Vision-Based Interfaces for Mobility

Mathias Kölsch Matthew Turk Tobias Höllerer
Department of Computer Science, University of California, Santa Barbara, CA 93106

Abstract

Vision-based user interfaces are a feasible and advantageous modality for wearable computers. To substantiate this claim, we present a robust real-time hand gesture recognition system that is capable of being the sole input provider for a demonstration application. It achieves usability and interactivity even when both the head-worn camera and the object of interest are in motion. We describe a set of general gesture-based interaction styles and explore their characteristics in terms of task suitability and the computer vision algorithms required for their recognition. Preliminary evaluation of our prototype system leads to the conclusion that vision-based interfaces have achieved the maturity necessary to help overcome some limitations of more traditional mobile user interfaces.

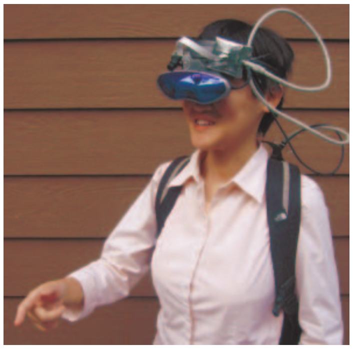

Figure 1: Our mobile user interface in action. All hardware components aside from display, camera, and microphone are in the backpack.

1. Introduction

Vision-based interfaces (VBI) in stationary installations have recently achieved a quality level acceptable to consumers. For example, Sony's Eye Toy, an accessory for the PlayStation 2, has topped the UK game sales charts for months. A USB camera recognizes the players' full-body motions and projects the player directly into the game. However, mobile computer vision (CV) systems have not seen the same level of maturity: the lack of robustness, speed, and accuracy has prevented reliable interface operation. In this paper, we present a CV system that makes hand gesture recognition feasible in mobile computing environments, using a head-worn camera.

Investigating free-hand gesture interfaces for mobile interaction is important because gestures can fill functionality gaps that other modalities such as speech and keypad input can not make accessible. For example, two-handed interaction to concurrently control two registered pointers is not conveniently and intuitively accessible with traditional input devices. CV is advantageous over gloves as an implementation technology because it requires no physical devices to be carried or worn on the hands.

The paper is organized as follows. In Section 3, we distinguish different types of gestures and their place in the mobile user interface (MUI). Section 4 introduces our main contribution, a CV system for hand detection, tracking, and posture recognition. We briefly discuss the maintenance application used for preliminary evaluation in Section 5, among others suggesting that our approach is capable of serving as the sole interface modality for mobile applications. Our hardware setup, shown in Figure 1 and detailed in Subsection 5.1, consists of a head-worn display with an attached camera as the only visible and interacting components. Overall, we hope to stimulate increased research and interest in using CV in the mobile systems arena.

2. Related work

This section reviews VBIs for mobile computers and applications. Work solely related to CV issues will be addressed in Section 4.

Starner et al. pioneered mobile VBIs for American Sign Language recognition [30]. A cap-mounted camera tracked skin-colored blobs whose spatial progression was analyzed over time with Hidden Markov Models. Their system worked with non-instrumented hands just as ours. However, our system integrates multiple image cues (skin color and texture information) to overcome the robustness limitations associated with relying on the accuracy of single-cue image segmentation. They recognized the need for a second modality and experiment now with accelerometers attached to the signer's wrists [1]. Recognizing a set of communicative gestures (that frequently exhibit distinct spatial trajec-

Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04)
0-7695-2208-4/04 $20.00 © 2004 **IEEE**

tories – see Quek [23] for a classification) requires more semantic post-processing, but manipulative and discrete postures as recognized by our methods are more demanding on the CV methods. Another color-based VBI was shown by Dominguez et al. [3] who implemented a compelling wearable VBI that enabled the user to encircle objects in view with a pointing gesture.

In a later project, Krum, Starner, et al. built a mobile system for recognizing gestures and speech [16]. It employed specialized imaging hardware with active infrared illumination and provided a small interactive area at sternum height in front of the wearer's body. Our vision hardware is entirely passive, that is, it does not include light sources. A related user study [15] found that the relatively static hand position for extended periods of time caused fatigue. We hope to avoid fatigue and discomfort symptoms even for long-term interaction through use of a much larger interaction area and less rigid hand postures.

Kurata et al.'s HandMouse [18] is a VBI for mobile users wearing a head-mounted display (HMD) and camera very similar to ours, allowing for the registered manipulation technique (see Section 3). It differs in that the hand has to be the visually prominent object in the camera image and that it relies solely on skin color. The robustness gained with our multi-modal approach makes it possible for the image of the hand to be much smaller. Via wireless networking, they employ a stationary cluster for processing [17] whereas our vision methods run on a single laptop without sacrificing speed. Going beyond the interaction methods they demonstrated, we characterize additional techniques and their suitability for mobility and the outdoors. Our system then shows how this improves MUI usability and effectiveness.

A few recent research projects use the ARtoolkit [10] software to obtain the hand's 6 degree-of-freedom (DOF) position purely by means of grey-level image processing. For example, Thomas and Piekarski [31] attach a fiducial (a marker that has distinct visual properties and can easily be detected) to the back of the hand. Our system requires no markers, tracks without restrictions on rotation, and can obtain posture information in addition to 2D location. The authors' Outdoor Tinmith Backpack Computer is an excellent example of a high-fidelity wearable computer, but also of the amount of equipment required to facilitate this functionality. We designed our system to minimize extraneous hardware requirements and instead make the computer disappear as much as possible. Only the head-worn devices are exposed, everything else is carried in a small backpack.

Wearable Augmented Reality systems such as described in Feiner et al. [5] are related in that they are a prime recipient for our interaction methods, as they lack sufficient interface capabilities and are thus still limited in their application.

An overview of using computer vision as user interface implementing technology can be found in [33].

3. Hand gesture interaction techniques

A hand *posture* is a static configuration of the fingers that can be recognized on a per-frame basis and might be described with a qualitative statement, for example "fist." *Gesture* is a more general term as it can involve dynamic aspects of movement, such as waving goodbye. Our CV methods recognize a set of postures from a certain view direction and track the hand in arbitrary configurations in two dimensions. We distinguish three styles of gesture interpretation for UI purposes; their characteristics and the manipulation techniques that they support are described in the following paragraphs.

Registered manipulation means that the pointer is colocated with the hand in the video see-through display. The hand can therefore virtually touch objects that it is interacting with. This style is especially suitable to interaction with virtual objects in mixed reality scenarios and for interaction with the view of the real world. However, it is hard to perform this kind of manipulation while walking.

Pointer-based manipulation describes gestures and their interpretation in the style of a computer mouse. See Fukumoto et al. [6] and Quek et al. [24] for early examples of interpreting the finger location as mouse input. Conceptually, movements in an *input plane* control a pointer on a distinct *manipulation plane*. The input plane is fixed relative to the camera coordinate system, while the ("direct") manipulation plane is fixed relative to the screen coordinate system. The transformation between the two planes requires some attention:

1) A method for "clutching" (see MacKenzie [20]) must be provided because with hand tracking, the user can not "pick up and reposition the mouse." Instead, clutching could happen automatically when the pointer reaches the confines of the screen. Further hand movements will then dynamically modify the translation offset between the two planes.

2) We found that constraining pointer movements to one dimension (for example, to a horizontal line) is very convenient as it reduces the required precision of hand movements. This in turn appears to reduce fatigue, sometimes caused by unnecessarily strict gesture requirements.

3) Larger-than-identity scaling factors avoid overly extensive hand movements while at the same time allowing for big, easily visible buttons. On the other hand, too large scaling factors again introduce unnecessarily strict requirements and might even subject the input to involuntary jitter during general body motion.

4) Snapping the pointer to the default button can ensure that in most cases no hand movement but only the selection

gesture has to be performed. While this behavior might be disruptive in a desktop environment, it is more convenient within the MUI context.

Location-independent interaction refers to hand postures that can be performed anywhere within the camera field-of-view (FOV) and produce a single event. As Hauptmann pointed out [7], pointer-based manipulation should not be the only mode of interaction. Location-independent gestures are thus an important mode of interaction, especially for people "on the move."

A selecting gesture (a "mouse click") is a necessary concept for many pointer-based and registered interfaces. It can be implemented with two techniques: **selection by action**, which involves a distinct posture to signal the desire to select. If the same hand is employed for both pointing and selection, some movement during the selection action must be expected and should not interfere with pointing precision. For high precision demands a **selection by suspension** technique might be more appropriate, in which the desire to select is conveyed by not moving the pointer for a threshold period of time. Requiring the user to be idle for a few seconds, or constantly move her hand to avoid selection, is usually unwise, particularly in mobile contexts.

In building the gesture interface, we took human factors about hand reach limits into account. In accordance with our definition [11], the interaction range was designed to be within the users' comfort zone. We also chose postures that were sufficiently distinguishable from background artifacts and from one another. That for example rules out using a fist from dorsal view, as shown in [14].

Table 3 summarizes which hand gesture interaction techniques we use for which application functionality (described in detail in Section 5).

manipulation technique	maintenance application's user interface component
pointer-based	voice recorder, image/video capture menu
registered	area image capture, number selection
location-independent	task switch, work order selection
selection by action	button click
selection by suspension	area image capture

Table 1: The different types of gestures and which application part utilizes this interaction technique.

Hauptmann [7] made another important observation: the importance of immediate feedback for the user's actions. We provide timely and direct feedback about the most important vision-level information – whether detection and tracking of the hand was successful – with a red dot on top of what the system thinks is the hand. All other feedback comes from the application space. For example, a red border is drawn around buttons that the user hovers over, signaling that executing a selection gesture will "click" that button. An iconic hand is drawn as cursor for the pointer-based manipulation techniques.

4. The computer vision system

The core of this paper's contribution – demonstrating feasibility of mobile VBIs and available choices – is based on the CV system. We use a combination of recently developed methods with novel algorithms to achieve real-time performance and robustness. A careful orchestration and automatic parameterization is largely responsible for the high speed performance while multi-modal cue integration guarantees robustness.

There are three stages: the first stage detects the presence of the hand in one particular posture. (It is undesirable to have the vision interface always active since coincidental gestures may be interpreted as commands. Also, processing is faster and more robust if only one gesture is to be detected.) After this gesture-based activation, the second stage serves as an initialization to the third stage, the main tracking stage.

This multi-stage approach makes it possible to take advantage of less general situations at each stage. Exploiting spatial and other constraints that limit the dimensionality and/or extent of the search space achieves better quality and faster processing speed. We use this at a number of places: the generic skin color model is adapted to the specifics of the observed user (see Subsection 4.2), and the search window for posture recognition is positioned with KLT tracking (see Subsection 4.3). However, staged systems are more prone to error propagation and failures at each stage. To avoid these, every stage makes conservative estimations and uses multiple image cues (grey-level texture and local color information) to increase confidence in the results.

The final output of the vision system consists of the 2D location and sometimes the posture of the hand, and at some occasions also the location of the second hand. The posture is described as a classification into a set of predefined, recognizable hand configurations. The diagram in Figure 2 and the following subsections detail the components of our vision system and their interactions.

4.1. Hand detection

Most earlier gesture recognition systems place restrictions on the environment, such as a uniform background (Segen and Kumar [27], Rehg and Kanade [25]), a static background (Morris and Elshehry [21]), or colored gloves or markers on the hand (Dorfmüller-Ulhaas et al. [4], Thomas and Piekarski [31]). Hand detection against arbitrary background was achieved for example by Triesch and Mals-

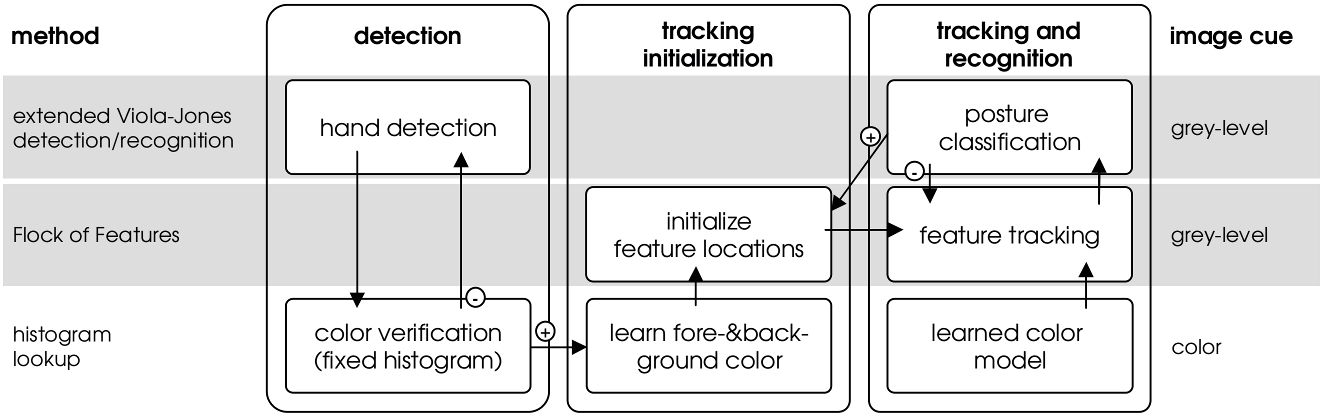

Figure 2: How the computer vision methods are arranged: only on successful hand detection will the tracking method start operating. Posture recognition is attempted after each tracking step. If successful, features and color are re-initialized.

burg [32] who are able to distinguish hand poses with 86.2% accuracy, or with a method by Cui and Weng [2] – however, neither method can perform the detection in real time as is required for UIs.

We customized an object detection method recently proposed by Viola and Jones [34]. Objects are learned during a training phase with AdaBoost of features that compare grey-level intensity in rectangular image areas. The decisive advantage of this over other appearance-based methods is that it can be implemented with "integral images," a method borrowed from database research, there known as *data cubes*. During detection, a pre-computation step produces a 2-dimensional brightness integral. The sum of pixel values in arbitrary rectangular areas can then be computed in constant time. Detection of hands of arbitrary scale (larger than 30x20 pixels) runs with about 10 frames per second on a 640x480 sized video stream on a 3GHz desktop computer.

The initial hand pose is a top-down view of the flat hand with the fingers touching each other (see Figure 4). We chose this posture/view combination due to its highly identifiable nature against background noise and therefore its good success rate as a fail-safe detection condition [14]. The recognition is executed in a part of the camera's FOV that corresponds to a natural reaching distance in front of the right shoulder. The original object detection method is very sensitive towards in-plane rotations. We trained the detector for multiple slight rotations of the same hand posture [12], allowing the posture to be performed at angles from 0 to 15 degrees, increasing usability. The same technique was used for posture recognition.

Upon detection of a hand area, it is tested for the amount of skin colored pixels it contains. To this end, we built a histogram-based statistical model in HSV space from a large collection of hand-segmented pictures from many imaging sources, similar to Jones and Rehg's approach [9]. We used a histogram-based method because they achieve better results in general, user-independent cases. If a sufficient amount of area pixels are classified as skin pixels, the hand detection is considered successful and control is passed to the second stage.

4.2. Tracking initialization

The very general statistical model of skin color is then refined by learning the observed hand color on the area detected. This color histogram is contrasted to a reference area that is assumed to not contain skin areas, located around the hand area to the left, top and right. This assumption always held in our experiments due to the camera FOV and angle. Note that other skin-colored objects, even other people, that might be in this reference area are – correctly – considered background. Figure 3 exemplifies the color segmentation and feature tracking operation.

Next, twenty KLT features (Shi and Tomasi [29]) are placed on "good-to-track" skin-colored spots in the detected area. KLT trackers are named after Kanade, Lucas, and Tomasi who found that a steep brightness gradient along at least two directions makes for a promising feature candidate to be tracked over time. In combination with image pyramids (a series of progressively smaller resolution interpolations of the original image, see Lucas and Kanade [19]), a feature's image area can be matched efficiently to a similar area in the following video frame.

KLT features do not encode object-level knowledge nor global information. To achieve consistency among the features, to improve tracking across changing backgrounds, and to better deal with short occlusions, we enforce global constraints on the features' locations with a "Flocks of Features" method that enforces conditions of minimum and maximum pairwise feature distances [13]. This was found to have superior performance over Condensation tracking

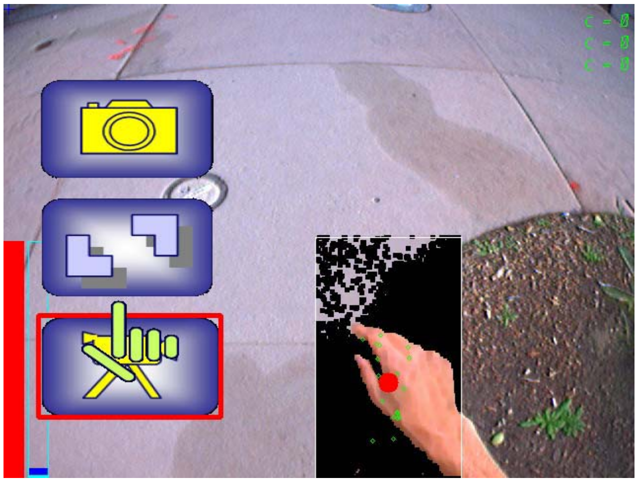

Figure 3: A screen capture with verbose output turned on, taken while walking. The image is partially color-segmented, illustrating how skin color by itself is not a reliable modality. In color prints also visible are the green KLT features.

(see Isard and Blake [8]) which we had used before.

4.3. Tracking and recognition

We found edge and shape based methods unsuitable for detection and tracking in complex environments. Their performance is largely determined by the amount of contrast between the foreground object and the background scene, which can not be guaranteed. Also, the frequently associated gradient-descent methods that enforce global constraints for the individual edges' locations encounter problems with the highly articulated hands whose appearance can trap the algorithms in deep local minima because of the multitude of strong edges. Texture-based features on the other hand, which have true spatial extent, contain more information than line features. We further decided not to extract hand configuration information, such as with kinematic 3D hand models, because the available methods do not exhibit the robustness and speed necessary for VBIs due to too many degrees of freedom and arising singularities during parameter estimation.

In our approach, called "Flocks of Features" and described to greater detail in [13], we first update the KLT features' positions with the traditional pyramid-based feature matching algorithm. From their locations we determine a small area that is scanned for the key postures that recognition is attempted for. If this classification succeeds, the feature location and the color lookup table are re-initialized as described in Subsection 4.2.

If no particular posture is recognized, tracking must continue with only the "cloud" of KLT features. Without additional effort, this would only work fine for rigid objects with a mostly invariant appearance. However, hands are a highly articulate object whose appearance can change vastly and rapidly. The feature match correlation between two consecutive frames can thus be very low so that the feature must be considered "lost." Also, features might gradually move off the hand onto background areas with more prominent grey-level gradients. To cope with this situation and to better track the object at hand, our algorithm removes features with low correlation, those far from the cloud's centroid, and those too close to other features from the set. They are resurrected at good-to-track locations that also have a high skin color probability and are close to the cloud's centroid.

This method leads to a very natural multi-modal integration, combining cues from feature movement based on grey-level image texture with cues from texture-less skin color probability. In addition, it enforces global constraints on the feature locations, keeping outliers at bay and following the main object of interest instead. Over time however, the only guarantee we get about the tracked object is that it has a high content of skin color since nothing prevents the features from moving onto other skin-colored objects.

Posture recognition counters this drift. We use the detection method described in Subsection 4.1 with cross-trained posture appearances to recognize currently five postures (see Figure 4) and a number of in-plane rotations. This proved to be sufficient to re-initialize tracking frequently enough during gesture transitions. As mentioned above, once any posture is recognized, the color model is relearned from the area known to be of skin color. Also, all features are located onto image parts that are known to belong to the hand.

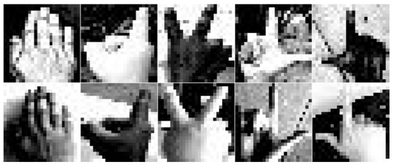

Figure 4: Two examples each of the five hand postures that are recognized, shown in the minimum resolution required for recognition, 25x25 pixels, some with a distorted aspect ratio for recognition performance reasons.

Some of the few papers that present methods for view-independent hand posture classification are Wu and Huang [35] and Rosales et al. [26]. Ong and Bowden [22] use an approach very similar to our classification method, yet no statement regarding real-time performance is made.

4.4. Performance

The quality and usability of a VBI is determined by four main aspects of the CV method(s): speed, accuracy, precision, and robustness. We have begun to evaluate the performance of the CV system, as well as its usability as a UI.

Below we report on the results of a preliminary user study to quantify these measures, conducted while users were standing at one indoor and two outdoor locations. A more extensive study and evaluation is in preparation.

Sheridan and Ferrell found a maximum latency between event occurrence and system response of 45ms to be experienced as “no delay” [28]. While we have not quite achieved that end-to-end latency, all methods combined require less than 100ms total processing time per frame (latency from frame capture to render completion time as reported by DirectShow on a 1.1GHz laptop). This is well below the threshold of 300ms for when interfaces start to feel sluggish, might provoke oscillations, and cause the “move and wait” symptom [28]. The system achieves frame rates of 10-15Hz, where 15Hz is the camera’s capture rate. In comparison to other mobile VBIs, our method is significantly more responsive than Kurata’s Hand Mouse [18], judging from a video available on their web site.

The object detection and posture recognition methods were trained to have a very low false positive rate (smaller than 1e-10, see [14]) and a medium detection rate between 85% and 95%. In practice, and in combination with the color cue, the detector produced around three false positives per hour during indoor and outdoor operation (or one false positive in about 10,000 frames). The recognition module performs almost as well: in a test set of 19,134 frames, 9137 postures were recognized. Of those, 93.76% were classified correctly. However, most misclassifications were due to one particular posture (*open*, not shown in Figure 4). If this posture is disregarded, 98.71% are classified correctly instead. It remains to be mentioned that the test frames were obtained without supervision from study participants that occasionally performed the wrong posture, resulting in a mislabeled frame and likely “misclassification”.

Hand tracking alone, without the re-initialization from Subsection 4.2, had a mean time to failure of 23 seconds, but it often succeeded for the maximum time tested of 60 seconds. The frequency of re-initializing posture recognitions depends entirely on the user task and has not been investigated yet. The accuracy of the KLT features’ average location (which we used as the pointer’s location) with respect to some fixpoint on the hand can not be guaranteed because of the entirely object-independent tracking method. However, this was only of concern for the registered manipulation tasks, as the other interaction techniques involve pointer location transformations or are location independent.

We evaluated hand tracking precision with an object-following task and found no significant differences to the performance of a handheld trackball (in terms of mean and median distance of pointer from object). Empirically, the tracking precision is excellent, even minute hand movements are tracked. Illustrating the naturalness of the interface, people frequently employed hand movements at first for the trackball task before remembering that now the hand was not being tracked anymore.

Our methods are generally robust to different environmental conditions, including different lighting, different users, cluttered backgrounds, and non-trivial motion. They are largely camera-independent and can cope with the automatic image quality adjustments of digital cameras. Two conditions will still violate our assumptions and might impact recognition and tracking negatively: an extremely over- or under-exposed hand appearance does not contain a sufficient amount of skin-colored pixels for successful detection. Second, if the color changes dramatically in between two consecutive successful posture classifications, the tracking degenerates into single-cue grey-level KLT tracking. Since the system updates its color model periodically, it is able to cope with slowly changing lighting conditions, however.

None of the system’s functionality explicitly detects or models hand occlusions. However, brief occlusions of the tracked hand with foreign objects or the other hand do generally not cause all KLT features to be lost. The detection and posture recognition classifiers were trained with images taken with different still picture cameras, while the system was successfully tested with three different digital video cameras. In addition, none of the training images was shot with as short a focal length lens as our mobile camera has. These facts suggest that the entire system will run with almost any color camera available.

5. The wearable computer

5.1. Hardware setup

The hardware setup of our system produces output through a head-worn display (HMD, Sony Glasstron LDI-A55), atop which we mounted a small digital camera (FireFly, Point Grey Research), see Figure 5. The camera has a horizontal FOV of 70° and its pitch for a normal head position is adjusted to cover the range from almost straight down to horizontal. The live video stream, augmented with the application overlay described in the following subsection, is fed into the display to achieve video see-through mixed reality. This alleviates problems with the HMD’s small 30° FOV because it makes 70° FOV available to the wearer. The resulting spatial compression takes users a few minutes to get used to, but no adaptation problems were reported after that time. Use of this fisheye-style lens reduced the tunnel effect that most optical see-through mixed reality displays exhibit. The high FOV is also important for interface functionality because both the hands and a more forward-facing view direction are within the FOV, which allows direct feedback as well as a registered interaction style (see Section 3).

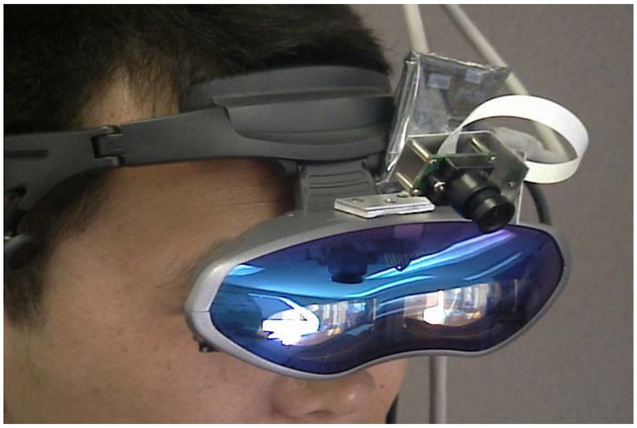

Figure 5: Closeup of display and camera. All other devices are stowed in a backpack and need not be accessed.

Note that no other input device such as a Twiddler keyboard or 3D mouse are used. Instead, the input- and output-interface is combined into a single head-worn unit. The other logical component of our system, a laptop plus a few adapters and batteries, is stored away in a conventional backpack. Overall, this makes for a fairly easy to assemble and relatively inexpensive mobile computer.

5.2. Maintenance application

We tested the functionality of our VBI with a custom built user interface component for a set of mock-up application "panes" for facilities personnel. It was designed to demonstrate the suitability of VBIs for the mobile use. Its suggested functionality supports building facilities managers in their daily tasks of performing maintenance operations and immediate-attention work requests, for example investigating a water leak or power failure in a particular room. The wearer of our mobile system can utilize three main panes: an audio recorder, a digital still and video camera, and a work order and communication pane. The active pane is selected by performing a location-independent task-switch gesture for a short period of time, which cycles through the applications and a "blank screen" mode, one by one.

Voice recorder: A small microphone clipped to the goggles allows auditory recordings, activated by gesture commands that start, pause, resume, and stop a sound recording. This interface utilized the pointer-based manipulation technique in combination with a location-independent "select" posture. Buttons are horizontally aligned and the pointer is restricted to moving along this dimension. A red border gives visual feedback whenever the hand pointer is in the area of a button (hovering above it).

Image and video capture: The image capture pane has three modes of operation which are selected via buttons. The interaction technique with the image/video capture menu is very similar to that of the voice recorder, only that the buttons are arranged in a vertical fashion and the pointer movement is constrained to that dimension. The first mode allows a user to take a picture of the entire visible area. A count-down timer is overlaid after activating this mode. A picture is taken and stored at the end of the count-down. The second mode records a video stream instead, stopping as soon as the hand is detected within the interaction initiation area.

The third mode allows taking snapshots of selective areas. The CV system searches for the left hand as the nearest skin-colored blob to the lower-left of the right hand. The rectangular area enclosed by both hands is highlighted in the display, shown in Figure 6. When the positions of both hands stabilized with respect to the camera, the snapshot is taken. Implementations of the same functionality that use only one pointer are conceivable but less convenient to use. This is the only task where hand suspension was the selection method of choice because the user will most likely have assumed a stationary body position and performing "action" selection gestures would interfere with the pointing precision.

Work order scheduler: With the aid of this pane, the person in the field can retrieve, view, and reply to work requests. Up to three work orders with title and status (open, closed, follow-up) are shown concurrently, automatic scrolling brings hidden orders into view (see Figure 6). Three dedicated, static hand gestures allow for selection and manipulation of work requests: One gesture selects the work order above the current one, another gesture selects the one below the current one. We choose the discrete posture technique over pointer-based manipulation because scrolling with a pointer and "scrollbars" is an unnatural, awkward operation, especially for MUIs. The third gesture facilitates activation of the currently selected work order. "Attachments" to a report can be selected from the previously recorded media clips (voice recording, still picture, or video) with "registered" hand movements. This was decided based upon the possibly large number of clips and the convenience of random access over access in a sequential fashion. The selection gesture picks the currently highlighted number.

6. Future work

We have not found a good solution to automatic detection of tracking loss. Heuristics based on KLT feature locations provide some clues, but in a few occasions the system would track some non-hand object.

Depth and world-referenced 3D information can supply additional input parameters, especially for manipulation of virtual 3D objects. We have so far restricted our work to 2D interaction, but a great benefit of hand gestures are nat-

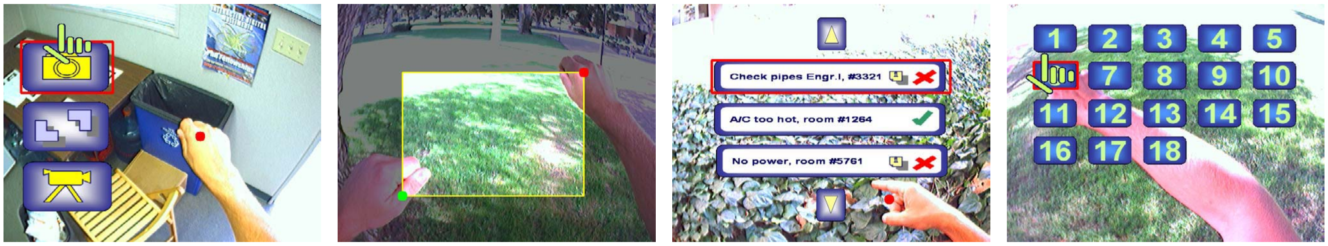

Figure 6: Left: Pointer-based interaction, constrained to the vertical dimension, for image and video capture pane. Left center: The user has selected an area, and the snapshot will be taken when the hands have settled for five seconds. Right center: Location-independent postures (*up, down*) change the highlighted work order. The gesture being performed in the picture selects the highlighted item. Right: Selecting from many items with registered manipulation. Note that the interface snapshots shown in the various figures were taken in different environments, illustrating the ability of our system to adjust to varying backgrounds and lighting conditions.

urally their inherent 3D capabilities. We have experimented only with limited two-handed manipulation, but would like to explore that interaction technique more. These issues and recognizing dynamic gestures such as clapping for interface purposes are on the horizon of interesting system extensions.

No multi-modal interface capability is currently provided because we wanted to focus on the CV aspect for this paper. However, the addition of a speech recognition component is certainly going to be beneficial for many aspects of usability and flexibility.

7. Conclusions

We showed different manipulation techniques for hand gesture-based mobile user interfaces and presented computer vision methods (CV) capable of detecting these gestures. We introduced a prototype system that we built to demonstrate robustness and usability of this vision-based interface when used as the sole input modality to a wearable computer. Robustness was evidenced with regard to environment conditions, in particular to indoor and outdoor lighting, cluttered backgrounds, concurrent movement of camera and user, user independence, and camera independence. The improved usability stems from from relatively low interaction latencies and the aforementioned robustness.

The contribution of the presented work is twofold: Firstly, the system shows the feasibility of vision-based hand gesture interfaces as the exclusive input modality for wearables. Secondly, it demonstrates enhanced interaction capabilities through hand gesture recognition, some of which are difficult to achieve with other modalities. We conclude that CV has reached a stage where it can effectively replace some physical-device interfaces and augment others, enabling new functionalities and novel applications for the mobile user.

8. Acknowledgments

Our thanks go to Keith Clarke for making equipment available to us and to James Chainey for the graphic art in the maintenance application. This research is supported in part by funds provided via NSF Digital Government grant EIA-9983289 and the following collaborating agencies: Bureau of the Census, Bureau of Labor Statistics, U.S. Department of Agriculture (Forest Service, National Agricultural Statistics Service, Natural Resources Conservation Service), and U.S. Geological Survey. This work was partially supported under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48.

References

[1] H. Brashear, T. Starner, P. Lukowicz, and H. Junker. Using Multiple Sensors for Mobile Sign Language Recognition. In *IEEE Intl. Symposium on Wearable Computers (ISWC)*, Oct 2003.

[2] Y. Cui and J. Weng. A Learning-Based Prediction and Verification Segmentation Scheme for Hand Sign Image Sequence. *Transactions on Pattern Analysis and Machine Intelligence*, pages 798–804, 1999.

[3] S. M. Dominguez, T. Keaton, and A. H. Sayed. Robust Finger Tracking for Wearable Computer Interfacing. In *ACM PUI 2001 Orlando, FL*, 2001.

[4] K. Dorfmüller-Ulhaas and D. Schmalstieg. Finger Tracking for Interaction in Augmented Environments. In *IFAR*, 2001.

[5] S. Feiner, B. MacIntyre, T. Höllerer, and T. Webster. A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment. In *Proc. First Int. Symp. on Wearable Computers*, October 1997.

[6] M. Fukumoto, Y. Suenaga, and K. Mase. Finger-Pointer: Pointing Interface by Image Processing. *Computers & Graphics*, 18(5):633–642, 1994.

[7] A. G. Hauptmann. Speech and Gesture for Graphic Image Manipulation. In *ACM CHI*, pages 241–245, May 1989.

[8] M. Isard and A. Blake. Condensation – Conditional Density Propagation for Visual Tracking. *Int. Journal of Computer Vision*, 1998.

[9] M. J. Jones and J. M. Rehg. Statistical Color Models with Application to Skin Detection. *Int. Journal of Computer Vision*, 46(1):81–96, Jan 2002.

[10] H. Kato and M. Billinghurst. Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, pages 85–94, October 1999.

[11] M. Kölsch, A. C. Beall, and M. Turk. The Postural Comfort Zone for Reaching Gestures. In *HFES Annual Meeting Notes*, October 2003.

[12] M. Kölsch and M. Turk. Analysis of Rotational Robustness of Hand Detection with a Viola-Jones Detector. In *IAPR International Conference on Pattern Recognition*, 2004.

[13] M. Kölsch and M. Turk. Fast 2D Hand Tracking with Flocks of Features and Multi-Cue Integration. In *IEEE Workshop on Real-Time Vision for Human-Computer Interaction (at CVPR)*, 2004.

[14] M. Kölsch and M. Turk. Robust Hand Detection. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, May 2004.

[15] D. M. Krum, O. Omoteso, W. Ribarsky, T. Starner, and L. F. Hodges. Evaluation of a Multimodal Interface for 3D Terrain Visualization. In *IEEE Visualization*, pages 411–418, October 27-November 1 2002.

[16] D. M. Krum, O. Omoteso, W. Ribarsky, T. Starner, and L. F. Hodges. Speech and Gesture Multimodal Control of a Whole Earth 3D Visualization Environment. In *VisSym '02, Joint Eurographics – IEEE TCVG Symposium on Visualization*, May 2002.

[17] T. Kurata, T. Kato, M. Kourogi, J. Keechul, and K. Endo. A Functionally-Distributed Hand Tracking Method for Wearable Visual Interfaces and Its Applications. In *Proc. IAPR Workshop on Machine Vision Applications*, pages 84–89, 2002.

[18] T. Kurata, T. Okuma, M. Kourogi, and K. Sakaue. The Hand Mouse: GMM Hand-color Classification and Mean Shift Tracking. In *Second Intl. Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems*, July 2001.

[19] B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proc. Imaging Understanding Workshop*, pages 121–130, 1981.

[20] I. S. MacKenzie. Input devices and interaction techniques for advanced computing. In W. Barfield and T. A. F. III, editors, *Virtual environments and advanced interface design*, pages 437–470. Oxford University Press, 1995.

[21] T. Morris and O. S. Elshehry. Hand segmentation from live video. In *The 2002 Intl. Conference on Imaging Science, Systems, and Technology*, UMIST, Manchester, UK, 2002.

[22] E. J. Ong and R. Bowden. A Boosted Classifier Tree for Hand Shape Detection. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, 2004.

[23] F. K. H. Quek. Eyes in the Interface. *Image and Vision Computing*, 13, August 1995.

[24] F. K. H. Quek, T. Mysliwiec, and M. Zhao. FingerMouse: A Freehand Pointing Interface. In *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, pages 372–377, June 1995.

[25] J. M. Rehg and T. Kanade. Visual Tracking of High DOF Articulated Structures: an Application to Human Hand Tracking. In *Third European Conf. on Computer Vision*, pages 35–46, May 1994.

[26] R. Rosales, V. Athitsos, and S. Sclaroff. 3D Hand Pose Reconstruction Using Specialized Mappings. In *Proc. Intl. Conference on Computer Vision*, July 2001.

[27] J. Segen and S. Kumar. GestureVR: Vision-Based 3D Hand Interface for Spatial Interaction. In *The Sixth ACM Intl. Multimedia Conference*, September 1998.

[28] T. Sheridan and W. Ferrell. Remote Manipulative Control with Transmission Delay. *IEEE Transactions on Human Factors in Electronics*, 4:25–29, 1963.

[29] J. Shi and C. Tomasi. Good features to track. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, June 1994.

[30] T. E. Starner, J. Weaver, and A. Pentland. Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 20(12):1371–1375, December 1998.

[31] B. H. Thomas and W. Piekarski. Glove Based User Interaction Techniques for Augmented Reality in an Outdoor Environment. *Virtual Reality: Research, Development, and Applications*, 6(3), 2002.

[32] J. Triesch and C. von der Malsburg. Robust Classification of Hand Postures against Complex Backgrounds. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, October 1996.

[33] M. Turk. Computer Vision in the Interface. *Communications of the ACM*, 47(1):60–67, 2004.

[34] P. Viola and M. Jones. Robust Real-time Object Detection. In *Intl. Workshop on Statistical and Computational Theories of Vision*, July 2001.

[35] Y. Wu and T. S. Huang. View-independent Recognition of Hand Postures. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 84–94, 2000.