

Perceptual Interfaces

Matthew Turk and Mathias Kölsch

University of California, Santa Barbara
UCSB Technical Report 2003-33

Abstract

In recent years, *perceptual interfaces* have emerged as an increasingly important research direction. The general focus of this area is to integrate multiple perceptual modalities (such as computer vision, speech and sound processing, and haptic I/O) into the user interface. Broadly defined, perceptual interfaces are highly interactive, multimodal interfaces that enable rich, natural, and efficient interaction with computers. More specifically, perceptual interfaces seek to leverage sensing (input) and rendering (output) technologies in order to provide interactions not feasible with standard interfaces and the common triumvirate of I/O devices: the keyboard, mouse and monitor.

In this report, we seek to communicate the motivations and goals of perceptual interfaces, to enumerate the relevant technologies, to discuss the integration of multiple modalities, and to describe in more detail the role of computer vision in human-computer interaction. We cover vision problems, constraints, and approaches that are apropos to the area, survey the state of the art in computer vision research and multi-modal interfaces, and take a look at other perceptual technologies such as brain-computer interfaces. We focus on their application to perceptual interfaces, describe several near-term applications, and suggest promising research directions.

Contents

1 PERCEPTUAL INTERFACES	1
1.1 Introduction	1
1.2 Perceptual Interfaces and HCI	2
1.3 Multimodal Interfaces	6
1.4 Vision Based Interfaces	11
1.4.1 Terminology	14
1.4.2 Elements of VBI	15
1.4.3 Computer Vision Methods for VBI	23
1.4.4 VBI Summary	30
1.5 Brain-Computer Interfaces	30
1.6 Summary	32
BIBLIOGRAPHY	35

PERCEPTUAL INTERFACES

A keyboard! How quaint.

— Scotty, in the film *Star Trek IV: The Voyage Home* (1986)

1.1 Introduction

Computer vision research has traditionally been motivated by a few main areas of application. The most prominent of these include biological vision modeling, robot navigation and manipulation, surveillance, medical imaging, and various inspection, detection, and recognition tasks. In recent years, a new area, often referred to as *perceptual interfaces*, has emerged to motivate an increasingly large amount of research within the machine vision community. The general focus of this effort is to integrate multiple perceptual modalities (such as computer vision, speech and sound processing, and haptic I/O) into the user interface. For computer vision technology in particular, the primary aim is to use vision as an effective input modality in human-computer interaction. Broadly defined, perceptual interfaces are highly interactive, multimodal interfaces that enable rich, natural, and efficient interaction with computers. More specifically, perceptual interfaces seek to leverage sensing (input) and rendering (output) technologies in order to provide interactions not feasible with standard interfaces and the common triumvirate of I/O devices: the keyboard, mouse and monitor.

The motivation behind perceptual interfaces is twofold: (1) the changing nature of computers and (2) the desire for a more powerful, compelling user experience than what has been available with graphical user interfaces (GUI) and the associated WIMP (windows, icons, menus, pointing devices) implementations. As computers evolve away from their recent past—desktop machines used primarily for word processing, spreadsheet manipulation, and information browsing—and move toward new environments with a plethora of computing form factors, uses, and interaction scenarios, the desktop metaphor will become less relevant and more cumbersome. Keyboard-based alphanumeric input and mouse-based 2D pointing and selection can be very limiting, and in some cases awkward and inefficient, modes of interaction. Neither mouse nor keyboard, for example, is very appropriate for communicating 3D information or the subtleties of human emotions.

Moore's Law has driven computer hardware over the decades, increasing performance (measured in various ways) exponentially. This observation predicts an improvement in chip density in five years by a factor of ten; in ten years by a factor of one hundred; and in twenty years by a factor of ten thousand. Unfortunately, human capacity does not grow at such a rate (if at all) so there is a serious problem in scaling human-computer interaction as machines evolve. It is unlikely that a user interface paradigm developed at an early point in the Moore's Law curve will continue to be appropriate much later on.

New computing scenarios, such as in automobiles and other mobile environments, rule out many traditional approaches to user interaction. Computing is becoming something that permeates daily life, rather than something people do only at distinct times and places. In order to accommodate a wider range of scenarios, tasks, users, and preferences, interfaces must become more natural, intuitive, adaptive, and unobtrusive. These are primary goals of research in perceptual interfaces.

We will certainly need new and different interaction techniques in a world of small, powerful, connected, ubiquitous computing. Since small, powerful, connected sensing and display technologies should be available, there has been increased interest in building interfaces that use these technologies to leverage the natural human capabilities to communicate via speech, gesture, expression, touch, etc. While these are unlikely to completely replace

tradition desktop and GUI-based interfaces, they will complement existing interaction styles and enable new functionality not otherwise possible or convenient.

In this chapter, we seek to communicate the motivations and goals of perceptual interfaces, to enumerate the relevant technologies, to discuss the integration of multiple modalities, and to describe in more detail the role of computer vision in human-computer interaction. We cover vision problems, constraints, and approaches that are apropos to the area, survey the state of the art in computer vision research applied to perceptual interfaces, describe several near-term applications, and suggest promising research directions.

1.2 Perceptual Interfaces and HCI

Human-computer interaction (HCI) is the study of people, computer technology, and the ways these influence each other. In practice, HCI involves the design, evaluation, and implementation of interactive computing systems for human use. The discipline has its early roots in studies of human performance in manufacturing industries. Human factors, or ergonomics, originally attempted to maximize worker productivity by designing equipment to reduce operator fatigue and discomfort. With the arrival of computers and their spread into the workforce, many human factors researchers began to specialize in the various issues surrounding the use of computers by people. Human-computer interaction is now a very broad interdisciplinary field involving computer scientists, psychologists, cognitive scientists, human factors researchers, and many other disciplines, and it involves the design, implementation, and evaluation of interactive computer systems in the context of the work or tasks in which a user is engaged [30].

As one element of human-computer interaction, the user interface is typically considered as the portion of a computer program with which the user interacts; i.e., the point of contact between the human and the computer. Shneiderman [124] describes five human factors objectives that should guide designers and evaluators of user interfaces:

1. Time to learn
2. Speed of performance
3. Rate of errors by users
4. Retention over time
5. Subjective satisfaction

Shneiderman also identifies the accommodation of human diversity as a major goal and challenge in the design of interactive systems, citing the remarkable diversity of human abilities, backgrounds, motivations, personalities, and work styles of users. People have a range of perceptual, cognitive, and motor abilities and limitations. In addition, different cultures produce different perspectives and styles of interaction, a significant issue in today's international markets. Users with various kinds of disabilities, elderly users, and children all have distinct preferences or requirements to enable a positive user experience.

In addition to human factors considerations for human-computer interaction in the context of typical workplace and consumer uses of computers, the cutting-edge uses of computer technology in virtual and augmented reality systems, wearable computers, ubiquitous computing environments, and other such scenarios demands a fresh view of usability and user interface design. Theoretical and experimental advances (such as the concept of Fitts' Law [39]) have to be translated into new arenas, which also require new analyses of usability. Despite the apparent ubiquity of graphical user interfaces, they are not the answer to all interactive system needs.

Historically, a few major user interface paradigms have dominated computing. Table 1.1 describes one view of the evolution of user interfaces. In the early days of computing, there was no real model of interaction—data was entered into the computer via switches or punched cards and the output was produced (some time later) via punched cards or lights. The second phase began with the arrival of command-line interfaces in perhaps the early 1960s, first using teletype terminals and later with electronic keyboards and text-based monitors. This “typewriter” model—where the user types a command (with appropriate parameters) to the computer, hits carriage return, and gets typed output from the computer—was spurred on by the development of timesharing systems, and continued with the popular Unix and DOS operating systems.

When	Implementation	Paradigm
1950s	Switches, punch cards, lights	None
1970s	Command-line interface	Typewriter
1980s	WIMP-based graphical user interface	Desktop
2000s	Perceptual interfaces	Natural interaction

Table 1.1. The evolution of user interface paradigms

In the 1970s and 1980s, the graphical user interface and its associated desktop metaphor arrived, often described by the acronym WIMP (windows, icons, menus, and a pointing device). For over two decades, graphical interfaces have dominated both the marketplace and HCI research, and for good reason: WIMP-based GUIs have provided a standard set of direct manipulation techniques that largely rely on recognition rather than recall. That is, GUI-based commands can typically be easily found and do not have to be remembered or memorized. Direct manipulation is appealing to novice users, it is easy to remember for occasional users, and it can be fast and efficient for frequent users [124]. Direct manipulation interfaces, in general, allow easy learning and retention, encourage exploration (especially with “Undo” commands) and they can give users a sense of accomplishment and responsibility for the sequence of actions leading to the completion of a task or subtask. The direct manipulation style of interaction with graphical user interfaces has been a good match with the office productivity and information access applications that have been the “killer apps” of computing to date.

However, as computing changes in terms of physical size and capacity, usage, and ubiquity, the obvious question arises: What is the next major generation in the evolution of user interfaces? Is there a paradigm (and its associated technology) that will displace GUI and become the dominant user interface model? Or will computer interfaces fragment into different models for different tasks and contexts? There is no shortage of HCI research areas billed as “advanced” or “future” interfaces—these include various flavors of immersive environments (virtual, augmented, and mixed reality), 3D interfaces, tangible interfaces, haptic interfaces, affective computing, ubiquitous computing, and multimodal interfaces. These are collectively called “post-WIMP” interfaces by van Dam [138], a general phrase describing interaction techniques not dependent on classical 2D widgets such as menus and icons.

The (admittedly grandiose) claim of this chapter is that the next dominant, long-lasting HCI paradigm is what many people refer to as *perceptual interfaces*:

Perceptual User Interfaces (PUIs) are characterized by interaction techniques that combine an understanding of natural human capabilities (particularly communication, motor, cognitive, and perceptual skills) with computer I/O devices and machine perception and reasoning. They seek to make the user interface more natural and compelling by taking advantage of the ways in which people naturally interact with each other and with the world—both verbal and non-verbal communications. Devices and sensors should be transparent and passive if possible, and machines should both perceive relevant human communication channels and generate output that is naturally understood. This is expected to require integration at multiple levels of technologies such as speech and sound recognition and generation, computer vision, graphical animation and visualization, language understanding, touch-based sensing and feedback (haptics), learning, user modeling, and dialog management. (Turk and Robertson [136])

There are two key features of perceptual interfaces. First, they are highly interactive. Unlike traditional passive interfaces that wait for users to enter commands before taking any action, perceptual interfaces actively sense and perceive the world and take actions based on goals and knowledge at various levels. (Ideally, this is an “active” interface that uses “passive,” or non-intrusive, sensing.) Second, they are multimodal, making use of multiple perceptual modalities (e.g., sight, hearing, touch) in both directions: from the computer to

the user, and from the user to the computer. Perceptual interfaces move beyond the limited modalities and channels available with a keyboard, mouse, and monitor, to take advantage of a wider range of modalities, either sequentially or in parallel.

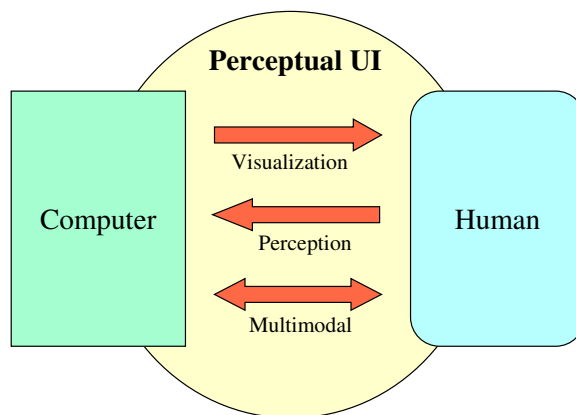


Figure 1.1. Information flow in perceptual interfaces.

The concept of perceptual interfaces is closely related to multimodal, multimedia, and recognition-based interfaces, as depicted in Fig. 1.1. Multimodal interfaces tend to focus on the input direction (input to the computer) and are most often extensions to current GUI and language based interfaces. Multimedia interfaces describe the presentation of various media to users, along with some (again, GUI-based) interaction techniques to control and query the media sources. Interfaces based on individual recognition technologies (such as speech, vision, pen-gesture) focus on the individual recognition technologies, with little integration across modalities. Although each of these classes has significant overlap with the idea of perceptual interfaces, none of them provides a clear conceptual model, or an overarching interaction paradigm, to the user.

The general model for perceptual interfaces is that of human-to-human communication. While this is not universally accepted in the HCI community as the ultimate interface model (e.g., see [121, 122, 123]), there are several practical and intuitive reasons why it makes sense to pursue this goal. Human interaction is natural and in many ways effortless; beyond an early age, people do not need to learn special techniques or commands to communicate with one another. There is a richness in human communication via verbal, visual, and haptic modalities, underscored by shared social conventions, shared knowledge, the ability to adapt and to model the other person's point of view, that is very different from current computer interfaces, which essentially implement a precise command-and-control interaction style. Figure 1.2 depicts natural interaction between people and, similarly, between humans and computers. Perceptual interfaces can potentially effect improvements in the human factors objectives mentioned earlier in the section, as they can be easy to learn and efficient to use, they can reduce error rates by giving users multiple and redundant ways to communicate, and they can be very satisfying and compelling for users.

People are adaptive in their interactions. Despite an abundance of ambiguity in natural language, people routinely pursue directions in conversation intended to disambiguate the content of the message. We do the same task in multiple ways, depending on the circumstances of the moment. A computer system that can respond to different modalities or interaction methods depending on the context would allow someone to perform a given task with ease whether he or she is in the office, in the car, or walking along a noisy street. Systems that are aware of the user and his or her activities can make appropriate decisions on how and when to best present information.

A number of studies by Reeves and Nass and their colleagues [96, 109, 95] provide compelling evidence that human interaction with computers and other communication technologies is fundamentally social and natural. These studies have produced similar (though typically reduced) social effects in human-computer interaction as are found in person to person interactions. The general approach to this work has been to choose a social science finding regarding people's behaviors or attitudes, and to determine if the relevant social rule still applies (and to what magnitude) when one of the roles is filled by a computer rather

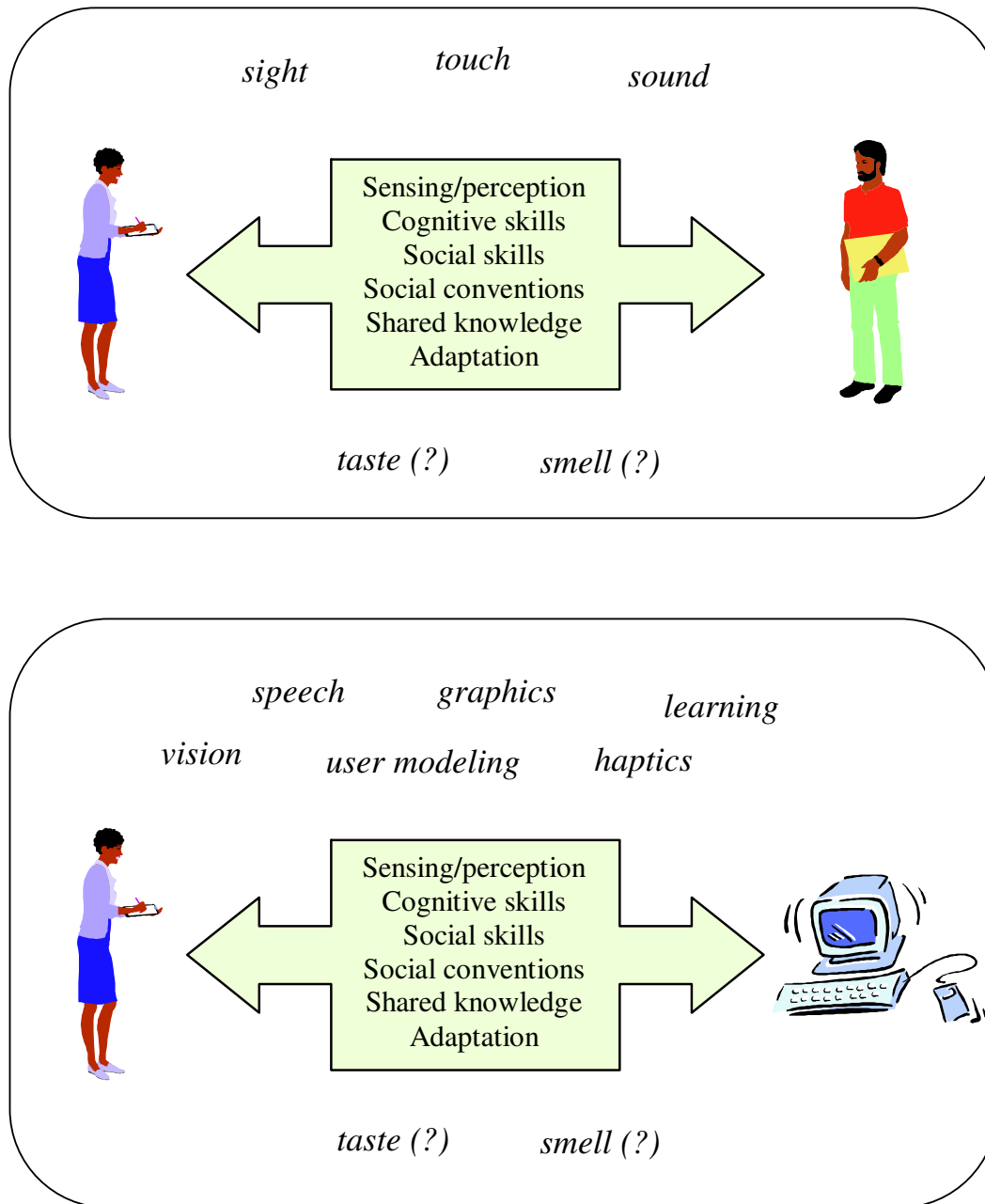


Figure 1.2. Models of interaction: human-human interaction and perceptual human-computer interaction.

than a human. For example, will users apply do norms of politeness or gender stereotypes to computers? In general, which social rules will people apply to computers, and how powerful are these rules?

Such studies have found that social responses are automatic and unconscious, and can be elicited by basic cues. People show social responses to cues regarding manners and politeness, personality, emotion, gender, trust, ethics, and other social concepts. Much of the research in “social interfaces” has focused on embodied conversational agents, or lifelike characters, that use speech and language processing to communicate with a human user [4, 3, 16, 1]. Although the application of these findings is not simple and straightforward, there are implications for future HCI technologies, and perceptual interfaces in particular. An interface that perceives, reasons, and acts in a social manner—even imperfectly—is not too far removed from people’s current conceptual models of the technology.

Despite technical advances in speech recognition, speech synthesis, computer vision, and

artificial intelligence, computers are still, by and large, deaf, dumb, and blind. Many have noted the irony of public restrooms that are “smarter” than computers because they can sense when people come and go, and act appropriately, while a computer may wait all night for a user (who has gone home for the evening) to respond to the dialog that asks “Are you sure you want to do this?” While general purpose machine intelligence is still a difficult and elusive goal, our belief is that much can be gained by pursuing an agenda of technologies to support the human-to-human model of interaction. Even if the Holy Grail of perceptual interfaces is far in the future, the near-term benefits may be transformational, as component technologies and integration techniques mature and provide new tools to improve the ways in which people interact with technology.

In addition to the general goal of interfaces that better match human capabilities and provide a more natural and compelling user experience, there are immediate application areas that are ripe for early perceptual interface technologies. The computer game industry is particularly compelling, as it is large and its population of users tend to be early adopters of new technologies. Game interfaces that can perceive the user’s identity, body movement, and speech, for example, are likely to become very popular. Another group of people who may have a lot to gain from early perceptual interfaces are users with physical disabilities. Interfaces that are more adaptable and flexible, and not limited to particular ways of moving a mouse or typing keys, will provide a significant benefit to this community.

Several other areas—such as entertainment, personal robotics, multimedia learning, and biometrics—would clearly seem to benefit from initial advances in perceptual interfaces. Eventually, the applications of technologies and sense, perceive, understand, and respond appropriately to human behavior appear to be unlimited.

What is necessary in order to bring about this vision of perceptual interfaces? A better understanding of human capabilities, limitations, and preferences in interaction is important, including physical, social, and cognitive aspects. Advances in several technical areas, some quite old and some relatively new, are also vital. Speech understanding and generation (i.e., speech recognition, natural language processing, speech synthesis, discourse modeling, and dialogue management) are vital to leverage the natural modality of spoken language, as well as other sound recognition and synthesis tools (e.g., in addition to words, systems should recognize a sneeze, a cough, a loud plane passing overhead, a general noisy environment, etc.). Computer graphics and information visualization are important to provide richer ways of communicating to users. Affective computing [105] may be vital to understand and generate natural interaction, especially subtle cues to aspects of humor and irony, and appropriate context-dependent displays of emotion (or lack thereof). Haptic and tangible interfaces (e.g., [7, 57]) which leverage physical aspects of interaction may also be important in building truly perceptual interfaces. User and task modeling (e.g., [52, 53]) is key to understanding the whole context of interaction.

Computer vision is also a vital element of perceptual interfaces. Whether alone or in conjunction with other perceptual modalities, visual information provides useful and important cues to interaction. The presence, location, and posture of a user may be important contextual information; a gesture or facial expression may be a key signal; the direction of gaze may disambiguate the object referred to in language as “this” or “that thing.” The next section of the chapter describes the scope of computer vision research and development as it relates to this area of perceptual interfaces.

In addition to advances in individual component areas, the integration of multiple modalities is of fundamental importance in perceptual interfaces. Both lower-level fusion techniques and higher-level integration frameworks will be necessary to build interactive, multimodal interfaces that provide a compelling, natural user experience.

1.3 Multimodal Interfaces

A multimodal interface is a system that combines two or more input modalities in a coordinated manner. Perceptual interfaces are inherently multimodal. In this section, we define more precisely what we mean by *modes* and *channels*, and discuss research in multimodal interfaces and how this relates to the more general concept of perceptual interfaces.

Humans interact with the world by way of information being sent and received, primarily through the five major senses of sight, hearing, touch, taste, and smell. A *modality*

(informally, a *mode*) refers to a particular sense. A communication *channel* is a course or pathway through which information is transmitted. In typical HCI usage, a channel describes the interaction technique that utilizes a particular combination of user and computer communication—i.e., the user output/computer input pair or the computer output/user input pair¹. This can be based on a particular device, such as the keyboard channel or the mouse channel, or on a particular action, such as spoken language, written language, or dynamic gestures. In this view, the following are all channels: text (which may use multiple modalities when typing in text or reading text on a monitor), sound, speech recognition, images/video, and mouse pointing and clicking.

Unfortunately, there is some ambiguity in the use of the word *mode* in HCI circles, as sometimes it is used to mean “modality” and at other times it means “channel.” So are multimodal interfaces “multi-modality” or “multi-channel?” Certainly every command line interface uses multiple modalities, as sight and touch (and sometimes sound) are vital to these systems. The same is true for graphical user interfaces, which in addition use multiple channels of keyboard text entry, mouse pointing and clicking, sound, images, etc.

What then distinguishes multimodal interfaces from other HCI technologies? As a research field, multimodal interfaces focus on integrating sensor recognition-based input technologies such as speech recognition, pen gesture recognition, and computer vision, into the user interface. The function of each technology is better thought of as a channel than as a sensing modality; hence, in our view, a multimodal interface is one that uses multiple modalities to implement multiple channels of communication. Using multiple modalities to produce a single interface channel (e.g., vision and sound to produce 3D user location) is multisensor fusion, not a multimodal interface. Similarly, using a single modality to produce multiple channels (e.g., a left-hand mouse to navigate and a right-hand mouse to select) is a multichannel (or multi-device) interface, not a multimodal interface.

An early prototypical multimodal interfaces was the “Put That There” prototype system demonstrated at MIT in the early 1980s [10]. In this system, the user communicated via speech and pointing gestures in a “media room.” The gestures served to disambiguate the speech (Which object does the word “this” refer to? What location is meant by “there?”) and effected other direct interactions with the system. More recently, the QuickSet architecture [18] is a good example of a multimodal system using speech and pen-based gesture to interact with map-based and 3D visualization systems. QuickSet is a wireless, handheld, agent-based, collaborative multimodal system for interacting with distributed applications. The system analyzes continuous speech and pen gesture in real time and produces a joint semantic interpretation using a statistical unification-based approach. The system supports unimodal speech or gesture as well as multimodal input.

Multimodal systems and architectures vary along several key dimensions or characteristics, including:

- The number and type of input modalities;
- The number and type of communication channels;
- Ability to use modes in parallel, serially, or both;
- The size and type of recognition vocabularies;
- The methods of sensor and channel integration;
- The kinds of applications supported.

There are many potential advantages of multimodal interfaces, including the following [101]:

- They permit the flexible use of input modes, including alternation and integrated use.
- They support improved efficiency, especially when manipulating graphical information.
- They can support shorter and simpler speech utterances than a speech-only interface, which results in fewer disfluencies and more robust speech recognition.
- They can support greater precision of spatial information than a speech-only interface, since pen input can be quite precise.
- They give users alternatives in their interaction techniques.
- They lead to enhanced error avoidance and ease of error resolution.

¹Input means *to the computer*; output means *from the computer*.

- They accommodate a wider range of users, tasks, and environmental situations.
- They are adaptable during continuously changing environmental conditions.
- They accommodate individual differences, such as permanent or temporary handicaps.
- They can help prevent overuse of any individual mode during extended computer usage.

Oviatt and Cohen and their colleagues at the Oregon Health and Science University (formerly Oregon Graduate Institute) have been at the forefront of multimodal interface research, building and analyzing multimodal systems over a number of years for a variety of applications. Oviatt's "Ten Myths of Multimodal Interaction" [100] are enlightening for anyone trying to understand the area. We list Oviatt's myths in italics, with our accompanying comments:

Myth #1. *If you build a multimodal system, users will interact multimodally.* In fact, users tend to intermix unimodal and multimodal interactions; multimodal interactions are often predictable based on the type of action being performed.

Myth #2. *Speech and pointing is the dominant multimodal integration pattern.* This is only one of many interaction combinations, comprising perhaps 14 all spontaneous multimodal utterances.

Myth #3. *Multimodal input involves simultaneous signals.* Multimodal signals often do not co-occur temporally.

Myth #4. *Speech is the primary input mode in any multimodal system that includes it.* Speech is not the exclusive carrier of important content in multimodal systems, nor does it necessarily have temporal precedence over other input modes.

Myth #5. *Multimodal language does not differ linguistically from unimodal language.* Multimodal language is different, and often much simplified, compared with unimodal language.

Myth #6. *Multimodal integration involves redundancy of content between modes.* Complementarity of content is probably more significant in multimodal systems than is redundancy.

Myth #7. *Individual error-prone recognition technologies combine multimodally to produce even greater unreliability.* In a flexible multimodal interface, people figure out how to use the available input modes effectively; in addition, there can be mutual disambiguation of signals that also contributes to a higher level of robustness.

Myth #8. *All users' multimodal commands are integrated in a uniform way.* Different users may have different dominant integration patterns.

Myth #9. *Different input modes are capable of transmitting comparable content.* Different modes vary in the type and content of their information, their functionality, the ways they are integrated, and in their suitability for multimodal integration.

Myth #10. *Enhanced efficiency is the main advantage of multimodal systems.* While multimodal systems may increase efficiency, this may not always be the case. The advantages may reside elsewhere, such as decreased errors, increased flexibility, or increased user satisfaction.

A technical key to multimodal interfaces is the specific integration levels and technique(s) used. Integration of multiple sources of information is generally characterized as "early," "late," or somewhere in between. In early integration (or "feature fusion"), the raw data from multiple sources (or data that has been processed somewhat, perhaps into component features) are combined and recognition or classification proceeds in the multidimensional space. In late integration (or "semantic fusion"), individual sensor channels are processed through some level of classification before the results are integrated. Figure 1.3 shows a view of these alternatives. In practice, integration schemes may combine elements of early and late integration, or even do both in parallel.

There are advantages to using late, semantic integration of multiple modalities in multimodal systems. For example, the input types can be recognized independently, and therefore

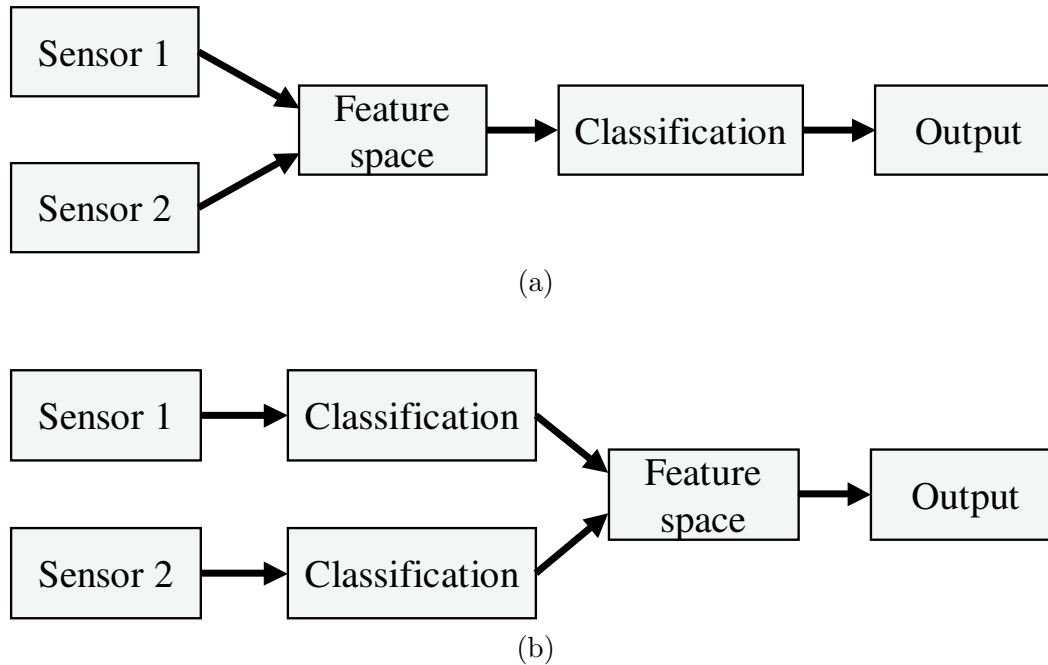


Figure 1.3. (a) Early integration, fusion at the feature level. (b) Late integration, fusion at the semantic level.

do not have to occur simultaneously. The training requirements are smaller, $O(2N)$ for two separately trained modes as opposed to $O(N^2)$ for two modes trained together. The software development process is also simpler in the late integration case, as exemplified by the QuickSet architecture [148]. Quickset uses temporal and semantic filtering, unification as the fundamental integration technique, and a statistical ranking to decide among multiple consistent interpretations.

Multimodal interface systems have used a number of non-traditional modes and technologies. Some of the most common are the following:

- **Speech recognition**
Speech recognition has a long history of research and commercial deployment, and has been a popular component of multimodal systems for obvious reasons. Speech is a very important and flexible communication modality for humans, and is much more natural than typing or any other way of expressing particular words, phrases, and longer utterances. Despite the decades of research in speech recognition and over a decade of commercially available speech recognition products, the technology is still far from perfect, due to the size, complexity, and subtlety of language, the limitations of microphone technology, the plethora of disfluencies in natural speech, and problems of noisy environments. Systems using speech recognition have to be able to recover from the inevitable errors produced by the system.
- **Language understanding**
Natural language processing attempts to model and understand human language, whether spoken or written. In multimodal interfaces, language understanding may be hand-in-hand with speech recognition (together forming a “speech understanding” component), or it may be separate, processing the user’s typed or handwritten input. Typically the more a system incorporates natural language, the more users will expect sophisticated semantic understanding from the system. Current systems are unable to deal with completely unconstrained language, but can do quite well with limited vocabularies and subject matter. Allowing for user feedback to clarify and disambiguate language input can help language understanding systems significantly.
- **Pen-based gesture**

Pen-based gesture has been popular in part because of computer form factors (PDAs and tablet computers) that include a pen or stylus as a primary input device. Pen input is particularly useful for deictic (pointing) gestures, defining lines, contours, and areas, and specially-defined gesture commands (e.g., minimizing a window by drawing a large “M” on the screen). Pen-based systems are quite useful in mobile computing, where a small computer can be carried, but a keyboard is impractical.

- Magnetic, inertial, etc. sensors for body tracking

Sturman’s 1991 thesis [131] thoroughly documented the early use of sensors worn on the hand for input to interactive systems. Magnetic tracking sensors such as the Ascension Flock of Birds² product, various instrumented gloves, and sensor- or marker-based motion capture devices have been used in multimodal interfaces, particularly in immersive environments (e.g., see [50]).

- Non-speech sound

Non-speech sounds have traditionally been used in HCI to provide signals to the user: e.g., warnings, alarms, and status information. (Ironically, one of the most useful sounds for computer users is rather serendipitous: the noise made by many hard drives that lets a user know that the machine is still computing, rather than hung.) However, non-speech sound can also be a useful input channel, as sound made by users can be meaningful events in human-to-human communication—e.g., utterances such as “uh-huh” used in *backchannel* communication (communication events that occur in the background of an interaction, rather than being the main focus), a laugh, a sigh, or a clapping of hands.

- Haptic input and force feedback

Haptic, or touch-based, input devices measure pressure, velocity, location—essentially perceiving aspects of a user’s manipulative and explorative manual actions. These can be integrated into existing devices (e.g., keyboards and mice that know when they are being touched, and possibly by whom). Or they can exist as standalone devices, such as the well-known PHANTOM device by SensAble Technologies, Inc.³ (see Fig. 1.4), or the DELTA device by Force Dimension.⁴ These and most other haptic devices integrate force feedback and allow the user to experience the “touch and feel” of simulated artifacts as if they were real. Through the mediator of a hand-held stylus or probe, haptic exploration can now receive simulated feedback including rigid boundaries of virtual objects, soft tissue, and surface texture properties. A tempting goal is to simulate all haptic experiences and to be able to recreate objects with all their physical properties in virtual worlds so they can be touched and handled in a natural way. The tremendous dexterity of the human hand makes this very difficult. Yet, astonishing results can already be achieved, for example with the CyberForce device which can produce forces on each finger and the entire arm. The same company, Immersion Corp.⁵, also supplies the iDrive, a hybrid of a rotary knob and joystick input interface to board computers of BMW’s flagship cars. This is the first attempt outside the gaming industry to bring haptic and force-feedback interfaces to the general consumer.

- Computer vision

Computer vision has many advantages as an input modality for multimodal or perceptual interfaces. Visual information is clearly important in human-human communication, as meaningful information is conveyed through identity, facial expression, posture, gestures, and other visually observable cues. Sensing and perceiving these visual cues from video cameras appropriately placed in the environment is the domain of computer vision. The following section describes relevant computer vision technologies in more detail.

²<http://www.ascension-tech.com>

³<http://www.sensable.com>

⁴<http://www.forcedimension.com>

⁵<http://www.immersion.com>

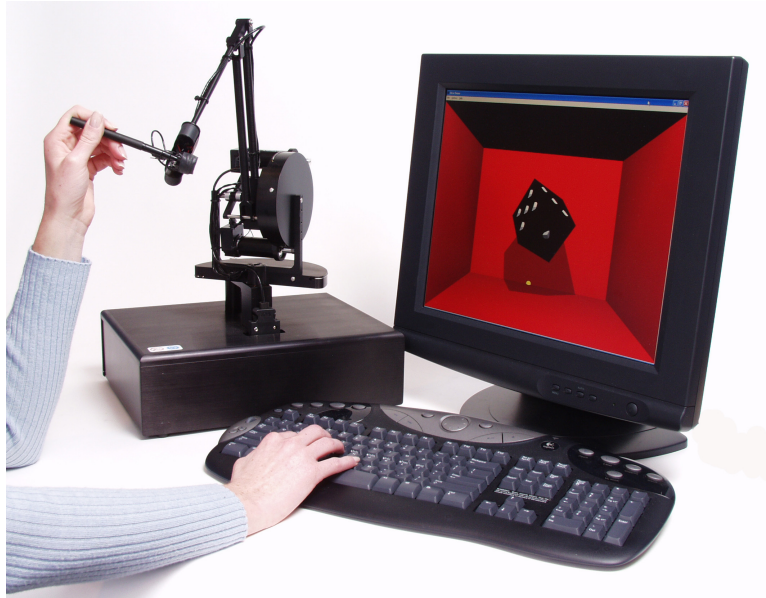


Figure 1.4. SensAble Technologies, Inc. PHANTOM haptic input/output device (reprinted with permission).

1.4 Vision Based Interfaces

Vision supports a wide range of human tasks, including recognition, navigation, balance, reading, and communication. In the context of perceptual interfaces, the primary task of computer vision (CV) is to detect and recognize meaningful visual cues to communication—i.e., to “watch the users” and report on their locations, expressions, gestures, etc. While vision is one of possibly several sources of information about the interaction to be combined multimodally in a perceptual interface, in this section we focus solely on the vision modality. Using computer vision to sense and perceive the user in an HCI context is often referred to as Vision Based Interaction, or Vision Based Interfaces (VBI).

In order to accurately model human interaction, it is necessary to take every observable behavior into account [69, 68]. The analysis of human movement and gesture coordinated with speech conversations has a long history in areas such as sociology, communication, and therapy (e.g., Scheffen and Birdwhistell’s Context Analysis [115]). These analyses, however, are often quite subjective and ill-suited for computational analysis. VBI aims to produce precise and real-time analysis that will be useful in a wide range of applications, from communication to games to automatic annotation of human-human interaction.

There is a range of human activity that has occupied VBI research over the past decade; Fig. 1.5 shows some of these from the camera’s viewpoint. Key aspects of VBI include the detection and recognition of the following elements:

- Presence and location – Is someone there? How many people? Where are they (in 2D or 3D)? [Face detection, body detection, head and body tracking]
- Identity – Who are they? [Face recognition, gait recognition]
- Expression – Is a person smiling, frowning, laughing, speaking...? [Facial feature tracking, expression modeling and analysis]
- Focus of attention – Where is a person looking? [Head/face tracking, eye gaze tracking]
- Body posture and movement – What is the overall pose and motion of the person? [Body modeling and tracking]
- Gesture – What are the semantically meaningful movements of the head, hands, body? [Gesture recognition, hand tracking]
- Activity – What is the person doing? [Analysis of body movement]

Surveillance and VBI are related areas with different emphases. Surveillance problems typically require less precise information and are intended not for direct interaction but to

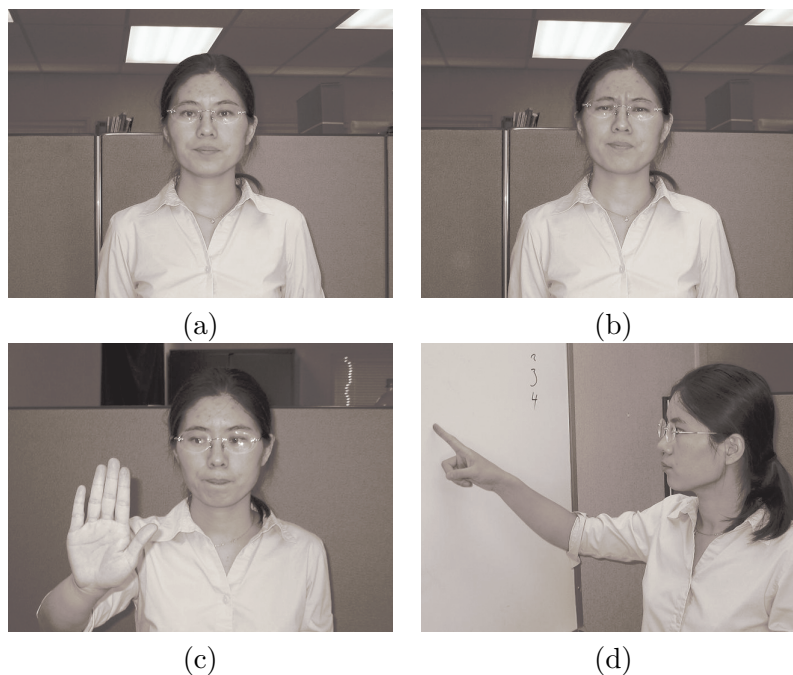


Figure 1.5. Some common visual cues for VBI. (a) User presence and identity. (b) Facial expression. (c) A simple gesture. (d) A pointing gesture and focus of attention.

record general activity or to flag unusual activity. VBI demands more fine-grained analysis, where subtle facial expressions or hand gestures can be very important.

These computer vision problems of tracking, modeling and analyzing human activities are quite difficult. In addition to the difficulties posed in typical computer vision problems by noise, changes in lighting and pose, and the general ill-posed nature of the problems, VBI problems add particular difficulties due to the fact that the objects to be modeled, tracked, and recognized are people, rather than simple, rigid, non-changing widgets. People change hairstyles, get sunburned, grow facial hair, wear baggy clothing, and in general make life difficult for computer vision algorithms. Robustness in the face of the variety of human appearances is a major issue in VBI research.

The main problem that computer vision faces is an overload of information. The human visual system effortlessly filters out unimportant visual information, attending to relevant details like fast moving objects, even if they are in the periphery of the visible hemisphere. But this is a very complex computational task. At the low level in human vision, a great deal of pre-processing is done in the retina in order to decrease the bandwidth requirements of the nervous channel into the visual cortex. At the high level, humans leverage *a priori* knowledge of the world in ways that are not well understood computationally. For example, a computer does not simply know that objects under direct sunlight cast sharp shadows. The difficult question is how to extract only relevant observations from the visual information so that vision algorithms can concentrate on a manageable amount of work. Researchers frequently circumvent this problem by making simplifying assumptions about the environment, which makes it possible to develop working systems and investigate the suitability of computer vision as a user interface modality.

In recent years, there has been increased interest in developing practical vision-based interaction methods. The technology is readily available, inexpensive, and fast enough for most real-time interaction tasks. CPU speed has continually increased following Moore's Law, allowing increasingly complex vision algorithms to run at frame rate (30 frames per second, *fps*). Figure 1.6 shows a history of available clock cycles per pixel of a VGA-sized video stream with a top-of-the-line CPU for the PC market over the last 35 years. Higher processing speeds, as well as the recent boom in digital imaging in the consumer market, could have far reaching implications for VBI. It is becoming more and more feasible to process large, high resolution images in near real-time, potentially opening the door for

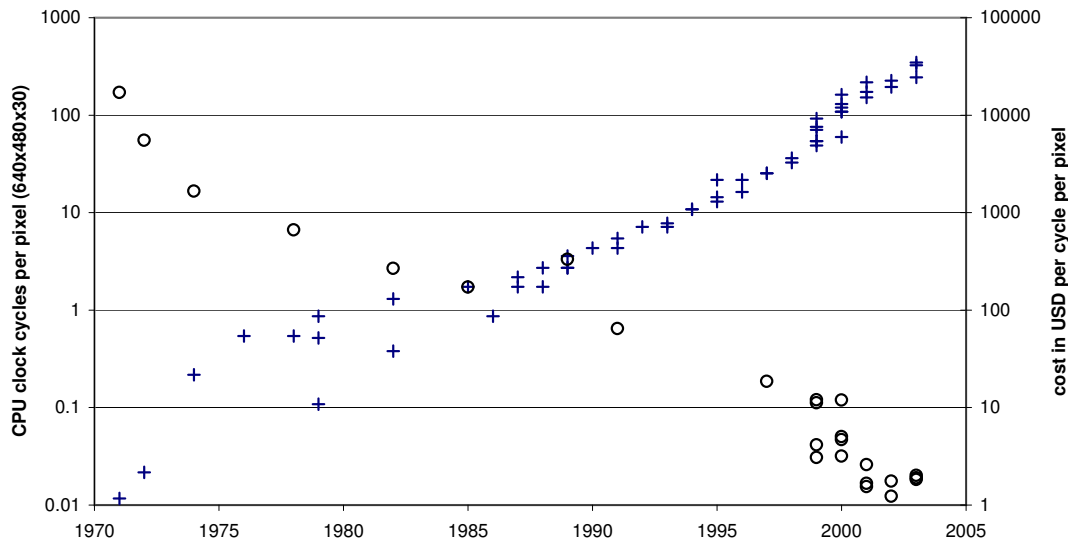


Figure 1.6. CPU processing power over the last 35 years. Each “+” data point denotes the release of the fastest CPU for the PC market from one of the major manufacturers. Multiple data points per year are shown if one manufacturer released multiple CPUs that year or competing manufacturers’ CPUs were released that year. The available clock cycles per pixel per frame of a video stream with 30 full frames per second of size 640x480 pixels determine the y-value. The “o” data points describe the cost in US \$ per MHz CPU speed.

numerous new applications and vision algorithms.

Fast, high-resolution digital image acquisition devices and fast processing power are only as effective as the link between them. The PC market has just recently seen a revolution in connector standards. Interface speeds to peripheral devices used to be orders of magnitude lower than the connection speed between the motherboard and internal devices. This was largely due to the parallel (32 bit or more) connector structure for internal boards and serial links to external devices. The introduction of Firewire (also called 1394 and i-Link) in 1995 and more recently USB 2.0 pushed interface speeds for peripheral devices into the same league as internal interfaces. While other high-speed interfaces for external devices exist (e.g., ChannelLink), they have not made inroads to the consumer market.

Using computer vision in human-computer interaction can enable interaction that is difficult or impossible to achieve with other interface modalities. As a picture is worth a thousand words to a person, a video stream may be worth a thousand words of explanation to a computer. In some situations, visual interaction is very important to human-human interaction—hence, people fly thousands of miles to meet face to face. Adding visual interaction to computer interfaces, if done well, may help to produce a similarly compelling user experience.

Entirely unobtrusive interfaces are possible with CV because no special devices must be worn or carried by the user (although special, easily-tracked objects may be useful in some contexts). No infrastructure or sensors need to be placed in the interaction space because cameras can cover a large physical range. In particular, no wires or active transmitters are required by the user. A camera operates extremely quietly, allowing input to a computer without disturbing the environment. Also, modern cameras can be very lightweight and compact, well-suited for mobile applications. Even in environments unsuitable for moving or exposed parts, cameras can be utilized since a camera can be completely encapsulated in its housing with a transparent window.

The cost of cameras and their supporting hardware and software has dropped dramatically in recent years, making it feasible to expect a large installed base in the near future. Software for image and video processing (e.g., for movie and DVD editing) has entered the consumer market and is frequently pre-installed, bundled with a computer’s operating system.

The versatility of a camera makes it reasonable and compelling to use as an interface

device. A camera may be used for several different purposes, sometimes simultaneously. For example, a single camera, affixed to a person's head or body, may function as a user interface device by observing the wearer's hands [73]; it can videotape important conversations or other visual memories at the user's request [58]; it can store and recall the faces of conversation partners and associate their names [112]; it can be used to track the user's head orientation [140, 2]; it can guide a museum visitor by identifying and explaining paintings [111].

As a particular example of VBI, hand gesture recognition offers many promising approaches for interaction. Hands can operate without obstructing high-level activities of the brain such as sentence-forming, thus being a good tool for interface tasks while thinking. Generating speech, on the other hand, is said to take up general-purpose brain resources, impeding the thought process [66]. Hands are very dextrous physical tools, and their capabilities have been quite successfully employed in the HCI context in conjunction with devices such as the keyboard and mouse. Human motor skills are, in many cases, easily trained to execute new tasks with high precision and incredible speeds. With the aid of computer vision, we have the chance to go beyond the range of activities that simple physical devices can capture and instead to let hands gesture with all their capabilities. The goal is to leverage the full range of both static hand postures and dynamic gestures in order to communicate (purposefully or otherwise) and perhaps to command and control. Data gloves accomplish some of this goal, yet they have an unnatural feel and are cumbersome to the user.

1.4.1 Terminology

This subsection reviews the essential terminology relevant to vision based interaction. Current VBI tasks focus on modeling (and detecting, tracking, recognizing, etc.) one or more body parts. These parts could be a face, a hand, a facial feature such as an eye, or an entire body. We will interchangeably call this the object of focus, the feature of attention, or simply the "body part."

Determining the presence (or absence) of an object of focus is the problem of *detection* and has so far primarily been applied to people detection [43] and face detection [155, 49]. Strictly speaking, the output is binary ("person present" versus "no person present"), but typically the location of the object is also reported. Object *localization* is sometimes used to describe the special case that the presence of an object is assumed and its location is to be determined at a particular point in time. *Registration* refers to the problem of aligning an object model to the observation data, often both object position and orientation (or *pose*). Object *tracking* locates objects and reports their changing pose over time [43].

Although tracking can be considered as a repeated frame-by-frame detection or localization of a feature or object, it usually implies more than discontinuous processing. Various methods improve tracking by explicitly taking temporal continuity into account and using prediction to limit the space of possible solutions and speed up the processing. One general approach uses filters to model the object's temporal progression. This can be as simple as linear smoothing, which essentially models the object's inertia. A Kalman filter [63] assumes a Gaussian distribution of the motion process and can thus also model non-constant movements. The frequently used Extended Kalman Filter (EKF) relieves the necessity of linearity. Particle filtering methods (or sequential Monte Carlo methods [34], and frequently called Condensation [55] in the vision community) make no assumptions about the characteristics of underlying probability distributions but instead sample the probability and build an implicit representation. They can therefore deal with non-Gaussian processes and also with multi-modal densities (caused by multiple, statistically independent sources) such as arise from object tracking in front of cluttered backgrounds. In addition to the filtering approach, different algorithms can be used for initialization (detection) and subsequent tracking. For example, some approaches detect faces with a learning-based approach [139] and then track with a shape and color based head tracker [8].

Recognition (or *identification*) involves comparing an input image to a set of models in a database. A recognition scheme usually determines confidence scores or probabilities that defines how closely the image data fits each model. Detection is sometimes called recognition, which makes sense if there are very different classes of objects (faces, cars, and books) and one of them (faces) must be recognized. A special case of recognition is

verification or *authentication*, which judges whether the input data belongs to one particular identity with very high confidence. An important application of verification is in biometrics, which has been applied to faces, fingerprints, and gait characteristics [17].

A *posture* is a static configuration of the human body—for example, sitting and thinking or holding a coffee cup or pen. *Gestures* are dynamic motions of the body or body parts, and can be considered as temporally consecutive sequences of postures. Hand and arm gestures in particular are covered extensively in the social sciences literature, especially in conversational and behavioral psychology [36, 69, 86, 87, 108]. As a result, the term gesture is often used to refer to the semantic interpretation that is associated with a particular movement of the body (e.g., happiness associated with a smile). We limit our attention in this chapter to a mostly syntactic view of “gesture” and “gesture recognition,” leaving the difficult problem of semantic interpretation and context [26, 86] to others.

Facial gestures are more commonly called *facial expressions*. Detecting and tracking facial features are typically the first steps of facial expression analysis, although holistic appearance-based approaches may also be feasible. Subsequent steps try to recognize known expressions (e.g., via FACS action units [37]) and to infer some meaning from them, such as the emotional state of the human [6, 9].

The parameter set for a rigid body consists of its *location* (x, y, z) in 3D space and its *orientation* (rx, ry, rz) with respect to a fixed coordinate frame. Deformable objects such as human faces require many parameters for an accurate description of their form, as do articulated objects such as human bodies. An object’s *appearance* describes its color and brightness properties at every point on its surface. Appearance is caused by texture, surface structure, lighting, and view direction. Since these attributes are view-dependent, it only makes sense to talk about appearance from a given viewpoint.

The *view sphere* is an imaginary sphere around the object or scene of interest. Every surface point of the sphere defines a different view of the object. When taking perspective into account, the object’s appearance changes even for different distances, despite a constant viewing angle. Vision techniques that can detect, track, or recognize an object regardless of the viewing angle are called *view-independent*; those that require a certain range of viewing angles for good performance, for example frontal views for face tracking, are called *view-dependent*.

1.4.2 Elements of VBI

VBI techniques apply computer vision to specific body parts and objects of interest. Depending on the application and environmental factors, it may be most useful to search for or track the body as a whole, individual limbs, a single hand, or an artifact such as a colored stick held by the user. VBI techniques for different body parts and features can complement and aid each other in a number of ways.

1. Coarse-to-fine hierarchy: Here, in a sequence of trackers which are targeted to successively more detailed objects, each stage benefits from the results of the previous stage by drastically reducing the search space. For example, whole-body detection limits the search space for face detection. After the face is detected in a scene, the search for particular facial features such as eyes can be limited to small areas within the face. After the eyes are identified, one might estimate the eye gaze direction [129].
2. Fine-to-coarse hierarchy: This approach works in the opposite way. From a large number of cues of potential locations of small features, the most likely location of a larger object that contains these features is deduced. For example, feature-based face recognition methods make use of this approach. They first try to identify prominent features such as eyes, nose, and lips in the image. This may yield many false positives, but knowledge about the features’ relative locations allows the face to be detected. Bodies are usually attached to heads at the same location: thus, given the position of a face, the search for limbs, hands, etc., becomes much more constrained and therefore simpler [91].
3. Assistance through more knowledge: The more elements in a scene that are modeled, the easier it is to deal with their interactions. For example, if a head tracking interface also tracks the hands in the image, although it does not need their locations directly,

it can account for occlusions of the head by the hands (e.g., [146]). The event is within its modeled realm, whereas otherwise occlusions would constitute an unanticipated event, diminishing the robustness of the head tracker.

Face tracking in particular is often considered a good anchor point for other objects [135, 24], since faces can be fairly reliably detected based on skin color or frontal view appearance.

Person-level, Whole Body & Limb Tracking

VBI at the person-level has probably produced the most commercial applications to date. Basic motion sensing is a perfect example how effective VBI can be in a constrained setting: the scene is entirely stationary, so that frame-to-frame differencing is able to detect moving objects. Another successful application is in traffic surveillance and control. A number of manufacturers offer systems that automate or augment the push button for pedestrian crossings. The effectiveness of this technology was demonstrated in a study [137] that compared the number of people crossing an intersection during the “Don’t Walk” signal with and without infrastructure that detected people in the waiting area on the curb or in the crossing. The study found that in all cases these systems can significantly decrease the number of dangerous encounters between people and cars.

Motion capture has frequently been used in the film industry to animate characters with technology from companies such as Adtech, eMotion, Motion Analysis, and Vicon Motion Systems. In optical motion capture, many IR-reflecting markers are placed on an artist’s or actor’s body. Typically more than five cameras observe the acting space from different angles, so that at least two cameras can see each point at any time. This allows for precise reconstruction of the motion trajectories of the markers in 3D and eventually the exact motions of the human body. This information is used to drive animated models and can result in much more natural motions than those generated automatically. Other common applications of motion capture technology include medical analysis and sports training (e.g., to analyze golf swings or tennis serves).

Detecting and tracking people passively using computer vision, without the use of markers, has been applied to motion detection and other surveillance tasks. In combination with artificial intelligence, it is also possible to detect unusual behavior, for example in the context of parking lot activity analysis [47]. Some digital cameras installed in classrooms and auditoriums are capable of following a manually selected person or head through pan-tilt-zoom image adjustments. Object-based image encoding such as defined in the MPEG-4 standard is an important application of body tracking technologies.

The difficulties of body tracking arise from the many degrees of freedom of the human body. Adults have 206 bones which are connected by over 230 joints. Ball and socket joints such as in the shoulder and hip have three degrees of freedom (DOF): They can abduct and adduct, flex and extend, and rotate around the limb’s longitudinal axis. Hinge joints have one DOF and are found in the elbow and between the phalanges of hands and feet. Pivot joints also allow for one DOF; they allow the head as well as radius and ulna to rotate. The joint type in the knee is called a condylar joint. It has two DOF because it allows for flexion and extension and for a small amount of rotation. Ellipsoid joints also have two DOF, one for flexion and extension, the other for abduction and adduction (e.g., the wrist’s main joint and the metacarpophalangeal joint as depicted in Fig. 1.8). The thumb’s joint is a unique type, called a saddle joint; in addition to the two DOF of ellipsoid joints, it also permits a limited amount of rotation. The joints between the human vertebrae each allow limited three DOF motion, and all together they are responsible for the trunk’s flexibility.

Recovering the degrees of freedom for all these joints is an impossible feat for today’s vision technology. Body models for CV purposes must therefore abstract from this complexity. They can be classified by their dimensionality and by the amount of knowledge versus learning needed to construct them.

The frequently used 3D kinematic models have between 20 and 30 DOF. Figure 1.7 shows an example, in which the two single-DOF elbow- and radioulnar joints have been combined into a two-DOF joint at the elbow. In fact, the rotational DOF of the shoulder joint is often transferred to the elbow joint. This is because the humerus shows little evidence of its rotation, while the flexed lower arm indicates this much better. Similarly,

the rotational DOF of the radioulnar joint can be attributed to the wrist. This transfer makes a hierarchical model parameter estimation easier.

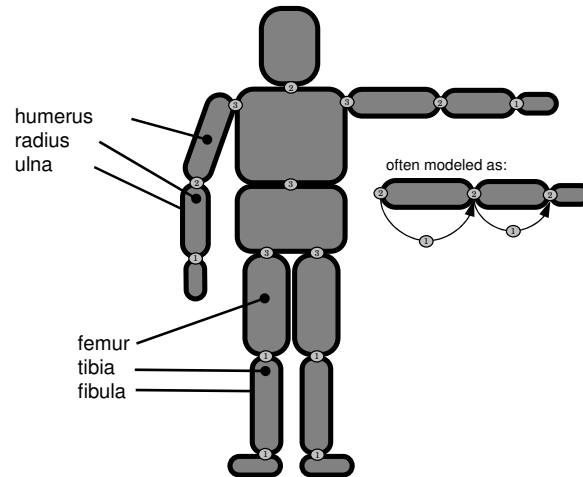


Figure 1.7. The “sausage link man” shows the structure of a 3D body model. The links can have cylindrical shape, but especially the trunk is more accurately modeled with a shape with non-circular cross section.

Most of the vision-based efforts to date have concentrated on detecting and tracking people while walking, dancing, or performing other tasks in a mostly upright posture. Pedestrian detection, for example, has seen methods employed that had previously shown success in face detection, such as wavelets [99] and a combination of depth information and a learning method [162].

Two systems with comprehensive functionality, Pfinder [145] and W4 [48], both show well how CV must be tailored to the task and properties of the particular environment. First, they rely on a static camera mounting, which gives the opportunity to model the background and achieve fast and reliable segmentation of moving foreground objects. Second, they make assumptions of the body posture; namely, they expect a mostly upright person. This can be easily distinguished from other moving objects such as cars or wind-blown objects. Third, heuristics about the silhouettes enable classification of a few typical postures or actions such as carrying an object, making use of the fact that only a small number of scenarios are likely for a person entering the field of view.

Hands

Hands are our most dextrous body parts, and they are heavily used in both manipulation and communication. Estimation of the hands’ configuration is extremely difficult due to the high degrees of freedom and the difficulties of occlusion. Even obtrusive data gloves⁶ are not able to acquire the hand state perfectly. Compared with worn sensors, CV methods are at a disadvantage. With a monocular view source, it is impossible to know the full state of the hand unambiguously for all hand configurations, as several joints and finger parts may be hidden from the camera’s view. Applications in VBI have to keep these limitations in mind and focus on obtaining information that is relevant to gestural communication, which may not require full hand pose information.

Generic hand detection is a largely unsolved problem for unconstrained settings. Systems often use color segmentation, motion flow, and background subtraction techniques, and especially a combination of these, to locate and track hands in images. In a second step and in settings where the hand is the prominent object in view, a shape recognition or appearance-based method is often applied for hand posture classification.

Anatomically, the hand is a connection of 18 elements: the five fingers with three elements each, the thumb-proximal part of the palm, and the two parts of the palm that extend

⁶Data gloves are gloves with sensors embedded in them that can read out the fingers’ flexion and abduction. Their locations and orientations in 3D space are often tracked with supplemental means such as electromagnetic trackers.

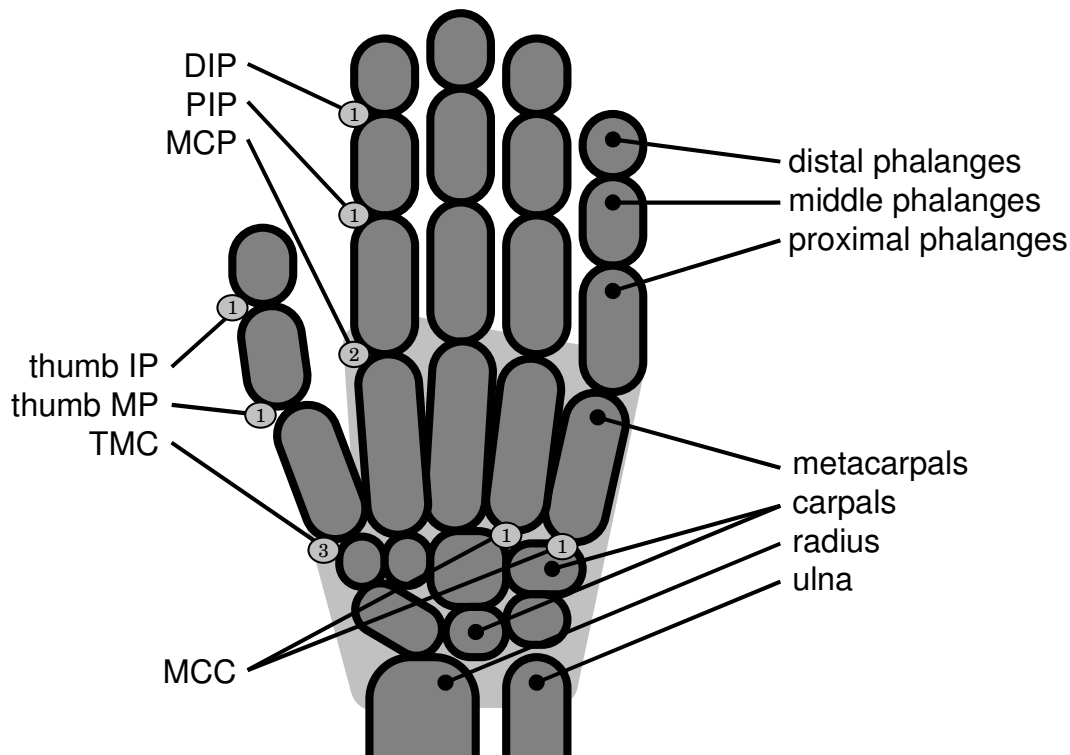


Figure 1.8. The structure of the hand. The joints and their degrees of freedom: distal interphalangeal joints (DIP, 1 DOF), proximal interphalangeal joints (PIP, 1 DOF), metacarpophalangeal joints (MCP, 2 DOF), metacarpocarpal joints (MCC, 1 DOF for pinky and ring fingers), thumb's interphalangeal joint (IP, 1 DOF), thumb's metacarpophalangeal joint (MP, 1 DOF), and thumb's trapeziometacarpal joint (TMC, 3 DOF).

from the pinky and ring fingers to the wrist (see Fig. 1.8). The 17 joints that connect the elements have one, two, or three degrees of freedom (DOF). There are a total of 23 DOF, but for simplicity the joints inside the palm are frequently ignored as well as the rotational DOF of the trapeziometacarpal joint, leaving 20 DOF. Each hand configuration is a point in this 20-dimensional configuration space. In addition, the hand reference frame has 6 DOF (location and orientation). See Braffort et al. [12] for an exemplary anthropomorphic hand model.

It is clearly difficult to automatically match a hand model to a point in such a high-dimensional space for posture recognition purposes. Lin et al. [80] suggest limiting the search to the interesting subspace of natural hand configurations and motions, and they define three types of constraints. Type I constraints limit the extent of the space by considering only anatomically possible joint angles for each joint (see also earlier work by Lee and Kunii [77]). Type II constraints reduces the dimensionality by assuming direct correlation between DIP and PIP flexion. Type III constraints limit the extent of the space again by eliminating generally impossible configurations and unlikely transitions between configurations. With a seven-dimensional space they cover 95% of configurations observed in their experiments.

An introduction to the state of the art in hand modeling and recognition can be found in a survey by Wu and Huang [152]. One of the early papers that described whole hand tracking and posture classification as real-time input was written by Fukumoto et al. [42]. They moved a cursor around on a projection screen by making a pointing hand posture and moving the hand within a space observed from two cameras. Two different postures can be distinguished (thumb up and down) with various interpretations to control a VCR and to draw. The paper also deals with the problem of estimating the pointing direction. Cutler and Turk [23] use a rule-based system for gesture recognition in which image feature are extracted by optical flow. The location and trajectory of the hand(s) constitutes the input to various simple interfaces such as controlling musical instruments. Mysliwiec [94, 107] tracks the hands by detecting the hands anew in every frame based on a skin color model.

Then a simple, hand-specific heuristic is used to classify the posture and find the index fingertip. Freeman and Roth [41] use histograms of edge orientations in hand images to distinguish different gestures. Moghaddam and Pentland [90] apply a density estimation to the PCA-transformed edge images and obtain a scale-invariant shape similarity measure. GREFIT [98] uses fingertip locations in 2D images to deduce possible 3D configurations of view-dependent hand images with an underlying anatomical hand model with Type I constraints. Zhu et al. [163] combine color segmentation with frame differencing to find hand-like objects, using higher-level dynamic models together with shape similarity based on image moments to distinguish gestures. They observed that it was easier and more reliable to classify gestures based on their motion trajectory than on finger configurations.

View independence is a significant problem for hand gesture interfaces. Wu and Huang [151] compared a number of classifiers for their suitability to view-independent hand posture classification.

The above approaches operate in the visible light spectrum and do not use any auxiliary aids such as marked gloves. Segmenting the hand from the background is much simplified by using infrared (IR) light. Since the skin reflects near-infrared light well, active IR sources placed in proximity to the camera in combination with an IR pass filter on the lens make it easy to locate hands that are within range of the light source. Dorfmueller-Ulhaas and Schmalstieg [32] use special equipment: users must wear gloves with infrared-reflecting markers, the scene is illuminated with IR light sources, and a pair of cameras is used for stereo vision. The system's accuracy and robustness are quite high even with cluttered backgrounds. It is capable of delivering the accuracy necessary to grab and move virtual checkerboard figures.

Hand and Arm Gestures in 4D

The temporal progression of hand gestures, especially those that accompany speech, are generally composable into three stages: pre-stroke, stroke, and post-stroke [86]. The pre-stroke prepares the movement of the hand. The hand waits in this ready state until the speech arrives at the point in time when the stroke is to be delivered. The stroke is often characterized by a peak in the hand's velocity and distance from the body. The hand is retracted during the post-stroke, but this phase is frequently omitted or strongly influenced by the gesture that follows (similar to coarticulation issues in speech processing).

Automatic sign language recognition has long attracted vision researchers. It offers enhancement of communication capabilities for the speech-impaired and deaf, promising improved social opportunities and integration. For example, the signer could wear a head-mounted camera and hand a device to his or her conversation partner that displays the recognized and translated text. Alternatively, a text-to-speech module could be used to output the sign language interpretation. Sign languages exist for several dozen spoken languages, such as American English (ASL), British English, French, German, and Japanese. The semantic meanings of language components differ, but most of them share common syntactic concepts. The signs are combinations of hand motions and finger gestures, frequently augmented with mouth movements according to the spoken language. Hand motions are distinguished by the spatial motion pattern, the motion speed, and in particular by which body parts the signer touches at the beginning, during, or the end of a sign. The finger configuration during the slower parts of the hand movements is significant for the meaning of the gesture. Usually, uncommon words can be spelled out as a concatenation of letter symbols and then be assigned to a context-dependent symbol for more efficient signing. Trained persons achieve speeds that equal that of conversational speech.

Most CV methods applicable to the task of sign language recognition have extracted feature vectors composed of hand location and contour. These feature vectors have their temporal evolution and variability in common with feature vectors stemming from audio data; thus, tools applied to the speech recognition domain may be suited to recognizing the visual counterpart of speech as well. An early system for ASL recognition [127] fed such a feature vector into a Hidden Markov Model (HMM) and achieved high recognition rates for a small vocabulary of mostly unambiguous signs from a constrained context. However, expansion into a wider semantic domain is difficult; the richness of syntactic signs is a big hurdle for a universal sign language recognition system. The mathematical methods that perform well for speech recognition need adaptation to the specifics of spatial data

with temporal variability. What is more, vision-based recognition must achieve precision in two complementary domains: very fast tracking of the position of the hand in 3D space and also exact estimation of the configuration of the fingers of the hand. To combine these requirements in one system is a major challenge. The theoretical capabilities of sign language recognition—assuming the CV methods are fast and precise enough—can be evaluated with glove-based methods in which research has a longer history (e.g., [93, 79]).

Wu and Huang [150] present a review of recognition methods for dynamic gestures up to 1999. Overall, vision-based hand gesture recognition has not yet advanced to a stage where it can be successfully deployed for user interfaces in consumer-grade applications. The big challenges are robustness, user independence, and some measure of view independence.

Head and Face

Head and face detection and tracking contributes an essential component to vision based interfaces. Heads and faces can safely be presumed to be present and visible for almost all kinds of human tasks. Heads are rarely occluded entirely, and they convey a good deal of information about the human, such as identity and focus of attention. In addition, this is an attractive area of research in computer vision because the appearance variability of heads and faces is limited yet complex enough to touch on many fundamental problems of CV. Methods that perform well on head or face detection or tracking may also perform well on other objects. This area of VBI has therefore received the most attention and its maturity can be observed by the existence of standard evaluation methods (test databases), the availability of software tools, and commercial developments and products. This progress raises the question of whether at least parts of the problem are solved to a degree that computers can satisfactorily perform these tasks.

Applications of the technology include face detection followed by face recognition for biometrics, for example to spot criminals at airports or to verify access to restricted areas. The same technology can be useful for personalized user interfaces, for example to recall stored car seat positions, car stereo volume, and car phone speed dial lists depending on the driver. Head pose estimation and gaze direction have applications for video conferencing. A common problem occurs when watching the video of one's interlocutor on a screen while the camera is *next to* the monitor. This causes an apparent offset of gaze direction which can be disturbing if eye contact is expected. CV can be used to correct for this problem [44, 156]. Head tracking has been used for automatically following and focusing on a speaker with fixed-mounted pan/tilt/zoom cameras. Future applications could utilize face recognition to aid human memory and attempts are already being made to use face detection and tracking for low bandwidth, object-based image coding. Face tracking is usually a prerequisite for efficient and accurate locating of facial features and expression analysis.

Face tracking methods can be characterized along two dimensions: whether they track a planar face or a 3D face, and whether they assume a rigid or a deformable face model. The usual tradeoffs apply: a model with more degrees of freedom (DOF) is harder to register with the image, but it can be more robust. For example, it may explicitly handle rotations out of the image plane. Planar methods can only deal with limited shape and appearance variation caused by out-of-plane rotations, for instance by applying learning methods. Fixed shape and appearance models such as polygons [130], ellipses [8], cylinders [74], and ellipsoids [84], are efficient for coarse head tracking, especially when combined with other image features such as color [8]. Models that can describe shape and/or appearance variations have the potential to yield more precise results and handle varying lighting conditions and even sideways views. Examples for 2D models are Snakes [147], Eigenfaces [134, 103], Active Shape Models [21] and Active Appearance Models (AAM) [20, 154], Gabor and other wavelets [75, 104, 160, 38], and methods based on Independent Component Analysis (ICA). 3D model examples are 3D AAM [33], point distribution models [45, 78], and meshes [27, 157].

The major difficulties for face detection arise from in-plane (tilted head, upside down) and out-of-plane (frontal view, side view) rotations of the head, facial expressions (see below), facial hair, glasses, and, as with all CV methods, lighting variation and cluttered backgrounds. There are several good surveys of head and face VBI⁷, including face detection

⁷See also the chapter *Face Detection and Recognition* in this volume.

[155, 49], face recognition [153, 46], and face tracking [54].

Facial Expression Analysis, Eye Tracking

Facial expressions are an often overlooked aspect of human-human communication. However, they make a rich contribution to our everyday life. Not only can they signal emotions and reactions to specific conversation topics, but on a more subtle level, they also regularly mark the end of a contextual piece of information and help in turn-taking during a conversation. In many situations, facial gestures reveal information about a person's true feelings, while other bodily signals can be artificially distorted more easily. Specific facial actions, such as mouth movements and eye gaze direction changes, have significant implications. Facial expressions serve as interface medium between the mental states of the participants in a conversation.

Eye tracking has long been an important facility for psychological experiments on visual attention. The quality of results of automatic systems that used to be possible only with expensive hardware and obtrusive head-mounted devices is now becoming feasible with off-the-shelf computers and cameras, without the need for head-mounted structures.⁸ Application deployment in novel environments such as cars is now becoming feasible.

Face-driven animation has begun to make an impact in the movie industry, just as motion capture products did a few years earlier. Some systems still require facial markers, but others operate without markers. The generated data is accurate enough to animate a virtual character's face with ease and with a degree of smoothness and naturalness that is difficult to achieve (and quite labor intensive) with conventional, scripted animation techniques.



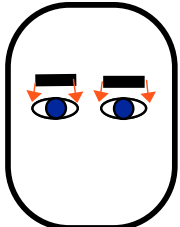
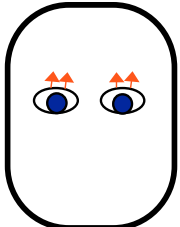
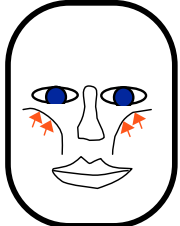
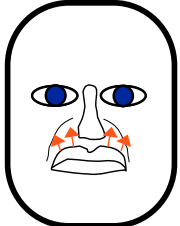
AU	Facial expression	FACS description	AU	Facial expression	FACS description
1		inner brow raiser	2		outer brow raiser
4		brow lower	5		upper lid raiser
6		brow lower	10		upper lip raiser

Figure 1.9. Some of the action units of the Facial Action Coding System (FACS) [37].

Facial expressions in general are an important class of human motion that behavioral psychology has studied for many decades. Much of the computer vision research on facial expression analysis has made use of the Facial Action Coding System (FACS) [37] (see

⁸See, for example, the Arrington Research, Inc. ViewPoint EyeTracker remote camera system (<http://www.tobii.se>).

Fig. 1.4.2), a fine-grained classification of facial expressions. It describes on the order of 50 individual “action units” (AU) such as raising the upper lip, stretching the lips, or parting the lips, most of which are oriented on a particular facial muscle and its movements. Some AU are combinations of two muscles’ movements, and some muscles have more than one AU associated with them if contracting different parts of the muscle results in very different facial expressions. Expressions such as a smile are composed of one or more AU. AU thus do not carry semantic meaning, they only describe physical deformations of skin and facial tissues. FACS was originally developed so that human observers could succinctly describe facial expressions for use in behavioral psychology. While many degrees of freedom allow for precise distinction of even the most subtle facial notions—even distinction between natural and purposefully forced smiles is said to be possible—the high expressiveness also poses additional challenges over less fine-grained classifications, such as the semantics-oriented classification into facial expressions caused by emotions (for example happiness and anger).

For eye tracking, a coarse-to-fine approach is often employed. Robust and accurate performance usually necessitates brief user training and constant lighting conditions. Under these conditions, the eye gaze direction can be accurately estimated despite moderate head translations and rotations.

According to a survey by Donato et al. [31], most approaches to automatic analysis of facial expression are either based on optical flow, global appearance, or local features. The authors re-implemented some of the most promising methods and compared their performance on one data set. Optical flow methods (such as [85]) were used early on to analyze the short term, dynamical evolution of the face. Naturally, these require an image sequence of the course of a facial expression, and do not apply to static images, due to the lack of motion information. These methods were found to perform significantly better when the image is not smoothed in a pre-processing step. This leads to the conclusion that small image artifacts are important when it comes to tracking and/or recognizing facial gestures. Another hint that high spatial frequencies contribute positively stems from comparing the performance of Gabor wavelets. Using only high-frequencies had a less detrimental effect than using only low-frequencies, compared to the baseline of unrestricted Gabor filters.

Approaches using Gabor filters and Independent Component Analysis (ICA), two methods based on spatially constrained filters, outperformed other methods such as those using PCA or Local Feature Analysis [102] in the investigation by Donato et al. [31]. The authors conclude that local feature extraction and matching is very important to good performance. This alone is insufficient, however, as the comparison between global and local PCA-based methods showed. Good performance with less precise feature estimation can be achieved with other methods (e.g., [161]).

Global methods that do not attempt to separate the individual contributors to visual appearances seem to be ill-suited to model multi-modal distributions. Refinement of mathematical methods for CV tasks, such as ICA, appears to be promising in order to achieving high accuracy in a variety of appearance-based applications such as detection, tracking, and recognition. Optical flow and methods that do not attempt to extract FACS action units are currently better suited to the real-time demands of VBI.

Handheld Objects

Vision based interfaces that detect and track objects other than human body parts—that is, handheld objects used in interaction—have a high potential for successful commercial development in the transition period from traditional to perceptual interfaces. Tracking such objects can be achieved more reliably than tracking high DOF body parts, and they may be easier to use than free-form gestures. Handheld artifacts can be used in much the same way as one’s hands, for example to make pointing gestures, to perform rhythmic commands, or to signal the spatial information content of sign languages. Possible useful objects include passive wands [143], objects with active transmitters such as LEDs⁹, and specially colored objects—in fact, anything that is easily trackable with CV methods. An alternative approach to having fixed-mounted cameras and tracking moving objects is to embed the camera in the moving object and recognize stationary objects in the environment

⁹Head trackers can utilize this technology to achieve high accuracy and short latencies. One commercial product is the “Precision Position Tracker,” available at <http://www.worldviz.com>.

or egomotion to enable user interaction. Since detecting arbitrary objects is very hard, fiducials can make this reverse approach practical [72].

1.4.3 Computer Vision Methods for VBI

In order for a computer vision method to be suitable for VBI, its performance must meet certain requirements with respect to speed, precision, accuracy, and robustness. A system is said to experience real-time behavior if no delay is apparent between an event (occurrence) and its effect. Precision concerns the repeatability of observations for identical events. This is particularly important for recognition tasks and biometrics: only if the VBI consistently delivers the same result for a given view can this information be used to identify people or scenes. Accuracy describes the deviation of an observation from ground truth. Ground truth must be defined in a suitable format. This format can, for example, be in the image domain, in feature space, or it can be described by models such as a physical model of a human body. It is often impossible to acquire ground truth data, especially if no straightforward translation from observation to the ground truth domain is known. In that case, gauging the accuracy of a VBI method may be quite difficult.

Robustness, on the other hand, is easier to determine by exposing the VBI to different environmental conditions, including different lighting (fluorescent and incandescent lighting and sunlight), different users, cluttered backgrounds, occlusions (by other objects or self-occlusion), and non-trivial motion. Generally, the robustness of a vision technique is inversely proportional to the amount of information that must be extracted from the scene.

With currently available hardware only a very specific set of fairly fast CV techniques can be used for truly interactive VBI. One of the most important steps is to identify constraints on the problem (regarding the environment or the user) in order to make simplifying assumptions for the CV algorithms. These constraints can be described by various means. Prior probabilities are a simple way to take advantage of likely properties of the object in question, both in image space and in feature space. When these properties vary significantly, but the variance is not random, principal component analysis, neural networks and other learning methods frequently do a good job in extracting these patterns from training data. Higher-level models can also be employed to limit the search space in the image or feature domain to physically or semantically possible areas.

Frequently, interface-quality performance can be achieved with multi-modal or multi-cue approaches. For example, combining the results from a stereo-based method with those from optical flow may overcome the restrictions of either method used in isolation. Depending on the desired tradeoff between false positives and false negatives, early or late integration (see Fig. 1.3) lends itself to this task. Application- and interface-oriented systems must also address issues such as calibration or adaptation to a particular user, possibly at runtime, and re-initialization after loss of tracking. Systems tend to become very complex and fragile if many hierarchical stages rely on each other. Alternatively, flat approaches (those that extract high-level information straight from the image domain) do not have to deal with scheduling many components, feedback loops from higher levels to lower levels, and performance estimation at each of the levels. Robustness in computer vision systems can be improved by devising systems that do not make irrevocable decisions in the lower layers but instead model uncertainties explicitly. This requires modeling of the relevant processes at all stages, from template matching to physical object descriptions and dynamics.

All computer vision methods need to specify two things. First, they need to specify the mathematical or algorithmic tool used to achieve the result. This can be PCA, HMM, a neural network, etc. Second, the domain to which this tool is applied must be made explicit. Sometimes this will be the raw image space with gray scale or color pixel information, and sometimes this will be a feature space that was extracted from the image by other tools in a pre-processing stage. One example would be using wavelets to find particular regions of a face. The feature vector, composed of the image-coordinate locations of these regions, is embedded in the domain of all possible region locations. This can serve as the input to an HMM-based facial expression analysis.

Edge and Shape Based Methods

Shape properties of objects can be used in three different ways. Fixed shape models such as an ellipse for head detection or rectangles for body limb tracking minimize the summative

energy function from probe points along the shape. At each probe, the energy is lower for sharper edges (in the intensity or color image). The shape parameters (size, ellipse foci, rectangular size ratio) are adjusted with efficient, iterative algorithms until a local minimum is reached. On the other end of the spectrum are edge methods that yield unconstrained shapes. Snakes [67] operate by connecting local edges to global paths. From these sets, paths are selected as candidates for recognition that resemble a desired shape as much as possible. In between these extremes lie the popular statistical shape models, e.g., the Active Shape Model (ASM) [21]. Statistical shape models learn typical deformations from a set of training shapes. This information is used in the recognition phase to register the shape to deformable objects. Geometric moments can be computed over entire images or alternatively over select points such as a contour.

These methods require sufficient contrast between the foreground object and the background, which may be unknown and cluttered. Gradients in color space [119] can alleviate some of the problems. Even with perfect segmentation, non-convex objects are not well suited for recognition with shape-based methods since the contour of a concave object can translate into a landscape with many, deep local minima in which gradient descent methods get stuck. Only near-perfect initialization allows the iterations to descend into the global minimum.

Color Based Methods

Compelling results have been achieved merely by using skin color properties, e.g., to estimate gaze direction [116] or for interface-quality hand gesture recognition [71]. This is because the appearance of skin color varies mostly in intensity while the chrominance remains fairly consistent [114]. According to Zarit et al. [159], color spaces that separate intensity from chrominance (such as the HSV color space) are better suited to skin segmentation when simple threshold-based segmentation approaches are used. However, some of these results are based on a few images only, while more recent work examined a huge number of images and found an excellent classification performance with a histogram-based method in RGB color space as well [61]. It seems that simple threshold methods or other linear filters achieve better results in HSV space, while more complex methods, particularly learning-based, nonlinear models do well in any color space. Jones et al. [61] also state that Gaussian mixture models are inferior to histogram-based approaches, which makes sense given the multi-modality of random image scenes and the fixed amount of Gaussians available to model this. The CAMShift algorithm (Continuously Adaptive Mean Shift) [11, 19] is a fast method to dynamically parameterize a thresholding segmentation which is able to deal with a certain amount of lighting and background changes. Together with other image features such as motion, patches or blobs of uniform color, this makes for a fast and easy way to segment skin-colored objects from backgrounds.

Infrared Light: One “color” is particularly well suited to segment human body parts from most backgrounds, and that is energy from the infrared (IR) portion of the EM spectrum. All objects constantly emit heat as a function of their temperature in form of infrared radiation, which are electromagnetic waves in the spectrum from about $700nm$ (visible red light) to about $1mm$ (microwaves). The human body emits the strongest signal at about $10\mu m$, which is called long wave IR or thermal infrared. Not many common background objects emit strongly at this frequency in modest climates, therefore it is easy to segment body parts given a camera that operates in this spectrum. Unfortunately, this requires very sensitive sensors that often need active cooling to reduce noise. While the technology is improving rapidly in this field, the currently easier path is to actively illuminate the body part with short wave IR. The body reflects it just like visible light, so the illuminated body part appears much brighter than background scenery to a camera that filters out all other light. This is easily done for short wave IR because most digital imaging sensors are sensitive this part of the spectrum. In fact, consumer digital cameras require a filter that limits the sensitivity to the visible spectrum to avoid unwanted effects. Several groups have used IR in VBI-related projects (e.g., [113, 32, 133]).

Color information can be used on its own for body part localization, or it can create attention areas to direct other methods, or it can serve as a validation and “second opinion” about the results from other methods (multi-cue approach). Statistical color as well as location information is thus often used in the context of Bayesian probabilities.

Motion Flow and Flow Fields

Motion flow is usually computed by matching a region from one frame to a region of the same size in the following frame. The motion vector for the region center is defined as the best match in terms of some distance measure (e.g., least-squares difference of the intensity values). Note that this approach is parameterized by both the size of the region (“feature”) as well as the size of the search neighborhood. Other approaches use pyramids for faster, hierarchical flow computation; this is especially more efficient for large between-frame motions. The most widely used feature tracking method is the “KLT” tracker. The tracker and the selection of good features to track (usually corners or other areas with high image gradients) are described by Shi and Tomasi [120]. KLT trackers have limitations due to the constancy assumption (no change in appearance from frame to frame), match window size (aperture problem), and search window size (speed of moving object, computational effort).

A flow field describes the apparent movement of entire scene components in the image plane over time. Within these fields, motion blobs are defined as pixel areas of (mostly) uniform motion, i.e., motion with similar speed and direction. Especially in VBI setups with static camera positions, motion blobs can be very helpful for object detection and tracking. Regions in the image that are likely locations for a moving body part direct other CV methods and the VBI in general to these “attention areas.”

The computational effort for tracking image features between frames increases dramatically with lower frame rates since the search window size has to scale according to the tracked object’s estimated maximal velocity. Since motion flow computation can be implemented as a local image operation that does not require a complex algorithm or extrinsic state information (only the previous image patch and a few parameters), it is suited for on-chip computation. Comprehensive reviews of optical flow methods can be found in Barron et al. [5] and Mitiche and Bouthemy [89].

Texture and Appearance Based Methods

Information in the image domain plays an important role in every object recognition or tracking method. This information is extracted to form image features: higher-level descriptions of what was observed. The degree of abstraction of the features and the scale of what they describe (small, local image artifacts or large, global impressions) have a big impact on the method’s characteristics. Features built from local image information such as steep gray-level gradients are more sensitive to noise; they need a good spatial initialization and frequently a large collection of those features is required. Once these features are found, they need to be brought into context with each other, often involving an iterative and computationally expensive method with multiple, interdependent and thus more fragile, stages.

If instead the features are composed of many more pixels, cover a larger region in the image, and abstract to more complex visuals, the methods are usually better able to deal with clutter and might flatten the hierarchy of processing levels (since they already contain much more information than smaller-scale features). The benefits do not come without a price, in this case increased computational requirements.

Appearance based methods attempt to identify the patterns that an object frequently produces in images. The simplest approach to comparing one appearance to another is to use metrics such as least squared difference on a pixel-by-pixel basis, i.e., the lowest-level feature vector. This is not very efficient and is too slow for object localization or tracking. The key is to encode as much information as possible in an as small as possible feature vector—to maximize the entropy.

One of the most influential procedures uses a set of training images and the Karhunen-Loève transform [65, 81]. This transformation is an orthogonal basis rotation of the training space that maximizes sample variance along the new basis vectors and is frequently known in the computer vision literature as principal component analysis (PCA) [59] and is directly related to singular value decomposition (SVD). In the well-known Eigenfaces approach, Turk and Pentland [134] applied this method to perform face detection and recognition, extending the work by Kirby and Sirovich for image representation and compression [70]. Active Appearance Models (AAMs) [20] encode shape and appearance information in one model, built in a two-step process with PCA. Active Blobs [118] are similar to AAM. In these approaches, the observed object appearance steers object tracking by guiding initialization

for subsequent frames, similar to the concept of the Kalman filter. During training, a parameterization is learned that correlates observation-estimation error with translational offsets.

A Gabor wavelet is a sine wave enveloped by a Gaussian, modeled after the function of the human visual cortical cell [106, 60, 29, 25]. Wavelets are well suited to encode both spatial and frequency information in a compact, parameterized representation. This alleviates problems of FFT approaches where all local spatial information is lost. Feris et al. [117] showed good face tracking performance with a hierarchical wavelet network, a collection of collections of wavelets. Each feature is represented by a set of wavelets, enabling tracking in the manner of KLT trackers, but comparatively more robust to lighting and deformation changes.

Another approach to learn and then test for common appearances of objects is to use neural networks; however, in some cases their performance (in terms of speed and accuracy) has been surpassed by other methods. One extremely fast detection procedure proposed by Viola and Jones [139] has attracted much attention. In this work, very simple features based on intensity comparisons between rectangular image areas are combined by Ada-boosting into a number of strong classifiers. These classifiers are arranged in sequence and achieve excellent detection rates on face images with a low false positive rate. The primary advantage of their method lies in the constant-time computation of features that have true spatial extent, as opposed to other techniques that require time proportional to the area of the extent. This allows for very high speed detection of complex patterns at different scales. However, the method is rotation-dependent.

A more complete review of appearance-based methods for detection and recognition of patterns (faces) can be found in Yang et al.'s survey on face detection [155].

Background Modeling

Background modeling is often used in VBI to account for (or subtract away) the non-essential elements of a scene. There are essentially two techniques: segmentation by thresholding and dynamic background modeling.

Thresholding requires that the foreground object has some unique property that distinguishes it from all or most background pixels. For example, this property can be foreground brightness, so that a pixels with values above a particular gray scale intensity threshold are classified as foreground, and values below as belonging to the background. Color restrictions on the background are also an effective means for simple object segmentation. There, it is assumed that the foreground object's color does not appear very frequently or in large blobs in the background scene. Of course, artificial coloring of the foreground object avoids problems induced by wildly varying or unknown object colors—e.g., using a colored glove for hand detection and tracking.

Dynamic Background Modeling requires a static camera position. The values of each pixel are modeled over time with the expectation to find a pattern of values that this pixel assumes. The values may be described by a single contiguous range, or it may be multi-modal (two or more contiguous ranges). If the value suddenly escapes these boundaries that the model describes as typical, the pixel is considered part of a foreground object that temporarily occludes the background. Foreground objects that are stationary for a long time will usually be integrated into the background model over time and segmentation will be lost. The mathematics to describe the background often use statistical models with one or more Gaussians [48].

Temporal Filtering and Modeling

Temporal filtering typically involves methods that go beyond motion flow to track on the object or feature level, rather than at the pixel or pattern level. For example, hand and arm gesture recognition (see subsection *Hand and Arm Gestures in 4D* on page 19) requires temporal filtering.

Once the visual information has been extracted and a feature vector has been built, general physics-based motion models are often used. For example, Kalman filtering in combination with a skeletal model can deal with resolving simple occlusion ambiguities [146]. Other readily available mathematical tools can be applied to the extracted feature vectors, independently of the preceding CV computation. Kalman filtering takes advantage

of smooth movements of the object of interest. At every frame, the filter predicts the object's location based on the previous motion history. The image matching is initialized with this prediction, and once the object is found, the Kalman parameters are adjusted according to the prediction error.

One of the limitations of Kalman filtering is the underlying assumption of a Gaussian probability. If this is not the case, and the probability function is essentially multi-modal as it is the case for scenes with cluttered backgrounds, Kalman filtering cannot cope with the non-Gaussian observations. The particle filtering or factored sampling method, often called Condensation (conditional density propagation) tracking, has no implicit assumption of a particular probability function but rather represents it with a set of sample points of the function. Thus, irregular functions with multiple "peaks"—corresponding to multiple hypotheses for object states—can be handled without violating the method's assumptions. Factored sampling methods [55] have been applied with great success to tracking of fast-moving, fixed-shape objects in very complex scenes [76, 28, 14]. Various models, one for each typical motion pattern, can improve tracking, as shown by Isard and Blake [56]. Partitioned sampling reduces the computational complexity of particle filters [82]. The modeled domain is usually a feature vector, combined from shape-describing elements (such as the coefficients of B-splines) and temporal elements (such as the object velocities).

Dynamic gesture recognition, i.e., recognition and semantic interpretation of continuous gestures and body motions, is an essential part of perceptual interfaces. Temporal filters exploit motion information only to improve tracking, while the following methods aim at detecting meaningful actions such as waving a hand for goodbye.

Discrete approaches can perform well at detecting spatio-temporal patterns [51]. Most methods in use, however, are borrowed from the more evolved field of speech recognition due to the similarity of the domains: multi-dimensional, temporal, and noisy data. Hidden Markov Models (HMMs) [15, 158] are frequently employed to dissect and recognize gestures due to their suitability to processing temporal data. However, the learning methods of traditional HMMs cannot model some structural properties of moving limbs very well.¹⁰ Brand [13] uses another learning procedure to train HMMs that overcomes these problems. This allows for estimation of 3D model configurations from a series of 2D silhouettes and achieves excellent results. The advantage of this approach is that no knowledge has to be hard-coded but instead everything is learned from training data. This has its drawbacks when it comes to recognizing previously unseen motion.

Higher-level Models

To model a human body or its parts very precisely such as is required for computer graphics applications, at least two models are necessary. One component describes the kinematic structure of the skeleton, the bone connections and the joint characteristics. Even complex objects such as the entire human body or the hand can thus—with reasonable accuracy—be thought of as a kinematic chain of rigid objects. The second component describes the properties of the flesh around the bones, either as a surface model or as a volumetric model. Very complex models that even describe the behavior of skin are commonplace in the animation industry. Additional models are frequently used to achieve even greater rendering accuracy, such as models of cloth or hair.

Computer vision can benefit from these models to a certain degree. A structural, anatomical 3D model of the human body has advantages over a 2D model because it has explicit knowledge of the limbs and the fact that they can occlude each other from a particular view [110]. Given a kinematic model of the human body (e.g., [77]), the task to estimate the limb locations becomes easier compared to the case when the knowledge of interdependency constraints is lacking. Since the locations and orientations of the individual rigid objects (limbs, phalanges) are constrained by their neighboring chain links, the effort to find them in the image decreases dramatically.

It is important for VBI applications to exploit known characteristics of the object of interest as much as possible. A kinematic model does exactly that, as do statistical models that capture the variation in appearance from a given view point or the variation of shape

¹⁰Brand [13] states that the traditional learning methods are not well suited to model state transitions since they do not improve much on the initial guess about connectivity. Estimating the structure of a manifold with these methods thus is extremely suboptimal in its results.

(2D) or form (3D). Often the model itself is *ad hoc*, that is, it is manually crafted by a human based on prior knowledge of the object. The model's parameter range is frequently learned from training data.

Higher-level models describe properties of the tracked objects that are not immediately visible in the image domain. Therefore a translation between image features and model parameters needs to occur. A frequently used technique is *analysis by synthesis* in which the model parameters are estimated by an iterative process. An initial configuration of the model is back-projected into the image domain. The difference between projection and observation then drives adjustment of the parameters, following another back-projection into the image domain and so forth until the error is sufficiently small (e.g., see Kameda et al. [64] and Ström et al. [130]). These methods lack the capability to deal with singularities that arise from ambiguous views [92]. When using more complex models that allow methods from projective geometry to be used to generate the synthesis, self-occlusion is modeled again and thus it can be dealt with. Stenger et al. [128]. use a recently proposed modification of the Kalman filter, the so-called an “unscented Kalman filter” [62] to align the model with the observations. Despite speed and accuracy advantages over more traditional approaches, the main drawbacks of all Kalman-based filters however is that they assume a unimodal distribution. This assumption is most likely violated by complex, articulated objects such as the human body or the hand.

Particle filtering methods for model adjustment are probably better suited for more general applications because they allow for any kind of underlying distribution, even multi-modal distributions. Currently, their runtime requirements are not immediately suitable for real-time operation yet, but more efficient modifications of the original algorithm are available.

Real-time Systems, Frame Rate, Latency, Processing Pipelines

Most user interfaces require real-time responses from the computer: for feedback to the user, to execute commands immediately or both. But what exactly does real-time mean for a computer vision application?

There is no universal definition for real-time in computer terms. However, a system that responds to user input is said to operate in real-time if the user of the system perceives no delay between action command and action. Hence, real-time pertains to a system's ability to respond to an event without noticeable delay. The opposite would be a delayed response, a response after a noticeable processing time. Mouse input, for example, is usually processed in real-time; that is, a mouse motion is immediately visible as a mouse pointer movement on the screen. Research has shown that delays as low as 50ms are noticeable for visual output [142, 83]. However, people are able to notice audible delays of just a few milliseconds since this ability is essential to sound source localization.

The terms *frame rate* and *latency* are well-suited to describe a CV system's performance. The (minimum, maximum, average) frame rate determines how many events can be processed per time unit. About five frames per second is the minimum for a typical interactive system. The system latency, on the other hand, describes the time between the event occurrence and the availability of the processed information. As mentioned above, about 50ms is tolerable for a system's performance to be perceived as real-time.

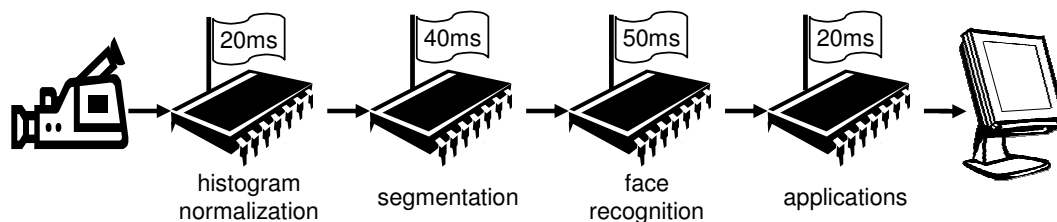


Figure 1.10. A four-stage pipeline processing of a video stream.

To illustrate the difference between frame rate and latency, imagine a pipeline of four processors as shown in Fig. 1.10. The first one grabs a frame from the camera and performs

histogram normalization. The normalized image frame is input to the second processor which segments the image into regions based on motion flow and color information, and then outputs the frame to the third processor. This processor applies an AAM for face recognition. The recognized face is input to the last processor which performs a saliency test and the uses the information to drive various applications. Altogether this process takes $20ms + 40ms + 50ms + 20ms = 130ms$ per frame. This is the latency: input is available after $130ms$ as output. The frame rate is determined by the pipeline's bottleneck, the third processor. A processed frame is available at the system output every $50ms$, that is, the frame rate is maximal $20fps$.

If the input occurs at a rate higher than $20fps$, there are two options for pipelined systems:

A) Every frame is processed. A 10 second input sequence with $30fps$ for example has 300 frames. Thus it requires $300f/20fps = 15s$ to process them all. The first frame is available $130ms$ after its arrival, thus the last frame is available as output $5s130ms$ after its input. It also means that there must be sufficient buffer space to hold the images, in our example for $5s * 30fps = 150$ frames. It is obvious that a longer input sequence increases the latency of the last frame and the buffer requirements. This model is therefore not suitable to real-time processing.

B) Frames are dropped somewhere in the system. In our pipeline example, a $30fps$ input stream is converted into at most a $20fps$ output stream. Valid frames (those that are not dropped) are processed and available in at most a constant time. This model is suitable to real-time processing.

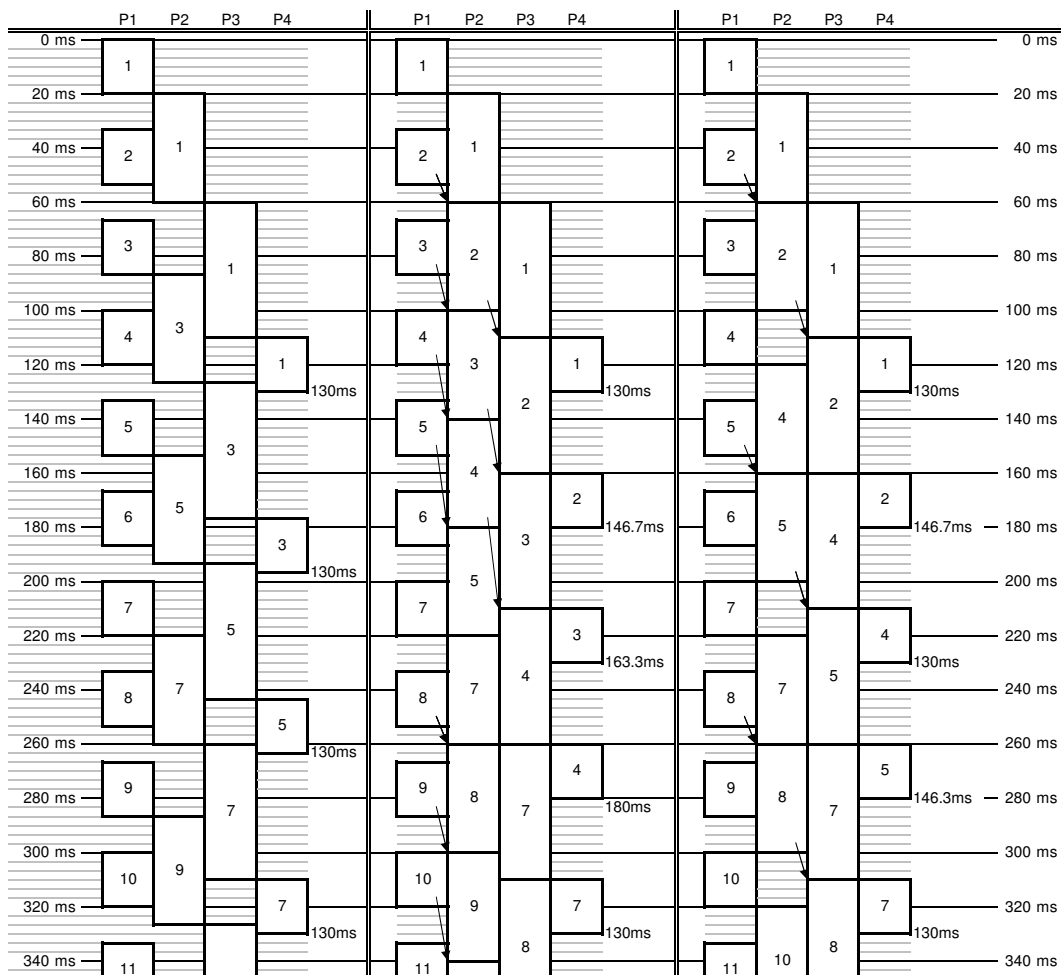


Figure 1.11. An example of pipelined processing of frames, with and without buffering inside the pipeline. Buffering is indicated with arrows. The example on the right uses feedback from stage 3 to avoid unnecessary processing and increased latency.

Dropping frames, however, brings about other problems. There are two cases: First, there is no buffer available anywhere in the system, not even for partial frames. This is shown on the left hand side in Fig. 1.11. In our example, the first processor has no problems keeping up with the input pace, but the second processor will still be working on the first frame when it receives the second frame $33.3ms$ after arrival of the first. The second frame needs to be dropped. Then the second processor would idle for $2 * 33.3ms - 40ms = 26.7ms$. Similar examples can be constructed for the subsequent pipeline stages.

Second, frames can be buffered at each pipeline stage output or input. This is shown in the center drawing of Fig. 1.11. It assumes no extra communication between the pipeline stages. Each subsequent stage requests a frame as soon as it completed processing one frame, and the preceding stage keeps the latest processed frame around in a buffer until it can replace it with the next processed frame. In the example, the second frame would be buffered for $6.7ms$ at the transition to the second processor, the third frame for $2 * 40ms - 2 * 33.3ms = 13.3ms$, the fourth for $3 * 40ms - 3 * 33.3ms = 20ms$, etc. The highest latency after the 2nd stage would be for every fifth frame, which is buffered for a total of $5 * 40ms - 5 * 33.3ms = 33.3ms$ before processing. Then in fact the next processed frame (frame number 7) is finished being processed already and is sent to stage 3 instead of the buffered frame, which is dropped. These latencies can accumulate throughout the pipeline. The frame rate is maximal ($20fps$) in this example, but some of the frames' latencies have increased dramatically.

The most efficient method in terms of system latency and buffer usage facilitates pipeline feedback from the later stages to the system input, as shown on the right in Fig. 1.11. The feedback is used to adjust every stages' behavior in order to maximize utilization of the most time critical stage in the pipeline. Alternatively to the scenario shown, it is also possible to buffer frames only buffered before the first stage, and feed them into the system with a speed that the pipeline's bottleneck stage can keep up with. This is not shown, but it completely avoids the need for in-pipeline buffering. Note that even in this efficient scenario with pipeline feedback the increased frame rates in comparison to the leftmost case are bought with an increase in average latency.

Examples of architectures that implement component communication and performance optimizing scheduling are the "Quality-Control Management" in the DirectShow subsystem of Microsoft's DirectX technology and the "Modular Flow Scheduling Middleware" [40].

1.4.4 VBI Summary

Vision-Based Interfaces have numerous applications; the potential of VBI has only begun to be explored. But computational power is getting to a stage where it can handle the vast amounts of data of live video streams. Progress has been made in many relevant areas of computer vision; many methods have been demonstrated that begin to provide human-computer interface quality translation of body actions into computer commands. While a large amount of work is still required to improve the robustness of these methods, especially in modeling and tracking highly articulated objects, the community has begun to take steps towards standardizing interfaces of popular methods and providing toolkits for increasingly higher level tasks. These are important steps in bringing the benefits of VBI to a wider audience.

The number of consumer-grade commercial applications of computer vision has significantly increased in recent years, and this trend will continue driven by ongoing hardware progress. To advance the state of the art of VBI—at the intersection of the disciplines of CV and HCI—it is vital to establish evaluation criteria, such as benchmarks for the quality and speed of the underlying methods and the resulting interfaces. Evaluation databases must be made accessible for all components of VBI (such as those already available for faces), both for static images and increasingly dynamic data for real-time video processing.

1.5 Brain-Computer Interfaces

Perhaps the ultimate interface to computers would be a direct link to the thoughts and intentions of the user, a "Your wish is my command" model of interaction, involving no physical action or interpretation of any kind. While this kind of mind-reading technology is not likely to be developed in the foreseeable future, the nascent research area of

Brain-Computer Interfaces (BCI) is perhaps a step in this direction. BCI technology attempts to perceive commands or control parameters by sensing relevant brain activity of the user. While not fitting completely within the perceptual interface model of natural human-human interaction, BCI may eventually be an integral component of perceptual interfaces. The computer vision community's extensive experience with learning, statistical, and other pattern recognition methods and techniques can be of tremendous value to this new field.

A Brain-Computer Interface does not depend on the brain's normal output channels of peripheral nerves and muscles, but instead measures electrical activity either at the scalp or in the cortex. By measuring the electroencephalographic (EEG) activity at the scalp, certain features of the EEG signal can be used to produce a control signal. Alternatively, implanted electrodes can be used to measure the activity of individual cortical neurons or an array of neurons. These technologies have primarily been targeted to be used by people with neuromuscular impairments that prevent them from communicating via conventional methods. In recent years, researchers have begun to consider more general uses of the technologies. A review by Wolpaw et al. [144] notes that while the rising interest in BCI technologies in recent years has produced exciting developments with considerable promise, they are currently low-bandwidth devices with a maximum information transfer rate of 10 to 25 bits per minute, and this rate is likely to improve only gradually.

Wolpaw et al. argue that, in order to make progress in brain-computer interfaces, researchers must understand that BCI is not simply mind reading or "wire-tapping" the brain, determining a person's thoughts and intentions by listening in on brain activity. Rather, BCI should be considered as a new output channel for the brain, one that is likely to require training and skill to master.

"Brain-Machine Interface" [97] is the traditional term as it grew out of initial uses of the technology: to interface to prosthetic devices. Sensors were implanted primarily in motoric nerves in the extremities and a one-to-one function was typically used to map the sensor outputs to actuator control signals. "Brain-Computer Interface" more accurately captures the necessity for computational power between the neuro-sensors and the controlled devices or application-specific software. As the sensors increasingly move into the brain (intracortical electrodes) and target not only motoric nerves but generic neurons, the mapping from neuron activity to (desired, normal, or pathologic) output becomes less direct. Complex mathematical models translate the activity of many neurons into a few commands—computational neuroscience focuses on such models and their parameterizations. Mathematical models that have proven to be well suited to the task of replicating human capabilities, in particular the visual sense, seem to perform well for BCIs as well—for example, particle and Kalman filters [149]. Two feature extraction and classification methods frequently used for BCIs are reviewed in Ebrahimi et al. [35].

Figure 1.12 schematically explains the principles of a BCI for prosthetic control. The independent variables, signals from one or many neural sensors, are processed with a mathematical method and translated into the dependent variables, spatial data that drives the actuators of a prosthetic device. Wolpaw et al. [144] stress that BCI should eventually comprise three levels of adaptation. In the first level, the computational methods (depicted in the right upper corner of Fig. 1.12) are trained to learn the correlation between the observed neural signals and the user's intention for arm movement. Once trained, the BCI then must translate new observations into actions. We quote from [144]: "However, EEG and other electro-physiological signals typically display short- and long-term variations linked to time of day, hormonal levels, immediate environment, recent events, fatigue, illness, and other factors. Thus, effective BCIs need a second level of adaptation: periodic online adjustments to reduce the impact of such spontaneous variations."

Since the human brain is a very effective and highly adaptive controller, adaptation on the third level means to benefit from the combined resources of the two adaptive entities brain and BCI. As the brain adapts to the demands and characteristics of the BCI by modifying its neural activity, the BCI should detect and exploit these artifacts and communicate back to the brain that it appreciates the effort, for example through more responsive, more precise, or more expressive command execution. This level of adaptation is difficult to achieve, but promises to yield vastly improved performance.

The number of monitored neurons necessary to accurately predict a task such as 3D arm movement is open to debate. Early reports employed open-loop (no visual feedback

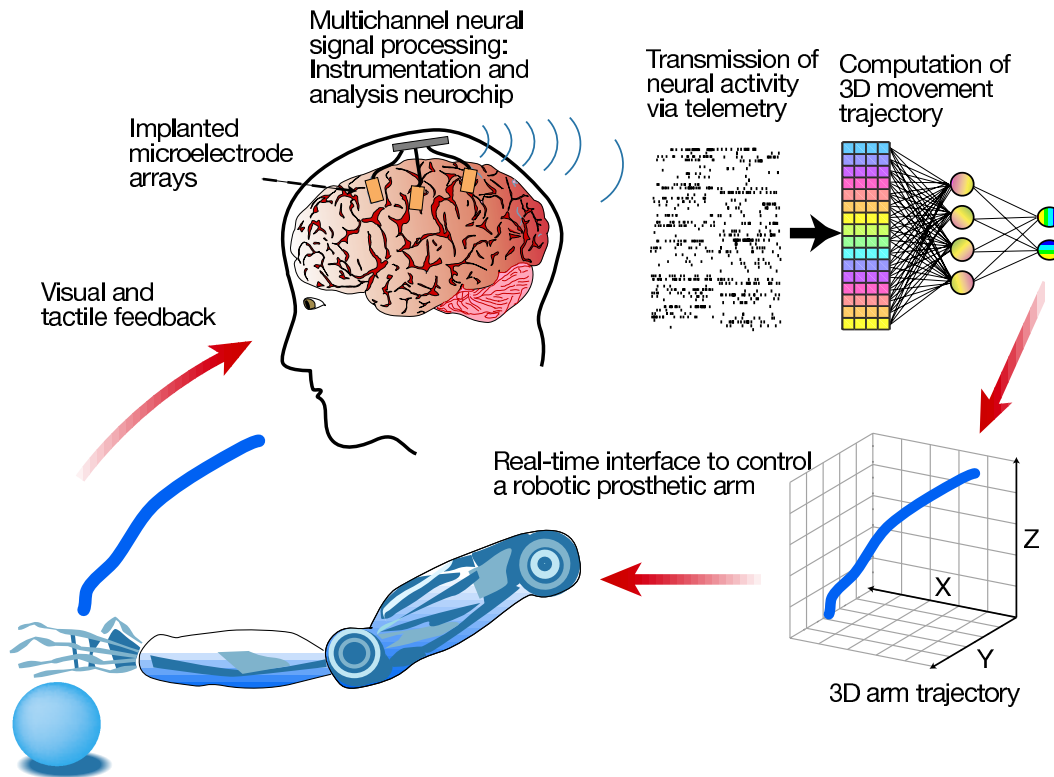


Figure 1.12. The control path of a closed-loop BCI. Figure reprinted with permission from Nicolelis et al. [97].

to the study subject) experiments with offline model building and parameterization. Those studies suggest by extrapolation that between 400 and 1350 neurons are necessary, depending on the brain area in which the sensors are implanted [141]. A more recent study by Taylor et al. provided real-time visual feedback and repeatedly updated the mathematical model underlying the translation function from neurons to the controlled object [132, 97]. They used only 18 neurons to achieve sufficient performance for a 2D cursor task, with the closed-loop method being significantly superior to the open-loop method. Currently, up to about 100 neurons can be recorded simultaneously. All currently used electro-physiological artifacts can be detected with a temporal resolution of $10ms$ to $100ms$, but some develop only over the course of many seconds.

In addition to the “input” aspect of BCIs, there are several examples of the reverse technology: computers connecting into the sensorimotor system providing motor output to the human (see [22]). Well-known examples include heart pace makers and cochlear implants, which directly stimulate auditory nerves, obviating the need for a mechanical hearing mechanism. Another device is able to prevent tremors caused by Parkinson’s disease or “Essential Tremor” by blocking erroneous nervous signals from reaching the thalamus where they would trigger involuntary muscle contractions [88].

1.6 Summary

The topic of perceptual interfaces is very broad, covering many technologies and their applications in advanced human-computer interaction. In this chapter we have attempted to give an overview of perceptual interfaces and go a bit deeper into how the field of computer vision can contribute to the larger goal of natural, adaptive, multimodal, interactive interfaces. Vision based interaction (VBI) is useful in itself, providing information about human identity, location, movement, and expression through non-invasive and non-intrusive methods. VBI has many near-term application areas, including computer games, accessibility, intelligent environments, biometrics, movement analysis, and social robots.

If the technical goal of building perceptual interfaces can be achieved to any reasonable degree, the ways in which people interact with computers—and with technology in general—

will be transformed significantly. In addition to computer vision, this will require advances in many areas, including speech and sound recognition and synthesis, natural language processing, user modeling, haptic and tangible interfaces, and dialogue modeling. More difficult yet, it will require collaboration and integration among these various research areas. In recent years, several workshops and conferences have begun to focus on these issues, including the Workshop on Perceptual/Perceptive User Interfaces (PUI), the International Conference on Multimodal Interfaces (ICMI), and International Conference on Intelligent User Interfaces (IUI). In addition, large major conference that attract a wide variety of participants—such as CHI and SIGGRAPH—now frequently showcase perceptual interface research or demonstrations.

As the separate technical communities continue to interact and work together on these common goals, there will be a great need for multimodal data sets for training and testing perceptual interfaces, with task data, video, sound, etc., and associated ground truth. Building such a database is not an easy task. The communities will also need standard benchmark suites for objective performance evaluation, similar to those that exist for individual modalities of speech, fingerprint, and face recognition. Students need to be trained to be conversant with multiple disciplines, and courses must be developed to cover various aspects of perceptual interfaces.

The fact that perceptual interfaces have great promise but will require herculean efforts to reach technical maturity leads to the question of short- and medium-term viability. One possible way to move incrementally toward the long-term goal is to “piggyback” on the current paradigm of graphical user interfaces. Such a “strawman perceptual interface” could start by adding just a few new events in the standard event stream that is part of typical GUI-based architectures. The event stream receives and dispatches events of various kinds: mouse movement, mouse button click and release, keyboard key press and release, window resize, etc. A new type of event—a “perceptual event”—could be added to this infrastructure that would, for example, be generated when a person enters the visual scene in front of the computer; or when a person begins to speak; or when the machine (or object of interest) is touched; or when some other simple perceptual event takes place. The benefit of adding to the existing GUI event-based architecture is that thousands upon thousands of developers already know how to deal with this architecture and how to write event handlers that implement various functionality. Adding even a small number of perceptual events to this structure would allow developers to come up with creative novel uses for them, and help lead to their acceptance in the marketplace.

This proposed development framework leads raises several questions. Which perceptual events would be most useful and feasible to implement? Is the event-based model the best way to bootstrap perceptual interfaces? Can we create perceptual events that are reliable enough to be useful? How should developers think about non-deterministic events (as opposed to current events, which are for all practical purposes deterministic)? For example, will visual events work when the lights are turned off, or if the camera lens is obstructed?

There are numerous issues, both conceptual and practical, surrounding the idea of perceptual interfaces. Privacy is one of the utmost importance. What are the implications of having microphones, cameras, and other sensors in computing environments? Where does the data go? What behavioral parameters are stored or sent elsewhere? To have any chance of success, these issues must be dealt with directly, and it must be made clear to users exactly where the data goes (and does not go). Acceptance of perceptual interfaces depends on instilling confidence that one’s privacy is not violated in any way.

Some argue against the idea of interface technologies that attempt to be intelligent or anthropomorphic, claiming that HCI should be characterized by direct manipulation, providing the user with predictable interactions that are accompanied by a sense of responsibility and accomplishment [121, 122, 123, 126, 125]. While these arguments seem quite appropriate for some uses of computers—particularly when a computer is used as a tool for calculations, word processing, and the like—it appears that future computing environments and uses will be well suited for adaptive, intelligent, agent-based perceptual interfaces.

Another objection to perceptual interfaces is that they just won’t work, that the problems are too difficult to be solved well enough to be useful. This is a serious objection—the problems are, indeed, very difficult. It would not be so interesting otherwise. In general, we

subscribe to the “If you build it, they will come” school of thought. Building it is a huge and exciting endeavor, a grand challenge for a generation of researchers in multiple disciplines.

BIBLIOGRAPHY

- [1] Elisabeth Andr and Thomas Rist. Presenting through performing: on the use of multiple lifelike characters in knowledge-based presentation systems. In *5th Intl. Conference on Intelligent User Interfaces*, pages 1–8. ACM Press, 2000.
- [2] Ronald Azuma, Jong Weon Lee, Bolan Jiang, Jun Park, Suya You, and Ulrich Neumann. Tracking in Unprepared Environments for Augmented Reality Systems. *ACM Computers & Graphics*, 23(6):787–793, December 1999.
- [3] G. Ball and J. Breese. Emotion and Personality in a Conversational Character. In *Workshop on Embodied Conversational Characters*, pages 83–84, October 1998.
- [4] G. Ball, D. Ling, D. Kurlander, J. Miller, D. Pugh, T. Skelly, A. Stankosky, D. Thiel, M. van Dantzich, and T. Wax. Lifelike computer characters: the persona project at Microsoft. In J. Bradshaw, editor, *Software Agents*. AAAI/MIT Press, 1997.
- [5] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of Optical Flow Techniques. *Int. Journal of Computer Vision*, 12(1):43–77, 1994.
- [6] J. N. Bassili. Emotion recognition: The role of facial movement and the relative importance of upper and lower areas of the face. *Journal of Personality and Social Psychology*, 37:2049–2059, 1979.
- [7] J. Biggs and M. A. Srinivasan. Haptic interfaces. In K. Stanney, editor, *Handbook of Virtual Environments*. Lawrence Earlbaum, Inc., 2002.
- [8] Stan Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 232–237, June 1998.
- [9] Michael J. Black and Yaser Yacoob. Recognizing Facial Expressions in Image Sequences, Using Local Parameterized Models of Image Motion. *Int. Journal of Computer Vision*, 25(1):23–48, 1997.
- [10] R. A. Bolt. Put-That-There: Voice and Gesture in the Graphics Interface. *Computer Graphics, ACM SIGGRAPH*, 14(3):262–270, 1980.
- [11] Gary R. Bradski. Real-time face and object tracking as a component of a perceptual user interface. In *IEEE Workshop on Applications of Computer Vision*, pages 142–149, 1998.
- [12] Annelies Braffort, Christophe Collet, and Daniel Teil. Anthropomorphic model for hand gesture interface. In *Proceedings of the CHI '94 conference companion on Human factors in computing systems*, April 1994.
- [13] Matthew Brand. Shadow Puppetry. In *ICCV*, 1999.
- [14] Lars Bretzner, Ivan Laptev, and Tony Lindeberg. Hand Gesture Recognition using Multi-Scale Colour Features, Hierarchical Models and Particle Filtering. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, pages 423–428, Washington D.C., 2002.
- [15] H. Bunke and T. Caelli, editors. *Hidden Markov Models in Vision*, volume 15(1) of *International Journal of Pattern Recognition and Artificial Intelligence*. World Scientific Publishing Company, 2001.

- [16] Justine Cassell. Embodied conversational interface agents. *Communications of the ACM*, 43(4):70–78, 2000.
- [17] T. Choudhury, B. Clarkson, T. Jebara, and A. Pentland. Multimodal Person Recognition using Unconstrained Audio and Video. In *Second Conference on Audio- and Video-based Biometric Person Authentication*, 1999.
- [18] P. R. Cohen, M. Johnston, D. McGee, S. Oviatt, J. Pittman, I. Smith, L. Chen, and J. Clow. Quickset: Multimodal interaction for distributed applications. In *Proceedings of the Fifth International Multimedia Conference (Multimedia '97)*, pages 31–40. ACM Press, 1997.
- [19] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-Time Tracking of Non-Rigid Objects Using Mean Shift. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 142–149, 2000.
- [20] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active Appearance Models. In *Proc. European Conference on Computer Vision*, pages 484–498, 1998.
- [21] Timothy F. Cootes and Christopher J. Taylor. Active Shape Models: Smart Snakes. In *Proceedings of the British Machine Vision Conference*, pages 9–18. Springer-Verlag, 1992.
- [22] William Craelius. Bionic Man: Restoring Mobility. *Science*, 295(5557):1018–1021, February 2002.
- [23] Ross Cutler and Matthew Turk. View-based Interpretation of Real-time Optical Flow for Gesture Recognition. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, pages 416–421, April 1998.
- [24] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated Person Tracking Using Stereo, Color, and Pattern Detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 601–609, June 1998.
- [25] J. G. Daugman. Complete Discrete 2D Gabor Transform by Neural Networks for Image Analysis and Compression. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 36:1169–1179, 1988.
- [26] James Davis and Mubarak Shah. Visual Gesture Recognition. In *Vision, Image, and Signal Processing*, volume 141, pages 101–106, April 1994.
- [27] Douglas DeCarlo and Dimitris N. Metaxas. Optical Flow Constraints on Deformable Models with Applications to Face Tracking. *Int. Journal of Computer Vision*, 38(2):99–127, 2000.
- [28] Jon Deutscher, Andrew Blake, and Ian Reid. Articulated Body Motion Capture by Annealed Particle Filtering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 126–133, 2000.
- [29] R. DeValois and K. DeValois. *Spatial Vision*. Oxford Press, 1988.
- [30] A. Dix, J. Finlay, G. Abowd, and R. Beale. *Human-Computer Interaction, Second Edition*. Prentice Hall Europe, 1998.
- [31] Gianluca Donato, Marian Stewart Bartlett, Joseph C. Hager, Paul Ekman, and Terrence J. Sejnowski. Classifying Facial Actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):974–989, October 1999.
- [32] Klaus Dorfmueller-Ulhaas and Dieter Schmalstieg. Finger Tracking for Interaction in Augmented Environments. In *IFAR*, 2001.
- [33] F. Dornaika and J. Ahlberg. Face Model Adaptation using Robust Matching and the Active Appearance Algorithm. In *IEEE Workshop on Applications of Computer Vision*, pages 3–7, December 2002.
- [34] A. Doucet, N. de Freitas, and N. J. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

- [35] Touradj Ebrahimi, Jean-Marc Vesin, and Gary Garcia. Brain-Computer Interface in Multimedia Communication. *IEEE Signal Processing Magazine*, January 2003.
- [36] David Efron. *Gesture, Race and Culture*. King's Crown Press, New York, 1941.
- [37] P. Ekman and W.V. Friesen. The facial action coding system: A technique for the measurement of facial movement. *Consulting Psychologists Press*, 1978.
- [38] R. Feris, V. Krueger, and R. Cesar Jr. Efficient Real-Time Face Tracking in Wavelet Subspace. In *ICCV'01 Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems*, Vancouver, BC, Canada, 2001.
- [39] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381–391, 1954.
- [40] Alexandre R.J. Franois and Grard G. Medioni. A Modular Software Architecture for Real-Time Video Processing. In *Proceedings of the International Workshop on Computer Vision Systems*, July 2001.
- [41] W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, pages 296–301. IEEE Computer Society, June 1995.
- [42] Masaaki Fukumoto, Yasuhito Suenaga, and Kenji Mase. Finger-Pointer: Pointing Interface by Image Processing. *Computers & Graphics*, 18(5):633–642, 1994.
- [43] D. M. Gavrila. The Visual Analysis of Human Movement: A Survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.
- [44] Jim Gemmell, Larry Zitnick, Thomas Kang, and Kentaro Toyama. Software-enabled Gaze-aware Videoconferencing. *IEEE Multimedia*, 7(4):26–35, Oct-Dec 2000.
- [45] Salih Burak Gokturk, Jean-Yves Bouguet, and Radek Grzeszczuk. A Data-Driven Model for Monocular Face Tracking. In *Proc. Intl. Conference on Computer Vision*, pages 701–708, 2001.
- [46] S. Gong, S. McKenna, and A. Psarrou. *Dynamic Vision: From Images to Face Recognition*. Imperial College Press, World Scientific Publishing, May 2000.
- [47] W.E.L. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 22–29, June 1998.
- [48] Ismail Haritaoglu, David Harwood, and Larry S. Davis. W⁴: Real-Time Surveillance of People and their Activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), August 2000.
- [49] Erik Hjelmås and Boon Kee Low. Face Detection: A Survey. *Computer Vision and Image Understanding*, 83(3):236–274, September 2001.
- [50] T. Höllerer. *User Interfaces for Mobile Augmented Reality Systems*. PhD thesis, Columbia University, Department of Computer Science, 2003.
- [51] Pengyu Hong, Matthew Turk, and Thomas S. Huang. Gesture Modeling and Recognition Using Finite State Machines. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, pages 410–415. IEEE Computer Society, March 2000.
- [52] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse. The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. In *Fourteenth Conference on Uncertainty in Artificial Intelligence*, July 1998.
- [53] E. Horvitz and T. Paek. Harnessing Models of Users' Goals to Mediate Clarification Dialog in Spoken Language Systems. In *Eighth International Conference on User Modeling*, July 2001.
- [54] Changbo Hu and Matthew Turk. Computer Vision Based Face Tracking. Technical report, UCSB Computer Science, 2003.

- [55] M. Isard and A. Blake. Condensation – Conditional Density Propagation for Visual Tracking. *Int. Journal of Computer Vision*, 1998.
- [56] Michael Isard and Andrew Blake. A mixed-state CONDENSATION tracker with automatic model-switching. In *ICCV*, pages 107–112, 1998.
- [57] H. Ishii and B. Ullmer. Tangible bits: Towards seamless interfaces between people, bits, and atoms. In *ACM CHI'97*, pages 234–241, 1997.
- [58] Tony Jebara, Bernt Schiele, Nuria Oliver, and Alex Pentland. DyPERS: Dynamic Personal Enhanced Reality System. In *Image Understanding Workshop*, November 1998.
- [59] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 1986.
- [60] J. Jones and L. Palmer. An Evaluation of the Two Dimensional Gabor Filter Methods of Simple Receptive Fields in Cat Striate Cortex. *J. Neurophysiology*, 58:1233–1258, 1987.
- [61] 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.
- [62] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proc. American Control Conference*, pages 1628–1632, June 1995.
- [63] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME Journal of Basic Engineering*, pages 34–45, 1960.
- [64] Yoshinari Kameda, Michihiko Minoh, and Katsuo Ikeda. Three dimensional pose estimation of an articulated object from its silhouette image. In *Proceedings of Asian Conference on Computer Vision*, pages 612–615, 1993.
- [65] K. Karhunen. Über Lineare Methoden in der Wahrscheinlichkeitsrechnung. *Annales Academiae Scientiarum Fennicae*, 37:3–79, 1946.
- [66] L. Karl, M. Pettey, and B. Shneiderman. Speech versus Mouse Commands for Word Processing Applications: An Empirical Evaluation. *Int. Journal on Man-Machine Studies*, 39(4):667–687, 1993.
- [67] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proc. Intl. Conference on Computer Vision*, pages 259–268, 1987.
- [68] Adam Kendon. Some methodological and theoretical aspects of the use of film in the study of social interaction. *Emerging Strategies in Social Psychological Research*, pages 67–91, 1979.
- [69] Adam Kendon. *Conducting interaction; Patterns of behavior in focused encounters*. Studies in Interactional Sociolinguistics 7. Cambridge University Press, 1990. edited by John J. Gumperz.
- [70] M. Kirby and L. Sirovich. Application of the Karhunen-Loève Procedure for the Characterization of Human Faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, January 1990.
- [71] Rick Kjeldsen and John Kender. Finding Skin in Color Images. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 312–317, October 1996.
- [72] Naohiko Kohtake, Jun Rekimoto, and Yuichiro Anzai. InfoPoint: A Device that Provides a Uniform User Interface to Allow Appliances to Work Together over a Network. *Personal and Ubiquitous Computing*, 5(4):264–274, 2001.
- [73] Takeshi Kurata, Takashi Okuma, Masakatsu Kouroggi, and Katsuhiko 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.

- [74] Marco La Cascia, Stan Sclaroff, and Vassilis Athitsos. Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(4), April 2000.
- [75] M. Lades, J. Vorbrüggen, J. Buhmann, J. Lange, W. Konen, C. von der Malsburg, and R. Würtz. Distortion Invariant Object Recognition In The Dynamic Link Architecture. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 42(3):300–311, March 1993.
- [76] I. Laptev and T. Lindeberg. Tracking of multi-state hand models using particle filtering and a hierarchy of multi-scale image features. Technical Report ISRN KTH/NA/P-00/12-SE, Department of Numerical Analysis and Computer Science, KTH (Royal Institute of Technology), September 2000.
- [77] Jintae Lee and Toshiyasu L. Kunii. Model-Based analysis of hand posture. *IEEE Computer Graphics and Applications*, 15(5):77–86, 1995.
- [78] Yongmin Li, Shaogang Gong, and Heather Liddell. Modelling Faces Dynamically Across Views and Over Time. In *Proc. Intl. Conference on Computer Vision*, 2001.
- [79] Rung-Huei Liang and Ming Ouhyoung. A Real-time Continuous Gesture Recognition System for Sign Language. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, pages 558–565. IEEE Computer Society, April 1998.
- [80] John Lin, Ying Wu, and Thomas S. Huang. Modeling the Constraints of Human Hand Motion. In *Proceedings of the 5th Annual Federated Laboratory Symposium*, 2001.
- [81] M. M. Loève. *Probability Theory*. Van Nostrand, 1955.
- [82] John MacCormick and Michael Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *Proc. European Conf. Computer Vision*, 2000.
- [83] I. S. MacKenzie and S. Ware. Lag as a determinant of human performance in interactive systems. In *Proceedings of the ACM Conference on Human Factors in Computing Systems - INTERCHI*, pages 488–493, 1993.
- [84] M. Malciu and F. Preteux. A Robust Model-Based Approach for 3D Head Tracking in Video Sequences. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, pages 169–174, 2000.
- [85] K. Mase. Recognition of Facial Expression from Optical Flow. *IEICE Trans.*, 74(10):3474–3483, 1991.
- [86] David McNeill. *Hand and Mind: What Gestures Reveal about Thoughts*. University of Chicago Press, 1992.
- [87] David McNeill, editor. *Language and Gesture*. Cambridge University Press, 2000.
- [88] Medtronics, Inc. Aactiva Tremor Control Therapy, 1997.
- [89] A. Mitiche and P. Bouthemy. Computation and analysis of image motion: A synopsis of current problems and methods. *Int. Journal of Computer Vision*, 19(1):29–55, 1996.
- [90] B. Moghaddam and Alex Pentland. Probabilistic visual learning for object detection. In *Proc. Intl. Conference on Computer Vision*, pages 786–793, June 1995.
- [91] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based Object Detection in Images by Components. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(4):349–361, April 2001.
- [92] Daniel D. Morris and James M. Rehg. Singularity Analysis for Articulated Object Tracking. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1998.
- [93] Kouichi Murakami and Hitomi Taguchi. Gesture Recognition Using Recurrent Neural Networks. In *ACM CHI Conference Proceedings*, pages 237–242, 1991.
- [94] Thomas A. Mysliwiec. FingerMouse: A Freehand Computer Pointing Interface. Technical Report VISLab-94-001, Vision Interfaces and Systems Lab, The University of Illinois at Chicago, October 1994.

- [95] C. Nass and Y. Moon. Machines and mindlessness: Social responses to computers. *Journal of Social Issues*, 56(1):81–103, 2000.
- [96] Clifford Nass, Jonathan Steuer, and Ellen Tauber. Computers are Social Actors. In *ACM CHI'94*, pages 72–78, 1994.
- [97] Miguel A. L. Nicolelis. Action from thoughts. *Nature*, 409:403–407, January 2001.
- [98] Claudia Nolker and Helge Ritter. GREFIT: Visual recognition of hand postures. In *Gesture-Based Communication in HCI*, pages 61–72, 1999.
- [99] Michael Oren, Constantine Papageorgiou, Pawan Sinha, Edgar Osuna, and Tomaso Poggio. Pedestrian Detection Using Wavelet Templates. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 1997.
- [100] S. L. Oviatt. Ten myths of multimodal interaction. *Communications of the ACM*, 42(11):74–81, November 1999.
- [101] S. L. Oviatt, P. R. Cohen, L. Wu, J. Vergo, L. Duncan, B. Suhm, J. Bers, T. Holzman, T. Winograd, J. Landay, J. Larson, and D. Ferro. Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research directions. *Human Computer Interaction*, 15(4):263–322, 2000.
- [102] P. S. Penev and J. J. Atick. Local Feature Analysis: A General Statistical Theory for Object Representation. *Netork: Computation in Neural Systems*, 7(3):477–500, 1996.
- [103] Alex Pentland, B. Moghaddam, and T. Starner. View-Based and Modular Eigenspaces for Face Recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 84–91, June 1994.
- [104] M. L. Phillips, A. W. Young, C. Senior, C. Brammer, M. Andrews, A. J. Calder, E. T. Bullmore, D. I Perrett, D. Rowland, S. C. R. Williams, A. J. Gray, and A. S. David. A Specific Neural Substrate for Perceiving Facial Expressions of Disgust. *Nature*, 389:495–498, 1997.
- [105] Rosalind W. Picard. *Affective Computing*. MIT Press, 1997.
- [106] D. A. Pollen and S. F. Ronner. Phase Relationship between Adjacent Simple Cells in the Visual Cortex. *Science*, 212:1409–1411, 1981.
- [107] 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.
- [108] Francis Quek, David McNeill, Robert Bryll, Cemil Kirbas, Hasan Arslan, Karl E. McCullough, Nobuhiro Furuyama, and Rashid Ansari. Gesture, Speech, and Gaze Cues for Discourse Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 247–254, June 2000.
- [109] Byron Reeves and Clifford Nass. *The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places*. Cambridge University Press, September 1996.
- [110] James M. Rehg and Takeo Kanade. Model-Based Tracking of Self-Occluding Articulated Objects. In *Proc. Intl. Conference on Computer Vision*, pages 612–617, June 1995.
- [111] Jun Rekimoto. NaviCam: A Magnifying Glass Approach to Augmented Reality Systems. *Presence: Teleoperators and Virtual Environments*, 6(4):399–412, 1997.
- [112] Bradley J. Rhodes. The wearable remembrance agent: a system for augmented memory. *Personal Technologies Journal; Special Issue on Wearable Computing*, pages 218–224, 1997.
- [113] Yoichi Sato, Yoshinori Kobayashi, and Hideki Koike. Fast Tracking of Hands and Fingertips in Infrared Images for Augmented Desk Interface. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, March 2000.

- [114] David Saxe and Richard Foulds. Toward robust skin identification in video images. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, pages 379–384, Sept. 1996.
- [115] A. E. Schefflen. Communication and regulation in psychotherapy. *Psychiatry*, 26(2):126–136, 1963.
- [116] Bernt Schiele and Alex Waibel. Gaze tracking based on face-color. In *Proceedings of the International Workshop on Automatic Face- and Gesture-Recognition*, pages 344–349, June 1995.
- [117] Rogério Schmidt-Feris, Jim Gemmell, Kentaro Toyama, and Volker Krüger. Hierarchical Wavelet Networks for Facial Feature Localization. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, May 2002.
- [118] S. Sclaroff and J. Isidoro. Active Blobs. In *Proc. Intl. Conference on Computer Vision*, 1998.
- [119] Kap-Ho Seo, Won Kim, Changmok Oh, and Ju-Jang Lee. Face Detection and Facial Feature Extraction Using Color Snake. In *Proc. IEEE Intl. Symposium on Industrial Electronics*, volume 2, pages 457–462, July 2002.
- [120] Jianbo Shi and Carlo Tomasi. Good features to track. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, June 1994.
- [121] B. Shneiderman. A nonanthropomorphic style guide: overcoming the humpty dumpty syndrome. *The Computing Teacher*, 16(7), 1989.
- [122] B. Shneiderman. Beyond intelligent machines: just do it! *IEEE Software*, 10(1):100–103, 1993.
- [123] Ben Shneiderman. Direct Manipulation for Comprehensible, Predictable and Controllable User Interfaces. In *Proceedings of IUI97, 1997 International Conference on Intelligent User Interfaces, Orlando, FL* [124], pages 33–39.
- [124] Ben Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison Wesley, 3rd edition, March 1998.
- [125] Ben Shneiderman. The Limits of Speech Recognition. *Communications of the ACM*, 43(9):63–65, September 2000.
- [126] Ben Shneiderman, Pattie Maes, and Jim Miller. Intelligent Software Agents vs. User-Controlled Direct Manipulation: A Debate, March 1997.
- [127] Thad E. Starner and Alex Pentland. Visual Recognition of American Sign Language Using Hidden Markov Models. In *AFGR, Zurich*, 1995.
- [128] B. Stenger, P. R. S. Mendonça, and R. Cipolla. Model-Based 3D Tracking of an Articulated Hand. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 310–315, December 2001.
- [129] Rainer Stiefelwagen, Jie Yang, and Alex Waibel. Estimating focus of attention based on gaze and sound. In *Workshop on Perceptive User Interfaces*. ACM Digital Library, November 2001. ISBN 1-58113-448-7.
- [130] J. Ström, T. Jebara, S. Basu, and A. Pentland. Real Time Tracking and Modeling of Faces: An EKF-based Analysis by Synthesis Approach. In *ICCV*, 1999.
- [131] D. J. Sturman. *Whole Hand Input*. PhD thesis, MIT, February 1992.
- [132] Dawn M. Taylor, Stephen I. Helms Tillery, and Andrew B. Schwartz. Direct Cortical Control of 3D Neuroprosthetic Devices. *Science*, June 2002.
- [133] Carlo Tomasi, Abbas Rafii, and Ilhami Torunoglu. Full-Size Projection Keyboard for Handheld Devices. *Communications of the ACM*, 46(7):70–75, July 2003.
- [134] M. Turk and A. Pentland. Eigenfaces for Recognition. *J. Cognitive Neuroscience*, 3(1):71–86, 1991.

- [135] Matthew Turk, Changbo Hu, Rogerio Feris, Farshid Lashkari, and Andy Beall. TLA Based Face Tracking. In *15th International Conference on Vision Interface*, May 2002.
- [136] Matthew Turk and George Robertson. Perceptual User Interfaces. *Communications of the ACM*, 43(3):32–34, March 2000.
- [137] U.S. Department of Transportation, Federal Highway Administration. Evaluation of Automated Pedestrian Detection at Signalized Intersections, August 2001.
- [138] Andries van Dam. Post-wimp user interfaces. *Communications of the ACM*, 40(2):63–67, 1997.
- [139] Paul Viola and Michael Jones. Robust Real-time Object Detection. *Int. Journal of Computer Vision*, 2002.
- [140] Greg Welch, Gary Bishop, Leandra Vicci, Stephen Brumback, Kurtis Keller, and D’ardo Colucci. The HiBall Tracker: High-Performance Wide-Area Tracking for Virtual and Augmented Environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST)*, December 1999.
- [141] Johan Wessberg, Christopher R. Stambaugh, Jerald D. Kralik, Pamela D. Beck, Mark Laubach, John K. Chapin, Jung Kim, S. James Biggs, Mandayam A. Srinivasan, and Miguel A. L. Nicolelis. Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature*, 408(361), 2000.
- [142] C. D. Wickens. The effects of control dynamics on performance. In K. Boff, K. Kaufman, and J. Thomas, editors, *Handbook on perception and human performance – cognitive processes and performance*, volume 2, chapter 39. Wiley Interscience, 1986.
- [143] Andrew Wilson and Steven Shafer. XWand: UI for Intelligent Spaces. In *ACM CHI*, 2003.
- [144] Jonathan R. Wolpaw, Niels Birbaumer, Dennis J. McFarland, Gert Pfurtscheller, and Theresa M. Vaughan. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, 113(6):767–791, June 2002.
- [145] Christopher Wren, Ali Azarbayejani, Trevor Darrell, and Alex Pentland. PFinder: Real-Time Tracking of the Human Body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997.
- [146] Christopher R. Wren and Alex P. Pentland. Dynamic Models of Human Motion. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, pages 22–27. IEEE Computer Society, April 1998.
- [147] Haiyuan Wu, Taro Yokoyama, Dadet Pramadihanto, and Masahiko Yachida. Face and Facial Feature Extraction from Color Image. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, pages 345–350, Oct 1996.
- [148] L. Wu, S. L. Oviatt, and P. R. Cohen. Multimodal integration – a statistical view. *IEEE Trans. Multimedia*, 1(4):334–331, December 1999.
- [149] W. Wu, M. J. Black, Y. Gao, E. Bienenstock, M. Serruya, A. Shaikhouni, and J. P. Donoghue. Neural decoding of cursor motion using a kalman filter. In *Neural Information Processing Systems, NIPS*, Dec 2002.
- [150] Ying Wu and Thomas S. Huang. Vision-based gesture recognition: A review. In Annelies Braffort, Rachid Gherbi, Sylvie Gibet, James Richardson, and Daniel Teil, editors, *Gesture-Based Communication in Human-Computer Interaction*, volume 1739 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, Berlin Heidelberg, 1999.
- [151] Ying Wu and Thomas S. Huang. View-independent Recognition of Hand Postures. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 84–94, 2000.
- [152] Ying Wu and Thomas S. Huang. Hand Modeling, Analysis, and Recognition. *IEEE Signal Processing Magazine*, May 2001.

- [153] W.Zhao, R.Chellappa, and A. Rosenfeld. Face Recognition: A Literature Survey. Technical Report Technical Report CAR-TR948, UMD CfAR, 2000.
- [154] Jing Xiao, Takeo Kanade, and Jeffrey Cohn. Robust Full-Motion Recovery of Head by Dynamic Templates and Re-registration Techniques. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, May 2002.
- [155] Ming-Hsuan Yang, David J. Kriegman, and Narendra Ahuja. Detecting Faces in Images: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 1 2002.
- [156] Ruigang Yang and Zhengyou Zhang. Eye Gaze Correction with Stereovision for Video-Teleconferencing. In *European Conference on Computer Vision*, May 2002.
- [157] Ruigang Yang and Zhengyou Zhang. Model-Based Head Pose Tracking With Stereo Vision. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, pages 255–260, 2002.
- [158] S. J. Young. HTK: Hidden Markov Model Toolkit V1.5, December 1993. Entropic Research Laboratories Inc.
- [159] Benjamin D. Zait, Boaz J. Super, and Francis K. H. Quek. Comparison of Five Color Models in Skin Pixel Classification. In *Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 58–63, Sept. 1999.
- [160] J. Zhang, Y. Yan, and M. Lades. Face Recognition: Eigenface, Elastic Matching, and Neural Nets. *Proceedings of the IEEE*, 85(9):1423–1435, 1997.
- [161] Z. Zhang. Feature-based facial expression recognition: Sensitivity analysis and experiments with a multilayer perceptron. *Intl. Journal of Pattern Recognition and Artificial Intelligence*, 13(6):893–911, 1999.
- [162] Liang Zhao and Charles E. Thorpe. Stereo- and Neural Network-Based Pedestrian Detection. *IEEE Tran. on Intelligent Transportation Systems*, 1(3), 2000.
- [163] Yuanxin Zhu, Haibing Ren, Guangyou Xu, and Xueyin Lin. Toward Real-Time Human-Computer Interaction with Continuous Dynamic Hand Gestures. In *Proceedings of the Conference on Automatic Face and Gesture Recognition*, pages 544–549, 2000.