KUNGL
TEKNISKA
HÖGSKOLAN

TRITA-NA-D0209 ● CID-172 ● ISSN 1403-0721 ● Department of Numerical Analysis and Computer Science

Computer Vision Based Hand Gesture Interfaces for Human-Computer Interaction

Sören Lenman, Lars Bretzner, Björn Thuresson

CID, CENTRE FOR USER ORIENTED IT DESIGN

Sören Lenman, Lars Bretzner, Björn Thuresson

Computer Vision Based Hand Gesture Interfaces for Human-Computer Interaction
Report number: TRITA-NA-D0209, CID-172
ISSN number: ISSN 1403 - 0721 (print) 1403 - 073 X (Web/PDF)
Publication date: June 2002
E-mail of author: lenman@nada.kth.se, bretzner@nada.kth.se, thure@nada.kth.se

Reports can be ordered from:

CID, Centre for User Oriented IT Design
NADA, Deptartment of Numerical Analysis and Computer Science
KTH (Royal Institute of Technology)
SE-100 44 Stockhom, Sweden
Telephone: + 46 (0) 8 790 91 00
Fax: + 46 (0) 8 790 90 99
E-mail: cid@nada.kth.se
URL: http://cid.nada.kth.se

Computer Vision Based Hand Gesture Interfaces for Human-Computer Interaction

Sören Lenman
Centre for User-Oriented IT-Design
Royal Institute of Technology
100 44 Stockholm, Sweden
lenman@nada.kth.se

Lars Bretzner
Comp. Vision and Active Perc. Lab
Royal Institute of Technology
100 44 Stockholm, Sweden
bretzner@nada.kth.se

Björn Thuresson
Centre for User-Oriented IT-Design
Royal Institute of Technology
100 44 Stockholm, Sweden
thure@nada.kth.se

ABSTRACT
The paper gives an overview of the field of computer vision based hand gesture interfaces for Human-Computer Interaction, and describes the early stages of a project about gestural command sets, an issue that has often been neglected. Currently we have built a first prototype for exploring the use of pie- and marking menus in gesture-based interaction. The purpose is to study if such menus, with practice, could support the development of autonomous gestural command sets. The scenario is remote control of home appliances, such as TV sets and DVD players, which in the future could be extended to the more general scenario of ubiquitous computing in everyday situations. Some early observations are reported, mainly concerning problems with user fatigue and precision of gestures. Future work is discussed, such as introducing flow menus for reducing fatigue, and control menus for continuous control functions. The computer vision algorithms will also have to be developed further.

Keywords
Hand gesture, computer vision, HCI, gesture command, marking menu.

INTRODUCTION
The trend towards embedded, ubiquitous computing in domestic environments creates a need for human-computer interaction forms that are experienced as natural, convenient, and efficient. The traditional desktop paradigm, building on a structured office work situation, and the use of keyboard, mouse and display, is no longer appropriate. Instead, natural actions in human-to-human communication, such as speak and gesture, seem more appropriate for what Abowd and Mynatt [1] have named *everyday computing*, and which should support the informal and unstructured activities of everyday life. Interaction in these situations implies that it should not be necessary to carry any equipment or to be in a specific location, e.g., at a desk in front of a screen. Interfaces based on computational perception and computer vision should be appropriate for accomplishing the goals of ubiquitous, everyday computing.

This paper presents an overview of the field of gesture-based interfaces in human-computer interaction as a background, and the first stages of a project concerning the development of such interfaces. Specifically, in the project we intend to study the use of *hand gestures* for interaction, in an approach based on computer vision. As a starting point, remote control of electronic appliances in a home environment, such as TV sets and DVD players, was chosen. This is an existing, common interaction situation, familiar to most. Normally it requires the use of a number of devices, which can be a nuisance, and there are clear benefits to an appliance-free approach. In the future the application could easily be extended to a more general scenario of ubiquitous computing in everyday situations. Currently we have implemented a first prototype for exploring the use of pie- and marking menus [9], [20] for gesture-based interaction. Our main purpose is not menu-based interaction, but to study if such menus, with practice, could support the development of an autonomous gestural command sets. The application will be described in more detail later in this paper.

Perceptive and Multimodal User Interfaces
Two main scenarios for gestural interfaces can be distinguished. One aims at developing *Perceptive User Interfaces* (PUI), as described by Turk [36], or *Perceptive Spaces* , e.g., Wren [42], striving for automatic recognition of natural, human gestures integrated with other human expressions, such as body movements, gaze, facial expression, and speech. The aim is to develop conversational interfaces, based on what is considered to be natural human-to-human dialog. For example, Bolt [4] suggested that in order to realize conversational computer interfaces, gesture recognition will have to pick up on unintended gestures, and interpret fidgeting and other body language signs, and Wexelblatt [41] argued that only the use of natural hand gestures is motivated, and that there might even be added cognitive load on the user by using gestures in any other way.

However, in this paper the focus is on using hand gestures given purposefully as instructions, and we restrict our work to deliberate, expressive movements. This falls within the second approach to gestural interfaces, *Multimodal User Interfaces,* where hand poses and specific gestures are used

as commands in a command language. The gestures need not be natural gestures but could be developed for the situation, or based on a standard sign language. In this approach, gestures are either a replacement for other interaction tools, such as remote controls and mice, or a complement, e.g., gestures used with speech and gaze input in a multimodal interface. Oviatt et al. [27] noted that there is a growing interest in designing multimodal interfaces that incorporate vision-based technologies. They also contrast the passive mode of PUI with the active input mode, addressed here, and claim that although passive modes may be less obtrusive, active modes generally are more reliable indicators of user intent, and not as prone to error.

Hand Gestures for Computer Vision

Detailed descriptions and taxonomies concerning hand gestures from the point of view of computer vision can be found in Quek [30], Pavlovic & Sharma [28] and Turk [36]. Here only a brief overview will be presented.

Gestures are expressive, meaningful body motions with the intent to convey information or interact with the environment [36]. According to Cadoz [8] hand gestures serve three functional roles, *semiotic*, *ergotic*, and *epistemic*. The *semiotic* function is to communicate information, the *ergotic* function corresponds to the capacity to manipulate objects in the real world, and the *epistemic* function allows us to learn from the environment through tactile experience. Based on this classification Quek [30] distinguishes *communicative gestures*, which are meant for visual interpretation and where no hidden part carries information critical to understanding, from *manipulative gestures*, which show no such constraints. Thus, it may be more appropriate to use special tools for interaction, like data gloves, rather than computer vision if the intent is realistic manipulation of objects in, e.g., a virtual environment. Pavlovic et al. [28] makes a similar classification, but also point out the distinction between *unintentional movements* and *gestures*.

For communicative, semiotic gestures, Kendon [14] distinguishes *gesticulation*, gestures that accompany speech, from *autonomous gestures.* These can be of four different kinds: *language-like gestures*, *pantomimes*, *emblems*, and *sign language*s. When moving forward in this list the association with speech diminishes, language properties increase, spontaneity decreases and social regulation increases.

Most work in computer vision and HCI has focused on emblems and signs because they carry more clear semantic meaning, and may be more appropriate for command and control interaction [37]. It is important to note, however, that they are largely symbolic, arbitrary in nature, and that universally understandable gestures of this kind hardly exist.

There is also one important exception worth mentioning. In the gesticulation category, McNeill [24] defines *deictic gestures* as pointing gestures that refer to people, objects, or events in space and time. Deictic gestures are potentially useful for all kinds of selections in human-computer interaction, as illustrated, e.g., by the early work of Bolt [4]. The deictic category itself can be further subdivided, but from a computer vision point of view all deictic gestures are performed as pointing, and the difference lies in the higher level of interpretation [30].

In the following we limit ourselves to *intentional, semiotic, hand gestures*. From a computer vision point of view, we focus on the recognition of static postures and gestures involving movements of fingers, hands and arm with the intent to convey information to the environment.

Gesture-Based Applications in HCI

Pavlovic [28] noted that, ideally, naturalness of the interface requires that any and every gesture performed by the user should be interpretable, but that the state of the art in vision-based gesture recognition is far from providing a satisfactory solution to this problem. A major reason obviously is the complexity associated with the analysis and recognition of gestures. A number of pragmatic solutions to gesture input in HCI exist, however, such as:

- use props or input devices (e.g., pen, or data glove)
- restrict the object information (e.g., silhouette of the hand)
- restrict the recognition situation (uniform background, restricted area)
- restrict the set of gestures

In traditional HCI, most attempts have used some device, such as an instrumented glove, for incorporating gestures into the interface. If the goal is natural interaction in everyday situations this might not be acceptable. However, a number of applications of hand gesture recognition for HCI exist, using the untethered, unencumbered approach of computer vision. Mostly they require restricted backgrounds and camera positions, and a small set of gestures, performed with one hand. They can be classified as applications for *pointing*, *presenting*, *digital desktops*, and *virtual workbenches and VR.*

Pointing: A number of applications that use computer vision for pointing (deictic) gestures have been developed, either in a scenario for some special kind of interaction situation, such as Put-That-There [4], or, as a replacement for some input device in general, mostly the mouse. An example is *Finger Mouse* [31], where a down-looking camera was used to create a virtual 2D mousepad above the keyboard, allowing users to perform pointing gestures to control the cursor. Mouse clicks were implemented by pressing the shift key. Kjeldsen and Kender [16] used a camera position below the screen, facing the user, to compute the x,y coordinates that control the cursor. For window control they used a neural network to classify hand poses (point, grasp, move, menu) with a simple grammar, based on pausing and retraction. They note that users had difficulties to remember the sequence of motions and poses and that there were unexpected interface actions, because gestures were dependent on timing. O´Hagan [25] used a

commercial system with a single video camera for *Finger Track*, which performed vision-based finger tracking on top of the workspace. A pointing gesture (one finger) and a click gesture (two-fingers extended) could be used. A similar application, *FingerMouse* [sic!] for controlling the mouse pointer was presented by von Hardenberg and Berard [39]. The finger, moving over a virtual touchscreen, is used as mouse and selection is indicated by a one sec delay in the gesture.

Presenting: Baudel et al. [2] used a glove-based system for controlling Microsoft PowerPoint-presentations. Even if the focus in this paper is on computer vision, their work should be mentioned, because it addresses the question of developing gestural command sets. They suggest that command gestures should be defined according to an articulatory scheme with a tense start position (e.g. all fingers outstretched), a relaxed dynamic phase (e.g. a hand movement to the right) and a tense end position (e.g. all fingers bent). In a similar application, based on computer vision, Lee & Kim [21] use hand movements for controlling presentations. The detection of the hand is entirely based on skin color, which requires a controlled background. The gesture-based virtual touchscreen of von Hardenberg et al. [39] included command gestures for slide changes and menu selection, in addition to general pointing gestures (see above). Hand detection relies on a time filtered background subtraction, i.e., it requires a reference image. In a more advanced multimodal scenario, Kettebekov and Sharma [15] performed an observational study to develop a gesture grammar for deictic gestures when presenting a weather map.

Digital Desks: A third kind of application aims at developing mixed reality desktops, using free hand pointing and manipulation of digital objects. Kruegers *VideoDesk* [19] was an early desk-based system in which an overhead camera and a horizontal light was used to provide hand gesture input for interactions, which were then displayed on a monitor at the far end of the desk. The work was built on the early research of the *VideoPlace* system [18]. Wellner [40] developed *DigitalDesk*, a more advanced digital desk system, mixing projected and electronic documents on a real desktop, and using an image processing system to determine the position of the users' hands, and to gather information from documents placed on the desk. Similarly, Maggioni and Kämmerer [23] explored pointing gestures in vision-based virtual touchscreens for office applications, public information terminals and medical applications. The detection is based on a skin segmentation step, and the approach requires controlled backgrounds. More recently, Koike et al. [17] developed an augmented desk interface, *EnhancedDesk*, with computer vision as a key technology. *EnhancedDesk* uses a projector for presenting information onto a physical desktop, an infrared camera for detecting users arms, hands, and hand poses, and a pan-tilt camera for giving detail. Users can manipulate digital information directly by using their hands and fingers. The system is reported to be able to track fingertip movements in real time under any lighting condition.

Virtual workbenches and VR: The distinction between virtual workbenches and digital desktops is not sharp. Here, a workbench is described as primarily intended for navigation and object manipulation in 3D environments. As mentioned earlier, computer vision might not be suitable for these tasks. Glove-based input might be better suited for intricate 3D manipulation tasks, due to the problem of occluded fingers. Recently, however, Utsumi and Ohya [38] proposed a multiple-viewpoint system for three-dimensional tracking of position, pose and shapes of human hands, as a step towards replacing glove-based input. Also, many gestures for navigation and object manipulation in virtual environments have a deictic component, i.e., are pointing gestures, which simplifies the problem from a computer vision point of view.

Segen and Kumar [33] investigated a vision-based system for 3D navigation, object manipulation and visualization. The system used stereo cameras against a plain background and with stable illumination, and has been used for movement control in a 3D virtual environment, for building 3D scenes, and for a 2D game. Fatigue is reported as an issue, especially when the system is used for object manipulation. Leibe et al. [22] experimented with 3D terrain navigation, games, and CSCW, using a *FakeSpace* immersive workbench with infrared illuminators placed next to the camera. IR light is reflected back to the camera by objects placed on the desk. A second IR camera provides a side view of users arms for recovering 3D pointing gestures. O'Hagan et al. [26] implemented a virtual, 3D workbench where two cameras were used to provide stereo images of the users' hand. As with Segen [33], the system could be used for object and scene translations, rotations, object resizing, and zoom. By combining feature-based tracking with a model-based system, tracking with cluttered backgrounds and changing illumination is claimed to be possible. O'Hagan et al. also point out user fatigue as a problem in this kind of application. Other examples of 3D object manipulation and navigation can be found in Sato et al. [32] and Bretzner and Lindeberg [6].

Finally, the work of Wren et al. regarding perceptive rooms and spaces [42] should be mentioned in this context, even if it might rather be characterized as an attempt at mixed reality, multimodality and ubiquitous computing in a PUI scenario. An interactive space is created in a room with constant lighting, controlled background, and a large projection screen. Stereo computer vision is used to track key features of body, hand and head motion. The authors point out that the possibility for users to enter the virtual environment just by stepping into the sensing area is very important, not having to spend time donning equipment. Also, the importance of social context is noted. Not only can the user see and hear a bystander, the bystander can easily take the users place for a few seconds, without any need to "suit up", as is the case with most scenarios requiring equipment.

CURRENT WORK

With the exception of Baudel et al. [2], very little attention has been paid to the selection of gestures in gesture-based interaction, and to the development on gestural command sets. Often the reason is that the gestures are deictic. However, even under circumstances when they are not, there has not been much discussion about what gestures or hand poses should be used.

Gestural Command Sets

The design space for gestural commands can be characterized along three dimensions: *Cognitive aspects, Articulatory aspects* and *Technological aspects.*

Cognitive aspects refer to how easy commands are to learn and to remember. It is often claimed that gestural command sets should be natural and intuitive, e.g. [4] [41], mostly meaning that they should inherently make sense to the user. This might be possible for manipulative gestures, but, as noted above, for communicative gestures there might not exist any shared stereotypes to build on, except in very specific situations. If the aim is gestural control of devices, there is no cultural or other context for most functions. Baudel et al. [2] recommend that ease of learning should be favored and that a compromise must be made between natural gestures that are immediately assimilated by the user and complex gestures that give more control. They define "natural gestures" as those that involve the least effort and differ the least from a rest position, i.e., that "naturalness" in part should be based on an articulatory component, according to the classification used here.

Articulatory aspects refer to how easy gestures are to perform, and how tiring they are for the user. Gestures involving complicated hand or finger poses should be avoided, because they are difficult to articulate and might even be impossible to perform for a substantial part of the population. They are common in current computer based approaches, because they are easy to recognize by computer vision. Repetitive gestures that require the arm to be held up and moved without support are also unsuitable from an articulatory point of view because of fatigue.

Technological aspects refer to the fact that in order to be appropriate for practical use, and not only in visionary scenarios and controlled laboratory situations, a command set for gestural interaction based on computer vision must take into account the state-of-the art of technology, now and in the near future. For example, Sign Language recognition might be desirable for a number of reasons, not least for people who need to use Sign Language for communication. Although difficult to learn, once learned a Sign Language is easy to remember because of its language properties, and might provide a good candidate framework for developing gestural languages for interaction. Some attempts to Sign Language recognition also exist. For example, recently Starner et al. [34] developed a recognition system for a subset of American Sign Language. However, Braffort [5] points out that if the real aim is to deal with Sign Language, then all the different varied and complex elements of language must be taken into account. This is currently far from feasible. Still, much work can be done with reduced sets of Sign Language, limited to standard signs, as a first step towards a long-term objective.

Menu-based Systems for Gesture-Based Interaction: Our current work represents the first stages in a research effort about computer vision based gesture interaction, primarily aimed at questions concerning gesture command sets. The point of departure is *cognitive*, leaving articulatory aspects aside for the moment, mainly for reasons of technical feasibility. We focus on the fact that the learning curve for a gestural interface of any complexity will be steeper than for a menu-based interface, because commands need to be recalled, rather than recognized. As noted earlier, there are very few natural, generally understandable signs and gestures that could be used. And, however desirable it might be to use some standard Sign Language it is not technically feasible, except at the level of isolated signs. Using signs from Sign Language, if not the language itself, will be addressed in this project in the future. Currently gestures and hand poses are kept simple, for technical reasons and for reasons of articulatory simplicity.

As was mentioned above, menu-based systems have the cognitive advantage that commands can be recognized rather than recalled. Traditional menu-based interaction, however, is not attractive in a gesture-based scenario for everyday situations. Menu navigation would be far from the directness that gestural interaction could provide. However, by using pie- and marking menus, it might be possible to support directness, and to provide a solution for developing gestural command sets.

Pie- and Marking Menus: Pie menus were first described by Callahan et al. [9]. They are pop-up menus with the alternatives arranged radially. Because the gesture to select an item is directional, users can learn to make selections without looking at the menu. In principle this could be learned also with linear menus, but it is much easier to move the hand without feedback in a given direction, as with a pie menu, than to a menu item at a given distance, as in a linear menu. This fact can support a smooth transition between novice and expert use. For an expert user, working at high speed, menus need not even be popped up. The direction of the gesture is sufficient to recognize the selection. If the user hesitates at some point in the interaction, the underlying menus could be popped up, always giving the opportunity to get feedback about the current selection. Hierarchic marking menus [20] is a development of pie menus that allow more complex choices by the use of submenus. The same principles apply: expert users could work by gesture alone, without feedback. The shape of the gesture with its movements and turns can be recognized as a selection, instead of the sequence of distinct choices between alternatives. A recent example can be found in Beaudouin-Lafon et al. [3].

Hierarchic Marking Menus for Gesture-Based Interaction: Here the assumption is that command sets for computer

vision based gesture interfaces can be created from hierarchical marking menus. As to articulatory characteristics, a certain hand pose, e.g., holding the hand up with all fingers outstretched, could be used for initiating a gesture and activating the menu system. This would correspond to the pen-down event in a pen-based system. The gesture could then be tracked by the computer vision algorithms, as the hand traverses the menu hierarchy. Finally, a certain hand pose could be used to actually make the selection, e.g., the index finger and thumb outstretched, corresponding to a pen-up event in pen-based interface. Put differently, the gestures in the command set would consist of *a start pose*, a *trajectory*, defined by menu organization, for each possible selection, and, lastly, a *selection pose.* Gestures ending in any other way than with the selection pose would be discarded, because either they could mean that the user abandoned the gesture, or simply that tracking of the hand was lost.

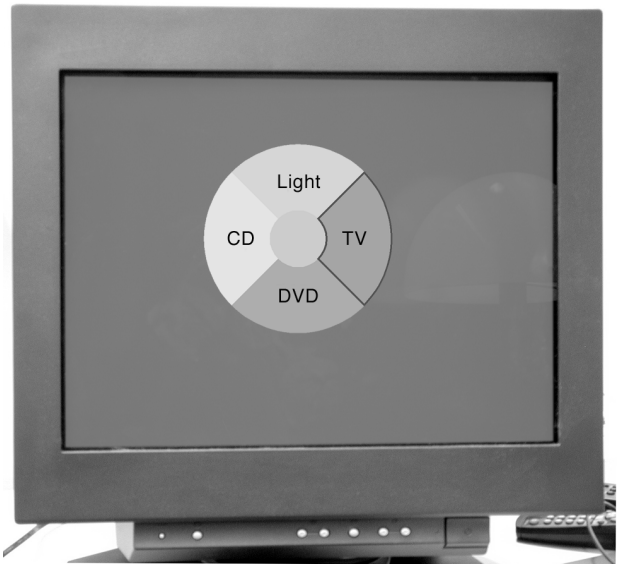

Fig. 1 An example of a pie menu in the prototype.

For a novice user, this would amount to a traditional menu-selection task, where selections are made by navigating through an hierarchical menu structure. This, as such, could provide for unencumbered interaction in remote control situations but, as noted above, the directness of a gesture-interface would be lost. The assumption here, however, is that over time users will learn the gesture corresponding to each selection and no longer need visual feedback. The interaction would develop into direct communication, using a gestural language. In addition to providing for a natural transition from novice to expert, such a gestural language makes no assumptions about naturalness or semantics of gestures, because it is defined by the menu structure. In principle, if not in practice, the command set is unlimited. A further advantage is that the demands put on the computer vision algorithms are reasonable. Fast and stable tracking of the hand will be required, however.

A PROTOTYPE FOR HAND GESTURE INTERACTION

The prototyping and experimental work is still in an early stage and only a brief overview and some early impressions can be given here. Inspired by Freeman et al. [11], [12], we chose remote control of appliances in a domestic environment as our first application. Freeman et al. used only one gesture to control a TV set: an open hand facing the camera. An icon on a computer display followed the users hand, and by moving the icon (hand) along one of two sliders, a user could control the volume or select channels. Our prototype is more intricate and intended to test the hypothesis, discussed above, that hierarchical marking menus can be used to develop gestural command sets. However, so far, we have only designed a first example of a hierarchic menu system for controlling some functions of a TV, a CD player, and a lamp. The prototype has been set up in a generally accessible, open lab/demo space at CID (fig. 2).

Fig. 2 The demo space at CID.

Technical Aspects

The Computer Vision System: We have chosen a view-based representation of the hand, including both color and shape cues. The system tracks and recognizes the hand poses based on a combination of multi-scale color feature detection, view-based hierarchical hand models and particle filtering. The hand poses, or hand states, are represented in terms of hierarchies of color image features at different scales, with qualitative inter-relations in terms of scale, position and orientation. These hierarchical models capture the coarse shape of the hand poses. In each image, detection of multi-scale color features is performed. The hand states are then simultaneously detected and tracked using particle filtering, with an extension of layered sampling referred to as hierarchical layered sampling. The particle filtering allows for the evaluation of multiple hypotheses about the hand position, state, orientation and scale, and a likelihood measure determines what hypothesis to chose. To improve the performance of the system, a prior on skin color is included in the particle filtering step. In fig. 3, yellow (white) ellipses show detected multi-scale features in a

complex scene and the correctly detected and recognized hand pose is superimposed in red (gray). A detailed description of the algorithms is given in [7].

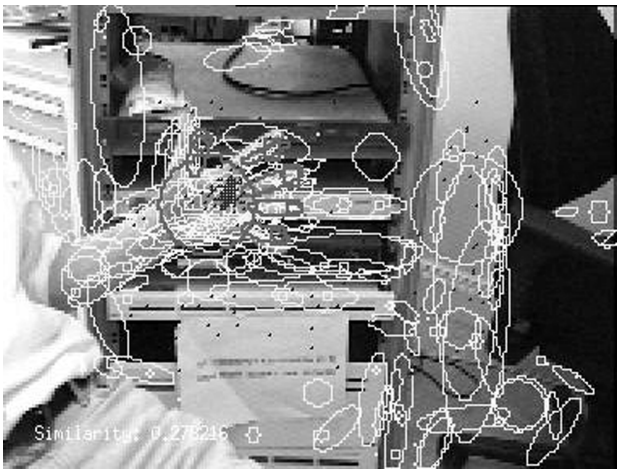

Fig. 3 Detected multi-scale features and the recognized hand pose superimposed in an image of a complex scene.

As the coarse shape of the hand is represented in the feature hierarchy, the system is able to reject other skin colored objects that can be expected in the image (the face, arm, etc). The hierarchical representation can easily be further extended to achieve higher discrimination to complex backgrounds, at the cost of a higher computational complexity. An advantage of the approach is that it is to a large extent user and scale (distance) invariant. To some extent, the chosen qualitative feature hierarchy also shows view invariance for rotations out of the image plane (up to approx. 20-30 degrees for the chosen gestures).

There is a large number of works on real-time hand pose recognition in the computer vision literature. Some of the most related to our approach are, e.g., Freeman and Weissman [11] (see above) who used normalized correlation of template images of hands for hand pose recognition. Though efficient, this technique can be expected to be more sensitive to different users, deformations of the pose and changes in view, scale, and background. Cui and Weng [10] showed promising results for hand pose recognition using an appearance based method. However, the performance was far from real-time. The approach closest to ours was presented by Triesch and von der Malsburg [35] representing the poses as elastic graphs with local jets of Gabor filters computed at each vertex.

Equipment: A Dell Workstation 530 with dual 1,7 GHz Intel Xeon P4 processors running Red Hat Linux was used. The menus were shown on a 19" Trinitron monitor, placed next to the TV screen. The menu system was developed in Smalltalk. An Mvdelta 2 framegrabber, IRdeo remote IR control, and a DI-01 Data interface (X10) was used for image acquisition and to control a table lamp, a Samsung 29" TV, and a Hitachi CD player. In order to maximize speed and accuracy, gesture recognition is currently tuned to work against a uniform background within a limited area, approximately 0,5 by 0,65 m in size, at a distance of approximately 3 m from the camera, and under relatively fixed lighting conditions**.**

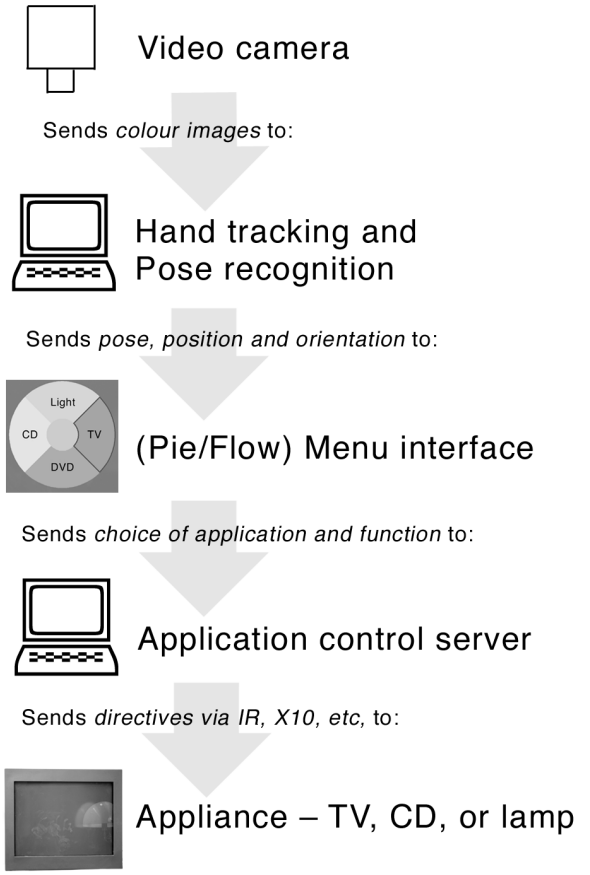

Fig. 4 An overview of the functional components and the information flow in the prototype.

Menu System

An overview of the functional components and the information flow in the prototype is presented in fig. 4 above. We have only recently begun working on the design, the arrangement, and the organization of the menus. An incomplete version with three hierarchical levels and four choices in each menu currently exists. Only a few of choices are active, however: *TV on/off*, *Previous/Next channel*, *CD Play/Stop/Back/Forward*, *Lamp on/off*. An example of a menu is shown in fig. 1.

A hand pose with the index finger and thumb outstretched is used as the start pose for activating the menus, corresponding to pen-down in a pen-based interface. A hand with five fingers outstretched is used as the selection pose, corresponding to pen-up. Evidently, any two hand poses could be used for these purposes. Menus are activated when

the start hand pose is detected by the computer vision system in the active area. The hand is tracked as long as the start pose is held. If the hand is moved over the periphery of a sector that has a submenu, the parent menu disappears, and the submenu appears. Showing the selection hand pose in an active field, e.g., *TV on*, makes a selection. All other ways of ending the interaction are ignored. The menus are currently shown on a computer screen, placed by the side of the TV (fig. 2). This is inconvenient, and in the future menus will be presented in an overlay on the TV screen.

Results and Discussion

Only a small number of informal user trials have been performed so far. From these it is obvious that a menu-based system requires some instructions to get people started. This was different from an earlier prototype, where we used four static hand poses for direct control, which was very easy to understand. Menu-based systems are more complex, and there is simply more to learn at the outset. However, learning the principles for using the menus was not a main issue, and the principles are the same no matter the number of choices in the menu system. There are major drawbacks with using static hand poses for direct control as in the earlier prototype. First, the number of usable poses is limited. Second, many people have difficulties using finger poses. Third, the association of poses to functions is arbitrary, and difficult to remember. There are also culturally specific hand poses (emblems) that have to be avoided.

We have not yet been able to bring the technical performance (speed and accuracy) of the menu-based system to a level where true gesture-based control without feedback can be accomplished. However, observations with the current system, as it is, indicate that gesture-based control with simple, single-level pie menus is feasible, but that gestures based on hierarchical menus create some problems. It is difficult for users to make the gestures for multiple-level selections sufficiently distinct, based on feedback only from the proprioceptive system of the arm. Thus, computer algorithms for recognition of fuzzy gestures might also be required. Another solution could be to reduce the number of choices at each level.

The current setup, with subjects seated facing the TV and making gestures with one arm and hand held out by the side of the body without support, is not suitable from an articulatory point of view. It is inconvenient and fatigue quickly sets in. This is also a consequence of the fact that gestures have to cover a relatively large area if the hierarchy is deep. Also, the gesture might end up outside of the recognition area. The problem of fatigue is known from earlier attempts with gesture-based interfaces and must be addressed. In the current application much could be gained by providing support for the arm, by making gestures smaller, and by making the recognition system more tolerant as to the whereabouts of the user and the hand.

Future Work

As to the computer vision algorithms there is ongoing work to increase the speed and performance of the system, to acquire more position independence for recognition of gestures, to increase the tolerance for varying lighting conditions, and to increase recognition performance with complex backgrounds. The main effort, however, is currently aimed at the design and organization of menus. Recently we have begun development of Flow Menus, a version of hierarchical marking menus in which successive levels of the hierarchy are shown in the same position [13]. In our application this would greatly reduce the area which the gestures have to cover when the hierarchy is deep. An additional problem we faced is that not all kinds of functions, e.g., increasing sound volume, are suitable for standard pie menus. Thus, we are working on including a version of control menus [29] into the hierarchy. With control menus, repeated control signals are sent as long as the hand is kept within the menu item in a selection pose.

We are also considering a different scenario in which a few gestures (hand poses or deictic gestures) are used for direct control of common functions, such as controlling the sound level or lighting, and menu-based gestures are used for more complex selections. In this situation it seems attractive to investigate if signs from Sign Language could be used for the static hand poses and poses for menu control.

ACKNOWLEDGMENTS

We thank Björn Eiderbäck, CID, who performed the Smalltalk programming for the menu system. Olle Sundblad, CID did Java programming for the Application control server.

REFERENCES

[1] Abowd, G.D. & Mynatt, E.D. (2000) Charting Past, Present, and Future Research in Ubiquitous Computing. *ACM ToCHI*, Vol. 7, No. 1, pp. 29-58

[2] Baudel, T. & Beaudouin-Lafon, M. Charade: (1993) *Communications of the ACM*, vol 36, no. 7, pp. 28-35.

[3] Beaudoin-Lafon, M., Mackay, W, Andersen, P. Janecek, P. Jensen, M., Lassen, M., Lund, K. Mortensen, K., Munck, S, Ravn, K. Ratzer, A. Christensen, S& Jensen, K. (2001) CPN/Tools: Revisiting the Desktop Metaphor with Post-WIMP Interaction Techniques. *Extended Abstracts from CHI2001*, pp. 11-12

[4] Bolt, R.A. (1980) Put-that-there: Voice and Gesture in the graphics interface. *Computer Graphics*, 14(3), pp. 262-270.

[5] Braffort, A. (2001) Research on Computer Science and Sign Language: Ethical Aspects. In Roy, D. & Panayi, M. (Eds.) *Lecture Notes in Artificial Intelligence*, Springer-Verlag

[6] Bretzner, L. and Lindeberg, T. (1998) Use Your Hand as a 3-D Mouse or Relative Orientation from Extended Sequences of Sparse Point and Line Correspondances Using the Affine Trifocal Tensor. In Burkhardt, H. and Neumann, B. (eds*.), Proc. 5th European Conference on Computer Vision*, Vol. 1406 pp. 141-157. Berlin: Springer Verlag.

[7] Bretzner, L., Laptev, I., & Lindeberg, T. (2002) Hand Gesture Recognition using Multi-Scale Colour Features, Hierarchical Models and Particle Filtering. To appear in the *5th International Conference on Automatic Face and Gesture Recognition*, Washington, D.C:, May 2002.

[8] Cadoz, C. (1994) *Les Réalites Virtuelles*. Dominos, Flammarion.

[9] Callahan, J., Hopkins, D, Weiser, M. & Shneiderman, B. (1988) An Empirical Comparision of Pie vs. Linear Menus. *Proceedings of CHI'88*, pp. 95-100.

[10] Y. Cui and J. Weng. (1996) Hand sign recognition from intensity image sequences with complex background. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 88-93.

[11] Freeman, W.T. & Weissman, C.D. (1994) Television Control by Hand Gestures. In *1st Intl. Conf. on Automatic Face and Gesture Recognition*.

[12] Freeman,W.T. Anderson, D. Beardsley, P. Dodge, C. Kage, H. Kyuma, K. Miyake, Y., Roth, M. Tanaka, K. Weissman, C. Yerazunis, W. (1998) *Computer Vision for Interactive Computer Graphics. IEEE Computer Graphics and Applications*, May-June, pp. 42-53.

[13] Guimbretière, F. & Winograd, T. (2000) FlowMenu: combining Command, Text and Data Entry. *Proceedings of UIST'2000*, pp. 213-216.

[14] Kendon, A. (1986) Current issues in the study of gesture. In Nespoulous, J.-L., Peron, P. & Lecours, A.R. (Eds.) *The biological Foundation of Gestures: Motor and Semiotic Aspects*. pp. 23-47. Lawrence – Erlbaum

[15] Kettebekov, S & Sharma, R. (2001) Toward Natural Gesture/Speech Control of a Large Display. In *Proceedings of EHCT'01. Lecture Notes in Computer Science*, Springer Verlag.

[16] R. Kjeldsen and J. Kender. (1996) Toward the use of gesture in traditional user interfaces. In *Proc. of IEEE International Conference on Automatic Face and Gesture Recognition*, pages 151-156.

[17] Koike, H. Sato, Y. & Kobauashi, Y. (2001) Integrating Paper and Digital Information on EnhancedDesk: A Method for Realtime Finger Tracking on an Augmented Desk System. *ACM ToCHI*, Vol 8, no. 4, pp.307-322

[18] Krueger, M. W., Gionfriddo, T., and Hinrichsen, K. (1985) Videoplace – an artificial reality. In *Proceedings of CHI'85*, pages 35-40.

[19] Krueger, M. (1991) *Artificial Reality II*. Addison-Wesley.

[20] Kurtenbach, G. & Buxton, W. (1994) The Limits of Expert Performance Using Hierarchic Marking Menus. *Proceedings of CHI'94*, pp. 482-487.

[21] Lee, H.-K. and Kim, J. H. (1999) An HMM-based threshold model approach for gesture recognition. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol 21, no 10.

[22] Leibe, B, Minnen, D., Weeks, J. & Starner, T. (2001) Integration of Wireless Gesture Tracking, Object tracking and 3D Reconstruction in the Perceptive Workbench. *Proceedings of 2nd International Workshop on Computer Vision Systems (ICVS 2001)*, Vancouver, BC, Canada, July 2001.

[23] Maggioni, C. and Kämmerer, B. (1998) Gesture Computer - History, Design and Applications. In Cipolla and Pentland (eds) *Computer Vision for Human-Computer Interaction*, Cambridge University Press, 1998. pp. 23-51

[24] McNeill, D. (1985) So you think gestures are nonverbal? *Psychological Review*, vol 92 (3), 350-373.

[25] O'Hagan, R. & Zelinsky, A. (1997) Finger Track – A Robust and Real-Time Gesture Interface. *Australian Joint Conference on AI*, Perth.

[26] O´Hagan, R.G., Zelinsky, A. & Rougeaux, S. (2002) Visual Gesture Interfaces for Virtual Environments. *Interacting with Computers*, Vol. 14, Nr 3, pp. 231-250

[27] Oviatt,, S., Cohen, P, Wu, L. Vergo, J, Duncan, L. Suhm, B, Bers, J, Holzman, T, Winograd, T. Landay, J. Larson, J. Ferro, D. (2000) Designing the User Interface for Multimodal Speech and Pen-Based Gesture Applications: State-of-the-Art Systems and Future Research Directions. *Human-Computer Interaction*, Vol 15, pp. 263-322

[28] Pavlovic, V.I., Sharma, R. & Huang, T.S. (1997) Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 19 (7) 677-695.

[29] Pook, S., Lecolinet, E., Vaysseix, G. & Barillot, E. (2000) Control Menus: Execution and Control in a Single Interactor. In *Extended Abstracts of CHI2000*, pp. 263-264.

[30] Quek, F. (1995) Eyes in the Interface. *International Journal of Image and Vision Computing*, vol 13, no 6, pp. 511-525

[31] Quek, F., Mysliwiec, T., and Zhao, M. (1995). FingerMouse: A Freehand Pointing Interface," *Proceedings of the International Workshop on Automatic Face- and Gesture-Recognition*, Zurich, Switzerland, pp. 372-377.

[32] Sato, Y., Saito, M., and Koike, H. (2001) Real-time input of 3D pose and gestures of a user's hand and its applications for HCI, *Proc. 2001 IEEE Virtual Reality Conference (IEEE VR2001)*, pp. 79-86.

[33] Segen, J. & Kumar, S. (2000) Look, Ma, No Mouse! *Communications of the ACM*, July 2 Vol 43, no 7.

[34] Starner, T, Weaver, J. & Pentland, A. (1998) Real-time American Sign Language recognition using desk and wearable computer-based video. *IEEE Transactions on Pattern. Analysis and Machine. Intelligence*.

[35] J. Triesch and C. von der Malsburg. (2001) A system for person-independent hand posture recognition against complex backgrounds. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 23, no 12.

[36] Turk, M. & Robertson, G. (2000) Perceptual User Interfaces. *Communications of the ACM*, vol. 43, no. 3, pp. 33-34

[37] Turk, M. (2002) Gesture Recognition. In Stanney, K. (Ed.): *Handbook of Virtual Environments. Design, Implementation, and Applications*. Lawrence–Erlbaum Assoc.

[38] Utsumi, A. & Ohya, J. (1999) Multiple-Hand-Gesture Tracking Using Multiple Cameras. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.473-478.

[39] Von Hardenberg, C. & Berard, F. (2001) Bare-Hand Human-Computer Interaction. In *Proc. of ACM Workshop on Perceptive User Interfaces*, Orlando, Florida.

[40] Wellner, P. (1991) The DigitalDesk Calculator: Ta c-tile Manipulation on a Desk Top Display. *Proceedings of UIST'91*, Nov. 11-13, pp.27-33

[41] Wexelblatt, A. (1995) An Approach to Natural Gesture in Virtual Environments. *ACM ToCHI*, Vol 2., No. 3, September, pp 179-200.

[42] Wren, C.R., Basu, S. Sparacino, F. & Pentland, A. (1999) Combining Audio and video in Perceptive Spaces. *1st International Workshop on Managing Interactions in Smart Environments*, Dec. 13-14, Dublin, Ireland