1 Definition of Gestures

Outside the HCI framework, hand gestures cannot be easily defined. The definitions, if they exist, are particularly related to the communicational aspect of the human hand and body movements. Webster's Dictionary, for example, defines gestures as "...the use of motions of the limbs or body as a means of expression; a movement usually of the body or limbs that expresses or emphasizes an idea, sentiment, or attitude." Psychological and social studies tend to narrow this broad definition and relate it even more to man's expression and social interaction [48]. However, in the domain of HCI the notion of gestures is somewhat different. In a computer controlled environment one wants to use the human hand to perform tasks that mimic both the natural use of the hand as a manipulator, and its use in human-machine communication (control of computer/machine functions through gestures). Classical definitions of gestures, on the other hand, are rarely, if ever, concerned with the former mentioned use of the human hand (so called practical gestures [48]).

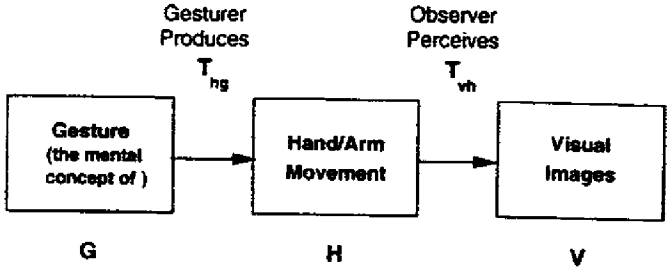


Fig. 2. Production and perception of gestures. Hand gestures originate as a mental concept G, are expressed (T_{hg}) through arm and hand motion H, and are perceived (T_{vh}) as visual images V.

Hand gestures are a means of communication, similar to spoken language. The production and perception of gestures can thus be described using a model commonly found in the field of spoken language recognition [85], [100]. An interpretation of this model, applied to gestures, is depicted in Fig. 2. According to the model, gestures originate as a gesturer's mental concept, possibly in conjunction with speech. They are expressed through the motion of arms and hands, the same way speech is produced by air stream modulation through the human vocal tract. Also, observers perceive gestures as streams of visual images which they interpret using the knowledge they possess about those gestures. The production and perception model of gestures can also be summarized in the following form:

$$H = T_{hg}G \tag{1}$$

$$V = T_{vh}H \tag{2}$$

$$V = T_{vh}(T_{hg}G) = T_{vg}G \tag{3}$$

Transformations T can be viewed as different *models*: T_{hg} is a model of hand or arm motion given gesture G, T_{vh} is a model of visual images given hand or arm motion H, and T_{vg} describes how visual images V are formed given some gesture G. The models are parametric, with the parameters belonging to their respective parameter spaces $\mathcal{M}_T$. In light of this notation, one can say that the aim of visual interpretation of hand gestures is to infer gestures G from their visual images V using a suitable gesture model T_{vg}, or

$$\hat{G} = T_{vg}^{-1}V \tag{4}$$

In the context of visual interpretation of gestures, it may then be useful to consider the following definition of gestures:

A hand gesture is a stochastic process in the gesture model parameter space $\mathcal{M}_T$ over a suitably defined time interval I.

Each realization of one gesture can then been seen as a *trajectory* in the model parameter space. For example, in performing a gesture the human hand's position in 3D space describes a trajectory in such space, Fig. 3. The stochastic property in the definition of gestures affirms their natural character: no two realizations of the same gesture will result in the same hand and arm motion or the same set of visual images. The presence of the time interval I suggests the gesture's dynamic nature.

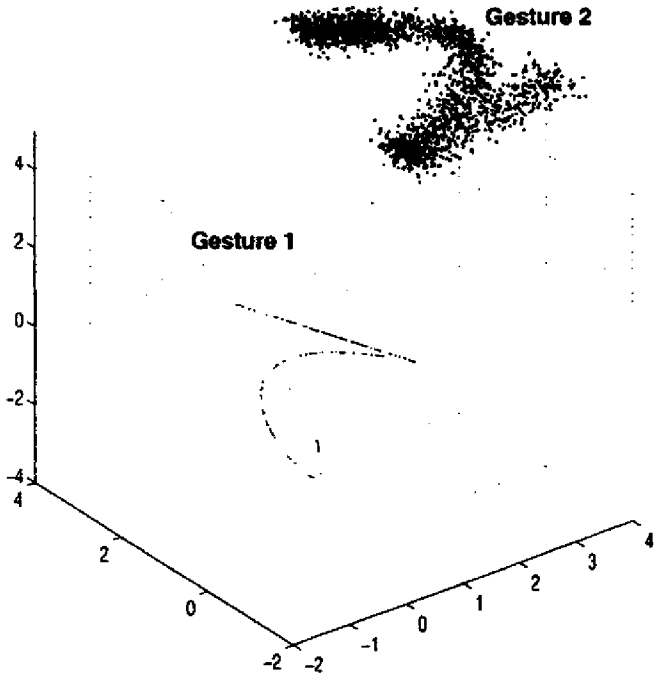


Fig. 3. Gesture as a stochastic process. Gestures can be viewed as random trajectories in parameter spaces which describe hand or arm spatial states. In this example, two different gestures are shown in a three dimensional parameter space. One realization of Gesture 1 is a trajectory in that space (solid line).

The gesture analysis and gesture recognition problems can then be posed in terms of the parameters involved in the above definition. For example, the problem of constructing the gestural model T over the parameter set $\mathcal{M}_T$, or the problem of defining the gesture interval I.

2.2 Gestural Taxonomy

Several alternative taxonomies have been suggested in the literature that deal with psychological aspects of gestures. Kendon [48] distinguishes "autonomous gestures" (that occur independently of speech) from "gesticulation" (gestures that occur in association with speech). McNeill and Levy [65] recognize three groups of gestures: iconic and metaphoric gestures, and "beats." The taxonomy that seems most appropriate within the context of HCI was recently developed by Quek [71], [72]. A slightly modified version of the taxonomy is given in Fig. 4.

All hand/arm movements are first classified into two major classes:

- gestures and
- unintentional movements.

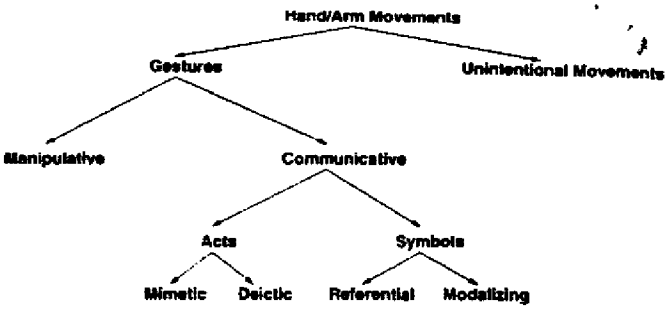


Fig. 4. A taxonomy of hand gestures for HCI. Meaningful gestures are differentiated from unintentional movements. Gestures used for manipulation (examination) of objects are separated from the gestures which possess inherent communicational character.

Unintentional movements are those hand/arm movements that do not convey any meaningful information. Gestures themselves can have two modalities:

- communicative and
- manipulative.

Manipulative gestures are the ones used to act on objects in an environment (object movement, rotation, etc.) Communicative gestures, on the other hand, have an inherent communicational purpose. In a natural environment they are usually accompanied by speech. Communicative gestures can be either acts or symbols. Symbols are those gestures that have a linguistic role. They symbolize some referential action (for instance, circular motion of index finger may be a referent for a wheel) or are used as modalizers, often of speech ("Look at that wing!" and a modalizing gesture specifying that the wing is vibrating, for example). In HCI context these gesture are, so far, one of the most commonly used gestures since they can often be represented by different static hand postures, as we will discuss further in Section 5. Finally, acts are gestures that are directly related to the interpretation of the movement itself. Such movements are classified as either mimetic (which imitate some actions) or deictic (pointing acts).

Taxonomy of gestures largely influences the way parameter space $\mathcal{M}_T$ and gesture interval I are determined. A related issue is the classification of gestural dynamics, which we consider next.

2.3 Temporal Modeling of Gestures

Since human gestures are a dynamic process, it is important to consider the temporal characteristics of gestures. This may help in the temporal segmentation of gestures from other unintentional hand/arm movements. In terms of our general definition of hand gestures, this is equivalent to determining the gesture interval I. Surprisingly, psychological studies are fairly consistent about the temporal nature of hand gestures. Kendon [48] calls this interval a "gesture phrase." It has been established that three phases make a gesture:

- preparation,
- nucleus (peak or stroke [65]), and
- retraction.

The preparation phase consists of a preparatory movement that sets the hand in motion from some resting position.

The nucleus of a gesture has some "definite form and enhanced dynamic qualities" [48]. Finally, the hand either returns to the resting position or repositions for the new gesture phase. An exception to this rule is the so called "beats" (gestures related to the rhythmic structure of the speech).

The above discussion can guide us in the process of temporal discrimination of gestures. The three temporal phases are distinguishable through the general hand/arm motion: "Preparation" and "retraction" are characterized by the rapid change in position of the hand, while the "stroke," in general, exhibits relatively slower hand motion. However, as it will be seen in Section 4, the complexity of gestural interpretation usually imposes more stringent constraints on the allowed temporal variability of hand gestures. Hence, a work in vision-based gesture HCI sometimes reduces gestures to their static equivalents, ignoring their dynamic nature.

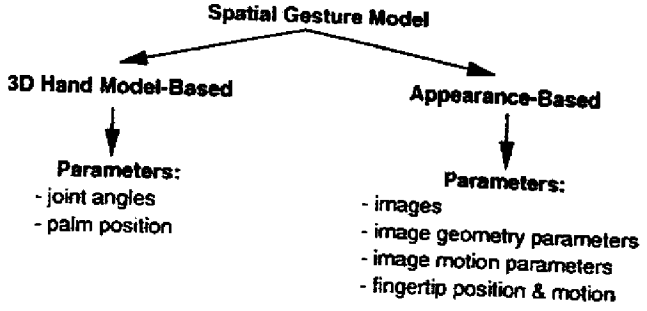


Fig. 5. Spatial models of gestures. 3D hand model-based models of gestures use articulated models of the human hand and arm to estimate the hand and arm movement parameters. Such movements are later recognized as gestures. Appearance-based models directly link the appearance of the hand and arm movements in visual images to specific gestures.

2.4 Spatial Modeling of Gestures

Gestures are observed as hand and arm movements, actions in 3D space. The description of gestures, hence, also involves the characterization of their spatial properties. In a HCI domain this characterization has so far been mainly influenced by the kind of application for which the gestural interface is intended. For example, some applications require simple models (like static image templates of the human hand in TV set control in [35]), while some others require more sophisticated ones (3D hand model used by [56], for instance).

If one considers the gesture production and perception model suggested in Section 2.1, two possible approaches to gesture modeling may become obvious. One approach may be to try to infer gestures directly from the visual images observed, as stated by (4). This approach has been often used to model gestures, and is usually denoted as *appearance-based* modeling. Another approach may result if the intermediate tool for gesture production is considered: the human hand and arm. In this case, a two step modeling process may be followed:

$$\hat{H} = T_{vh}^{-1} V \tag{5}$$

$$\hat{G} = T_{he}^{-1} \hat{H} \tag{6}$$

In other words, one can first model the motion and posture of the hand and arm $\hat{H}$ and then infer gestures $\hat{G}$ from the motion and posture model parameters. A group of models which follows this approach is known as *3D-model-based*.

Fig. 5 shows the two major approaches used in the spatial modeling of gestures. We examine the two approaches more closely in the following subsections.

2.4.1 3D Hand/Arm Model

The 3D hand and arm models have often been a choice for hand gesture modeling. They can be classified in two large groups:

- *volumetric models* and
- *skeletal models*.

Volumetric models are meant to describe the 3D visual appearance of the human hand and arms. They are commonly found in the field of computer animation [64], but have recently also been used in computer vision applications. In the field of computer vision volumetric models of the human body are used for *analysis-by-synthesis* tracking and recognition of the body's posture [52], [105]. Briefly, the idea behind the analysis-by-synthesis approach is to analyze the body's posture by synthesizing the 3D model of the human body in question and then varying its parameters until the model and the real human body appear as the same visual images. Most of the volumetric models used in computer animation are complex 3D surfaces (NURBS or nonuniform rational B-splines) which enclose the parts of the human body they model [64]. Even though such models have become quite realistic, they are too complex to be rendered in real-time. A more appealing approach, suitable to real-time computer vision, lies in the use of simple 3D geometric structures to model the human body [68]. Structures like *generalized cylinders* and *super-quadrics* which encompass cylinders, spheres, ellipsoids and hyper-rectangles are often used to approximate the shape of simple body parts, like finger links, forearm, or upperarm [6], [20], [29], [31], [37]. The parameters of such geometric structures are quite simple. For example, a cylindrical model is completely described with only three parameters: height, radius, and color. The 3D models of more complex body parts, like hands, arms, or legs, are then obtained by connecting together the models of the simpler parts [46]. In addition to the parameters of the simple models, these structures contain the information on connections between the basic parts. The information may also include constraints which describe the interaction between the basic parts in the structure. There are two possible problems in using such elaborate hand and arm models. First, the dimensionality of the parameter space is high (more than 23×3 parameters per hand). Second, and more importantly, obtaining the parameters of those models via computer vision techniques may prove to be quite complex.

Instead of dealing with all the parameters of a volumetric hand and arm model, models with a reduced set of equivalent joint angle parameters together with segment lengths are often used. Such models are known as *skeletal models*. Skeletal models are extensively studied in the human hand morphology and biomechanics [92], [95]. We briefly describe the basic notions relevant to our discussion. The human hand skeleton consists of 27 bones, divided in three groups:

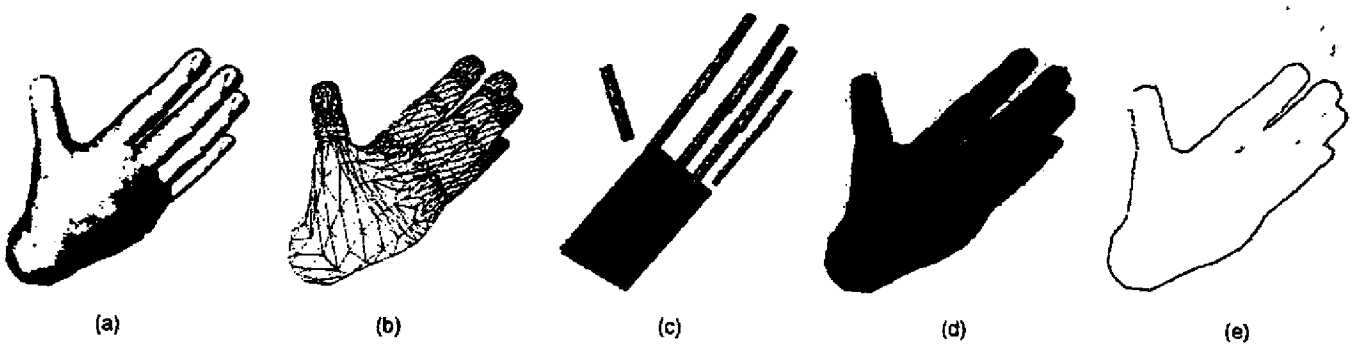


Fig. 6. Hand models. Different hand models can be used to represent the same hand posture. (a) 3D Textured volumetric model. (b) 3D wireframe volumetric model. (c) 3D skeletal model. (d) Binary silhouette. (e) Contour.

- carpals (wrist bones—eight),
- metacarpals (palm bones—five), and
- phalanges (finger bones—14).

The joints connecting the bones naturally exhibit different *degrees of freedom* (DoF). Most of the joints connecting carpals have very limited freedom of movement. The same holds for the carpal-metacarpal joints (except for the TM, see Fig. 7). Finger joints show the most flexibility: For instance, the MCP and the TM joint have two DoFs (one for extension/flexion and one for adduction/abduction), while the PIP and the DIP joints have one DoF (extension/flexion). Equally important to the notion of DoF is the notion of dependability between the movements in neighboring joints. For instance, it is natural to most people to bend (flex/extend) their fingers such that both PIP and DIP joints flex/extend. Also, there is only a certain range of angles that the hand joints can naturally assume. Hence, two sets of constraints can be placed on the joint angle movements: static (range) and dynamic (dependencies). One set of such constraints was used by Kuch [55] in his 26 DoF hand model:

Static Constraints	
Fingers	Thumb
$0 \le \theta^{\vee}_{MCP,\wedge} \le 90°$ $-15° \le \theta^{\wedge}_{MCP,\wedge} \le 15°$	
Dynamic Constraints	
$\theta^{\vee}_{PIP} = \frac{3}{2}\theta^{\vee}_{DIP}$ $\theta^{\vee}_{MCP} = \frac{1}{2}\theta^{\vee}_{PIP}$ $\theta^{\wedge}_{MCP} =$ $\frac{\theta^{\vee}_{MCP}}{90}\left(\theta^{\wedge}_{MCP,converse} - \theta^{\wedge}_{MCP,\wedge}\right) +$ $\theta^{\wedge}_{MCP,\wedge}$	$\theta^{\vee}_{IP} = \theta^{\wedge}_{MCP}$ $\theta^{\wedge}_{TM} = \frac{1}{3}\theta^{\wedge}_{MCP}$ $\theta^{\wedge}_{TM} = \frac{1}{2}\theta^{\wedge}_{MCP}$

where superscripts denote flexions/extensions ("y") or adduction/abduction ("x") movements in local, joint centered coordinate systems. In another example, Lee and Kunii [59], [60] developed a 27 degree of freedom hand skeleton model with an analogous set of constraints. Similar skeleton-based models of equal or lesser complexity have been used by other authors [4], [66], [76], [77], [97].

2.4.2 Appearance-Based Model

The second group of models is based on appearance of hands/arms in the visual images. This means that the model parameters are not directly derived from the 3D spatial description of the hand. The gestures are modeled by relating the appearance of any gesture to the appearance of the set of predefined, template gestures.

A large variety of models belong to this group. Some are based on deformable 2D templates of the human hands, arms, or even body [18], [21], [45], [49], [58]. Deformable 2D templates are the sets of points on the outline of an object, used as interpolation nodes for the object outline approximation. The simplest interpolation function used is a piecewise linear function. The templates consist of the average point sets, point variability parameters, and so-called external deformations. Average point sets describe the "average" shape within a certain group of shapes. Point variability parameters describe the allowed shape deformation (variation) within that same group of shapes. These two types of parameters are usually denoted as internal. For instance, the human hand in open position has one shape on the average, and all other instances of any open posture of the human hand can be formed by slightly varying the average shape. Internal parameters are obtained through *principal component analysis* (PCA) of many of the training sets of data. External parameters or deformations are meant to describe

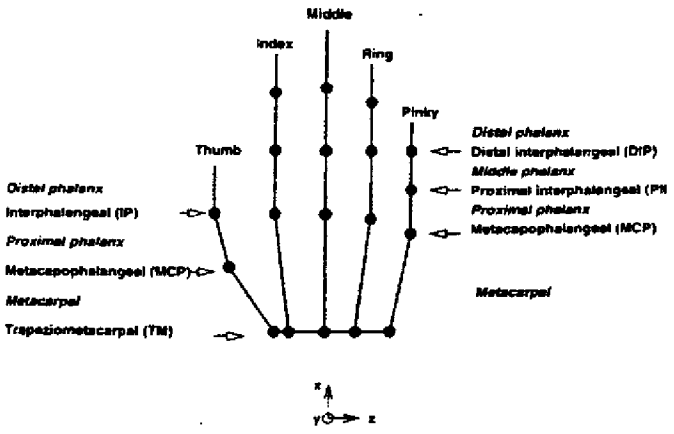


Fig. 7. Skeleton-based model of the human hand. The human hand skeleton consists of 27 bones. This model, on the other hand, approximates the anatomical structure using five serial link chains with 19 links.

the global motion of one deformable template. Rotations and translations are used to describe such motion. Template-based models are used mostly for hand-tracking purposes [18], [49]. They can also be used for simple gesture classification based on the multitude of classes of templates [58]. Trajectories of external parameters of deformable templates have also been used for simple gesture recognition [45]. Extensions of the 2D template approach to 3D deformable models have also been recently explored. For example, *3D point distribution model* has been employed for gesture tracking [42].

A different group of appearance-based models uses 2D hand image sequences as gesture templates. Each gesture from the set of allowed gestures is modeled by a sequence of representative image n-tuples. Furthermore, each element of the n-tuple corresponds to one view of the same hand or arm. In the most common case, only one (monoscopic) or two (stereoscopic) views are used. Parameters of such models can be either images themselves or some features derived from the images. For instance, complete image sequences of the human hands in motion can be used as templates per se for various gestures [25], [26]. Images of fingers only can also be employed as templates [22] in a finger tracking application. Another recently pursued approach has been to model different gestural actions by *motion history images* or MHIs [12]. MHIs are 2D images formed by accumulating the motion of every single pixel in the visual image over some temporal window. This way the intensity of the pixel in the MHI relates to how much prolonged motion is observed at that pixel.

The majority of appearance-based models, however, use parameters derived from images in the templates. We denote this class of parameters as *hand image property parameters*. They include: contours and edges, image moments, and image eigenvectors, to mention a few. Many of these parameters are also used as features in the analysis of gestures (see Section 3). Contours as a direct model parameter are often used: simple edge-based contours [17], [81] or "signatures" (contours in polar coordinates) [14] are some possible examples. Contours can also be employed as the basis for further eigenspace analysis [23], [67]. Other parameters that are sometimes used are image moments [80], [86]. They are easily calculated from hand/arm silhouettes or contours. Finally, many other parameters have been used: Zernike moments [79] and orientation histograms [34], for example.

Another group of models uses fingertip positions as parameters. This approach is based on the assumption that the position of fingertips in the human hand, relative to the palm, is almost always sufficient to differentiate a finite number of different gestures. The assumption holds in 3D space under several restrictions; some of them were noted by Lee and Kunii [59], [60]: The palm must be assumed to be rigid, and the fingers can only have a limited number of DoFs. However, most of the models use only 2D locations of fingertips and the palm [3], [28], [57]. Applications that are concerned with deictic gestures usually use only a single (index) fingertip and some other reference point on the hand or body [36], [57], [73].

3 Gesture Analysis

In the previous section, we discussed different approaches for modeling gestures for HCI. In this section we consider the analysis phase where the goal is to estimate the parameters of the gesture model using measurements from the video images of a human operator engaged in HCI. Two generally sequential tasks are involved in the analysis (see Fig. 8). The first task involves "detecting" or extracting relevant image features from the raw image or image sequence. The second task uses these image features for computing the model parameters. We discuss the different approaches used in this analysis.

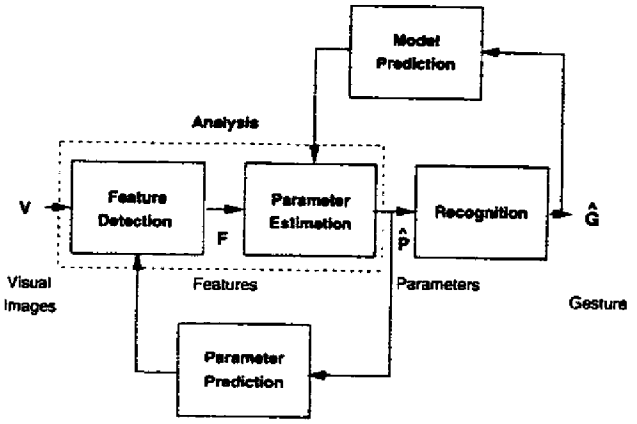


Fig. 8. Analysis and recognition of gestures. In the analysis stage, features F are extracted from visual images V. Model parameters $\hat{P}$ are estimated and possibly predicted. Gestures $\hat{G}$ are recognized in the recognition stage. Recognition may also influence the analysis stage by predicting the gesture model at the next time instance.

3.1 Feature Detection

Feature detection stage is concerned with the detection of features which are used for the estimation of parameters of the chosen gestural model. In the detection process it is first necessary to localize the gesturer. Once the gesturer is localized, the desired set of features can be detected.

3.1.1 Localization

Gesturer localization is a process in which the person who is performing the gestures is extracted from the rest of the visual image. Two types of cues are often used in the localization process:

- *color cues* and
- *motion* cues.

Color cues are applicable because of the characteristic color footprint of the human skin. The color footprint is usually more distinctive and less sensitive to illumination changes in the hue-saturation space than in the standard (camera capture) RGB color space. Most of the color segmentation techniques rely on histogram matching [4] or employ a simple look-up table approach [51], [73] based on the training data for the skin and possibly its surrounding areas. The major drawback of color-based localization techniques is the variability of the skin color footprint in different lighting conditions. This frequently results in undetected skin

regions or falsely detected nonskin textures. The problem can be somewhat alleviated by considering only the regions of a certain size (scale filtering) or at certain spatial position (positional filtering). Another common solution to the problem is the use of restrictive backgrounds and clothing (uniform black background and long dark sleeves, for example.) Finally, many of the gesture recognition applications resort to the use of uniquely colored gloves or markers on hands/fingers [19], [28], [57], [60], [62]. The use of background restriction or colored gloves makes it possible to localize the hand efficiently and even in real-time, but imposes the obvious restriction on the user and the interface setup. On the other hand, without these restrictions some of the color-based localization techniques such as the ones that use histogram matching are computationally intensive and currently hard to implement in real-time.

Motion cue is also commonly applied for gesturer localization and is used in conjunction with certain assumptions about the gesturer. For example, in the HCI context, it is usually the case that only one person gestures at any given time. Moreover, the gesturer is usually stationary with respect to the (also stationary) background. Hence, the main component of motion in the visual image is usually the motion of the arm/hand of the gesturer and can thus be used to localize her/him. This localization approach is used in [35], [72]. The disadvantage of the motion cue approach is in its assumptions. While the assumptions hold over a wide spectrum of cases, there are occasions when more than one gesturer is active at a time (active role transition periods) or the background is not stationary.

To overcome the limitations of the individual cues for localization, several approaches have been suggested. One approach is the *fusion* of color, motion and other visual cues [7] or the fusion of visual cues with nonvisual cues like speech or gaze [83]. The potential advantage of the so-called *multimodal* approach has not yet been fully exploited for hand localization though it has been explored for face localization in video [38]. We discuss the multimodal approach further in Section 6. Another way in which the localization problem can be substantially eased is by the use of *prediction* techniques. These techniques provide estimates of the future feature locations based on the model dynamics and the previously known locations. We will discuss this further in Section 3.2.

3.1.2 Features and Detection

Even though different gesture models are based on different types of parameters, the image features employed to compute those parameters are often very similar. For example, some 3D hand/arm models and models that use finger trajectories all require fingertips to be extracted first. Color or gray scale images which encompass hands and arms or gesturers themselves are often used as the features. This choice of features is very common in the appearance-based models of gestures where sequences of images are used to form *temporal templates* of gestures [25]. The computational burden of the detection of these features is relatively low and is associated mostly with the gesturer localization phase. Another approach to using whole images as features is related to building of the so-called *motion energy (history) images* or MEI (MHI). MEIs are 2D images which unify the motion information of a sequence of 2D images by accumulating the motion of some characteristic image points over the sequence [30]. One simple yet effective choice of characteristic points is the whole image itself [12]. As discussed in Section 3.2, such features can represent a valid choice for the recognition of communicative gestures. However, their applicability to hand and arm tracking and recognition of manipulative gestures seems to be limited.

Hand and arm silhouettes are among the simplest, yet most frequently used features. Silhouettes are easily extracted from local hand and arm images in the restrictive background setups. In the case of complex backgrounds, techniques that employ color histogram analyses, as described in the gesturer localization phase, can be used. Examples of the use of silhouettes as features are found in both 3D hand model-based analyses [56] as well as in the appearance-based techniques (as in [54], [69]). Naturally, the use of such binary features results in a loss of information which can effect the performance especially for 3D hand posture estimators. For example, in the 3D hand posture estimation problem of [56], the binary silhouette prevents the accurate estimation of the positions of some fingers.

Contours represent another group of commonly used features. Several different edge detection schemes can be used to produce contours. Some are extracted from simple hand-arm silhouettes, and thus, are equivalent to them, while the others come from color or gray-level images. Contours are often employed in 3D model-based analyses. In such cases, contours can be used to select finger and arm link candidates through the clustering of the sets of parallel edges [29], [31], or through image-contour-to-model-contour matching [37], for example. In appearance-based models, on the other hand, many different parameters can be associated with contours: for instance "signatures" (description in polar coordinates of the points on the contour [14]) and "size functions" [96].

A frequently used feature in gesture analysis is the fingertip. Fingertip locations can be used to obtain parameters of both the 3D hand models and the 2D appearance-based gestural models (see Section 3.2). However, the detection of fingertip locations in either 3D or 2D space is not trivial. A simple and effective solution to the fingertip detection problem is to use marked gloves or color markers to designate the characteristic fingertips (see [19], [28], [57], [60], [93], for instance). Extraction of fingertip location is then fairly simplified and can be performed using color histogram-based techniques. A different way to detect fingertips is to use pattern matching techniques: templates can be images of fingertips [22] or fingers [77] or generic 3D cylindrical models [27]. Such pattern matching techniques can be enhanced by using additional image features, like contours [76]. Some fingertip extraction algorithms are based on the characteristic properties of fingertips in the image. For instance, curvature of a fingertip outline follows a characteristic pattern (low-high-low) which can be used for the feature detection [63], [97]. Other heuristics can be used as well. For example, for deictic gestures it can be assumed that the finger represents the foremost point of the hand [63], [73]. Finally, many other indirect approaches in detection of

fingertips have be employed in some instances, like image analysis using specially tuned Gabor kernels [66]. The main hindrance in the use of fingertips as features is their susceptibility to occlusions. Very often one or more fingers are occluded by the palm from a given camera viewpoint and direction. The most obvious solution to this occlusion problem involves the use of multiple cameras [60], [76]. Other solutions are based on the estimation of the occluded fingertip positions based on the knowledge of the 3D model of the gesture in question [77]. More often, however, restrictions are placed on the user to posture her/his hand so that the occlusions are minimized.

3.2 Parameter Estimation

Computation of the model parameters is the last stage of the gesture analysis phase. In the gesture recognition systems, this is followed by the recognition stage, as shown in Fig. 8. For hand or arm tracking systems, however, the parameter computation stage usually produces the final output. The type of computation used depends on both the model parameters and the features that were selected.

3.2.1 Estimation of 3D Model Parameters

As mentioned in Section 2.4.1, two sets of parameters are used in 3D hand models—angular (joint angles) and linear (phalangae lengths and palm dimensions). The estimation of these *kinematic* parameters from the detected features is a complex and cumbersome task. The process involves two steps:

- the initial parameter estimation and
- the parameter update as the hand gesture evolves in time.

All of the 3D hand models employed so far assume that all the linear parameters are known a priori. This assumption reduces the problem of finding the hand joint angles to an *inverse kinematics* problem. Given a 3D position of the end-effectors and the base of a kinematic chain, the inverse kinematic's task is to find the joint angles between the links in the chain. The 3D model of the hand can then be viewed as a set of five serial kinematic chains (finger links) attached to a common base (palm). The finger tips now play the role of the end-effectors in the chains. Inverse kinematic problems are in general ill-posed, allow for multiple solutions, and are computationally expensive. The use of *constraints on parameter values* (see Section 2.4.1) somewhat alleviates those problems. Nevertheless, alternative approaches to 3D hand parameter estimation have been often sought. One automated solution to the initial parameter estimation problem was proposed by [59] through a two phase procedure using the accumulated displacement torque approach. The first phase involves the initial wrist positioning while the second phase deals with palm/finger adjustment. The procedure is applied recursively until the accumulated torque excerpted on all links reaches a local minimum, constrained on a set of static and dynamic joint angle constraints. Even though this approach produces accurate parameter estimates, it is computationally very expensive and thus not applicable to real-time problems. Some simpler solutions involve a user interactive model parameter initialization [56]. Another approach is to use interpolation of the discretized *forward kinematics* mappings to approximate the inverse kinematics [4]. Given a table of the discrete values of the joint angles and the resulting fingertip positions it is possible to estimate the values of the joint angles for a nontable value of the fingertip position.

Once the hand model parameters are initially estimated, the parameter estimates can be updated using some kind of prediction/smoothing scheme. A commonly used scheme is *Kalman filtering and prediction*. This scheme works under the assumption of small motion displacements and a known parameter update (motion) model. Such a model can be derived from a known hand kinematics model, using the inverse Jacobian mapping from the space of measurable linear displacements into the space of desired angular displacements. A variation of this approach was used by [76] in a real-time 27 degree of freedom hand tracker. On the other hand, when the dynamics are not explicitly available a simple scheme like the one reported in [56] may be employed. In this scheme, a simple silhouette matching between the 3D hand model and the real hand image was used to obtain satisfactory parameter estimation and update.

It is necessary to stress three major drawbacks associated with the mentioned 3D hand model parameter estimation approach. One has to do with the obvious computational complexity of any task involving the inverse kinematics. The other, potentially more serious problem, is due to occlusions of the fingertips used as the model features. An obvious, yet expensive, solution is to use multiple cameras. Another possible solution was developed by [77], and involves the use of finger links as features built upon a set of rules designed to resolve the finger occlusions. The last drawback stems from the employed assumption that the linear dimensions of the hand are known, which is necessary in the inverse kinematics problems. Thus, any change in scale of the hand images always results in inaccurate estimates of the hand joint angles. Finally, it should be pointed out that the knowledge of the exact hand posture parameters seem unnecessary for the recognition of communicative gestures [71] although the exact role of 3D hand parameter in gesture recognition is not clear.

The motion of the arm and hand also plays a role in gesture recognition although again the exact nature of this role is controversial [71]. The estimates of such motion can be made using either 3D space or 2D space. The 3D arm parameters are similar to the ones used in the 3D hand model description—joint angles and links. Hence, similar techniques could be used for the 3D arm parameter estimation. However, because of the simpler macro structure of the arm (the arm can be viewed as a serial kinematic chain with only three links) and fewer occlusions, it is possible to use less complex approaches to the arm parameter estimation. Most of the approaches match simplified geometrical 3D models of the arm (see Section 2.4.1) to the visual images of a real arm. The commonly used features are edges and contours which are used to estimate the link axes. For example, [29], [31] used sets of symmetry axes of line segments to estimate the axes of generalized cylinders which modeled the arm links and the upper body. In another example, [37] used chamfer matching to align the 3D tapered super-quadrics model of the upper body to two camera

visual images. In a slightly different approach [105] used fusion of color "blob" features and contours to detect elements of the "blob" representation of the human body.

As is the case in the 3D hand model parameter estimation, a good initialization of arm parameters is crucial for many of these techniques to work. This is because these techniques often rely on dynamic updates of the parameters through a Kalman-based filtering/prediction scheme rather than a global initial search. Also, the introduction of constraints on the position and motion of the arm links, as in [37], can greatly improve the estimation process.

3.2.1 Estimation of Appearance Parameters

Many different appearance-based models have been reported. The estimation of the parameters of such models usually coincides with the estimation of some compact description of the image or image sequence.

Appearance models based on the visual images per se are often used to describe gestural actions. These models are often known as the *temporal* models. Various different parameters of such models are used. In the simplest case the parameters can be selected as the sets of key visual frames, as in [25]. Another possibility is to use the eigendecomposition representation of visual images in the sequence with respect to an average image [104]. A promising direction has recently been explored: accumulation of *spatio/temporal* information of a sequence of visual images into a single 2D image, a so-called *motion history image* (MHI) [12]. Such a 2D image can then be easily parameterized using one of 2D image description techniques, such as the geometric moment description or eigendecomposition. A major advantage of using these appearance models is the inherent simplicity of their parameter computation. However, this advantage may be outweighed by the loss of precise spatial information which makes them especially less suited for manipulative gestures.

Deformable 2D template-based models are often employed as the *spatial* models of hand and arm contours or even the whole human body [45]. They are usually specified through a pair of mean values of the template nodes $\mathbf{m}$ and their covariances $\mathbf{v}$ [21], [49]. The parameter estimates are obtained through *principal component analysis* (PCA) on sets of training data. Different parameters are then used to describe individual gestures. The variation of the node parameters allows for the same gesture to be recognized despite the fact that it takes on slightly different appearance when performed by different gesturers. An extension of this approach to 3D deformable templates or *point distribution models* (PDM) was recently suggested in [42]. Associated with the deformable template model parameters are also the so called external deformations or global motion parameters (rotation and translation of the hand or body in the workspace). The updates of the model parameters can then be estimated in a framework similar to the one used for rigid motion estimation. The main difference is that in the case of deformable templates an additional displacement due to the template variability $d\mathbf{v}$ also needs to be estimated [42], [49]. While the parameter computation for such deformable models is not extensive in the parameter update phase, it can be overwhelming during the initialization. On the other hand, deformable models can provide sufficient information for the recognition of both classes of gestures: manipulative and communicative.

Finally, a wide class of appearance models uses silhouettes or gray level images of the hands. In such cases, the model parameters attempt to capture a description of the shape of the hand while being relatively simple. A very commonly employed technique is built upon the geometric moment description of hand shapes [14], [69], [86]. Usually, moments of up to the second order are used. Some other techniques use Zernike moments [79] whose magnitudes are invariant to rotation, thus allowing for rotation invariant shape classification. Many other shape descriptors have also been tested—orientation histograms [34], for example, represent summary information of small patch orientations over the whole image. This parameter tends to be invariant under changes in the lighting conditions which often occur during the hand motion. Even though the parameters of the above mentioned models are easy to estimate, they are also very sensitive to the presence of other, nonhand objects in the same visual image. This means that tight "bounding boxes" around the hand need to be known at all times during the hand motion. This in turn implies either the use of good motion prediction or restriction to the hand postures. Like the other parameter estimation tasks, the reported estimation of motion parameters are usually based on simple Newtonian dynamics models and Kalman-based predictors.

4 Gesture Recognition

Gesture recognition is the phase in which the data analyzed from the visual images of gestures is recognized as a specific gesture. Analogously, using the notation we established in Section 2, the trajectory in the model parameter space (obtained in the analysis stage) is classified as a member of some meaningful subset of that parameter space. Two tasks are commonly associated with the recognition process:

- Optimal partitioning of the parameter space and
- Implementation of the recognition procedure.

The task of optimal partitioning is usually addressed through different *learning-from-examples* training procedures. The key concern in the implementation of the recognition procedure is computational efficiency. We discuss each of the above issues in more detail.

The task of optimal partitioning of the model parameter space is related to the choice of the gestural models and their parameters, as mentioned in Section 2. However, most of the gestural models are not implicitly designed with the recognition process in mind. This is especially true for the models of static gestures or hand postures. For example, most of the static models are meant to accurately describe the visual appearance of the gesturer's hand as they appear to a human observer. To perform recognition of those gestures, some type of parameter clustering technique stemming from *vector quantization* (VQ) is usually used. Briefly, in vector quantization, an n-dimensional space is partitioned into convex sets using n-dimensional hyperplanes, based on training examples and some metric for determining

the nearest neighbor. If the parameters of the model are chosen especially to help with the recognition, as for example in [23], [90], the separation of classes belonging to different gestures can be done easily. However, if the model parameters are not chosen to properly describe the desired classes, the separation of the classes, and thus, accurate recognition in that parameter space may not be possible. For example, with contour descriptors, several hand postures would be confused during classification and recognition. Therefore, contours are often used for hand or arm tracking rather than for the recognition of hand postures. Parameters of other appearance-based static hand models often suffer from the same problem. For example, it is known that geometric moment parameters are not rotationally invariant. Thus, a small change in rotation of the same hand posture can cause it to be classified as a different posture. This problem can be somewhat alleviated if the chosen training hand postures classes are either very distinct or somehow normalized with respect to rotation. Another approach is to introduce different model parameters, such as Zernike moments [79] or orientation histograms [34], which posses 2D rotational invariance property. Some other models, based on eigenspace decompositions, are more discriminant and hence produce higher recognition accuracies under classical clustering techniques [103]. The problem of accurate recognition of postures which use model parameters that cluster in nonconvex sets can also be solved by selecting nonlinear clustering schemes. Neural networks are one such option, although their use for gesture recognition has not been fully explored [50]. Such nonlinear schemes are often sensitive to training and may be computationally expensive. Further, there is an inherent limitation in the discrimination capability by considering a 2D projection (or appearance) of a 3D hand when trying to capture a wide class of natural gestures. On the other hand, the use of 3D hand and gesture models offers the possibility of improving recognition, but because of the complexity of model parameter computation they are not often used for hand posture recognition.

Gestural actions, as opposed to static gestures, involve both the temporal and the spatial context [16]. As in the case of static posture recognition, the recognition of gestural actions depends on the choice of gestural models. Most of the gestural models, as seen in Section 2, produce trajectories in the model's parameter space. Since gestural action possess temporal context, the main requirement for any clustering technique used in their classification is that it be *time instance invariant* and *time scale invariant*. For example, a clapping gesture should be recognized as such whether it is performed slowly or quickly, now, or in 10 minutes. Numerous signal recognition techniques deal with such problems, the most prominent of these being automatic speech recognition (ASR). Since both speech as well as gestures are a means of natural human communication, an analogy is drawn between them and computational tools developed for ASR are frequently used in gesture recognition.

In speech recognition problems, a long standing task has been to recognize spoken words independent of their duration and variation in pronunciation. A tool called the *Hidden Markov Models* or HMM [74] has shown tremendous success is such tasks. HMM is a doubly stochastic process, a probabilistic network with *hidden* and *observable states*. The hidden states "drive" the model dynamics—at each time instance the model is in one of its hidden states. Transitions between the hidden states are governed by probabilistic rules. The observable states produce outcomes during hidden state transitions or while the model is in one of its hidden states. Such outcomes are measurable by an outside observer. The outcomes are governed by a set of probabilistic rules. Thus, an HMM can be represented as a triplet (A, b, π), where A is called the (hidden) state transition matrix, b describes the probabilities of the observation states, and π is the initial hidden state distribution. It is common to assume that the hidden state space is discrete, and that the observables are allowed to assume a continuum of values. In such cases, b is usually represented as a mixture of Gaussian (MOG) probability density functions. In automatic speech recognition, one HMM is associated with each different unit of speech (phoneme or sometimes word). Analogously, in the recognition of gestural actions, one HMM can be associated with each different gesture. In speech, the observables take on values of the linear prediction cepstrum coefficients (LPC cepstrum). In gestures, the observable is a vector of the spatial model parameters, like geometric moments [69], Zernike moments [79], or eigen image coefficients [103]. The process of association of different HMMs with different gestures (speech) units is denoted as training. In this process the parameters of the HMM (A, b, π) are modified so that the chosen model "best" describes the spatio/temporal dynamics of the desired gestural action. The training is usually achieved by optimizing the *maximum likelihood* measure $\log(Pr(observation \mid model))$ over a set of training examples for the particular gesture associated with the model. Such optimization involves the use of computationally expensive *expectation-maximization* or EM procedures, like the Baum-Welch algorithm [74]. However, any such training procedure involves a step based on *dynamic programming* or DP which in turn has a *dynamic time warping* or DTW property. This means that the variability in duration of training samples is accounted for in the model. The same is true for the recognition or model evaluation process. In that process, a gesture trajectory is tested over the set of trained HMMs in order to decide which one it belongs to. A probability of the gesture being produced by each HMM is evaluated using the *Viterbi* algorithm [74]. Obviously, the larger the number of trained HMMs (gestures) is, the more computationally demanding the recognition procedure. Problems like this one have successfully been solved by imposing an external set of rules or *grammar* which describes the language sentence structure or how the trained units (gestures or spoken) can be "connected" in time [69], [86]. Several problems are related to the use of the HMM as a recognition tool. For example, in its original formulation, an HMM is a first order stochastic process. This implies that the (hidden) state of the model at time instance i depends only on the state at time $i - 1$. While this model may be sufficient for some processes, it often results in lower recognition rates for the processes which do not follow the first order Markov property. As in speech, such problems can be somewhat reduced by extending the parameter vectors with the time

derivatives of the original parameters [15]. It is also possible to derive a higher order HMMs, however such models do not share the computational efficiency of the first order models [75]. Another possible drawback of classical HMMs is the assumption that probability distribution functions or pdfs of the observables can be modeled as mixtures of Gaussians. The main reason for modeling the observables as MOGs is in training. In such cases, the HMM parameters can be efficiently computed using the Baum-Welch algorithm. Extensions in this direction have been achieved in ASR by using neural networks to model the observation pdfs [13]. Unfortunately, the training procedure in that case is computationally overwhelming. Also, in the original formulation a HMM is assumed to be *stationary*. This means that the observation probabilities do not vary in time. Such assumption may hold over short time intervals. However, since a complete gestural action is often modeled as a single HMM, the stationarity of observation pdfs may not hold true in this case. Nonstationary HMMs have been formulated for ASR [89], but have not yet been used for gesture recognition. Finally, it is interesting to note that hidden states of the HMM may possibly be viewed as the temporal phases known from the psychological studies of gesture (Section 2.3).

Another approach to recognition of gestural actions proposed recently is based on *temporal templates, so called motion energy* [30] or *motion history images* (MHIs) [12] (see Section 3.2). Such motion templates accumulate the motion history of a sequence of visual images into a single 2D image. Each MHI is parameterized by the length of the time history window that was used for its computation. To achieve time duration invariance, the templates are calculated for a set of history windows of different durations, ranging between two predefined values. The recognition is then simply achieved using any of the 2D image clustering techniques, based on the sets of trained templates. An advantage of a such temporal template approach is in its extreme computational simplicity. However, the fact that the motion is accumulated over the *entire* visual image can result in artifacts being introduced by motions of unrelated objects or body parts present in the images.

A successful recognition scheme should also consider the time-space context of any specific gesture. This can be established by introducing a grammatical element into recognition procedure. The grammar should reflect the linguistic character of communicative gestures as well as spatial character of manipulative gestures. In other words, only certain subclasses of gestural actions with respect to the current and previous states of the HCI environment are (naturally) plausible. For example, if a user reaches (performs a valid manipulative gesture) for the coffee cup handle and the handle is not visible from the user's point of view, the HCI system should discard such a gesture. Still, only a small number of the systems so far exploits this. The grammars are simple and usually introduce artificial linguistic structures: they build their own "languages" that have to be learned by the user [36], [50], [73], [80].

The computational complexity of a recognition approach is important in the context of HCI. The trade-offs involved across various approaches is a classical one—model complexity, versus richness of the gesture classes, versus recognition time. The more complex the model is, the wider class of gestures to which it can be applied. The computational complexity increases, and, hence, the recognition time. Most of the 3D model-based gesture models are characterized by more than 10 parameters. Their parameter calculation (gesture analysis) requires computationally expensive successive approximation procedures (the price of which is somewhat lowered using prediction-type analysis). The systems based on such models rarely show close to real-time performance. For example, the time performance ranges from 45 minutes per single frame in [59] (although it does not use any prediction element) to 10 frames per second in [76]. Yet potentially, the 3D models can be used to capture the richest sets of hand gestures for HCI. The appearance-based models are usually restricted in their applicability to a narrower subclass of HCI applications, enhancements of the computer mouse concept [22], [35], [36], [50], [73], or hand posture classification [43], [63], [66], [67], [81], [86]. On the other hand, because of the lower complexity of the appearance-based models they are easier to implement in real-time and more widely used.

5 Applications and Systems

Recent interest in gestural interface for HCI has been driven by a vast number of potential applications (Fig. 9). Hand gestures as a mode of HCI can simply enhance the interaction in "classical" desktop computer applications by replacing the computer mouse or similar hand-held devices. They can also replace joysticks and buttons in the control of computerized machinery or be used to help the physically impaired to communicate more easily with others. Nevertheless, the major impulse to the development of gestural interfaces has come from the growth of applications situated in *virtual environments* (VEs) [2], [53].

Hand gestures in natural environments are used for both manipulative actions and communication (see Section 2). However, the communicative role of gestures is subtle, since hand gestures tend to be a supportive element of speech (with the exception of deictic gestures, which play a major role in human communication). Manipulative aspect of gestures also prevails in their current use for HCI. However, some applications have emerged recently which take advantage of the communicative role of gestures. We present a brief overview of several application driven systems with interfaces based on hand gestures.

Most applications of hand gestures portray them as the manipulators of *virtual objects* (VOs). This is depicted in Fig. 9. VOs can be computer generated graphics, like simulated 2D and 3D objects [14], [22], [44], [36] or windows [50], [73], or abstractions of computer-controlled physical objects, such as device control panels [4], [35], [36] or robotic arms [18], [43], [47], [94]. To perform manipulations of such objects through HCI a combination of coarse tracking and communicative gestures is currently being used. For example, to direct the computer to rotate an object a user of such an interface may issue a two-step command: *<select object> <rotate object>*. The first action uses coarse hand tracking to move a pointer in the VE to the vicinity of the object.

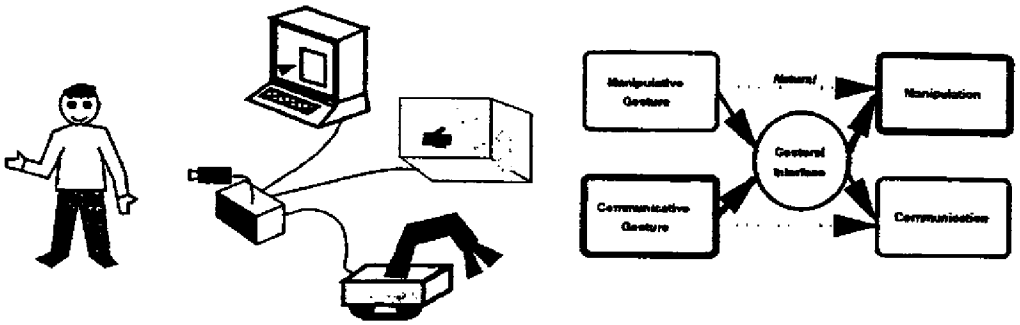


Fig. 9. Applications of gestural interface for HCI. Unlike the gestures in a natural environment, both manipulative and communicative gestures in HCI can be employed to direct manipulations of objects or to convey messages.

To rotate the object, the user rotates his/her hand back and forth producing a *metaphor* for rotational manipulation [14]. One may then pose the question: "Why use the communicative gestures for manipulative actions?" Communicative gestures imply a finite (and usually small) vocabulary of gestures that has to be learned, whereas the manipulative ones are natural hand/arm movements. To answer this question, one has to consider the complexity of analysis and recognition of each type of gestural models (Section 3 and Section 4). The 3D hand model-based gestural models are well suited for modeling of both manipulative and communicative gestures, while the appearance-based models of gestures are mostly applicable to the communicative ones. However, the use of 3D hand model-based gesture models are computationally more expensive than that of the appearance-based models (see Section 4). Therefore, to achieve a usable (real-time) performance one has to usually resort to the less desirable appearance-based models of gestures. Recently, however, with the increase in computing power, simplified hand/head blob models [6], [105] have been considered for applications which use communicative gesture recognition [10]. Such models are simple enough to be analyzed in real-time and are used for recognition of a small set of communicative gestures. For example, [10] used such a model followed by a HMM classifier to recognize eighteen *T'ai Chi* gestures. The system was intended to provide a virtual environment for the relaxation of cancer patients.

A brief summary of characteristics of some of the systems aimed at the application of hand gestures for HCI is given in Table 1. It summarizes the basic modeling technique used for the gestures, the class of gesture commands that are interpreted, and the reported performance in terms of the speed of processing.

Not all of the applications of hand gestures for HCI are meant to yield manipulative actions. Gestures for HCI can also be used to convey messages for the purpose of their analysis, storage or transmission. Video-teleconferencing (VTC) and processing of American sign language (ASL) provide such opportunities. In VTC applications, reduction of bandwidth is one of the major issues. A typical solution is to use different coding techniques. One such technique is model-based coding where image sequences are described by the states (e.g., position, scale, and orientation) of all physical objects in the scene (human participants in the case of VTC) [1], [40]. Only the updates of descriptors are sent while at the receiving end a computer generated model of physical objects is driven using the received data. Model-based coding for VTC, therefore, requires that the human bodies be modeled appropriately. Depending on the amount of detail desired, this can be achieved by only coarse models of the upper body and limbs [20], or finely tuned models of human faces or hands. Modeling of hand/arm gestures can then be of substantial value for such applications.

Recognition of ASL is often considered as another application that naturally employs human gestures as means of communication. Such applications could play a vital role in communication with people with a communication impairment like deafness. A device which could automatically translate ASL hand gestures into speech signals would undoubtedly have a positive impact on such individuals. However, the more practical reason for using the ASL as a test bed for the present hand gesture recognition systems is its well-defined structure compared to other natural gestures humans use. This fact implies that the appearance-based modeling techniques are particularly suited for such ASL interpretation, as was proven in several recent applications [86], [96].

There are numerous prospective applications of vision-based hand gesture analysis. The applications mentioned so far are only the first steps toward using hand gestures in HCI. The need for further development is thus quite clear. We discuss several important research issues that need to be addressed toward incorporating natural hand gestures into the HCI.

6 Future Directions

To fully exploit the potential of gestures in HCI environments, the class of recognizable gestures should be as broad as possible. Ideally, any and every gesture performed by the user should be unambiguously interpretable, thus allowing for *naturalness* of the interface. However, the state of the art in vision-based gesture recognition does not provide a satisfactory solution for achieving this goal. Most of the gesture-based HCI systems at the present time address a very narrow group of applications: mostly symbolic commands based on hand postures or 3D-mouse type of pointing (see Section 5). The reason for this is the complexity associated with the analysis (Section 3) and recognition (Section 4) of gestures. Simple gesture models make it possible to build real-time gestural interfaces—for example, pointing direction can be quickly found from the silhou-

TABLE 1
SYSTEMS THAT EMPLOY HAND GESTURES FOR HCI

Application	Gestural Modeling Technique	Gestural Commands	Complexity (Speed)
CD Player Control Panel [4]	Hand silhouette moments	Tracking only	30 fps[1]
Staying Alive [10]	3D hand / head blob model [6]	Tracking & HMM-based recognition	real time
Virtual Squash [14]	Hand silhouette moments & contour "signature"	Tracking & three metaphors	10.6 fps
FingerPaint [22]	Fingertip template	Tracking only	n.a.[2]
ALIVE [26]	Template correlation	Tracking combined with recognition of facial expressions	real-time
Computer Game Control [33]	Image moments using dedicated hardware	Hand & body posture recognition	real-time
TV Display Control [35]	Template correlation	Tracking only	5 fps
FingerPointer [36]	Heuristic detection of pointing action	Tracking and one metaphor combined with speech	real-time
Window Manager [50]	Hand pose recognition using neural networks	Tracking & four metaphors	real-time
GestureComputer [63]	Image moments & fingertip position	Tracking and six metaphors	10-25 fps
FingerMouse [73]	Heuristic detection of pointing action	Tracking only	real-time
DigitEyes [76]	27 DoF 3D hand model	Tracking only	10 fps
Robot manipulator guidance [18]	active contour	pointing	real-time
ROBOGEST [43]	Silhouette Zernike moments	Six metaphors	1/2 fps
Automatic robot instruction	Fingertip position in 2D	Grasp tracking	n.a.
Robot manipulator control [94]	Fingertip positions in 3D	Six metaphors	real-time
Hand sign recognition [24]	Most expressive featur cameres (MEF) of images	40 signs	n.a.
ASL recognition [86]	Silhouette moments & grammar	40 words	5 fps

1. *Frames per second.*
2. *Not available.*

We choose speed as the measure of complexity of interpretation given the lack of any other accurate measure. Note, however, that different applications may be implemented on different computer systems with different levels of optimization.

(a) (b)

Fig. 10. Silhouettes and gray-scale images of two different hand postures. The silhouette in (a) can also be interpreted as the reflection about the vertical axes of the silhouette in (b). Hence, the two silhouettes do not unambiguously define the hand posture.

ettes of the human hand in relatively nonrestrictive environments ([36],[73]). However, as it can be seen from Fig. 10 to find the hand posture and thus distinguish between the two gestures from the simple image appearance (silhouettes) is sometimes quite difficult.

Real-time interaction based on 3D hand model-based gesture analysis is yet to be demonstrated. The use of 3D models are mostly confined to hand tracking and hand posture analysis. Yet, the analysis of the parameters of the 3D hand model-based models can result in a wider class of hand gestures that can be identified than the analysis linked with the appearance-based models. This leads us to the conclusion that, from the point of the naturalness of HCI, the 3D hand model-based approaches offer more promise than the appearance-based models. However, this prospect is presently hindered by a lack of speed and the restrictiveness of the background in the 3D hand model-based approaches. The first problem is associated with the complexity of the model and the feature extraction. Fingertip positions seem to be a very useful feature (see Section 3.1.2), yet sometimes difficult to extract. A possible solution to this problem may employ the use of skin and nail texture to distinguish the tips of the fingers. Additionally, the computational complexity of estimating the

model parameters (Section 3.2) can be reduced by choosing an optimal number of parameters that satisfies a particular level of naturalness and employing parallelization of the computations involved.

Several other aspects that pertain to the construction of a natural HCI need to be adequately addressed in the future. One of the aspects involves the two-handed gestures. Human gestures, especially communicative, naturally employ actions of both hands. Yet, many of the vision-based gesture systems focus their attention on single-hand gestures. Until recently, the single-hand gesture approach has been almost inevitable. First, many analysis techniques require that the hands be extracted from global images. If the two-handed gestures are allowed, several ambiguous situations that do not occur in single-hand case may occur that have to be dealt with (occlusion of hands, distinction between or indexing of left/right hand). Second, the most versatile gesture analysis techniques (namely, 3D model-based techniques) currently exhibit one major drawback: speed. Some 3D model-based techniques which use coarse upper body and arm models [6] have reached the near real-time speeds and have been utilized for basic two-hand gesture analysis. Appearance-based techniques can, in principle, handle two-handed gestures. Nevertheless, their applicability has been usually restricted to simple (symbolic) gestures that do not require two hands. A more recent work on appearance-based motion templates [12] has indirectly addressed the issue of two-handed gestures. Another notable exception is an early system developed by Krueger [54]. Thus, to adequately address the issue of two-handed gestures in the future, more effective analysis techniques should be considered. These techniques should not only rely on the improvements of the classical techniques used in single-hand gestures, but also exploit the interdependence between the two hands performing a gesture since in many case the two hands performing a single gesture assume symmetrical postures.

An issue related to two-handed gestures is the one of multiple gesturers. Successful interaction in HCI-based environments has to consider multiple users. For example, a virtual modeling task can benefit enormously if several designers simultaneously participate in the process. However, the implementation of the multi-user interface has several difficult issues to face, the foremost one being the analysis of gestures. The analysis at the present assumes that there is a well-defined workspace associated with the gesturer. However, in the case of multiple users the intersection of workspaces is a very probable event. The differentiation between the users can then pose a serious problem. The use of active computer vision [11], [82], [5], [8], in which the cameras adaptively focus on some area of interest, may offer a solution to this problem. Another approach would be to optimize the parameters of the stationary camera(s) for a given interface; related issues are studied under *sensor planning* [39], [91].

Hand gestures are, like speech, body movement, and gaze, a means of communication (see Section 2.1). Moreover, almost any natural communication among humans concurrently involves several modes of communication that accompany each other. For instance, the "come here" gesture is usually accompanied by the words "Come here." Another example is the sentence "Notice *this* control panel." and a deictic gesture involving an index finger pointing at the particular control panel and a gaze directed at the panel. As seen from the above examples, the communicative gestures can be used both to *affirm* and to *complement* the meaning of a speech message. In fact, in the literature that reports psychological studies of human communication, the interaction between the speech and gestures as well as the other means of communication is often explored [48], [61], [87]. This leads to the conclusion that any such *multimodal* interaction can also be rendered useful for HCI (see Fig. 11 and [84]). The affirmative hand gesture (speech) can be used to reduce the uncertainty in speech (hand gesture) recognition and, thus, provide a more robust interface. Gestures that complement speech, on the other hand, carry a complete communicational message only if they are interpreted together with speech and, possibly, gaze. The use of such multimodal messages can help reduce the complexity and increase the naturalness of the interface for HCI (see Fig. 12). For example, instead of designing a complicated gestural command for the object selection which may consist of a deictic gesture followed by a symbolic gesture (to symbolize that the object that was pointed at by the hand is supposed to be selected) a simple concurrent deictic gesture and verbal command "this" can be used [84]. The number of studies that explore the use of multimodality in HCI has been steadily increasing over the past couple of years [36], [83], [98], [99], [102]. At the present time, the integration of communication modes in such systems is performed after the commands portions of different modes have been independently recognized. Although the interface structure is simplified in this way, the information pertaining to the interaction of the modes at lower levels is probably lost. To utilize the multimodal interaction at all levels, new approaches that fuse the multimodal input analysis as well as recognition should be considered in the future.

Fig. 11. A possible situation where speech/gesture integration may be particularly effective: A 3D visualization facility for structural biologist where researchers could be examining and discussing the results of a simulation.

Photograph courtesy of Rich Saal, Illinois State Journal-Register, Springfield, Ill.

7 Conclusions

Human-computer interaction is still in its infancy. Visual interpretation of hand gestures would allow the development of potentially natural interfaces to computer controlled environments. In response to this potential, the number of different approaches to video-based hand gesture recognition has grown tremendously in recent years. Thus there is a growing need for systematization and analysis of many aspects of gestural interaction. This paper surveys the different approaches to modeling, analysis, and recognition of hand gestures for visual interpretation. The discussion recognizes two classes of models employed in the visual interpretation of hand gestures. The first relies on 3D models of the human hand, while the second utilizes the appearance of the human hand in the image. The 3D hand models offer a rich description and discrimination capability that would allow a wide class of gestures to be recognized leading to natural HCI. However, the computation of 3D model parameters from visual images under real-time constraints remains an elusive goal. Appearance-based models are simpler to implement and use for real-time gesture recognition, but suffer from inherent limitations which could be a drawback for natural HCI.

Several simple HCI systems have been proposed that demonstrate the potential of vision-based gestural interfaces. However, from a practical standpoint, the development of such systems is in its infancy. Though most current systems employ hand gestures for the manipulation of objects, the complexity of the interpretation of gestures dictates the achievable solution. For example, the gestures used to convey manipulative actions today are usually of the communicative type. Further, hand gestures for HCI are mostly restricted to single-handed and produced only by a single user in the system. This consequently downgrades the effectiveness of the interaction. We suggest several directions of research for raising these limitations toward gestural HCI. For example, integration of hand gestures with speech, gaze and other naturally related modes of communication in a multimodal interface. However, substantial research effort that connects advances in computer vision with the basic study of human-computer interaction will be needed in the future to develop an effective and natural hand gesture interface.

Acknowledgments

This work was supported in part by the U. S. Army Research Laboratory under Cooperative Agreement No. DAAL01-96-2-0003 and in part by the National Science Foundation Grant IRI-96-34618. The authors would like to acknowledge Yusuf Azoz and Lalitha Devi for their help with the references, and Karin Pavlovic for her help in reviewing the manuscript. They would also like to thank the anonymous reviewers for their comments which greatly helped in improving this survey.

References

[1] J.F. Abramatic, P. Letellier, and M. Nadler, "A Narrow-Band Video Communication System for the Transmission of Sign Language Over Ordinary Telephone Lines," *Image Sequences Processing and Dynamic Scene Analysis*, T.S. Huang, ed., pp. 314-336. Berlin and Heidelberg: Springer-Verlag, 1983.

[2] J.A. Adam, "Virtual Reality," *IEEE Spectrum*, vol. 30, no. 10, pp. 22-29, 1993.

[3] S. Ahmad and V. Tresp, "Classification With Missing and Uncertain Inputs," *Proc. Int'l Conf. Neural Networks*, vol. 3, pp. 1,949-1,954, 1993.

[4] S. Ahmad, "A Usable Real-Time 3D Hand Tracker," *IEEE Asilomar Conf.*, 1994.

[5] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active Vision," *Int'l J. Computer Vision*, vol. 1, pp. 333-356, 1988.

[6] A. Azarbayejani, C. Wren, and A. Pentland, "Real-Time 3D Tracking of the Human Body," *Proc. IMAGE'COM 96*, Bordeaux, France, 1996.

[7] Y. Azoz, L. Devi, and R. Sharma, "Vision-Based Human Arm Tracking for Gesture Analysis Using Multimodal Constraint Fusion," *Proc. 1997 Advanced Display Federated Laboratory Symp.*, Adelphi, Md., Jan. 1997.

[8] R. Bajcsy, "Active Perception," *Proc. IEEE*, vol. 78, pp. 996-1,005, 1988.

[9] T. Baudel and M. Baudouin-Lafon, "Charade: Remote Control of Objects Using Free-Hand Gestures," *Comm. ACM*, vol. 36, no. 7, pp. 28-35, 1993.

[10] D.A. Becker and A. Pentland, "Using a Virtual Environment to Teach Cancer Patients T'ai Chi, Relaxation, and Self-Imagery," *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, Killington, Vt., Oct. 1996.

[11] A. Blake and A. Yuille, *Active Vision*. Cambridge, Mass.: MIT Press, 1992.

[12] A.F. Bobick and J.W. Davis, "Real-Time Recognition of Activity Using Temporal Templates," *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, Killington, Vt., Oct. 1996.

[13] H.A. Boulard and N. Morgan, *Connectionnist Speech Recognition. A Hybrid Approach*. Norwell, Mass.: Kluwer Academic Publishers, 1994.

[14] U. Bröckl-Fox, "Real-Time 3D Interaction With Up to 16 Degrees of Freedom From Monocular Image Flows," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, pp. 172-178, June 1995.

[15] L.W. Campbell, D.A. Becker, A. Azarbayejani, A.F. Bobick, and A. Pentland, "Invariant Features for 3D Gesture Recognition," *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, Killington, Vt., pp. 157-162, Oct. 1996.

[16] C. Cedras and M. Shah, "Motion-Based Recognition: A Survey," *Image and Vision Computing*, vol. 11, pp. 129-155, 1995.

[17] K. Cho and S.M. Dunn, "Learning Shape Classes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, pp. 882-888, Sept. 1994.

[18] R. Cipolla and N.J. Hollinghurst, "Human-Robot Interface by Pointing With Uncalibrated Stereo Vision," *Image and Vision Computing*, vol. 14, pp. 171-178, Mar. 1996.

[19] R. Cipolla, Y. Okamoto, and Y. Kuno, "Robust Structure From Motion Using Motion Parallax," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 374-382, 1993.

[20] E. Clergue, M. Goldberg, N. Madrane, and B. Merialdo, "Automatic Face and Gestural Recognition for Video Indexing," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, pp. 110-115, June 1995.

[21] T. F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham, "Active Shape Models—Their Training and Application," *Computer Vision and Image Understanding*, vol. 61, pp. 38-59, Jan. 1995.

[22] J.L. Crowley, F. Berard, and J. Coutaz, "Finger Tacking As an Input Device for Augmented Reality," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, pp. 195-200, June 1995.

[23] Y. Cui and J. Weng, "Learning-Based Hand Sign Recognition," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, pp. 201-206, June 1995.

[24] Y. Cui and J. J. Weng, "Hand Segmentation Using Learning-Based Prediction and Verification for Hand Sign Recognition," *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, Killington, Vt., pp. 88-93, Oct. 1996.

[25] T. Darrell, I. Essa, and A. Pentland, "Task-Specific Gesture Analysis in Real-Time Using Interpolated Views," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 12, pp. 1,236-1,242, Dec. 1996.

[26] T. Darrell and A.P. Pentland, "Attention-Driven Expression and Gesture Analysis in an Interactive Environment," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, pp. 135-140, June 1995.

[27] J. Davis and M. Shah, "Determining 3D Hand Motion," *Proc. 28th Asilomar Conf. Signals, Systems, and Computer*, 1994.

[28] J. Davis and M. Shah, "Recognizing Hand Gestures," *Proc. European Conf. Computer Vision*, Stockholm, Sweden, pp. 331-340, 1994.

[29] A. C. Downton and H. Drouet, "Image Analysis for Model-Based Sign Language Coding," *Progress in Image Analysis and Processing II: Proc. Sixth Int'l Conf. Image Analysis and Processing*, pp. 637-644, 1991.

[30] I. Essa and S. Pentland, "Facial Expression Recognition Using a Dynamic Model and Motion Energy," *Proc. IEEE Int'l Conf. Computer Vision*, 1995.

[31] M. Etoh, A. Tomono, and F. Kishino, "Stereo-Based Description by Generalized Cylinder Complexes From Occluding Contours," *Systems and Computers in Japan*, vol. 22, no. 12, pp. 79-89, 1991.

[32] S.S. Fels and G.E. Hinton, "Glove-Talk: A Neural Network Interface Between a Data-Glove and a Speech Synthesizer," *IEEE Trans. Neural Networks*, vol. 4, pp. 2-8, Jan. 1993.

[33] W.T. Freeman, K. Tanaka, J. Ohta, and K. Kyuma, "Computer Vision for Computer Games," *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, Killington, Vt., pp. 100-105, Oct. 1996.

[34] W.T. Freeman and M. Roth, "Orientation Histograms for Hand Gesture Recognition," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, June 1995.

[35] W.T. Freeman and C.D. Weissman, "Television Control by Hand Gestures," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, pp. 179-183, June 1995.

[36] M. Fukumoto, Y. Suenaga, and K. Mase, "Finger-Pointer": Pointing Interface by Image Processing," *Computers and Graphics*, vol. 18, no. 5, pp. 633-642, 1994.

[37] D.M. Gavrila and L.S. Davis, "Towards 3D Model-Based Tracking and Recognition of Human Movement: A Multi-View Approach," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, pp. 272-277, June 1995.

[38] H.P. Graf, E. Cosatto, D. Gibbon, M. Kocheisen, and E. Petajan, "Multi-Modal System for Locating Heads and Faces," *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, Killington, Vt., pp. 88-93, Oct. 1996.

[39] G. D. Hager, *Task Directed Sensor Fusion and Planning*. Kluwer Academic Publishers, 1990.

[40] H. Harashima and F. Kishino, "Intelligent Image Coding and Communications With Realistic Sensations—Recent Trends," *IEICE Trans.*, vol. E 74, pp. 1,582-1,592, June 1991.

[41] A.G. Hauptmann and P. McAvinney, "Gesture With Speech for Graphics Manipulation," *Int'l J. Man-Machine Studies*, vol. 38, pp. 231-249, Feb. 1993.

[42] T. Heap and D. Hogg, "Towards 3D Hand Tracking Using a Deformable Model," *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, Killington, Vt., pp. 140-145, Oct. 1996.

[43] E. Hunter, J. Schlenzig, and R. Jain, "Posture Estimation in Reduced-Model Gesture Input Systems," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, June 1995.

[44] K. Ishibuchi, H. Takemura, and F. Kishino, "Real Time Hand Gesture Recognition Using 3D Prediction Model," *Proc. 1993 Int'l Conf. Systems, Man, and Cybernetics*, Le Touquet, France, pp. 324-328, Oct. 17-20, 1993.

[45] S.X. Ju, M.J. Black, and Y.Y. oob, "Cardboard People: A Parameterized Model of Articulated Image Motion," *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, Killington, Vt., pp. 38-43, Oct. 1996.

[46] I.A. Kakadiaris, D. Metaxas, and R. Bajcsy, "Active Part-Decomposition, Shape and Motion Estimation of Articulated Objects: A Physics-Based Approach," *Proc. IEEE C.S. Conf. Computer Vision and Pattern Recognition*, pp. 980-984, 1994.

[47] S.B. Kang and K. Ikeuchi, "Toward Automatic Robot Instruction for Perception—Recognizing a Grasp From Observation," *IEEE Trans. Robotics and Automation*, vol. 9, pp. 432-443, Aug. 1993.

[48] A. Kendon, "Current Issues in the Study of Gesture," *The Biological Foundations of Gestures: Motor and Semiotic Aspects*, J.-L. Nespoulous, P. Peron, and A. R. Lecours, eds., pp. 23-47. Lawrence Erlbaum Assoc., 1986.

[49] C. Kervrann and F. Heitz, "Learning Structure and Deformation Modes of Nonrigid Objects in Long Image Sequences," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, June 1995.

[50] R. Kjeldsen and J. Kender, "Visual Hand Gesture Recognition for Window System Control," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, pp. 184-188, June 1995.

[51] R. Kjeldsen and J. Kender, "Finding Skin in Color Images," *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, Killington, Vt., pp. 312-317, Oct. 1996.

[52] R. Koch, "Dynamic 3D Scene Analysis Through Synthetic Feedback Control," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp. 556-568, 1993.

[53] M.W. Krueger, *Artificial Reality II*. Addison-Wesley, 1991.

[54] M.W. Krueger, "Environmental Technology: Making the Real World Virtual," *Comm. ACM*, vol. 36, pp. 36-37, July 1993.

[55] J.J. Kuch, "Vision-Based Hand Modeling and Gesture Recognition for Human Computer Interaction," master's thesis, Univ. of Illinois at Urbana-Champaign, 1994.

[56] J.J. Kuch and T.S. Huang, "Vision-Based Hand Modeling and Tracking," *Proc. IEEE Int'l Conf. Computer Vision*, Cambridge, Mass., June 1995.

[57] Y. Kuno, M. Sakamoto, K. Sakata, and Y. Shirai, "Vision-Based Human Computer Interface With User Centered Frame," *Proc. IROS'94*, 1994.

[58] A. Lanitis, C.J. Taylor, T.F. Cootes, and T. Ahmed, "Automatic Interpretation of Human Faces and Hand Gestures Using Flexible Models," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, pp. 98-103, June 1995.

[59] J. Lee and T.L. Kunii, "Constraint-Based Hand Animation," *Models and Techniques in Computer Animation*, pp. 110-127. Tokyo: Springer-Verlag, 1993.

[60] J. Lee and T.L. Kunii, "Model-Based Analysis of Hand Posture," *IEEE Computer Graphics and Applications*, pp. 77-86, Sept. 1995.

[61] E.T. Levy and D. McNeill, "Speech, Gesture, and Discourse," *Discourse Processes*, no. 15, pp. 277-301, 1992.

[62] C. Maggioni, "A Novel Gestural Input Device for Virtual Reality," *1993 IEEE Annual Virtual Reality Int'l Symp.*, pp. 118-124, IEEE, 1993.

[63] C. Maggioni, "GestureComputer—New Ways of Operating a Computer," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, pp. 166-171, June 1995.

[64] N. Magnenat-Thalmann and D. Thalman, *Computer Animation: Theory and Practice*. New York: Springer-Verlag, 2nd rev. ed., 1990.

[65] D. McNeill and E. Levy, "Conceptual Representations in Language Activity and Gesture," *Speech, Place and Action: Studies in Deixis and Related Topics*, J. Jarvella and W. Klein, eds. Wiley, 1982.

[66] A. Meyering and H. Ritter, "Learning to Recognize 3D-Hand Postures From Perspective Pixel Images," *Artificial Neural Networks 2*, I. Alexander and J. Taylor, eds. North-Holland: Elsevier Science Publishers B.V., 1992.

[67] B. Moghaddam and A. Pentland, "Maximum Likelihood Detection of Faces and Hands," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, pp. 122-128, June 1995.

[68] O'Rourke and N.L. Badler, "Model-Based Image Analysis of Human Motion Using Constraint Propagation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 2, pp. 522-536, 1980.

[69] V.I. Pavlovic, R. Sharma, and T.S. Huang, "Gestural Interface to a Visual Computing Environment for Molecular Biologists," *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, Killington, Vt., pp. 30-35, Oct. 1996.

[70] D.L. Quam, "Gesture Recognition With a DataGlove," *Proc. 1990 IEEE National Aerospace and Electronics Conf.*, vol. 2, 1990.

[71] F.K.H. Quek, "Toward a Vision-Based Hand Gesture Interface," *Virtual Reality Software and Technology Conf.*, pp. 17-31, Aug. 1994.
[72] F.K.H. Quek, "Eyes in the Interface," *Image and Vision Computing*, vol. 13, Aug. 1995.
[73] F.K.H. Quek, T. Mysliwiec, and M. Zhao, "Finger Mouse: A Free-hand Pointing Interface," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, pp. 372-377, June 1995.
[74] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE*, vol. 77, pp. 257-286, Feb. 1989.
[75] L.R. Rabiner and B. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, N.J.: Prentice Hall, 1993.
[76] J.M. Rehg and T. Kanade, "DigitEyes: Vision-Based Human Hand Tracking," Technical Report CMU-CS-93-220, School of Computer Science, Carnegie Mellon Univ., 1993.
[77] J.M. Rehg and T. Kanade, "Model-Based Tracking of Self-Occluding Articulated Objects," *Proc. IEEE Int'l Conf. Computer Vision*, Cambridge, Mass., pp. 612-617, June 20-23 1995.
[78] H. Rheingold, *Virtual Reality*. Summit Books, 1991.
[79] J. Schlenzig, E. Hunter, and R. Jain, "Vision-Based Hand Gesture Interpretation Using Recursive Estimation," *Proc. 28th Asilomar Conf. Signals, Systems, and Computer*, 1994.
[80] J. Schlenzig, E. Hunter, and R. Jain, "Recursive Identification of Gesture Inputs Using Hidden Markov Models," *Proc. Second IEEE Workshop on Applications of Computer Vision*, Sarasota, Fla., pp. 187-194, Dec. 5-7, 1994.
[81] J. Segen, "Controlling Computers With Gloveless Gestures," *Proc. Virtual Reality Systems*, Apr. 1993.
[82] R. Sharma, "Active Vision for Visual Servoing: A Review," *IEEE Workshop on Visual Servoing: Achievements, Applications and Open Problems*, May 1994.
[83] R. Sharma, T.S. Huang, and V.I. Pavlovic, "A Multimodal Framework for Interacting With Virtual Environments," *Human Interaction With Complex Systems*, C.A. Ntuen and E.H. Park, eds., pp. 53-71. Kluwer Academic Publishers, 1996.
[84] R. Sharma, T.S. Huang, V.I. Pavlovic, Y. Zhao, Z. Lo, S. Chu, K. Schulten, A. Dalke, J. Phillips, M. Zeller, and W. Humphrey, "Speech/Gesture Interface to a Visual Computing Environment for Molecular Biologists," *Proc. Int'l Conf. Pattern Recognition*, 1996.
[85] K. Shirai and S. Furui, "Special Issue on Spoken Dialogue," *Speech Communication*, vol. 15, pp. 3-4, 1994.
[86] T.E. Starner and A. Pentland, "Visual Recognition of American Sign Language Using Hidden Markov Models," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, pp. 189-194, June 1995.
[87] J. Streeck, "Gesture as Communication I: Its Coordination With Gaze and Speech," *Communication Monographs*, vol. 60, pp. 275-299, Dec. 1993.
[88] D.J. Sturman and D. Zeltzer, "A Survey of Glove-Based Input," *IEEE Computer Graphics and Applications*, vol. 14, pp. 30-39, Jan. 1994.
[89] D.X. Sun and L. Deng, "Nonstationary Hidden Markov Models for Speech Recognition," *Image Models (and Their Speech Model Cousins)*, S.E. Levinson and L. Shepp, eds., pp. 161-182. New York: Springer-Verlag, 1996.
[90] D.L. Swets and J. Weng, "Using Discriminant Eigenfeatures for Image Retrieval," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 831-836, 1996.
[91] K.A. Tarabanis, P.K. Allen, and R.Y. Tsai, "A Survey of Sensor Planning in Computer Vision," *IEEE Trans. Robotics and Automation*, vol. 11, pp. 86-104, 1995.
[92] D. Thompson, "Biomechanics of the Hand," *Perspectives in Computing*, vol. 1, pp. 12-19, Oct. 1981.
[93] Y.A. Tijerino, K. Mochizuki, and F. Kishino, "Interactive 3D Computer Graphics Driven Through Verbal Instructions: Previous and Current Activities at ATR," *Computers and Graphics*, vol. 18, no. 5, pp. 621-631, 1994.
[94] A. Torige and T. Kono, "Human-Interface by Recognition of Human Gestures With Image Processing. Recognition of Gesture to Specify Moving Directions," *IEEE Int'l Workshop on Robot and Human Communication*, pp. 105-110, 1992.
[95] R. Tubiana, ed., *The Hand*, vol. 1. Philadelphia, Penn.: Sandgers, 1981.
[96] C. Uras and A. Verri, "Hand Gesture Recognition From Edge Maps," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, pp. 116-121, June 1995.
[97] R. Vaillant and D. Darmon, "Vision-Based Hand Pose Estimation," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, pp. 356-361, June 1995.
[98] M.T. Vo, R. Houghton, J. Yang, U. Bub, U. Meier, A. Waibel, and P. Duchnowski, "Multimodal Learning Interfaces," *ARPA Spoken Language Technology Workshop 1995*, Jan. 1995.
[99] M.T. Vo and A. Waibel, "A Multi-Modal Human-Computer Interface: Combination of Gesture and Speech Recognition," *Adjunct Proc. InterCHI'93*, Apr. 26-29 1993.
[100] A. Waibel and K.F. Lee, *Readings in Speech Recognition*. Morgan Kaufmann, 1990.
[101] C. Wang and D.J. Cannon, "A Virtual End-Effector Pointing System in Point-and-Direct Robotics for Inspection of Surface Flaws Using a Neural Network-Based Skeleton Transform," *Proc. IEEE Int'l Conf. Robotics and Automation*, vol. 3, pp. 784-789, May 1993.
[102] K. Watanuki, K. Sakamoto, and F. Togawa, "Multimodal Interaction in Human Communication," *IEICE Trans. Information and Systems*, vol. E78-D, pp. 609-614, June 1995.
[103] A.D. Wilson and A.F. Bobick, "Configuration States for the Representation and Recognition of Gesture," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, pp. 129-134, June 1995.
[104] A.D. Wilson and A.F. Bobick, "Recovering the Temporal Structure of Natural Gestures," *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, Killington, Vt., pp. 66-71, Oct. 1996.
[105] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, Killington, Vt., pp. 51-56, Oct. 1996.

Vladimir I. Pavlovic received the Dipl Eng degree in electrical engineering from the University of Novi Sad, Yugoslavia, in 1991. In 1993, he received the MS degree in electrical engineering and computer science from the University of Illinois at Chicago. He is currently a doctoral student in electrical engineering at the Beckman Institute and the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign. His research interests include vision-based computer interaction, multimodal signal fusion, and image coding.

Rajeev Sharma is an assistant professor in the Department of Computer Science and Engineering at the Pennsylvania State University, University Park. After receiving a PhD in computer science from the University of Maryland, College Park, in 1993, he spent three years at the University of Illinois, Urbana-Champaign as a Beckman Fellow and adjunct assistant professor in the Department of Electrical and Computer Engineering.

He is a recipient of the Association of Computing Machinery Samuel Alexander Doctoral Dissertation Award and the IBM pre-doctoral fellowship.

His research interests lie in studying the role of computer vision in robotics and advanced human-computer interfaces.

T.S. Huang received his B.S. Degree in electrical engineering from National Taiwan University, Taipei, Taiwan, China; and his MS and ScD degrees in Electrical Engineering from the Massachusetts Institute of Technology, Cambridge, Massachusetts. He was on the faculty of the Department of Electrical Engineering at MIT from 1963 to 1973; and on the faculty of the School of Electrical Engineering and director of its Laboratory for Information and Signal Processing at Purdue University from 1973 to 1980. In 1980, he joined the University of Illinois at Urbana-Champaign, where he is now William L. Everitt Distinguished Professor of Electrical and Computer Engineering, Research Professor at the Coordinated Science Laboratory, and head of the Image Formation and Processing Group at the Beckman Institute for Advanced Science and Technology.

During his sabbatical leaves Dr. Huang has worked at the MIT Lincoln Laboratory, the IBM Thomas J. Watson Research Center, and the Rheinishes Landes Museum in Bonn, West Germany, and held visiting professor positions at the Swiss Institutes of Technology in Zurich and Lausanne, University of Hannover in West Germany, INRS-Telecommunications of the University of Quebec in Montreal, Canada, and University of Tokyo, Japan. He has served as a consultant to numerous industrial firms and government agencies both in the US and abroad.

Dr. Huang's professional interests lie in the broad area of information technology, especially the transmission and processing of multidimensional signals. He has published 11 books, and over 300 papers in network theory, digital filtering, image processing, and computer vision. He is a Fellow of the International Association of Pattern Recognition, IEEE and the Optical Society of America and has received a Guggenheim Fellowship, an A.V. Humboldt Foundation Senior U.S. Scientist Award, and a Fellowship from the Japan Association for the Promotion of Science. He received the IEEE Acoustics, Speech, and Signal Processing Society's Technical Achievement Award in 1987 and the Society Award in 1991. He is a Founding Editor of the *International Journal Computer Vision, Graphics, and Image Processing*, and Editor of the *Springer Series in Information Sciences*, published by Springer Verlag.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

PCT

(43) International Publication Date
8 September 2006 (08.09.2006)

(10) International Publication Number
WO 2006/094308 A3

(51) International Patent Classification:
G06F 1/16 (2006.01) *G06F 3/033* (2006.01)

(21) International Application Number:
PCT/US2006/008349

(22) International Filing Date: 3 March 2006 (03.03.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/658,777 4 March 2005 (04.03.2005) US
60/663,345 16 March 2005 (16.03.2005) US

(71) Applicant *(for all designated States except US)*: **APPLE COMPUTER, INC.** [US/US]; 1 Infinite Loop, Cupertino, CA 95014 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **HOTELLING, Steve, P.** [US/US]; 1351 Hidden Mine Road, San Jose, CA 95210 (US).

(74) Agent: **ALLEN, Billy, C., III.**; Wong, Cabello, Lutsch, Rutherford &, Brucculeri LLP, 20333 Sh 249, Suite 600, Houston, TX 77070 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

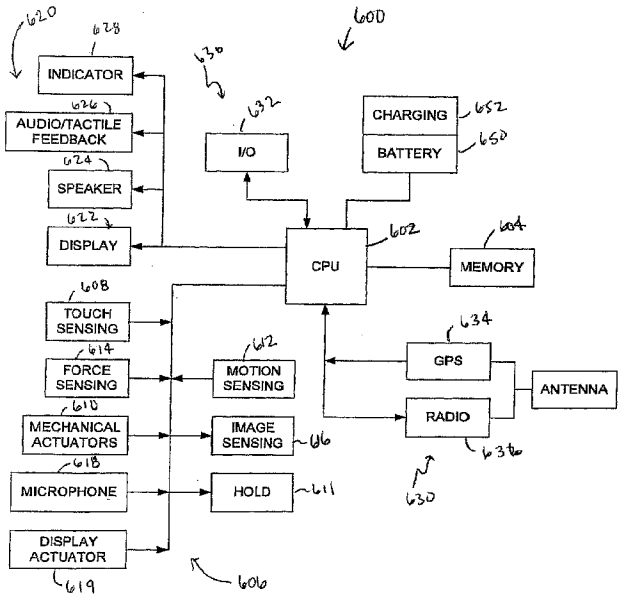
(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

(88) Date of publication of the international search report:
14 December 2006

[Continued on next page]

(54) Title: MULTI-FUNCTIONAL HAND-HELD DEVICE



(57) Abstract: Disclosed herein is a multi-functional hand-held device capable of configuring user inputs based on how the device is to be used. Preferably, the multifunctional hand-held device has at most only a few physical buttons, keys, or switches so that its display size can be substantially increased. The multifunctional hand-held device also incorporates a variety of input mechanisms, including touch sensitive screens, touch sensitive housings, display actuators, audio input, etc. The device also incorporates a user-configurable GUI for each of the multiple functions of the devices.

WO 2006/094308 A3

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2006/008349

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F1/16 G06F3/033

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2003/234768 A1 (REKIMOTO JUNICHI ET AL) 25 December 2003 (2003-12-25) paragraph [0031] - paragraph [0068]	1-12,18, 19,21,22
Y	figures	13-17
A		20,23
X	US 2003/206202 A1 (MORIYA TAKASHIRO) 6 November 2003 (2003-11-06)	20,23
Y	paragraph [0012]	15-17
A	paragraph [0022] - paragraph [0039]; figures 1-6	1,21,22
X	US 2004/263484 A1 (MANTYSALO TAPIO ET AL) 30 December 2004 (2004-12-30) paragraph [0004] - paragraph [0013] paragraph [0051] - paragraph [0068]; figures	1-12,18, 19,21,22

-/--

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
26 September 2006	06/10/2006

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Semple, Mark

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2006/008349

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2003/095095 A1 (PIHLAJA PEKKA JUHANA) 22 May 2003 (2003-05-22)	13,14
A	paragraph [0007] - paragraph [0008] paragraph [0020] - paragraph [0034]; figures 1,2,4,5	1,3,4,18
A	US 2003/006974 A1 (CLOUGH JAMES ET AL) 9 January 2003 (2003-01-09) paragraph [0004] - paragraph [0013] paragraph [0025] - paragraph [0053]; figures	1,4,7,8, 11,12, 20-23

Form PCT/ISA/210 (continuation of second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2006/008349

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2003234768 A1	25-12-2003	JP 2003330611 A	21-11-2003
US 2003206202 A1	06-11-2003	CN 1455615 A	12-11-2003
		GB 2391060 A	28-01-2004
		JP 2003323259 A	14-11-2003
US 2004263484 A1	30-12-2004	CN 1813462 A	02-08-2006
		EP 1642445 A1	05-04-2006
		WO 2004114636 A1	29-12-2004
US 2003095095 A1	22-05-2003	US 2006017711 A1	26-01-2006
US 2003006974 A1	09-01-2003	NONE	

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/01454

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :G09G 5/00
US CL :345/173

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 345/173

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,543,591 A (GILLESPIE ET AL) 06 August 1996, see abstract, column 5, lines 50-67, column 6, lines 1-5, 43-50, see figure 1.	17, 22-24
Y	US 5,305,017 A (GERPHEIDE et al) 19 April 1994, column 5, lines 12-50, see figure 1.	17, 22-24
A	US 5,563,632 A (ROBERTS) 08 October 1996, see abstract, column 3, lines 18-41, see figure 1.	1-16, 18-21
A,E	US 5,880,411 A (GILLESPIE et al) 09 March 1999, column 5, lines 29-67, column 6, lines 1-9, column 10, lines 18-37, see figure 1.	25-121
A	US 5,376,948 A (ROBERTS) 27 December 1994, see abstract and figure 1.	1-16, 18-21

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search
29 MARCH 1999

Date of mailing of the international search report
14 MAY 1999

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703) 305-3230

Authorized officer
RONALD LANEAU James R. Matthews
Telephone No. (703) 305-3973

Form PCT/ISA/210 (second sheet)(July 1992)*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/01454

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A,P	US 5,821,930 A (HANSEN) 13 October 1998, see abstract.	1-16, 18-21, 25-121

(12) UK Patent Application (19) GB (11) 2 330 670 (13) A

(43) Date of A Publication 28.04.1999

(21) Application No 9722541.1

(22) Date of Filing 24.10.1997

(51) INT CL[6]
G06F 3/033 , H04H 7/00

(52) UK CL (Edition Q)
G4A AKS
H4R RSX
U1S S1939

(56) Documents Cited
EP 0689122 A1 US 5523774 A

(58) Field of Search
UK CL (Edition P) G4A AKS , H4R RSX
INT CL[6] G06F 3/02 3/023 3/033 , H04H 7/00 , H04S 7/00

(71) Applicant(s)
Sony United Kingdom Limited
(Incorporated in the United Kingdom)
The Heights, Brooklands, WEYBRIDGE, Surrey, KT13 0XW, United Kingdom

(72) Inventor(s)
Peter Charles Eastty
Peter Damien Thorpe
Christopher Sleight

(74) Agent and/or Address for Service
D Young & Co
21 New Fetter Lane, LONDON, EC4A 1DA, United Kingdom

(54) Abstract Title
User interface for data processing apparatus

(57) Data processing apparatus, in particular an audio mixing console, has a fader panel (30, fig. 1) comprising an array of touch-sensitive controls 350, adjustable by movement of the user's hand while touching a control. Each control corresponds to a channel strip (300, fig. 4) on a display screen (10, fig. 1), which shows the processing controls and devices for that channel, and the current control settings (fig. 6). When a touch-sensor is touched, and so open to adjustment, the fader display on the corresponding channel strip is coloured in a contrasting colour, 400. Proximity sensors also enable contrasting colours to be displayed, 410, as the user's hand approaches one or more of the controls. Thus the user can track his hands across the fader panel (30, fig 1) from the display screen (10, fig. 1) rather than having to look at the fader panel itself.

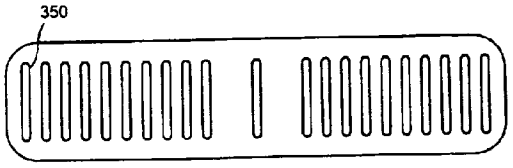


FIG. 5

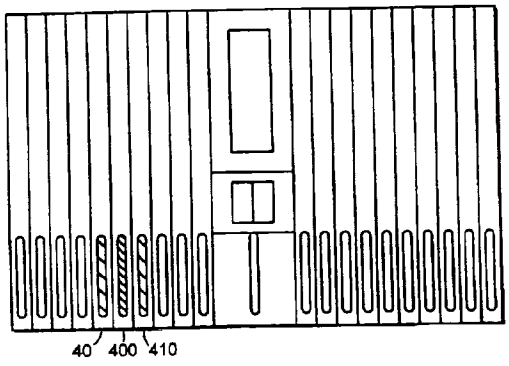


FIG. 7

GB 2 330 670 A

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

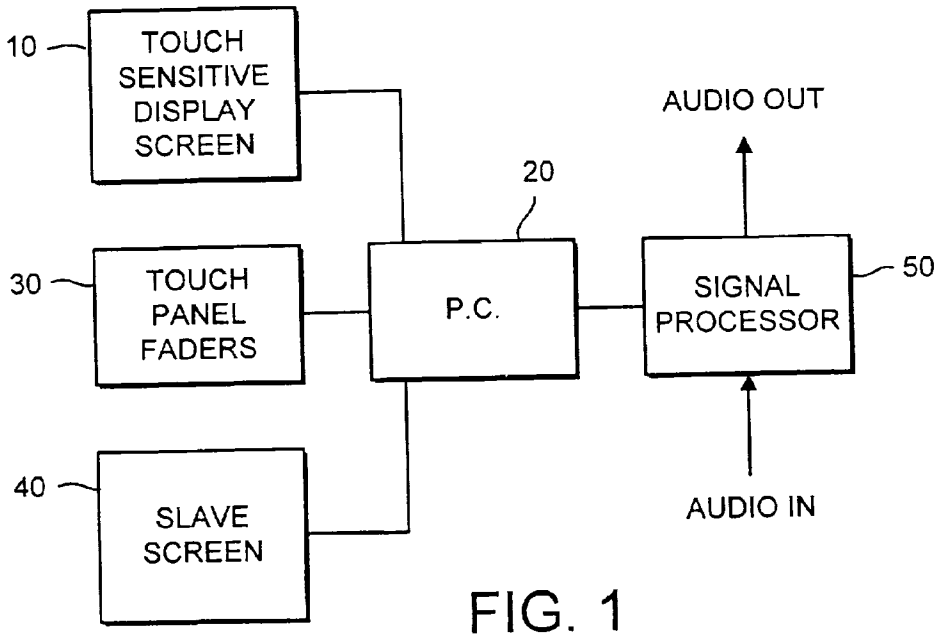

10
TOUCH SENSITIVE DISPLAY SCREEN
AUDIO OUT
20
30
TOUCH PANEL FADERS
P.C.
SIGNAL PROCESSOR
50
40
SLAVE SCREEN
AUDIO IN

FIG. 1

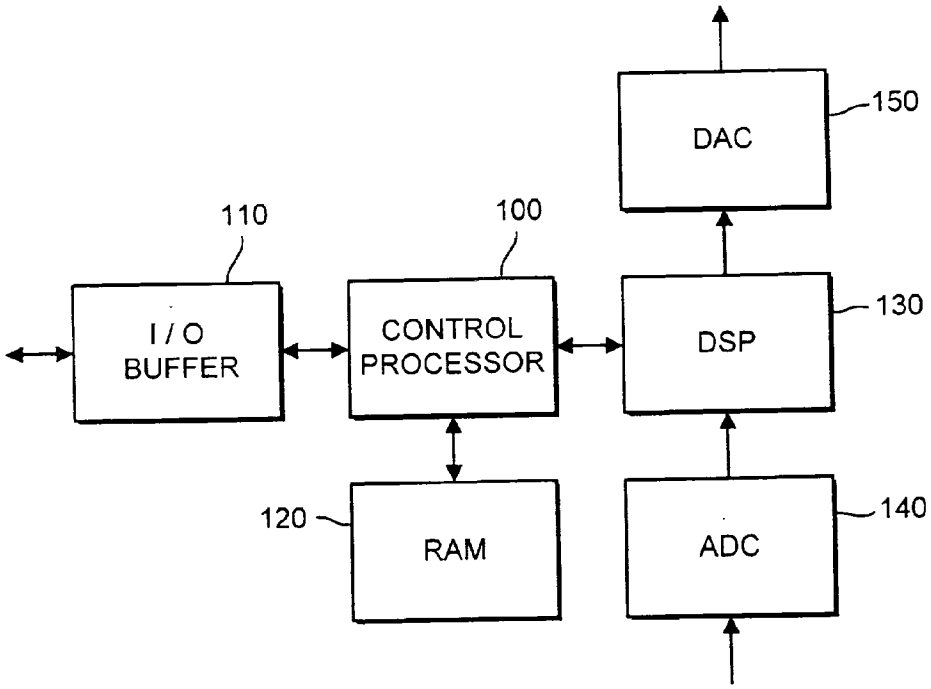

DAC
150
110
100
I / O BUFFER
CONTROL PROCESSOR
DSP
130
120
RAM
ADC
140

FIG. 2

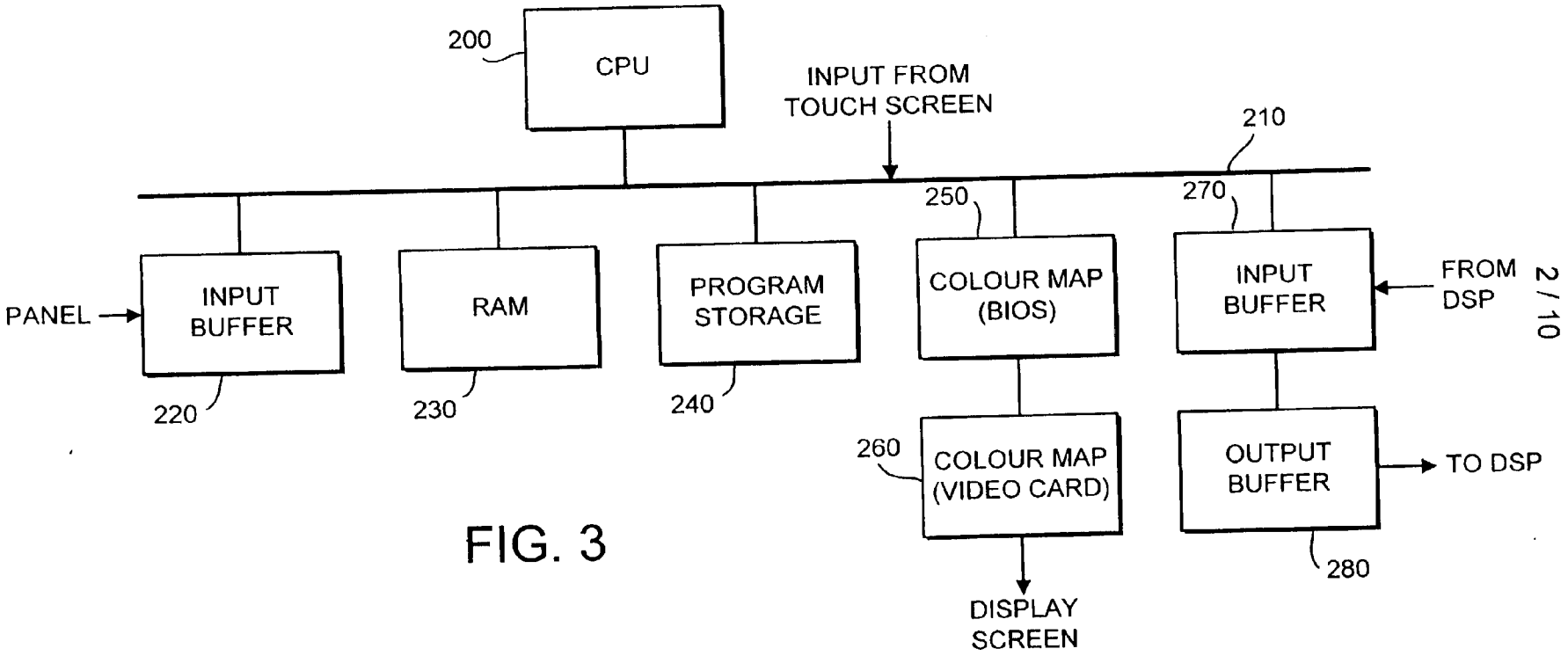

200
CPU
INPUT FROM
TOUCH SCREEN
210
250
270
PANEL
INPUT
BUFFER
220
RAM
230
PROGRAM
STORAGE
240
COLOUR MAP
(BIOS)
INPUT
BUFFER
FROM
DSP
260
COLOUR MAP
(VIDEO CARD)
OUTPUT
BUFFER
TO DSP
280
DISPLAY
SCREEN

FIG. 3

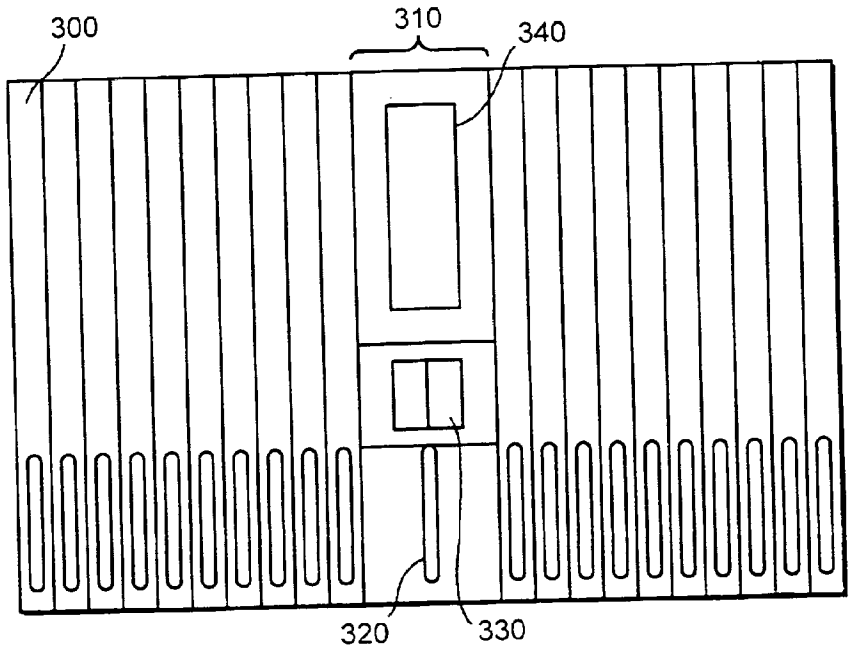


FIG. 4

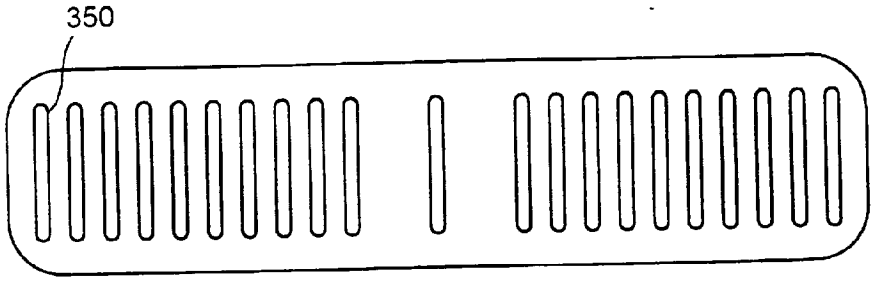


FIG. 5

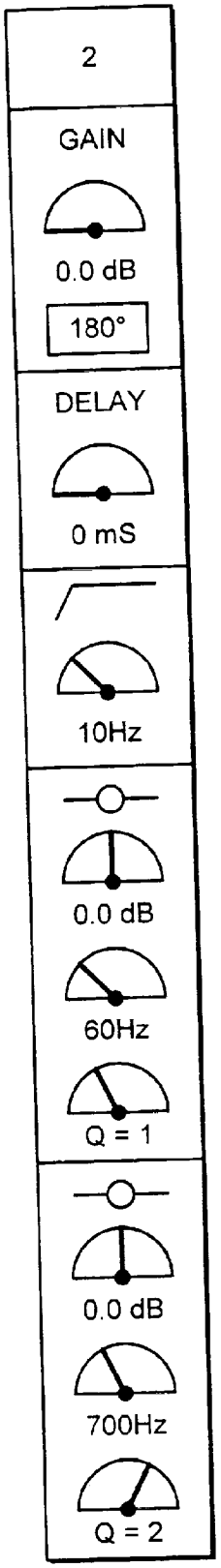


FIG. 6A

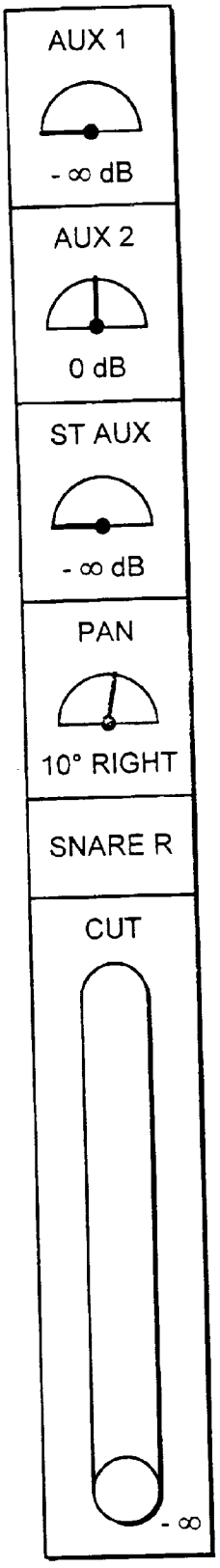


FIG. 6B

FIG. 6A

FIG. 6B

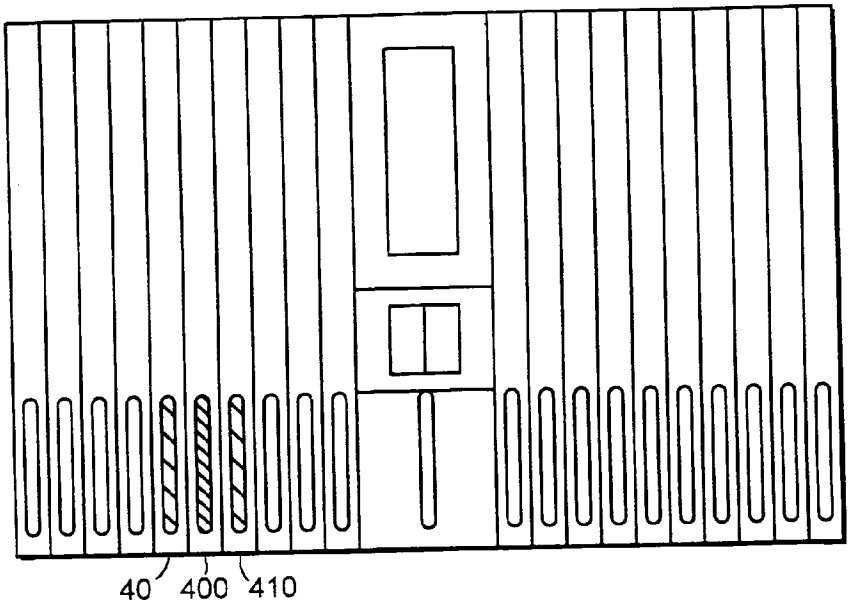


FIG. 7

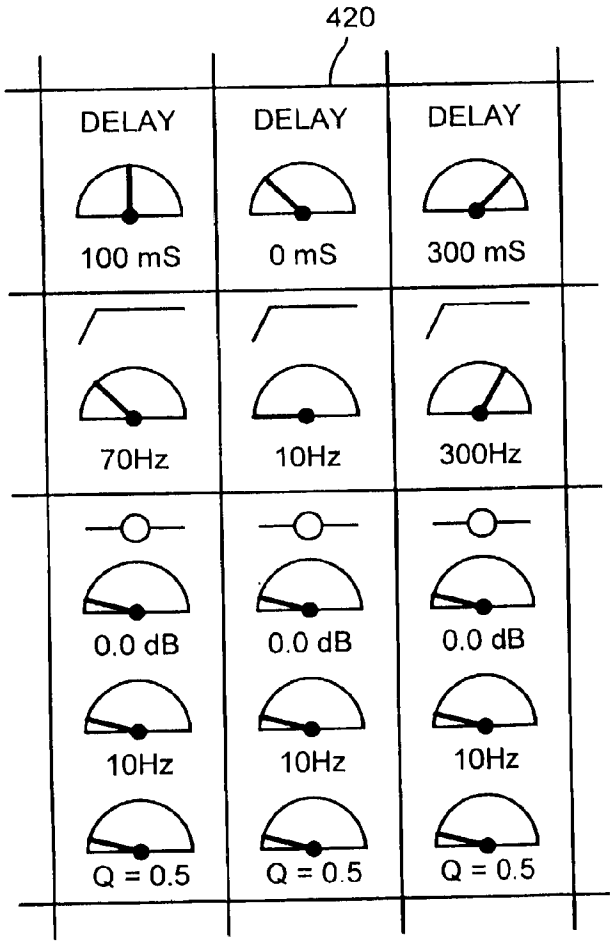

420
DELAY
DELAY
DELAY
100 mS
0 mS
300 mS
70Hz
10Hz
300Hz
0.0 dB
0.0 dB
0.0 dB
10Hz
10Hz
10Hz
Q = 0.5
Q = 0.5
Q = 0.5

FIG. 8A

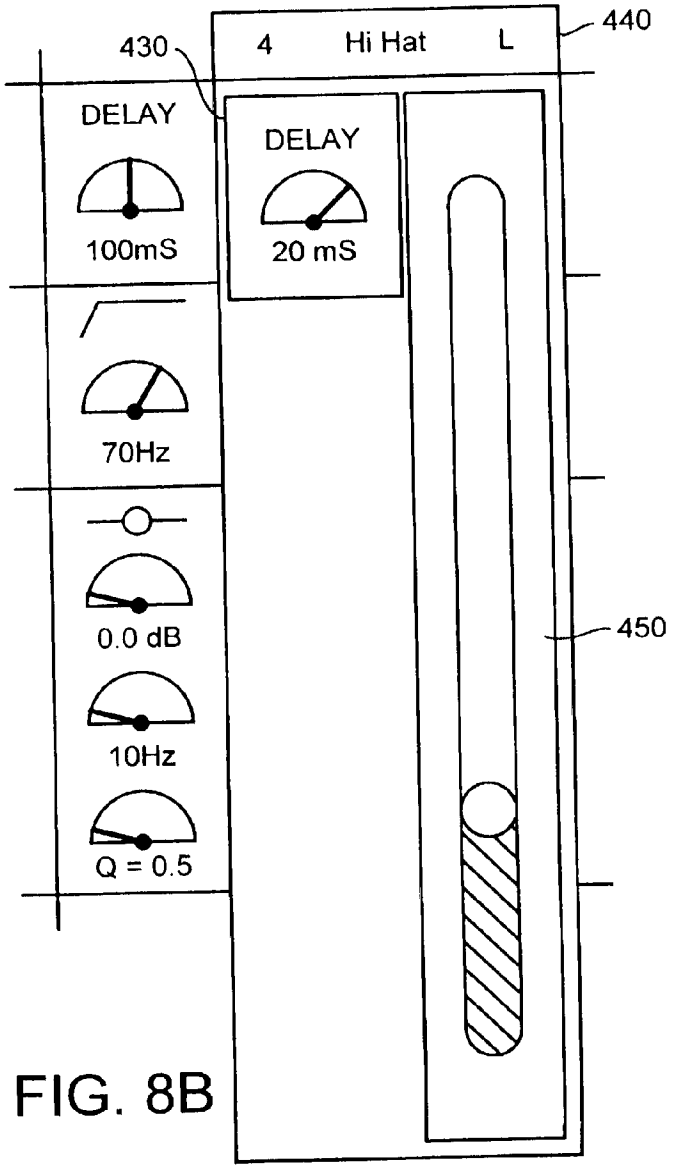

430
4
Hi Hat
L
440
DELAY
DELAY
100mS
20 mS
70Hz
0.0 dB
10Hz
Q = 0.5
450

FIG. 8B

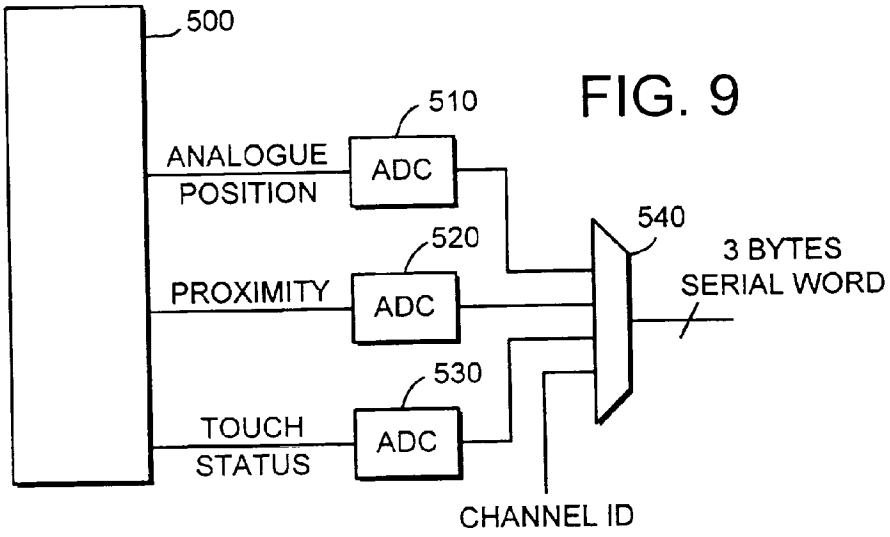


FIG. 9

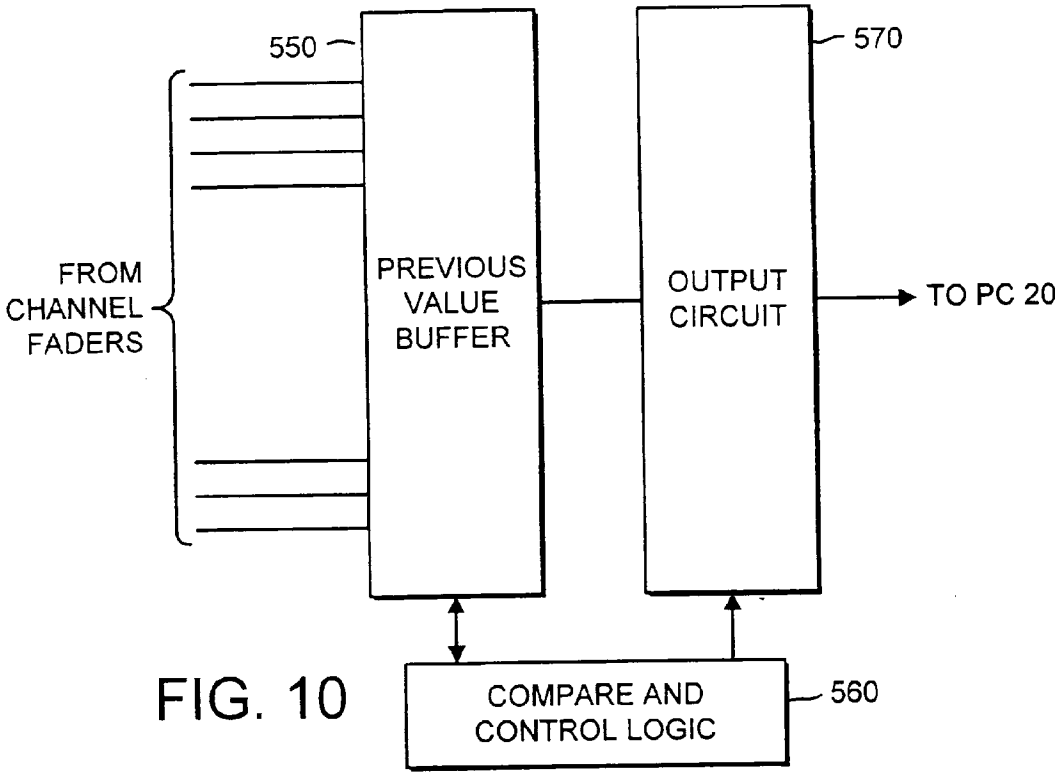


FIG. 10

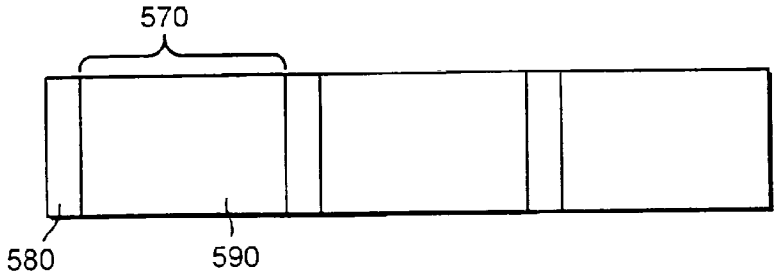


FIG. 11

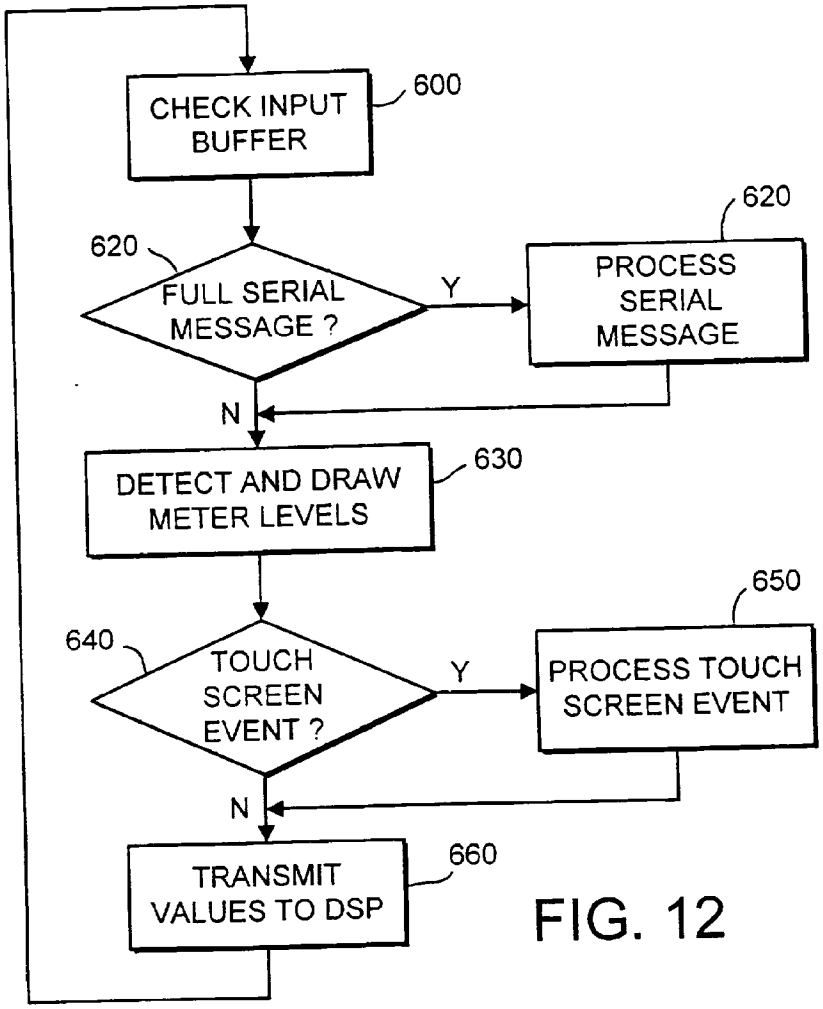


FIG. 12

	R	G	B
0	17	223	56
1	60	60	60
...			
255	255	200	0

FIG. 14

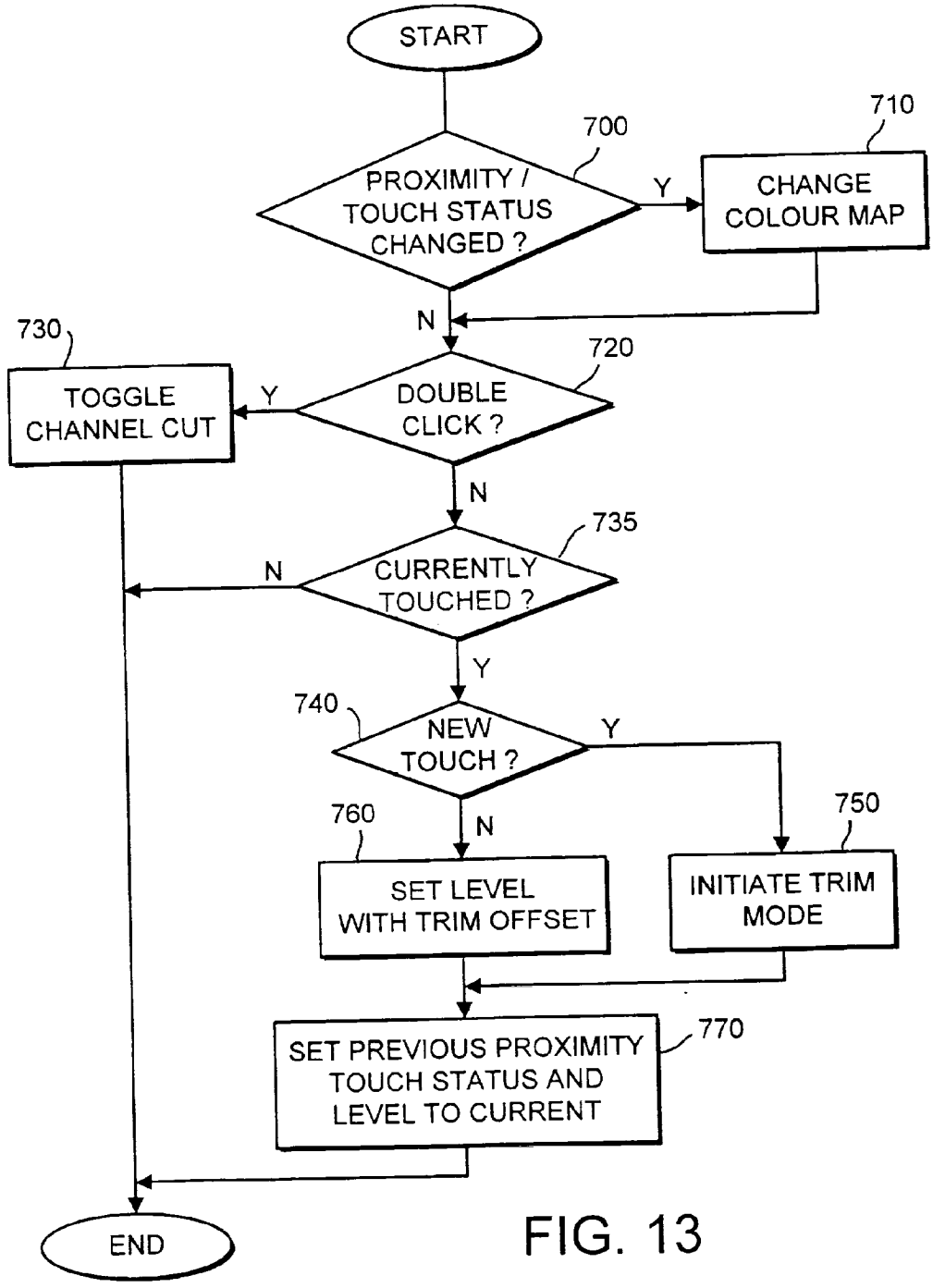

START
700
PROXIMITY / TOUCH STATUS CHANGED ?
Y
710
CHANGE COLOUR MAP
N
730
TOGGLE CHANNEL CUT
Y
720
DOUBLE CLICK ?
N
735
CURRENTLY TOUCHED ?
N
Y
740
NEW TOUCH ?
Y
760
N
750
SET LEVEL WITH TRIM OFFSET
INITIATE TRIM MODE
770
SET PREVIOUS PROXIMITY TOUCH STATUS AND LEVEL TO CURRENT
END

FIG. 13

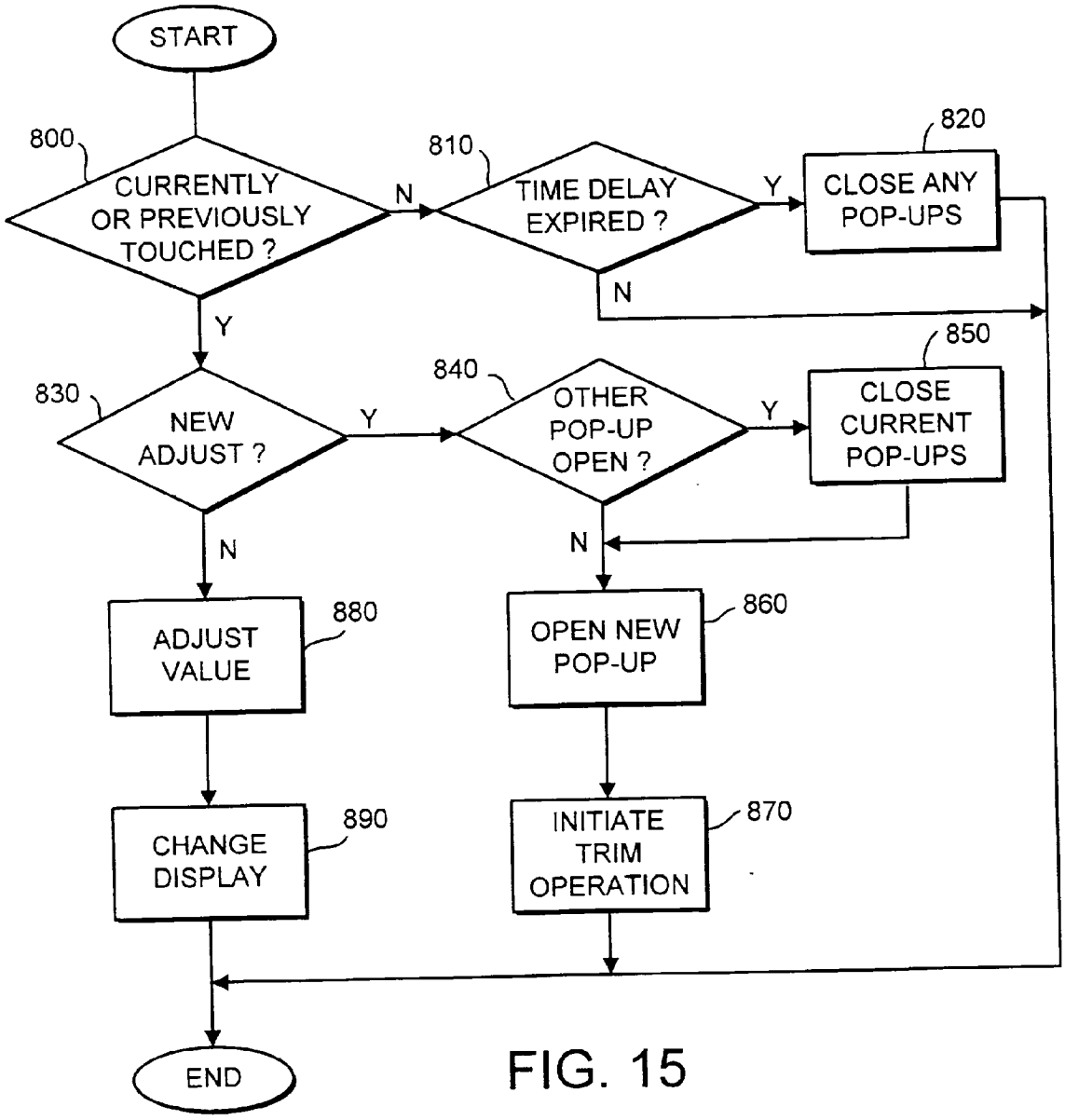

START
800
CURRENTLY OR PREVIOUSLY TOUCHED ?
N
810
TIME DELAY EXPIRED ?
Y
820
CLOSE ANY POP-UPS
N
Y
830
NEW ADJUST ?
Y
840
OTHER POP-UP OPEN ?
Y
850
CLOSE CURRENT POP-UPS
N
N
880
ADJUST VALUE
860
OPEN NEW POP-UP
890
CHANGE DISPLAY
870
INITIATE TRIM OPERATION
END

FIG. 15

DATA PROCESSING

This invention relates to data processing.

Data processing apparatus including display screens (e.g. PC computers) are generally controlled by external control devices such as keyboards, mice etc.

If the user wishes to concentrate on data displayed on the screen, it is difficult to look at the control devices while the user is operating such devices.

One solution is to make the control devices in a predetermined configuration, such as that used for standard "QWERTY" keyboards. This then allows touch typing to be learned. However, learning the layout of a complicated control device can be time-consuming and difficult.

This invention provides data processing apparatus comprising:

an array of user-operable controls, the controls being adjustable by movement of a user's hand while touching a control;

a detector for detecting when a user's hand is touching a control;

a display screen for displaying respective screen icons associated with the controls; and

a display processor, responsive to a detection that a user's hand is touching one of the controls, for altering the screen icon associated with that control.

The invention allows the user to operate a potentially complicated array of controls without the need to learn the control layout by heart (which may be impossible in some applications) or to look at the controls while he is operating them. The controls are adjustable by movement once the user is touching the controls, so the invention indicates to the user as soon as he touches a control in order that the user can check that he is touching the correct control before initiating an adjustment.

An embodiment of the invention will now be described, by way of example only, with reference to the accompanying drawings, throughout which like parts are referred to by like references, and in which:

Figure 1 schematically illustrates an audio mixing console;

Figure 2 schematically illustrates a digital signal processor forming part of the audio mixing console of Figure 1;

Figure 3 schematically illustrates a control computer forming part of the audio

mixing console of Figure 1;

Figure 4 schematically illustrates the display on a display screen forming part of the audio mixing console of Figure 1;

Figure 5 schematically illustrates a fader panel forming part of the audio mixing console of Figure 1;

Figures 6A and 6B schematically illustrate a channel strip;

Figure 7 schematically illustrates a proximity and touch display;

Figures 8A and 8B schematically illustrate a screen pop-up display;

Figures 9 and 10 schematically illustrate circuitry within the fader panel of Figure 5;

Figure 11 schematically illustrates the format of a data word transmitted by the fader panel to the control computer;

Figure 12 is a flow chart summarising the operation of the control computer;

Figure 13 is a flow chart illustrating the processing of a serial message;

Figure 14 schematically illustrates a colour map; and

Figure 15 is a flow chart illustrating processing of a touch screen event.

Figure 1 schematically illustrates an audio mixing console comprising a touch-sensitive display screen 10, a control computer 20, a touch-fader panel 30, a slave display screen 40 and a signal processor 50.

The basic operation of the audio mixing console is that the signal processor 50 receives audio signals, in analogue or digital form, and processes them according to parameters supplied by the control computer 20. The user can adjust the parameters generated by the control computer 20 either by touching the display screen 10 or by operating the touch panel faders 30. Both of these modes of parameter adjustment will be described in detail below.

The slave screen 40 is provided to display various metering information such as audio signals levels at different points within the mixing console.

Figure 2 schematically illustrates the digital signal processor 50. The digital signal processor 50 comprises a control processor 100 for controlling data and filter coefficient flow within the digital signal processor 50, an input/output (I/O) buffer 110 for receiving parameter information and filter coefficients from the control computer 20 and for returning metering information back to the control computer 20,

a random access memory (RAM) 120 for storing current parameter data, a programmable DSP unit 130, an input analogue-to-digital converter 140 for converting input analogue audio signals into digital audio signals (where required) and an output digital-to-analogue converter 150 for converting digital audio signals into output analogue audio signals (where required).

Figure 3 schematically illustrates the structure of the control computer 20. The control computer 20 comprises a central processor 200 connected to a communications bus 210. Also connected to the communications bus are: an input buffer 220 for receiving data from the fader panel 30, a random access memory (RAM) 230, program storage memory 240, a BIOS colour map 250, a video card 260 including a video card colour map, an input buffer 270 for receiving data from the digital signal processor 50 and an output buffer 280 for transmitting data to the digital signal processor 50.

Figure 4 schematically illustrates the display on the touch-sensitive display screen 10.

Running vertically on each side of the display are two groups of ten channel strips 300, laid out in an arrangement similar to the physical layout of a conventional (hardware) audio mixing console. Each channel strip is identical to the others (apart from adjustments which are made by the user to the various parameters defined thereby) and the channel strips will be described with reference to Figures 6A and 6B below.

In a central part of the display 310 is provided a main fader 320, routing and equalisation controls 330 and display meters 340.

The channel strips include controls which are adjustable by the user, along with visual indications of the current state of the controls (rather like a hardware rotary potentiometer is adjustable by the user, with its current rotary position giving visual feedback of the current state of adjustment). This feature will be shown in more detail in Figures 6A and 6B. Accordingly, as a parameter is adjusted by the user, the control computer 20 makes corresponding changes to the displayed value on the display screen 10, and also generates a replacement set of filter or control coefficients to control the corresponding processing operation carried out by the signal processor 50.

The meters 340 provide simple level indications for, for example, left and right channels output by the DSP 130. (In the case, the level information is transmitted from the DSP 130, via the control processor 100 and the I/O buffer 110, to the input buffer 270 of the control computer.)

Figure 5 schematically illustrates the fader panel 30.

The fader panel 30 is primarily a substantially linear array of elongate touch-sensors. The touch-sensors will be described in more detail below, but briefly they are arranged to output three pieces of information to the control computer:

(a) whether the sensor is touched at any position along its length;

(b) the position along the length of the fader at which it is touched;

(c) a signal indicating the proximity of a user's hand to the sensor.

Suitable sensors are described in WO 95/31817.

The fader panel comprises one such sensor 350 for each channel strip on the display screen, plus an extra sensor corresponding to the main fader control 320 on the display screen.

The current level or state of a parameter control is thus shown on the screen. The touch-screen and fader touch-sensors can be used to adjust that current level in either direction, but this is only a relative adjustment form the current level. In other words, a particular finger position on a fader touch-sensor is not mapped to a particular gain value for the corresponding channel, but instead finger movements on a touch-sensor are mapped to adjustments up or down in the gain value.

So, when an adjustment is to be made via the fader panel, the user touches the appropriate fader touch-sensor (for the particular channel or the main fader to be adjusted). The user then moves his finger up or down that touch-sensor. Whatever linear position along the sensor the user's finger starts at, the adjustment is made with respect to the current level of the gain control represented by that fader.

Figures 6A and 6B taken together illustrate a channel strip.

The channel strip is a schematic illustration on the display screen of a number of audio processing controls and devices which can be placed in the signal processing path for each of the channels. From the top of Figure 6A, there is an input pre-amplifier, a variable delay control, a high-pass filter, two band-splitting filters, three controls relating to output feeds from the channel, a so-called panpot, a channel label,

and a channel fader. For all of the controls shown in Figure 6A, i.e. those which process different attributes of the audio signal, the controls can be displayed either in bold or faint colour on the display screen. Where a control is displayed in bold colour, this indicates that the control is "in circuit". Where a control is displayed in faint colour (so-called "greyed out"), the control can still be adjusted but it is not currently in the audio circuit.

As an example of the "greying out" feature, consider the "delay" control at the second-to-top control position in the channels strip (Figure 6A). The delay can be set to values between, say, 0 milliseconds (mS) and 1000 mS whether or not the delay processor is in the audio circuit, but the delay period is applied to the audio signal only if the delay processor is in circuit.

The channel strip of Figures 6A and 6B also illustrates how a visual feedback of a current control setting is given to the user. All of the controls except for the channel fader have an associated numerical value giving their current setting (e.g. 60Hz for a filter centre frequency, 0.0dB for a gain), as well as a semicircle with a pointer schematically illustrating the current setting with respect to the available range of settings in a manner similar to the hand of a clock from a lowest possible value (pointer horizontal and to the left) to a highest possible value (pointer horizontal and to the right). So, for the centre frequency of upper the band splitting filter in Figure 6A, the pointer is a third of the way around the semicircle, indicating that the current value of 60Hz is nearer to the lower extreme than to the higher extreme. The scales used to map current settings to rotary positions on the semicircles need not be linear, but could be logarithmic or otherwise.

Figure 7 schematically illustrates the way in which proximity and touch is displayed on the display screen with regard to the faders.

When one of the sensors on the fader panel 30 is touched, the corresponding fader display on the display screen (in this example, a particular fader 400) is coloured in a contrasting colour to the rest of the screen - e.g. red. This shows that that particular fader is currently being touched and so is open to adjustment.

Similarly, when the user's hand is near to one of the faders (as detected by the proximity detector - see above), that fader is coloured in one of several shades of a further contrasting colour, for example getting more saturated as the user's hand gets

closer to that fader touch-sensor. Examples are shown as faders 410 in Figure 7.

This system allows the user to track his hands across the fader panel 30 without having to look down at the fader panel itself, since he can see the proximity of his hands to different faders on the screen. Furthermore, because several degrees of proximity are available for display, it is possible to work out the location of the user's hand from the distribution of the different colours representing different degrees of proximity.

Figures 8A and 8B schematically illustrate a so-called screen pop-up display.

Figure 8A illustrates a part of the display screen illustrated in Figure 4, in particular a short vertical section of three channel strips. If one of the controls on the channel strips is touched on the screen (which is a touch-sensitive screen), the screen detects the position of the touch. This position is translated by the control computer (using a look-up table - not shown) into the identification of the corresponding control in one of the channel strips. A pup-up display, including that control, is shown and the control can be adjusted using icons on the pop-up display.

For example if the delay control 420 in Figure 8A is touched, a corresponding "pop-up" display appears and remains displayed until the user selects another control for adjustment or a time delay since the pop-up was touched expires. This is illustrated in Figure 8B.

The pop-up display includes the icon representing the control which was touched, shown in Figure 8B as the icon 430, but to clarify that this control is under adjustment the icon is shifted diagonally downwards and to the right by a few (e.g. 1-10) pixels. The pop-up also includes the title of the channel and the channel number 440, together with a fader 450 allowing the value of the particular control to be adjusted.

Two modes of adjustment are available to the user. In a first mode, the user touches the control and keeps his finger on the touch-sensitive screen. Once the pop-up has appeared, a vertical component of movement of the user's finger from the position at which he first touched the screen will cause a corresponding movement of the schematic fader 450 and a corresponding adjustment of the attribute controlled by that control.

In a further mode of operation, the user can touch and release a particular

control without moving the finger position between touch and release. The pop-up then appears. The user can then touch the screen within the pop-up and move his finger up or down to adjust the fader 450. If the user touches a non-active area of the pop-up, the pop-up disappears.

Again, adjustment is via a so-called "trim" mode, whereby the adjustment is relative to a current setting of the control, whatever position the user's finger starts at on the screen.

Figures 9 and 10 schematically illustrate circuitry within the fader panel 30. In Figure 9, a particular fader sensor 500 supplies three outputs to respective analogue-to-digital converters 510, 520, 530. These three outputs are: the analogue position at which the fader has been touched (if it has indeed been touched), a proximity signal indicating the proximity of a user's hand to the fader, and a touch status indicating whether or not the fader has been touched.

Digital equivalents of these signals are multiplexed together by a multiplexer 540, with an additional, fixed, signal indicating the identity of the channel to which the fader 500 relates. The multiplexed output of the multiplexer 540 is a three byte serial data word.

All of the these data words from the various channel faders are stored then in a previous value buffer 550 (Figure 10). Whenever a new serial word is received, it is compared by a compare-and-control logic circuit 560 with the previously buffered value. If a change is detected, the compare-and-control logic 560 causes an output circuit to transmit the three bytes representing the channel which has changed to the control computer 20.

So, a three byte word is transmitted to the control computer 20 only when the status of the fader corresponding to that channel has changed.

Figure 11 schematically illustrates the format of a data word transmitted by the fader panel to the control computer. Each byte 570 of the three byte data word comprises a byte header 580 and a payload 590 carrying information about the channel. The byte header 580 for each byte identifies which of the three bytes in the serial word is represented by the currently transmitted data. This enables the control computer 20 to detect when it has received all three bytes of a data word.

Figure 12 is a flow chart summarizing the operation of the control computer

20.

The control computer 20 operates a repetitive loop, which starts with a check of the input buffer 220 (at a step 600). At a step 610, the contents of the input buffer are examined to see whether a full three byte serial word is present. If such a word is present, the serial word is processed at a step 620. The processing associated with step 620 will be described in more detail with reference to Figure 13 below.

At a step 630, metering information is read from the signal processor 50 and the meters displayed on the display screen are redrawn.

At a step 640, a detection is made as to whether the touch screen has been touched or an existing touch has been removed or changed in position. If such a touch screen event is detected, the touch screen event is processed at a step 650. The processing associated with the step 650 will be described in more detail below with reference to Figure 15.

Finally, if any attributes associated with signal processing operations have changed throughout the operation of the loop, the new values are transmitted to the digital signal processor 50.

Figure 13 is a flow chart illustrating the processing of a serial message.

At a step 700, a detection is made as to whether the proximity or touch status of a channel has changed, i.e. is the channel touched where it was not touched before or has the proximity value changed. If the answer is yes, the colour map associated with particular areas of the fader corresponding to that channel is changed at a step 710. This process will be described in more detail with reference to Figure 14.

At a step 720, a detection is made as to whether a double click action has taken place. In other words, has the touch panel been touched, released, touched and released within a predetermined period. If such an event is detected, a channel cut control is toggled at a step 730 and the process ends. The channel cut control switches on or off the output of that channel. By toggling the control, if the control is currently off it toggles on, and vice versa.

If a double click event is not detected, a detection is made at a step 735 as to whether the panel is currently touched. If the answer is yes, a further detection is made 740 as to whether the touch is a new touch. This detection is made by examining a stored touch attribute from a previous operation of this flow chart.

If this is a new touch, a so-called trim mode is initiated at a step 750. This involves storing the position along the fader at which the new touch has been made and mapping it to the current value of the gain parameter controlled by that fader. Thus, when (in subsequent operations of this flow chart) the user's hand might be moved up or down the fader, adjustment is made from the current gain attribute controlled by the fader. If this is not a new touch, then at a step 760 an adjustment might have to be made to the gain attribute controlled by the fader, if the user's finger has moved up or down the fader since the last operation of the flow chart.

Finally, the stored previous proximity touch status and level attributes are set to those detected during the current operation of the flow chart at a step 770.

Figure 14 schematically illustrates a colour map.

The colour map provides a mapping between so-called logical colours (indexed from 0 to 255) and values of red, green and blue for actual display on the screen. So, for example, the logical colour 1 is mapped to 60R,60G,60B for display.

The R,G and B values are each adjustable between 0 and 255 (i.e. 8 bits) so the colour map defines a subset of 256 of the 16.7 million combinations of R, G and B values.

The control computer maintains two copies of the colour map. A first copy, the so-called "BIOS" copy, is alterable by the control computer under program control. Alterations can then be copied across into the video card colour map which is actually used to map logical colours onto display parameters for the display screen.

In the present embodiment, areas of the screen such as each of the channel faders are assigned a different logical colour, even though the R, G and B values specified by those logical colours may all be initially the same. When the display colour of an area is to be changed rapidly, for example when the touch or proximity status of a fader changes, then instead of redrawing the area using a standard but (in this context) relatively slow Microsoft Windows redraw command, a simple change is made to the colour map entry for the logical colour used for that particular area of the screen. This has almost instant effect on the actual displayed colour.

As described above, the change is made first to the BIOS colour map and then the change is propagated (using a standard command) to the video card colour map.

Figure 15 illustrates the processing relating to step 650 of Figure 12, namely

the processing of a touch screen event.

At a step 800, a check is made as to whether the screen is currently or previously (i.e. at the last operation of the flowchart) touched. If the answer is yes, then processing proceeds to step 830. If the answer is no, then a check is made at a step 810 as to whether a time delay has expired since the screen was last touched. If not, the process ends. If so, then any open pop-ups are closed at a step 820 and the process ends.

At step 830 a check is made as to whether the current touch represents a new adjustment. If so, processing proceeds to steps 840 and 850 where any existing pop-ups are closed. At a step 860 a new pop-up for the new adjustment is opened, and at a step 870 a trim operation is initiated by mapping the current setting of the selected control to the current finger position, so that adjustments are made in a relative, rather than an absolute, manner as described above. The process then ends.

If this is an existing adjustment, i.e. if the finger has not left the screen since the trim mode was set up (on a previous operation of the flow chart) then at a step 880 the current value of the control is altered (if the finger has moved) and the corresponding display within the pop-up is altered at a step 890.

In further embodiments of the invention, a detection (not shown) can be made of the average proximity value over those sensors detecting the proximity of a user's hand. The sensitivity of the proximity measurement can be adjusted as a result of this detection. For example, if the average value is that of a very weak detection (suggesting that the user's hand is far away) then the sensitivity can be increased.

CLAIMS

1. Data processing apparatus comprising:

an array of user-operable controls, the controls being adjustable by movement of a user's hand while touching a control;

a detector for detecting when a user's hand is touching a control;

a display screen for displaying respective screen icons associated with the controls; and

a display processor, responsive to a detection that a user's hand is touching one of the controls, for altering the screen icon associated with that control.

2. Apparatus according to claim 2, the apparatus being audio processing apparatus and the array of controls relating at least in part to the control of audio processing parameters.

3. Apparatus according to claim 2, in which the screen icon relating to a control displays, at least in part, an audio processing parameter associated with that control.

4. Apparatus according to any one of the preceding claims, in which the display processor is operable to change a display colour of at least a part of the icon associated with a touched control.

5. Data processing apparatus substantially as hereinbefore described with reference to the accompanying drawings.

The Patent Office

12

Application No: GB 9722541.1
Claims searched: 1-5

Examiner: Melanie Gee
Date of search: 6 May 1998

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.P): G4A (AKS); H4R (RSX)

Int Cl (Ed.6): G06F 3/02, 3/023, 3/033; H04H 7/00; H04S 7/00

Other:

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
A	EP 0689122 A1	(MATSUSHITA ELECTRIC INDUSTRIAL CO.), see abstract and col. 10.	
A	US 5523774 A	(SIEMENS MEDICAL SYSTEMS), see whole document	

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 :		(11) International Publication Number: WO 92/02000
G06K 9/46, 9/64, G07C 9/00	A1	(43) International Publication Date: 6 February 1992 (06.02.92)

(21) International Application Number: PCT/GB91/01183

(22) International Filing Date: 17 July 1991 (17.07.91)

(30) Priority data:

9015694.4	17 July 1990 (17.07.90)	GB
9019827.6	11 September 1990 (11.09.90)	GB

(71) Applicant *(for all designated States except US)*: BRITISH TELECOMMUNICATIONS PUBLIC LIMITED COMPANY [GB/GB]; 81 Newgate Street, London EC1A 7AJ (GB).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)* : SHACKLETON, Mark, Andrew [GB/GB]; 323 Cauldwell Hall Road, Ipswich, Suffolk IP4 5AH (GB). WELSH, William, John [GB/GB]; 47 Fountains Road, Ipswich, Suffolk IP2 9EF (GB).

(74) Agent: GREENWOOD, John, David; Intellectual Property Department, British Telecom, 151 Gower Street, London WC1E 6BA (GB).

(81) Designated States: AT (European patent), BE (European patent), CA, CH (European patent), DE (European patent), DK (European patent), ES (European patent), FR (European patent), GB (European patent), GR (European patent), IT (European patent), JP, LU (European patent), NL (European patent), SE (European patent), US.

Published

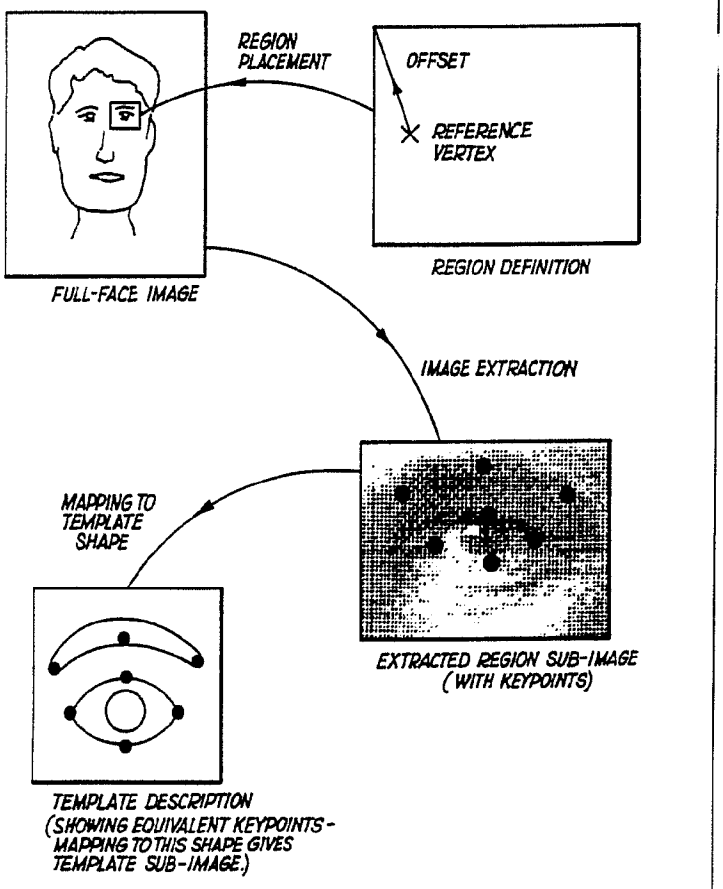
With international search report.

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: A METHOD OF PROCESSING AN IMAGE

(57) Abstract

A method of processing an image comprising the steps of: locating within the image the position of at least one predetermined feature; extracting from said image data representing each said feature; and calculating for each feature a feature vector representing the position of the image data of the feature in an N-dimensional space, said space being defined by a plurality of reference vectors each of which is an eigenvector of a training set of like features in which the image data of each feature is modified to normalise the shape of each feature thereby to reduce its deviation from a predetermined standard shape of said feature, which step is carried out before calculating the corresponding feature vector.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	**ES**	Spain	**MG**	Madagascar
AU	Australia	**FI**	Finland	**ML**	Mali
BB	Barbados	**FR**	France	**MN**	Mongolia
BE	Belgium	**GA**	Gabon	**MR**	Mauritania
BF	Burkina Faso	**GB**	United Kingdom	**MW**	Malawi
BG	Bulgaria	**GN**	Guinea	**NL**	Netherlands
BJ	Benin	**GR**	Greece	**NO**	Norway
BR	Brazil	**HU**	Hungary	**PL**	Poland
CA	Canada	**IT**	Italy	**RO**	Romania
CF	Central African Republic	**JP**	Japan	**SD**	Sudan
CG	Congo	**KP**	Democratic People's Republic of Korea	**SE**	Sweden
CH	Switzerland			**SN**	Senegal
CI	Côte d'Ivoire	**KR**	Republic of Korea	**SU+**	Soviet Union
CM	Cameroon	**LI**	Liechtenstein	**TD**	Chad
CS	Czechoslovakia	**LK**	Sri Lanka	**TG**	Togo
DE	Germany	**LU**	Luxembourg	**US**	United States of America
DK	Denmark	**MC**	Monaco		

+ It is not yet known for which States of the former Soviet Union any designation of the Soviet Union has effect.

A METHOD OF PROCESSING AN IMAGE

This invention relates to a method of processing an image and particularly, but not exclusively, to the use of such a method in the recognition and encoding of images of objects such as faces.

One field of object recognition which is potentially useful is in automatic identity verification techniques for restricted access buildings or fund transfer security, for example in the manner discussed in our UK application GB9005190.5. In many such fund transfer transactions a user carries a card which includes machine-readable data stored magnetically, electrically or optically. One particular application of face recognition is to prevent the use of such cards by unauthorised personnel by storing face identifying data of the correct user on the card, reading the data out, obtaining a facial image of the person seeking to use the card by means of a camera, analyzing the image, and comparing the results of the analysis with the data stored on the card for the correct user.

The storage capacity of such cards is typically only a few hundred bytes which is very much smaller than the memory space needed to store a recognisable image as a frame of pixels. It is therefore necessary to use an image processing technique which allows the image to be characterised using the smaller number of memory bytes.

Another application of image processing which reduces the number of bytes needed to characterise an image is in hybrid video coding techniques for video telephones as disclosed in our earlier filed application published as US patent 4841575. In this and similar applications the perceptually important parts of the image are located and the available coding data is preferentially allocated to those parts.

A known method of such processing of an image comprises the steps of: locating within the image the position of at least one predetermined feature; extracting image data from

said image representing each said feature; and calculating for each feature a feature vector representing the position of the image data of the feature in an N-dimensional space, said space being defined by a plurality of reference vectors each of which is an eigenvector of a training set of of images of like features.

The Karhunen-Loeve transform (KLT) is well known in the signal processing art for various applications. It has been proposed to apply this transform to identification of human faces (Sirovitch and Kirby, J. Opt. Soc. Am. A vol 4 no 3, pp 519-524 "Low Dimensional Procedure for the Characterisation of Human Faces", and IEEE Trans on Pattern Analysis and Machine Intelligence Vol 12, no 1 pp103-108 "Application of the Karhunen-Loeve Procedure for the Characterisation of Human Faces"). In these techniques, images of substantially the whole face of members of a reference population were processed to derive a set of N eigenvectors each having a picture-like appearance (eigen-pictures, or caricatures). These were stored. In a subsequent recognition phase, a given image of a test face (which need not belong to the reference population) was characterised by its image vector in the N-dimensional space defined by the eigenvectors. By comparing the image vector, or data derived therefrom, with the identification data one can generate a verification signal in dependence upon the result of the comparison.

However, despite the ease with which humans "never forget a face", the task for a machine is a formidable one because a person's facial appearance can vary widely in many respects over time because the eyes and mouth, in the case of facial image processing, for example, are mobile.

According to a first aspect of the present invention a method of processing an image according to the preamble of claim 1 is characterised in that the method further comprises the step of modifying the image data of each feature to normalise the shape of each feature thereby to reduce its deviation from a predetermined standard shape of said feature, which step is carried out before calculating the corresponding feature vector.

We have found that recognition accuracy of images of faces, for example, can be improved greatly by such a modifying step which reduces the effects of a persons's changing facial expression.

In the case of an image of a face, for example, the predetermined feature could be the entire face or a part of it such as the nose or mouth. Several predetermined features may be located and characterised as vectors in the corresponding space of eigenvectors if desired.

It will clear to those skilled in this field that the present invention is applicable to processing images of objects other than the faces of humans notwithstanding that the primary application envisaged by the applicant is in the field of human face images and that the discussion and specific examples of embodiments of the invention are directed to such images.

The invention also enables the use of fewer eigen-pictures, and hence results in a saving of storage or of transmission capacity.

Further, by modifying the shape of the feature towards a standard (topologically equivalent) feature shape, the accuracy with which the feature can be located is improved.

Preferably the training set of of images of like features are modified to normalise the shape of each of the training set of images thereby to reduce their deviation from a predetermined standard shape of said feature, which step is carried out before calculating the eigenvectors of the training set of images.

The method is useful not only for object recognition, but also as a hybrid coding technique in which feature position data and feature representative data (the N-dimensional vector) are transmitted to a receiver where an image is assembled by combining the eigen-pictures corresponding to the image vector.

Eigen-pictures provide a means by which the variation in a set of related images can be extracted and used to represent those images and others like them. For instance, an

eye image could be economically represented in terms of 'best' coordinate system 'eigen-eyes'.

The eigen-pictures themselves are determined from a training set of representative images, and are formed such that the first eigen-picture embodies the maximum variance between the images, and successive eigen-pictures have monotonically decreasing variance. An image in the set can then be expressed as a series, with the eigen-pictures effectively forming basis functions:

$$I = M + w_1P_1 + w_2P_2 + \ldots + w_mP_m$$

where M = mean over entire training set of images

w_i = component of the i'th eigen-picture

P_i = i'th eigen-picture, of m,

I = original image

If we truncate the above series we still have the best representation we could for the given number of eigen-pictures, in a mean-square-error sense.

The basis of eigen-pictures is chosen such that they point in the directions of maximum variance, subject to being orthogonal. In other words, each training image is considered as a point in n-dimensional space, where 'n' is the size of the training images in pels; eigen-picture vectors are then chosen to lie on lines of maximum variance through the cluster(s) produced.

Given training images $I_1, \ldots, I_m$, we first form the mean image M, and then the difference images (a.k.a. 'caricatures') $D_i=I_i-M$.

The first paragraph (above) is equivalent to choosing our eigen-picture vectors P_k such that

$$\lambda_k=(\frac{1}{m})\sum_j (P_k^T D_j)^2 \quad \textit{is maximised}$$

$$\textit{with } P_i^T P_k=0, \quad i<k$$

The eigen-pictures P_k above are in fact the eigenvectors of a very large covariance matrix, the solution of which would

be intractable. However, the problem can be reduced to more manageable proportions by forming the matrix L where

$$L_{ij} = D_i^T D_j$$

and solving for the eigenvectors v_k of L.

The eigen-pictures can then be found by

$$P_k = \sum_j (v_{kj} D_j)$$

The term 'representation vector' has been used to refer to the vector whose components (w_i) are the factors applied to each eigen-picture (P_i) in the series. That is

$$\Omega = (w_1, w_2, \ldots, w_m)^T$$

The representation vector equivalent to an image I is formed by taking the inner product of I's caricature with each eigen-picture:

$w_i = (I-M)^T P_i$, for $1 \leq +i \leq m$.

Note that a certain assumption is made when it comes to representing an image taken from outside the training set used to create eigen-pictures; the image is assumed to be sufficiently 'similar' to those in training set to enable it to be well represented by the same eigen-pictures.

The representation of two images can be compared by calculating the Euclidean distance between them:

$d_{ij} = /\Omega_i - \Omega_j/$.

Thus, recognition can be achieved via a simple threshold, $d_{ij} < T$ means recognised, or d_{ij} can be used as a sliding confidence scale.

Deformable templates consist of parametrically defined geometric templates which interact with images to provide a best fit of the template to a corresponding image feature. For example, a template for an eye might consist of a circle for the iris and two parabolas for the eye/eyelid boundaries, where size, shape and location parameters are variable.

An energy function is formed by integrating certain image attributes over template boundaries, and parameters are iteratively updated in an attempt to minimise this function.

This has the effect of moving the template towards the best available fit in the given image.

The location within the image of the position of at least one predetermined feature may employ a first technique to provide a coarse estimation of position and a second, different, technique to improve upon the coarse estimation. The second technique preferably involves the use of such a deformable template technique.

The deformable template technique requires certain filtered images in addition to the raw image itself, notably peak, valley and edge images. Suitable processed images can be obtained using morphological filters, and it is this stage which is detailed below.

Morphological filters are able to provide a wide range of filtering functions including nonlinear image filtering, noise suppression, edge detection, skeletonization, shape recognition etc. All of these functions are provided via simple combinations of two basic operations termed erosion and dilation. In our case we are only interested in valley, peak and edge detection.

Erosion of greyscale images effectively causes bright areas to shrink in upon themselves, whereas dilation causes bright areas to expand. An erosion followed by a dilation causes bright peaks to be lost (operator called 'open'. Conversely, a dilation followed by an erosion causes dark valleys to be filled (operator called 'close'). For specific details see Maragos P, (1987), "Tutorial on Advances in Morphological Image Processing and Analysis", Optical Engineering. Vol 26. No. 7.

In image processing systems of the kind to which the present invention relates, it is often necessary to locate the object, eg head or face, within the image prior to processing.

Usually this is achieved by edge detection, but traditional edge detection techniques are purely local - an edge is indicated whenever a gradient of image intensity occurs - and hence will not in general form an edge that is completely closed (ie. forms a loop around the head) but will instead create a number of edge segments which together

outline or partly outline the head. Post-processing of some kind is thus usually necessary.

We have found that the adaptive contour model, or "snake", technique is particularly effective for this purpose. Preferably, the predetermined feature of the image is located by determining parameters of a closed curve arranged to lie adjacent a plurality of edge features of the image, said curve being constrained to exceed a minimum curvature and to have a minimum length compatible therewith. The boundary of the curve may be initially calculated proximate the edges of the image, and subsequently interactively reduced.

Prior to a detailed description of the physical embodiment of the invention, the 'snake' signal processing techniques mentioned above will now be described in greater detail.

Introduced by Kass et al [Kass m, Witkin A, Terpozopoulus d. "Snakes: Active Contour Models", International Journal of Computer Vision, 321-331, 1988], snakes are a method of attempting to provide some of the post-processing that our own visual system performs. A snake has built into it various properties that are associated with both edges and the human visual system (Eg continuity, smoothness and to some extent the capability to fill in sections of an edge that have been occluded).

A snake is a continuous curve (possibly closed) that attempts to dynamically position itself from a given starting position in such a way that it 'clings' to edges in the image. The form of snake that will be considered here consists of curves that are piecewise polynomial. That is, the curve is in general constructed from N segments $\{x_i(s), y_i(s)\}i=1,\ldots,N$ where each of the $x_i(s)$ and $y_i(s)$ are polynomials in the parameter s. As the parameter s is varied a curve is traced out.

From now on snakes will be referred to as the parametric curve $\underline{u}(s)=(x(s),y(s))$ where s is assumed to vary between 0 and 1. What properties should an 'edge hugging' snake have?

The snake must be 'driven' by the image. That is, it must be able to detect an edge in the image and align itself with the edge. One way of achieving this is to try to position the snake such that the average 'edge strength' (however that may be measured) along the length of the snake is maximised. If the measure of edge strength is $F(x,y) \geq 0$ at the image point (x,y) then this amounts to saying that the snake $\underline{u}(s)$ is to be chosen in such a way that the functional

$$\int_{s=0}^{1} F(x(s),y(s))\,ds \qquad \ldots(1)$$

is maximised. This will ensure that the snake will tend to mould itself to edges in the image if it finds them, but does not guarantee that it will find them in the first place. Given an image, the functional may have many local minima- static problem: finding them is where the 'dynamics' arise.

An edge detector applied to an image will tend to produce an edge map consisting of mainly thin edges. This means that the edge strength function tends to be zero at most places in the image, apart from on a few lines. As a consequence a snake placed some distance from an edge may not be attracted towards the edge because the edge strength is effectively zero at the snakes initial position. To help the snake come under the influence of an edge, the edge image is blurred to broaden the width of the edges.

If an elastic band were held around a convex object and then let go, the band would contract until the object prevented it from doing so further. At this point the band would be moulded to the object, thus describing the boundary. Two forces are at work here. Firstly that providing the natural tendency of the band to contract; secondly the opposing force provided by the object. The band contracts because it tries to minimise its elastic energy due to stretching. If the band were described by the parametric curve $\underline{u}(s)=(x(s),y(s))$ then the elastic energy at any point $\hat{s}$ is proportional to

$$\left(\left.\frac{d\underline{u}}{ds}\right|_{\hat{s}}\right)^2 = \left(\left.\frac{dx}{ds}\right|_{\hat{s}}\right)^2 + \left(\left.\frac{dy}{ds}\right|_{\hat{s}}\right)^2$$

That is, the energy is proportional to the square of how much the curve is being stretched at that point. The elastic energy along its entire length, given the constraint of the object, is minimised. Hence the elastic band assumes the shape of the curve $\underline{u}(s)=(x(s),y(s))$ where the $\underline{u}(s)$ minimises the functional

$$\int_{s=0}^{1}\left\{\left(\frac{dx}{ds}\right)^2 + \left(\frac{dy}{ds}\right)^2\right\}ds \qquad \ldots(2)$$

subject to the constraints of the object. We would like closed snakes to have analogous behaviour. That is, to have a tendency to contract, but to be prevented from doing so by the objects in an image. To model this behaviour the parametric curve for the snake is chosen so that the functional (2) tends to be minimised. If in addition the forcing term (1) were included then the snake would be prevented from contracting 'through objects' as it would be attracted toward their edges. The attractive force would also tend to pull the snake into the hollows of a concave boundary, provided that the restoring 'elastic force' was not too great.

One of the properties of the edges that is difficult to model is their behaviour when they can no longer be seen. If one were looking at a car and a person stood in front of it, few would have any difficulty imagining the contours of the edge of the car that were occluded. They would be 'smooth' extensions of the contours either side of the person. If the above elastic band approach were adopted it would be found that the band formed a straight line where the car was occluded (because it tries to minimise energy, and thus length in this situation). If however the band had some stiffness (that is a resistance to bending, as for example displayed by a flexible bar) then it would tend to form a smooth curve in the occluded region of the image and be tangential to the boundaries on either side.

Again a flexible bar tends to form a shape so that its elastic energy is minimised. The elastic energy in bending is dependent on the curvature of the bar, that is the second derivatives. To help force the snake to emulate this type of behaviour the parametric curve $\underline{u}(s)=(x(s),y(s))$ is chosen so it tends to minimise the functional

$$\int_{s=0}^{1}\left\{\left(\frac{d^2x}{ds^2}\right)^2+\left(\frac{d^2y}{ds^2}\right)^2\right\}ds \qquad \ldots(3)$$

which represents a pseudo-bending energy term. Of course, if a snake were made too stiff it would be difficult to force it to conform to highly curved boundaries under the action of the forcing term (1).

Three desirable properties of snakes have now been identified. To incorporate all three into the snake at once the parametric curve $\underline{u}(s)=(x(s),y(s))$ representing the snake is chosen so that it minimises the functional

$$I(x(s),y(s)) =$$

$$\int_{s=0}^{1}\left\{\alpha(s)\left[\left(\frac{d^2x}{ds^2}\right)^2+\left(\frac{d^2y}{ds^2}\right)^2\right]+\beta(s)\left[\left(\frac{dx}{ds}\right)^2+\left(\frac{dy}{ds}\right)^2\right]-F(x(s),y(s))\right\}ds \qquad \ldots(4)$$

Here the terms $\alpha(s)>0$ and $\beta(s)\geq 0$ represent respectively the amount of stiffness and elasticity that the snake is to have. It is clear that if the snake approach is to be successful then the correct balance of these parameters is crucial. Too much stiffness and the snake will not correctly hug the boundaries; too much elasticity and closed snakes will be pulled across boundaries and contract to a point or may even break away from boundaries at concave regions. The negative sign in front of the forcing term is because minimising $-\int F(x,y)ds$ is equivalent to maximising $\int F(x,y)ds$.

As it stands, minimising the functional (4) is trivial. If the snake is not closed then the solution degenerates into a single point $(x(s),y(s))=constant$, where the point is chosen to minimise the edge strength $F(x(s),y(s))$. Physically, this

is because the snake will tend to pull its two end points together in order to minimise the elastic energy, and thus shrink to a single point. The global minimum is attained at the point in the image where the edge strength is largest. To prevent this from occurring it is necessary to fix the positions of the ends of the snake in some way. That is, 'boundary conditions' are required. It turns out to be necessary to fix more than just the location of the end points and two further conditions are required for a well posed problem. A convenient condition is to impose zero curvature at each end point.

Similarly, the global minimum for a closed-loop snake occurs when it contracts to a single point. However, in contrast to an fixed-end snake, additional boundary conditions cannot be applied to eliminate the degenerate solution. The degenerate solution in this case is the true global minimum.

Clearly the ideal situation is to seek a local minimum in the locality of the initial position of the snake. In practice the problem that is solved is weaker than this: find a curve $\hat{\underline{u}}(s)=(\hat{x}(s),\hat{y}(s)) \in H^2[0,1]xH^2[0,1]$ such that

$$\left.\frac{\partial I(\hat{\underline{u}}(s)+\epsilon\underline{v}(s))}{\partial\epsilon}\right|_{\epsilon=0} = 0 \; ; \quad \underline{v}(s) \in H_0^2[0,1] \; X \; H_0^2[0,1] \qquad \ldots (5)$$

Here $H^2[0,1]$ denotes the class of real valued functions defined on [0,1] that have 'finite energy' in the second derivatives (that is the integral of the square of the second derivatives exists [Keller HB. Numerical Methods for Two-Point Boundary Value Problems, Blaisdell, 1968] and $H_0^2[0,1]$ is the class of functions in $H^2[0,1]$ that are zero at s=0 and s=1. To see how this relates to finding a minimum consider $\hat{\underline{u}}(s)$ to be a local minimum and $\hat{\underline{u}}(s)+\epsilon\underline{v}(s)$ to be a perturbation about the minimum that satisfies the same boundary conditions (ie $\underline{v}(0)=\underline{v}(1)=0$).

Clearly, considered as a function of ϵ, $I(\epsilon)=I\hat{\underline{u}}(s)+\epsilon\underline{v}(s))$ is a minimum at $\epsilon=0$. Hence the derivative of $I(\epsilon)$ must be zero at $\epsilon=0$. Equation (5) is therefore a necessary condition for a local minimum. Although solutions to (5) are not guaranteed to be minima for completely general

edge strength functions, it has been found in practice that solutions are indeed minima.

Standard arguments in the calculus of variations show that problem (5) is equivalent to another problem, which is simpler to solve: find a curve $(\hat{x}(s),\hat{y}(s)) \in C^4[0,1]xC^4[0,1]$ that satisfies the pair of fourth order ordinary differential equations

$$-\frac{d^2}{ds^2}\left\{\alpha(s)\frac{d^2\hat{x}}{ds^2}\right\} + \frac{d}{ds}\left\{\beta(s)\frac{d\hat{x}}{ds}\right\} + \frac{1}{2}\frac{\partial F}{\partial x}\bigg|_{(\hat{x},\hat{y})} = 0 \qquad \ldots(6)$$

$$-\frac{d^2}{ds^2}\left\{\alpha(s)\frac{d^2\hat{y}}{ds^2}\right\} + \frac{d}{ds}\left\{\beta(s)\frac{d\hat{y}}{ds}\right\} + \frac{1}{2}\frac{\partial F}{\partial y}\bigg|_{(\hat{x},\hat{y})} = 0 \qquad \ldots(7)$$

together with the boundary conditions

$\hat{x}(0),\hat{y}(0),\hat{x}(1),\hat{y}(1)$ *given, and*

$$\frac{d^2\hat{x}}{ds^2}\bigg|_{s=0} = \frac{d^2\hat{y}}{ds^2}\bigg|_{s=0} = \frac{d^2\hat{x}}{ds^2}\bigg|_{s=1} = \frac{d^2\hat{y}}{ds^2}\bigg|_{s=1} = 0 \qquad \ldots(7)$$

The statement of the problem is for the case of a fixed-end snake, but if the snake is to form a closed loop then the boundary conditions above are replaced by periodicity conditions. Both of these problems can easily be solved using finite differences.

The finite difference approach starts by discretising the interval [0,1] into N-1 equispaced subintervals of length h=1\(N-1) and defines a set of nodes

$\{(s_i)\}_{i=1}^{i=N}$ where $s_i=(i-1)h$.

The method seeks a set of approximations

$\{(x_i,y_i)\}_{i=1}^{i=N}$ *to* $\{(x(s_i),y(s_i))\}_{i=1}^{i=N}$

by replacing the differential equations (6) and (7) in the continuous variables with a set of difference equations in the discrete variables [Keller HB., *ibid.*]. Replacing the derivatives in (6) by difference approximations at the point s_i gives

$$-\frac{1}{h^2}\left\{\alpha_{i+1}\frac{(x_{i+2}-2x_{i+1}+x_i)}{h^2} - 2\alpha_i\frac{(x_{i+1}-2x_i+x_{i-1})}{h^2} + \alpha_{i-1}\frac{(x_i-2x_{i-1}+x_{i-2})}{h^2}\right\}$$

$$+\frac{1}{h}\left\{\beta_{i+1}\frac{(x_{i+1}-x_i)}{h} - \beta_i\frac{(x_i-x_{i-1})}{h}\right\} + \frac{1}{2}\frac{\partial F}{\partial x}\bigg|_{(x_i,y_i)} = 0; \quad \textit{for } i=3,4\ldots,N-2$$

... (9)

where $\alpha_i=\alpha(s_i)$ and $\beta_i=\beta(s_i)$. Similarly a difference approximation to (7) may be derived. Note that the difference equation only holds at internal modes in the interval where the indices referenced lie in the range I to N. Collecting like terms together, (9) can be written as

$$a_i x_{i-2} + b_i x_{i-1} + c_i x_i + d_i x_{i+1} + e_i x_{i+2} = f_i$$

where

$$a_i = -\frac{\alpha_{i-1}}{h^4}$$

$$b_i = \frac{2\alpha_i}{h^4} + \frac{2\alpha_{i-1}}{h^4} + \frac{\beta_i}{h^2}$$

$$c_i = -\left\{\frac{\alpha_{i+1}}{h^4} + \frac{4\alpha_i}{h^4} + \frac{\alpha_{i-1}}{h^4} + \frac{\beta_{i+1}}{h^2} + \frac{\beta_i}{h^2}\right\}$$

$$d_i = \frac{2\alpha_{i+1}}{h^4} + \frac{2\alpha_i}{h^4} + \frac{\beta_{i+1}}{h^2}$$

$$e_i = -\frac{\alpha_{i+1}}{h^4}$$

$$f_i = -\frac{1}{2}\frac{\partial F}{\partial x}\bigg|_{(x_i,y_i)}$$

Discretising both the differential equations (6) and (7) and taking boundary conditions into account, the finite difference approximations $\underline{x}=\{x_i\}$ and $\underline{y}=\{y_i\}$ to $\{x_i\}$ and $\{y_i)\}$, respectively, satisfy the following system of the algebraic equations

$K\underline{x}=\underline{f}(\underline{x},\underline{y})$, $K\underline{y}=\underline{g}(\underline{x},\underline{y})$ ... (10)

The structure of the matrices K and the right hand vectors $\underline{f}$ and $\underline{g}$ are different depending on whether closed or open snake boundary conditions are used. If the snake is closed then fictitious nodes at S_0, S_{-1}, S_{N+1} and S_{N+2} are introduced and the difference equation (9) is applied at nodes 0, 1, N-1 and N.

Periodicity implies that $x_0=x_N$, $x_{-1}=x_{N-1}$, $x_{N+1}=x_1$ and $x_2=x_{N+2}$. With these conditions in force the coefficient matrix becomes

$$K=\begin{pmatrix} c_1 & d_1 & e_1 & & & & a_i & b_i \\ b_2 & c_2 & d_2 & e_2 & & & & a_2 \\ a_3 & b_3 & c_3 & d_3 & e_3 & & & \\ & a_4 & b_4 & c_4 & d_4 & e_4 & & \\ & & . & . & . & . & . & \\ & & & . & . & . & . & . \\ e_{N-1} & & & & a_{N-1} & b_{N-1} & c_{N-1} & d_{N-1} \\ d_N & e_N & & & & a_N & b_N & c_N \end{pmatrix}$$

and the right hand side vector is

$(f_1, f_2, \ldots, f_N)^T$

For fixed-end snakes fictitious nodes at S_0 and S_{N+1} are introduced and the difference equation (9) is applied at nodes S_1 and S_{N+1}. Two extra difference equations are introduced to approximate the zero curvature boundary conditions:

$$\left.\frac{d^2x}{ds^2}\right|_{s_1} = \left.\frac{d^2x}{ds^2}\right|_{s_N} = 0$$

namely $x_0-2x_1+x_2=0$ and $x_{-1}-2x_N+x_{N+1}=0$. The coefficient matrix is now

$$K=\begin{pmatrix} (c_2-a_2) & d_2 & e_2 & & & & \\ b_3 & c_3 & d_3 & e_3 & & & \\ a_4 & b_4 & c_4 & d_4 & e_4 & & \\ & a_5 & b_5 & c_5 & d_5 & e_5 & \\ & & \cdot & \cdot & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & \cdot & \cdot \\ & & & & a_{N-2} & b_{N-2} & c_{N-2} & d_{N-2} \\ & & & & & a_{N-1} & b_{N-1} & (c_{N-1}-e_{N-1}) \end{pmatrix}$$

and the right hand side vector is

$(f_2-(2a_2+b_2)x_1,\ f_3-a_3x_1,\ \ldots$

$\ldots\ f_{N-3},\ f_{N-2}e_{N-2}x_N,\ f_{N-1}-(2e_{N-1}+d_{N-1})x_N)^T$

The right hand side vector for the difference equations corresponding to (7) is derived in a similar fashion.

The system (10) represents a set of non-linear equations that has to be solved. The coefficient matrix is symmetric and positive definite, and banded for the fixed-end snake. For a closed-loop snake with periodic boundary conditions it is banded, apart from a few off-diagonal entries. As the system is non-linear it is solved iteratively. The iteration performed is

$$\frac{(x_{n+1}-x_n)}{\gamma}+K\underline{x}_{n+1}=f(\underline{x}_n,\underline{y}_n) \quad for\ n=0,1,2,...$$

$$\frac{(y_{n+1}-y_n)}{\gamma}+K\underline{y}_{n+1}=g(\underline{x}_n,\underline{y}_n) \quad for\ n=0,1,2,...$$

where $\gamma>0$ is a stabilisation parameter. This can be rewritten as

$$\left(K+\frac{1}{\gamma}I\right)\underline{x}_{n+1}=\frac{1}{\gamma}\underline{x}_n+f(\underline{x}_n,\underline{y}_n) \quad for\ n=0,1,2,...$$

$$\left(K+\frac{1}{\gamma}I\right)\underline{y}_{n+1}=\frac{1}{\gamma}\underline{y}_n+g(\underline{x}_n,\underline{y}_n) \quad for\ n=0,1,2,...$$

This system has to be solved for each n. For a closed-loop snake the matrix on the left hand side is difficult to invert directly because the terms that are outside the main diagonal band destroy the band structure. In general, the coefficient matrix K can be split into the sum of a banded matrix B plus a non-banded matrix A; K=A+B. For a fixed-end snake the matrix A would be zero. The system of equations is now solved for each n by performing the iteration

$$\left(B+\frac{1}{\gamma}I\right)\underline{x}_{n+1}^{(k+1)} = -A\underline{x}_{n+1}^{(k)} + \frac{1}{\gamma}\underline{x}_n + f(\underline{x}_n,\underline{y}_n) \qquad for\ k=0,1,2,...$$

$$\left(B+\frac{1}{\gamma}I\right)\underline{y}_{n+1}^{(k+1)} = -A\underline{y}_{n+1}^{(k)} + \frac{1}{\gamma}\underline{y}_n + g(\underline{x}_n,\underline{y}_n) \qquad for\ k=0,1,2,...$$

The matrix $(B + I/\gamma)$ is a band matrix and can be expressed as a product of Cholesky factors LL^T [Johnson Reiss, *ibid.*]. The systems are solved at each stage by first solving

$$L\tilde{\underline{x}}_{n+1}^{(k+1)} = -A\underline{x}_{n+1}^{(k)} + \frac{1}{\gamma}\underline{x}_n + f(\underline{x}_n,\underline{y}_n)$$

$$L\tilde{\underline{y}}_{n+1}^{(k+1)} = -A\underline{y}_{n+1}^{(k)} + \frac{1}{\gamma}\underline{y}_n + g(\underline{x}_n,\underline{y}_n)$$

followed by

$$L^T\underline{x}_{n+1}^{(k+1)} = \tilde{\underline{x}}_{n+1}^{(k+1)}$$

$$L^T\underline{y}_{n+1}^{(k+1)} = \tilde{\underline{y}}_{n+1}^{(k+1)}$$

Notice that the Cholesky decomposition only has to be performed once.

Model-based coding schemes use 2-D or 3-D models of scene objects in order to reduce the redundancy in the information needed to encode a moving sequence of images. The location and tracking of moving objects is of fundamental importance of this. Videoconference and videophone type scenes may present difficulties for conventional machine

vision algorithms as there can often be low contrast and 'fuzzy' moving boundaries between a person's hair and the background. Adaptive contour models or 'snakes' form a class of techniques which are able to locate and track object boundaries; they can fit themselves to low contrast boundaries and can fill in across boundary segments between which there is little or no local evidence for an edge. This paper discusses the use of snakes for isolating the head boundary in images as well as a technique which combines block motion estimation and the snake: the 'power-assisted' snake.

The snake is a continuous curve (possibly closed) which attempts to dynamically position itself from a given starting position in such a way that it clings to edges in the image. Full details of the implementation for both closed and 'fixed-end' snakes are given in Waite JB, Welsh WJ, "Head Boundary Location using Snakes", British Telecom Technology Journal, Vol 8, No 3, July 1990, which describes two alternative implementation strategies: finite elements and finite differences. We implemented both closed and fixed-end snakes using finite differences. The snake is initialised around the periphery of a head-and-shoulders image and allowed to contract under its own internal elastic force. It is also acted on by forces derived from the image which are generated by first processing the image using a Laplacian-type operator with a large space constant the output of which is rectified and modified using a smooth non-linear function. The rectification results in isolating the 'valley' features which have been shown to correspond to the subjectively important boundaries in facial images; the non-linear function effectively reduces the weighting of strong edges relative to weaker edges in order to give the weaker boundaries a better chance to influence the snake. After about 200 iterations of the snake it reaches the position hugging the boundary of the head. In a second example, a fixed-end snake with its end points at the bottom corners of the image was allowed to contract in from the sides and top of the image. The snake stabilises on the boundary between hair and background although this is a relatively low-contrast boundary in the

image. As the snake would face problems trying to contract across a patterned background, it might be better to derive the image forces from a moving edge detector.

In Kass et al, *ibid.*, an example is shown of snakes being used to track the moving lips of a person. First, the snake is stabilised on the lips in the first frame of a moving sequence of images; in the second frame it is initialised in the position corresponding to its stable position in the previous frame and allowed to achieve equilibrium again. There is a clear problem with the technique in this form in that if the motion is too great between frames, the snake may lock on to different features in the next frame and thus lose track. Kass suggests a remedy using the principle of 'scale-space continuation': the snake is allowed to stabilise first on an image which has been smoothed using a Gaussian filter with a large space constant; this has the effect of pulling the snake in from a large distance. After equilibrium has occurred, the snake is presented with a new set of image forces derived by using a Gaussian with slightly smaller space constant and the process is continued until equilibrium has occurred in the image at the highest level of resolution possible.

This is clearly a computationally expensive process; a radically simpler technique has been developed and found to work well and this will now be described. After the snake has reached equilibrium in the first frame of a sequence, block motion estimation is carried out at the positions of the snake nodes (the snake is conventionally implemented by approximating it with a set of discrete nodes - 24 in our implementation). The motion estimation is performed from one frame into the next frame which is the opposite sense to that conventionally performed during motion compensation for video coding. If the best match positions for the blocks are plotted in the next frame then, due to the 'aperture problem', a good match can often be found at a range of points along a boundary segment which is longer than the side of the block being matched. The effect is to produce a non-uniform spacing of the points. The snake is then initialised in the

frame with its nodes at the positions of the best match block positions and allowed to run for a few iterations, typically ten. The result is the nodes are now uniformly distributed along the boundary; the snake has successfully tracked the boundary, the block motion estimation having acted as a sort of 'power-assist' which will ensure tracking is maintained as long as the overall motion is not greater than the maximum displacement of the block search. As the motion estimation is performed at only a small set of points, the computation time is not increased significantly.

Both fixed-end and closed snakes have been shown to perform object boundary location even in situations where they may be a low contrast between the object and its background.

A composite technique using both block motion estimation and snake fitting has been shown to perform boundary tracking in a sequence of moving images. The technique is simpler to implement than an equivalent coarse-to-fine resolution technique. The methods described in the paper have so far been tested in images where the object boundaries do not have very great or discontinuous curvature at any point; if these conditions are not met, the snake would fail to conform itself correctly to the boundary contour. One solution, currently being pursued, is to effectively split the boundaries into a number of shorter segments and fit these segments with several fixed-end snakes.

According to a second aspect of the present invention a method of verifying the identity of the user of a data carrier comprises: generating a digital facial image of the user; receiving the data carrier and reading therefrom identification data; performing the method of the first aspect of the present invention; comparing each feature vector, or data derived therefrom, with the identification data; and generating a verification signal in dependence upon the comparison.

According to a yet further aspect of the present invention apparatus for verifying the identity of the user of a data carrier comprises: means for generating a digital facial image of the user; means for receiving the data carrier

and reading therefrom identification data; means for performing the method of the first aspect of the present invention; and means for comparing each feature vector, or data derived therefrom, with the identification data and generating a verification signal in dependence upon the comparison.

An embodiment of the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Figure 1 is flow diagram of the calculation of a feature vector;

Figure 2 illustrates apparatus for credit verification using the method of the present invention; and

Figure 3 illustrates the method of the present invention.

Referring to Figure 1, an overview of an embodiment incorporating both aspects of the invention will be described.

An image of the human face is captured in some manner - for example by using a video camera or by a photograph - and digitised to provide an array of pixel values. A head detection algorithm is employed to locate the position within the array of the face or head. This head location stage may comprise one of several known methods but is preferably a method using the above described "snake" techniques. Pixel data lying outside the boundaries thus determined are ignored.

The second step is carried out on the pixel data lying inside the boundaries to locate the features to be used for recognition - typically the eyes and mouth. Again, several location techniques for finding the position of the eyes and mouth are known from the prior art, but preferably a two stage process of coarse location followed by fine location is employed. The coarse location technique might, for example, be that described in US 4841575.

The fine location technique preferably uses the deformable template technique described by Yuille et al "Feature Extraction from Faces using Deformable Templates", Harvard Robotics Lab Technical Report Number; 88/2 published in Computer Vision and Pattern Recognition, June 1989 IEEE.

In this technique, which has been described above, a line model topologically equivalent to the feature is positioned (by the coarse location technique) near the feature and is iteratively moved and deformed until the best fit is obtained. The feature is identified as being at this position.

Next, the shape of the feature is changed until it assumes a standard, topologically equivalent, shape. If the fine location technique utilised deformable templates as disclosed above, then the deformation of the feature can be achieved to some extent by reversing the deformation of the template to match the feature to the initial, standard, shape of the template.

Since the exact position of the feature is now known, and its exact shape is specified, recognition using this information can be employed as identifiers of the image supplementary to the recognition process using the feature vector of the feature. All image data outside the region identified as being the feature are ignored and the image data identified as being the feature are resolved into its orthogonal eigen-picture components corresponding to that feature. The component vector is then compared with a component vector corresponding to a given person to be identified and, in the event of substantial similarity, recognition may be indicated.

Referring to Figure 2, an embodiment of the invention suitable for credit card verification will now be described.

A video camera 1 receives an image of a prospective user of a credit card terminal. Upon entry of the card to a card entry device 2, the analogue output of the video camera 1 is digitised by an AD converter 3, and sequentially clocked into a framestore 4. A video processor 5 (for example, a suitable processed digital signal processing chip such as that AT&T DSP 20) is connected to access the framestore 4 and processes the digital image therein to form and edge-enhanced image. One method of doing this is simply to subtract each sample from its predecessor to form a difference picture, but a better method involves the use of a Laplacian type of operator, the output of which is modified by a sigmoidal

function which suppresses small levels of activity due to noise as well as very strong edges whilst leaving intermediate values barely changed. By this means, a smoother edge image is generated, and weak edge contours such as those around the line of the chin are enhanced. This edge picture is stored in an edge picture frame buffer 6. The processor then executes a closed loop snake method, using finite differences, to derive a boundary which encompasses the head. Once the snake algorithm has converged, the position of the boundaries of the head in the edge image and hence the corresponding image in the frame store 4 is now in force.

The edge image in the framestore 6 is then processed to derive a coarse approximation to the location of the features of interest - typically the eyes and the mouth. The method of Nagao is one suitable technique (Nagoa M, "Picture Recognition and Data Structure", Graphic Languages - Ed. Rosenfield) as described in our earlier application EP0225729. The estimates of position thus derived are used as starting positions for the dynamic template process which establishes the exact feature position.

Accordingly, processor 5 employs the method described in Yuille et al [Yuille A, Cohen D, Hallinan P, (1988), "Facial Feature Extraction by Deformable Templates", Harvard Robotics Lab. Technical Report no. 88-2] to derive position data for each feature which consists of a size (or resolution) and a series of point coordinates given as fractions of the total size of the template. Certain of these points are designated as keypoints which are always internal to the template, the other points always being edge points. These key point position data are stored, and may also be used as recognition indicia. This is indicated in Figure 3.

Next, the geometrical transformation of the feature to the standard shape is performed by the processor 5. This transformation takes the form of a mapping between triangular facets of the regions and the templates. The facets consist of local collections of three points and are defined in the template definition files. The mapping is formed by considering the x,y values of template vertices with each of

the x′,y′ values of the corresponding region vertices - this yields two plane equations from which each x′,y′ point can be calculated given any x,y within the template facet, and thus the image data can be mapped from the region sub-image.

The entire template sub-image is obtained by rendering (or scan-converting) each constituent facet pel by pel, taking the pel's value from the corresponding mapped location in the equivalent region's sub-image.

The processor 5 is arranged to perform the mapping of the extracted region sub-images to their corresponding generic template size and shape. The keypoints on the regions form a triangular mesh with a corresponding mesh defined for the generic template shape; mappings are then formed from each triangle in the generic mesh to its equivalent in the region mesh. The distorted sub-images are then created and stored in the template data structures for later display.

The central procedure in this module is the 'template stretching' procedure. This routine creates each distorted template sub-image facet by facet (each facet is defined by three connected template points). A mapping is obtained from each template facet to the corresponding region facet and then the template facet is filled in pel by pel with image data mapped from the region sub-image. After all facets have been processed in this way the distorted template sub-image will have been completely filled in with image data.

The standardised feature image thus produced is then stored in a feature image buffer 7. An eigen-picture buffer 8 which contains a plurality (for example 50) of eigen-pictures of increasing sequency which have previously been derived in known manner from a representative population (preferably using an equivalent geometric normalisation technique to that disclosed above). A transform processor 9 (which may in practice be realised as processor 5 acting under suitable stored instructions) derives the co-ordinates or components of the feature image with regard to each eigen-picture, to give a vector of 50 numbers, using the method described above. The card entry device 2 reads from the inserted credit card the 50 components which characterise the

correct user of that card, which are input to a comparator (which may again in practice be realised as part of a single processing device) which measures the distance in pattern space between the two connectors. The preferred metric is the Euclidian distance, although other distance metrics (eg. "a city block" metric) could equally be used. If this distance is less than a predetermined threshold, correct recognition is indicated to an output 11; otherwise, recognition failure is signalled.

Other data may also be incorporated into the recognition process; for example, data derived during template deformation, or head measurements (e.g. the ratio of head height to head width derived during the head location stage) or the feature position data as mentioned above. Recognition results may be combined in the manner indicated in our earlier application GB9005190.5.

Generally, some preprocessing of the image is provided (indicated schematically as 12 in Figure 2); for example, noise filtering (spatial or temporal) and brightness or contrast normalisation.

Variations in lighting can produce a spatially variant effect on the image brightness due to shadowing by the brows etc. It may be desirable to further pre-process the images to remove most of this variation by using a second derivative operator or morphological filter in place of the raw image data currently used. A blurring filter would probably also be required.

It might also be desirable to reduce the effects of variations in geometric normalisation on the representation vectors. This could be accomplished by using low-pass filtered images throughout which should give more stable representations for recognition purposes.

CLAIMS

1. A method of processing an image comprising the steps of:

locating within the image the position of at least one predetermined feature;

extracting image data from said image representing each said feature; and

calculating for each feature a feature vector representing the position of the image data of the feature in an N-dimensional space, said space being defined by a plurality of reference vectors each of which is an eigenvector of a training set of images of like features;

characterised in that the method further comprises the step of:

modifying the image data of each feature to normalise the shape of each feature thereby to reduce its deviation from a predetermined standard shape of said feature, which step is carried out before calculating the corresponding feature vector.

2. A method according to claim 1, wherein the step of modifying the image data of each feature involves the use of a deformable template technique.

3. A method according to claim 1 or 2, wherein the step of locating within the image the position of at least one predetermined feature employs a first technique to provide a coarse estimation of position and a second, different, technique to improve upon the coarse estimation.

4. A method according to claim 3 wherein the second technique involves the use of a deformable template technique.

5. A method according to any preceeding claim in which the training set of of images of like features are modified to normalise the shape of each of the training set of images thereby to reduce their deviation from a predetermined standard shape of said feature, which step is carried out before calculating the eigenvectors of the training set of images.

6. A method according to any preceding claim comprising locating a portion of the image by determining parameters of a closed curve arranged to lie adjacent a plurality of edge features of the image, said curve being constrained to exceed a minimum curvature and to have a minimum length compatible therewith.

7. A method as claimed in claim 6 in which the boundary of the curve is initially calculated proximate the edges of the image, and subsequently interactively reduced.

8. A method according to either one of claims 6 and 7 in which the portion of the image is a face or a head of a person.

9. A method according to any preceding claim further comprising determining the position of each feature within the image.

10. A method of verifying the identity of the user of a data carrier comprising:

generating a digital facial image of the user;

receiving the data carrier and reading therefrom identification data;

performing the method of any one of claims 1 to 8;

comparing each feature vector, or data derived therefrom, with the identification data; and

generating a verification signal in dependence upon the comparison.

11. Apparatus for verifying the identity of the user of a data carrier comprising:

means for generating a digital facial image of the user;

means for receiving the data carrier and reading therefrom identification data;

means for performing the method of any one of claims 1 to 8; and

means for comparing each feature vector, or data derived therefrom, with the identification data and generating a verification signal in dependence upon the comparison.

12. Apparatus as claimed in claim 11 in which the means for generating a digital facial image of the user comprises a

video camera the output of which is connected to an AD convertor.

Fig. 1.

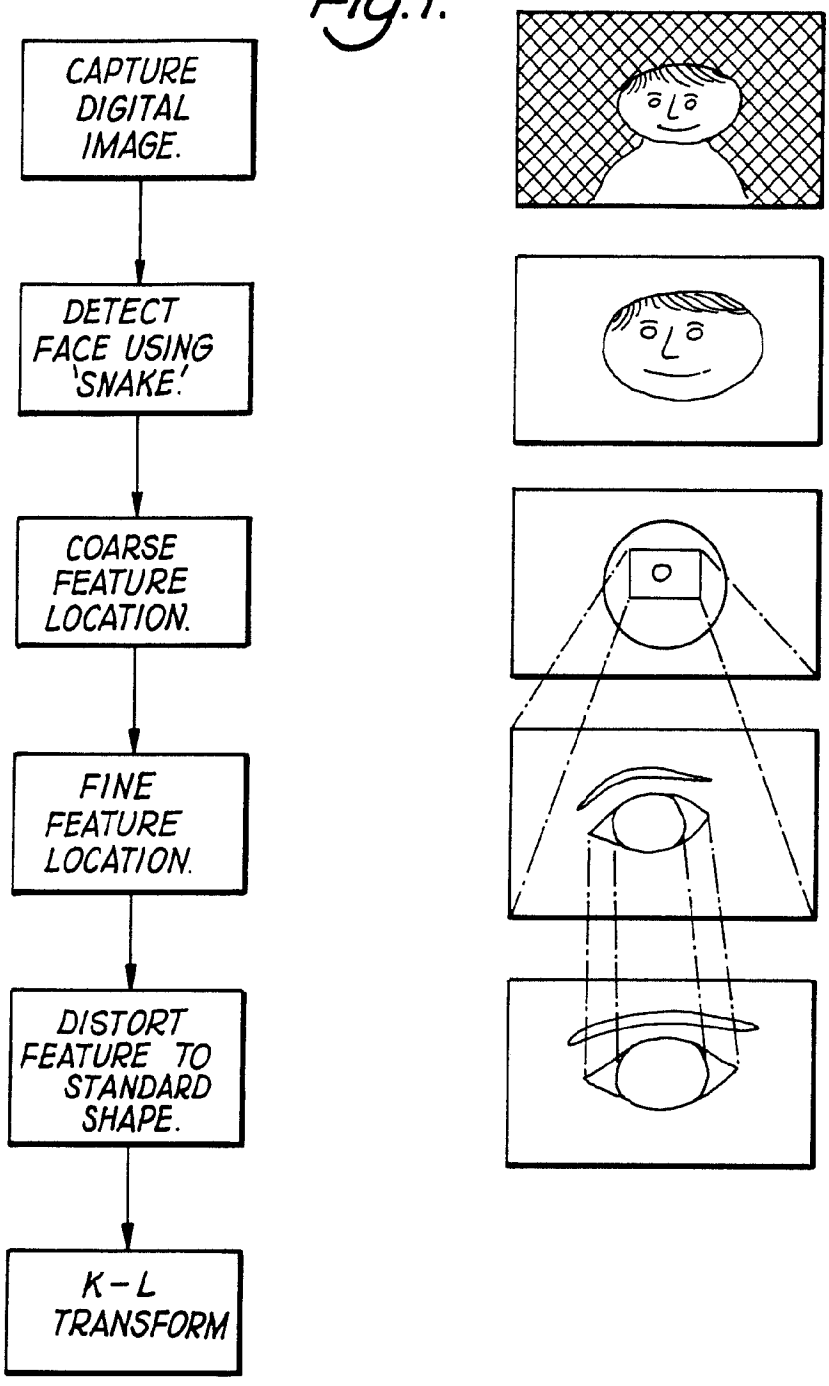

CAPTURE DIGITAL IMAGE.
DETECT FACE USING 'SNAKE.'
COARSE FEATURE LOCATION.
FINE FEATURE LOCATION.
DISTORT FEATURE TO STANDARD SHAPE.
K-L TRANSFORM

Fig. 2.

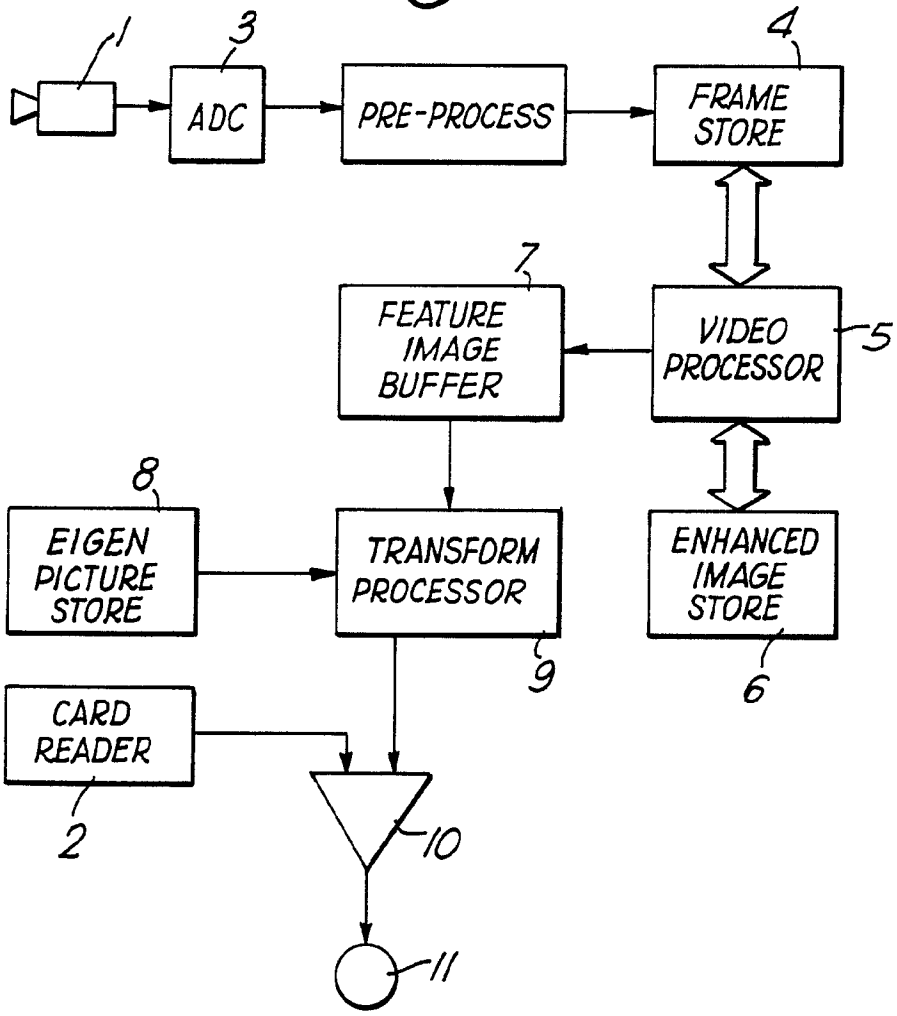
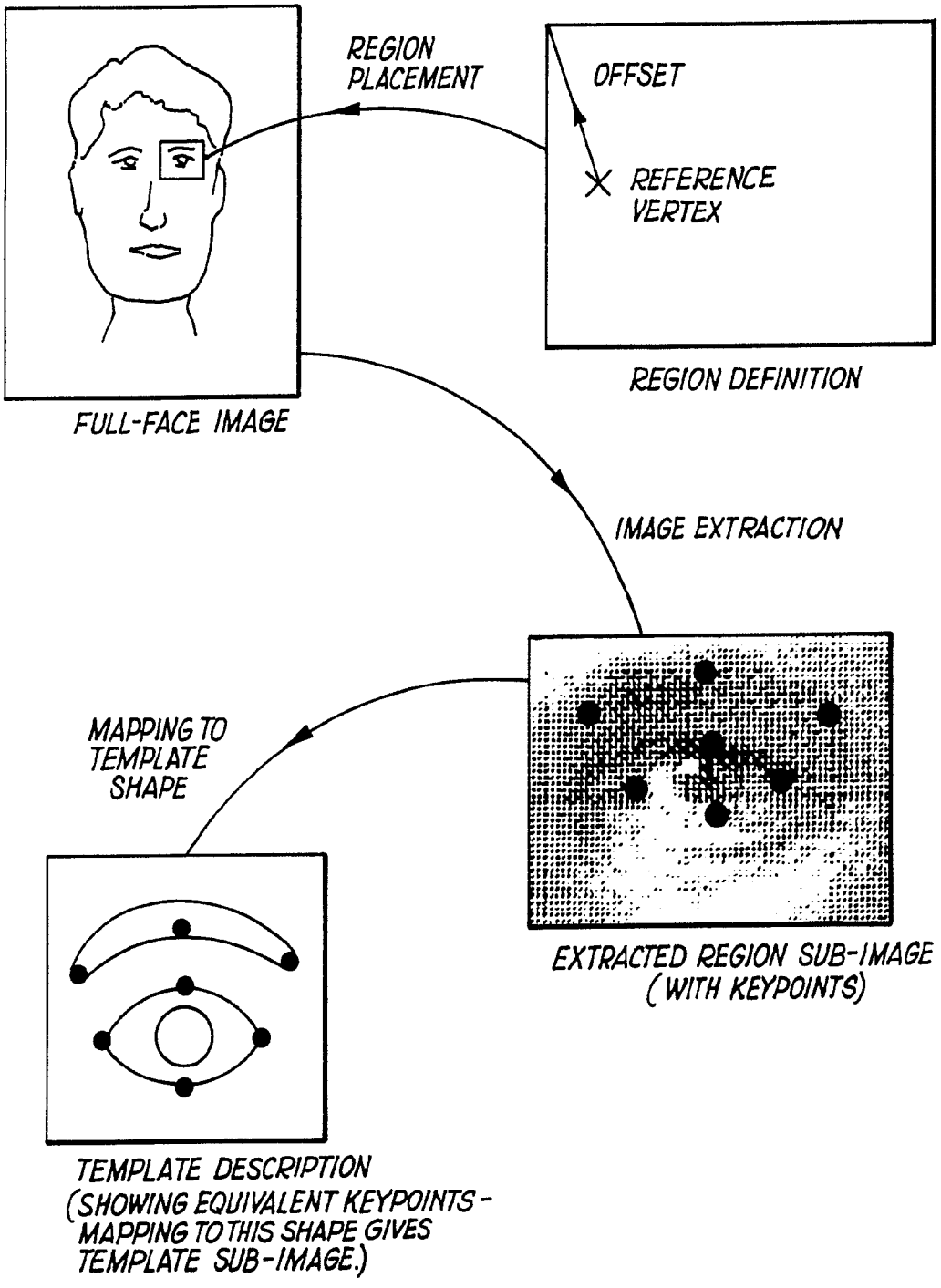

1
3
ADC
PRE-PROCESS
4
FRAME STORE
7
FEATURE IMAGE BUFFER
VIDEO PROCESSOR
5
8
EIGEN PICTURE STORE
TRANSFORM PROCESSOR
9
ENHANCED IMAGE STORE
6
CARD READER
2
10
11

Fig.3.



I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all)[6]

According to International Patent Classification (IPC) or to both National Classification and IPC

Int.Cl. 5 G06K9/46; G06K9/64; G07C9/00

II. FIELDS SEARCHED

Minimum Documentation Searched[7]

Classification System	Classification Symbols
Int.Cl. 5	G06K ; G07C ; G06F

Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched[8]

III. DOCUMENTS CONSIDERED TO BE RELEVANT[9]

Category °	Citation of Document,[11] with indication, where appropriate, of the relevant passages[12]	Relevant to Claim No.[13]
Y	US,A,4 906 940 (GREENE ET AL.) 6 March 1990 see abstract see column 5, line 50 - line 52 see column 8, line 10 - column 15, line 9	1,2,10, 11
Y	IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE vol. 12, no. 1, January 1990, NEW YORK US pages 103 - 108; M. KIRBY ET AL.: 'Application of the Karhoenen-Loève Procedure for the Characterization of Human Faces' cited in the application see page 105, right column, line 48 - page 106, right column, line 5; figures PLATE,3	1,2,10, 11
A		5

-/--

° Special categories of cited documents :[10]

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

IV. CERTIFICATION

Date of the Actual Completion of the International Search	Date of Mailing of this International Search Report
15 NOVEMBER 1991	28. 11. 91
International Searching Authority	**Signature of Authorized Officer**
EUROPEAN PATENT OFFICE	CHATEAU J.P.

Form PCT/ISA/210 (second sheet) (January 1985)

III. DOCUMENTS CONSIDERED TO BE RELEVANT (CONTINUED FROM THE SECOND SHEET)

Category °	Citation of Document, with indication, where appropriate, of the relevant passages	Relevant to Claim No.
Y	US,A,3 805 238 (ROLF ERIC ROTHFJELL) 16 April 1974 see abstract see column 8, line 55 - column 10; figures 1-3	1,2,10, 11
A	US,A,4 644 584 (SUMIO NAGASHIMA ET AL.) 17 February 1987 see abstract	1,2
A	EP,A,0 247 788 (NATIONAL BUSINESS SYSTEMS CORPORATION) 2 December 1987 see column 1 - column 5 see column 7, line 5 - line 7	1,8,10, 11,12
A	WO,A,8 707 058 (COSTELLO,BRENDAN,DAVID) 19 November 1987 see abstract	9

ANNEX TO THE INTERNATIONAL SEARCH REPORT ON INTERNATIONAL PATENT APPLICATION NO. GB 9101183 SA 49469

This annex lists the patent family members relating to the patent documents cited in the above-mentioned international search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information. 15/11/91

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US-A-4906940	06-03-90	None	
US-A-3805238	16-04-74	SE-B- 365325	18-03-74
		AT-B- 321618	10-04-75
		AU-A- 4827972	27-09-73
		BE-A- 790510	15-02-73
		CA-A- 1001761	14-12-76
		CH-A- 560537	15-04-75
		DE-A,B,C 2254597	17-05-73
		FR-A- 2159977	22-06-73
		GB-A- 1403765	28-08-75
		JP-A- 48055638	04-08-73
		LU-A- 66388	05-02-73
		NL-A- 7214882	08-05-73
US-A-4644584	17-02-87	JP-C- 1357926	13-01-87
		JP-A- 58201185	22-11-83
		JP-B- 61020035	20-05-86
		DE-A,C 3318303	24-11-83
EP-A-0247788	02-12-87	AU-B- 586778	20-07-89
		AU-A- 7338887	03-12-87
		JP-A- 63040969	22-02-88
		US-A- 4754487	28-06-88
WO-A-8707058	19-11-87	AU-B- 603801	29-11-90
		AU-A- 7353987	01-12-87
		EP-A- 0310603	12-04-89
		JP-T- 1502368	17-08-89
		US-A- 4947443	07-08-90

EPO FORM P0479

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 3/033, G06K 11/16

A1

(11) International Publication Number: **WO 97/36225**

(43) International Publication Date: 2 October 1997 (02.10.97)

(21) International Application Number: PCT/US97/05333

(22) International Filing Date: 26 March 1997 (26.03.97)

(30) Priority Data:
08/623,483 28 March 1996 (28.03.96) US

(71) Applicant: SYNAPTICS, INC. [US/US]; 2698 Orchard Parkway, San Jose, CA 95134 (US).

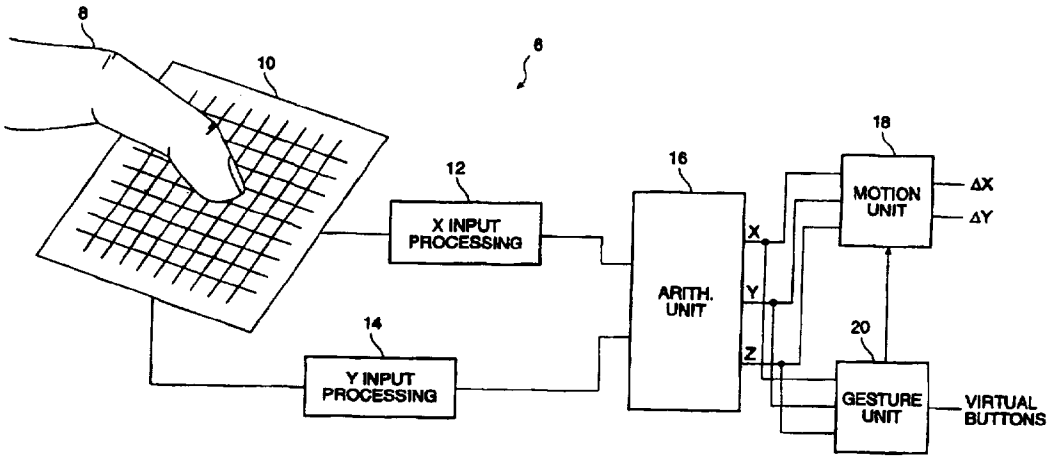
(72) Inventors: GILLESPIE, David, W.; 220 Ventura Avenue #8, Palo Alto, CA 94306 (US). ALLEN, Timothy, P.; 16100 Soda Springs Road, Los Gatos, CA 95030 (US). WOLF, Ralph; 2194 Nobili Avenue, Santa Clara, CA 95051 (US). DAY, Shawn; 379 Sun Ridge Lane, San Jose, CA 95123 (US).

(74) Agents: D'ALESSANDRO, Kenneth et al.; D'Alessandro & Ritchie, P.O. Box 640640, San Jose, CA 95164-0640 (US).

(81) Designated States: CN, JP, KR, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: OBJECT POSITION DETECTOR WITH EDGE MOTION FEATURE AND GESTURE RECOGNITION



(57) Abstract

Methods for recognizing gestures made by a conductive object on a touch-sensor pad and for cursor motion are disclosed. Tapping, drags, pushes, extended drags and variable drags gestures are recognized by analyzing the position, pressure, and movement of the conductive object on the sensor pad during the time of a suspected gesture, and signals are sent to a host indicating the occurrence of these gestures. Signals indicating the position of a conductive object and distinguishing between the peripheral portion and an inner portion of the touch-sensor pad are also sent to the host.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		