Available online at www.sciencedirect.com
ScienceDirect
Computer Vision and Image Understanding 108 (2007) 52–73
Computer Vision and Image Understanding
www.elsevier.com/locate/cviu

Vision-based hand pose estimation: A review

Ali Erol [a,*], George Bebis [a], Mircea Nicolescu [a], Richard D. Boyle [b], Xander Twombly [b]

[a] Computer Vision Laboratory, University of Nevada, Reno, NV 89557, USA
[b] BioVis Laboratory, NASA Ames Research Center, Moffett Field, CA 94035, USA

Received 13 September 2005; accepted 13 October 2006
Available online 19 January 2007
Communicated by Mathias Kolsch

Abstract

Direct use of the hand as an input device is an attractive method for providing natural human–computer interaction (HCI). Currently, the only technology that satisfies the advanced requirements of hand-based input for HCI is glove-based sensing. This technology, however, has several drawbacks including that it hinders the ease and naturalness with which the user can interact with the computer-controlled environment, and it requires long calibration and setup procedures. Computer vision (CV) has the potential to provide more natural, non-contact solutions. As a result, there have been considerable research efforts to use the hand as an input device for HCI. In particular, two types of research directions have emerged. One is based on gesture classification and aims to extract high-level abstract information corresponding to motion patterns or postures of the hand. The second is based on pose estimation systems and aims to capture the real 3D motion of the hand. This paper presents a literature review on the latter research direction, which is a very challenging problem in the context of HCI.

Keywords: Hand pose estimation; Gesture recognition; Gesture-based HCI

1. Introduction

There has been a great emphasis lately in HCI research to create easier to use interfaces by directly employing natural communication and manipulation skills of humans. Adopting direct sensing in HCI will allow the deployment of a wide range of applications in more sophisticated computing environments such as Virtual Environments (VEs) or Augmented Reality (AR) systems. The development of these systems involves addressing challenging research problems including effective input/output techniques, interaction styles and evaluation methods. In the input domain, the direct sensing approach requires capturing and interpreting the motion of head, eye gaze, face, hand, arms or even the whole body.

Among different body parts, the hand is the most effective, general-purpose interaction tool due to its dexterous functionality in communication and manipulation. Various interaction styles tend to import both modalities to allow intuitive, natural interaction (see Appendix A). Gesture languages made up of hand postures (i.e., static gestures) or motion patterns (i.e., dynamic gestures) have been employed to implement command and control interfaces [1–4]. Gesticulations, which are spontaneous movements of the hand and arms that accompany speech, have shown to be very effective tools in Multimodal User Interfaces [5–9]. Object manipulation interfaces [10–12] utilize the hand for *navigation*, *selection*, and *manipulation* tasks in VEs. In many applications such as complex machinery or manipulator control, computer-based puppetry or musical performance [13], the hand serves as an efficient, high degree of freedom (DOF) control device. Finally, some immersive VE applications, such as surgical simulations [14] and training systems [15], have intricate object

* Corresponding author.
E-mail address: aerol@cse.unr.edu (A. Erol).

1077-3142/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2006.10.012

manipulation in their definitions. Broad deployment of hand gesture-based HCI requires the development of general purpose-hand motion capture and interpretation systems.

Currently, the most effective tools for capturing hand motion are electro-mechanical or magnetic sensing devices (data gloves) [16,17]. These devices are worn on the hand to measure the location of the hand and the finger joint angles. They deliver the most complete, application-independent set of real-time measurements that allow importing all the functionality of the hand in HCI. However, they have several drawbacks in terms of casual use as they are very expensive, hinder the naturalness of hand motion, and require complex calibration and setup procedures to be able to obtain precise measurements.

CV represents a promising alternative to data gloves because of its potential potential to provide more natural, unencumbered, non-contact interaction. However, several challenges including accuracy, processing speed, and generality have to be overcome for the widespread use of this technology. Recovering the full DOF hand motion from images with unavoidable self-occlusions is a very challenging and computationally intensive problem. As a result, current implementations of CV-based systems do not have much in common with glove-based ones. Dating back to late 70s [18], the dominant method pursued in the implementation of CV-based interaction has been *appearance-based modeling* of hand motion [19,20]. These models have been successfully applied to build gesture classification engines for detecting elements of a gesture vocabulary. However, 3D motion information delivered by these systems is limited to rough estimates of fingertip positions, finger orientations and/or palm frame obtained using appearance-specific features that affect the generality of the approach.

In this study, we review a more general problem, which aims to recover the full kinematic structure of the hand by bridging the gap between CV-based and glove-based sensing. This is a very challenging, high dimensional problem. Since the hand is a very flexible object, its projection leads to a large variety of shapes with many self-occlusions. Nevertheless, there are several good reasons for tackling this problem. First, there are various types of interaction styles and applications that explicitly rely on 3D hand pose information. Second, 3D hand pose forms an effective feature to be used in gesture classification, as it is view independent and directly related to hand motion. Finally, in contrast to appearance-based methods, full DOF hand pose estimation can provide general, principled methods that can be easily adapted to process simple, lower DOF tasks such as pointing, resizing, navigation etc. [21–23].

1.1. Related literature

There exist several reviews on hand modeling, pose estimation, and gesture recognition [24–27,19,28], the latest of which covers studies up to 2000. However, none of these surveys addresses the pose estimation problem in detail as they mainly concentrate on the gesture classification problem. In this study, we provide a detailed review on pose estimation together with recent contributions in the hand modeling domain including new shape and motion models and the kinematic fitting problem.

It should be mentioned that hand pose estimation has a close relationship to human body or articulated object pose estimation. Human body pose estimation is a more intensive research field. Many algorithms used in hand tracking have their roots in methods proposed previously in human body tracking. However, there are also many differences in operation environments, related applications and the features being used [29]. For example, clothing on human body introduces extra difficulties in segmentation but it also makes color or texture features more reliable for tracking compared to the weakly textured, uniformly colored surface of the hand. Another example is the possibility of estimating human body pose part-by-part or hierarchically (first head, then torso and so on), to break the problem into smaller dimensional ones. In the case of the hand hierarchical processing is limited to two stages for palm, and fingers. It would be difficult, if not impossible, to go any further because of the lack of texture, the proximity of the limbs and the mostly concave shape of the hand that produces severe occlusions. Therefore, we have limited the content of this paper to studies directly addressing the problem of hand pose estimation. Some reviews covering human body pose estimation can be found in [30–35].

1.2. Outline

In Section 2, we define the problem of hand pose estimation, discuss the challenges involved, and provide a categorization of the methods that have appeared in the literature. Hand modeling is an important issue to be considered for any model-based method and it is reviewed in Section 3. Sections 4–6 provide a detailed review of the methods mentioned in Section 2. In Section 7, we summarize the systems reviewed and discuss their strengths and weaknesses. In Section 8, we discuss potential problems for future research. Finally, our conclusions are provided in Section 9.

2. CV-based pose estimation

The dominant motion observed in hand image sequences is articulated motion. There is also some elastic motion but recovering it does not have any major practical use in most applications. Therefore, hand pose estimation corresponds to estimating all (or a subset of) the kinematic parameters of the skeleton of the hand (see Fig. 2). Using visual data for this purpose, however, involves solving challenging image analysis problems in real-time.

In this section, we first discuss some major difficulties associated with the hand pose estimation problem and the restrictions applied on the user or the environment to

alleviate some of them. Then we provide a taxonomy of existing methods based on their technical characteristics.

2.1. Difficulties

The main difficulties encountered in the design of hand pose estimation systems include:

- *High-dimensional problem*: The hand is an articulated object with more than 20 DOF. Although natural hand motion does not have 20 DOF due to the interdependences between fingers, studies have shown that it is not possible to use less than six dimensions (see Section 3). Together with the location and orientation of the hand itself, there still exist a large number of parameters to be estimated.
- *Self-occlusions*: Since the hand is an articulated object, its projection results in a large variety of shapes with many self-occlusions, making it difficult to segment different parts of the hand and extract high level features.
- *Processing speed*: Even for a single image sequence, a real-time CV system needs to process a huge amount of data. On the other hand, the latency requirements in some applications are quite demanding in terms of computational power. With the current hardware technology, some existing algorithms require expensive, dedicated hardware, and possibly parallel processing capabilities to operate in real-time.
- *Uncontrolled environments*: For widespread use, many HCI systems would be expected to operate under non-restricted backgrounds and a wide range of lighting conditions. On the other hand, even locating a rigid object in an arbitrary background is almost always a challenging issue in computer vision.
- *Rapid hand motion*: The hand has very fast motion capabilities with a speed reaching up to 5 m/s for translation and 300°/s for wrist rotation. Currently, off-the-shelf cameras can support 30–60 Hz frame rates. Besides, it is quite difficult for many algorithms to achieve even a 30 Hz tracking speed. In fact, the combination of high speed hand motion and low sampling rates introduces extra difficulties for tracking algorithms (i.e., images at consecutive frames become more and more uncorrelated with increasing speed of hand motion).

Since it is hard to satisfy all the issues listed above simultaneously, some studies on hand pose estimation apply restrictions on the user or the environment. For example, it is usually assumed that the background is uniform or static and that the hand is the only skin-colored object. However, such restrictions may not be applicable in many real-life systems. Although, it is usually acceptable to ask the users to avoid rapid hand motions, there are some applications that can not tolerate low tracking speed. For example, Sturman [13] recommends at least 100 Hz tracking speed to allow effective interaction, based on his practical experience with high DOF control and manipulation tasks.

The alleviation of the first two challenges listed above requires restricting the motion of the hand, which is more difficult to justify. One purpose of restricting the pose is to minimize occlusions. The most common restriction is to assure that the palm is parallel to the image plane. The purpose is to avoid out of plane rotations that cause fingers to occlude each other. In single camera systems, such a restriction leads to *posed interaction*. Quek [1] has justified this type of interaction for communicative gestures. During communicative gesturing, the user inherently makes sure that salient features of the gesture are visible to the observer. The use of multiple cameras located at critical view points is necessary to enable *non-posed interaction*.

Pose restrictions can be also applied to reduce the dimension of the problem by exploiting interaction styles that combine low DOF motion primitives to perform complex tasks (see Appendix A). In this case, it becomes possible to attack the pose estimation problem with dedicated, goal-oriented, appearance-based algorithms described in Section 4. Full DOF pose estimation algorithms can be adapted to these interfaces by simply fixing some of the DOF thereby achieving a faster processing speed. However, the existence of many applications and interaction styles that rely on unrestricted hand motion should also be considered.

2.2. Taxonomy

There are two main approaches in hand pose estimation as illustrated in Fig. 1. The first one consists of "partial pose estimation" methods that can be viewed as extensions of appearance-based systems that capture the 3D motion of specific parts of the hand such as the fingertip(s) or the palm. These systems rely on appearance-specific 2D image analysis to enable simple, low DOF tasks such as pointing or navigation. Because of their dedicated, goal-oriented nature, these algorithms do not have any straightforward extensions for estimating other kinematic parameters that fall out of the scope of the task. The second approach is the full DOF hand pose estimation that targets all the kinematic parameters (i.e., joint angles, hand position or orientation) of the skeleton of the hand, leading to a full reconstruction of hand motion.

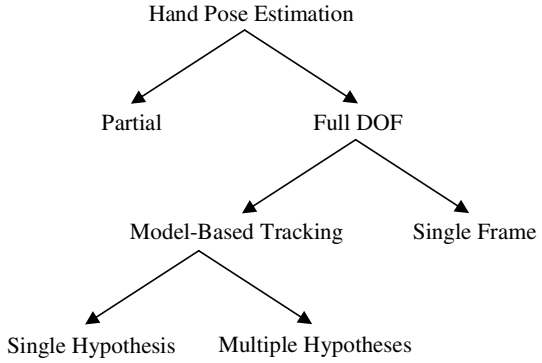

Fig. 1. Different approaches to hand pose estimation.

Solutions to the full DOF hand pose estimation problem can be classified in two main categories: (1) *Model-based tracking*, and (2) *Single frame pose estimation*. The former refers to a top-down tracking method based on parametric models of the 3D hand shape and its kinematic structure. Model-based tracking is common in many studies for tracking various types of objects in 2D and 3D [36–39] (see Section 5 for details). It corresponds to a search executed at each frame of an image sequence to find the pose of the shape model that best matches the features extracted from the image(s). The search is initiated using a prediction based on the object's motion history and dynamics. Some systems rely on a local search around the prediction to produce a single best estimate at each frame. However, imperfections due to occlusions and complexity of hand motion do not allow this type of tracker to work well over long sequences. The alternative approach is keeping multiple hypotheses at each frame to improve the robustness of tracking.

The second solution (i.e., single frame pose estimation) is a more recent one that attacks the problem without making any assumptions on time coherence, resulting in a more difficult problem. This approach can lead to algorithms for initialization or re-initialization in tracking-based systems. Another motivation for this approach is the rapid motion of the hand and fingers. Images of consecutive frames can be very different, making time coherence assumptions useless.

3. Hand modeling

In this section, we provide a review on hand modeling in the context of model-based vision. First, we describe the kinematic model that forms the basis of all types of hand models. A kinematic hand model represents the motion of hand skeleton, but is also a redundant model in the sense that it does not capture the correlation between joints. After a review on modeling the natural hand motion, we present some hand shape models that allow generating appearances of the hand in arbitrary configurations. Finally, the kinematic fitting problem, which involves the calibration of the user specific parameters of the hand model, is discussed.

3.1. Kinematic hand model

The human hand consists of 27 bones, 8 of which are located in the wrist. The other 19 constitute the palm and fingers as shown in Fig. 2a. The bones in the skeleton form a system of rigid bodies connected together by joints with one or more degrees of freedom for rotation. Joints between the bones are named according to their location on the hand as metacarpophalangeal (MCP) (i.e., joining fingers to the palm), interphalangeal (IP) (i.e., joining finger segments) and carpometacarpal (CMC) (i.e., connecting the metacarpal bones to the wrist). The nine IP joints can be accurately described as having only one DOF, flexion-extension. All five MCP joints, however, are described in the literature as saddle joints with two DOF: abduction/adduction (i.e., spreading fingers apart) in the plane defined by the palm, and flexion/extension. The CMC of the index and middle fingers are static while the CMC of the pinky and the ring finger have limited motion capability reflecting palm folding or curving, which is often discarded yielding a rigid palm. The CMC of the thumb, which is also called trapeziometacarpal (TM), is the most difficult to model. Biomechanical studies [40] have shown that the TM joint has two non-orthogonal and non-intersecting rotation axes. The two DOF saddle joint is a restrictive model but it has been used in many studies. Extending it to a three

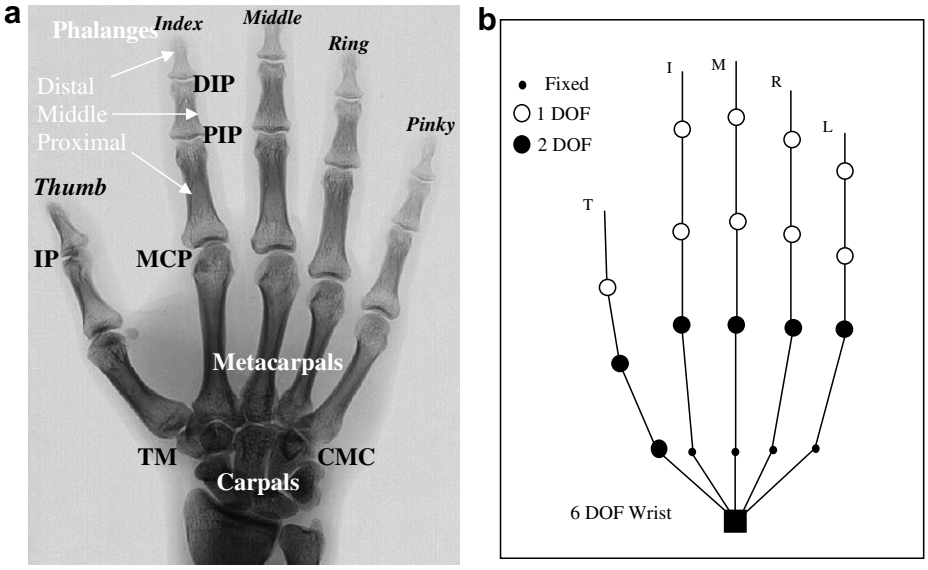

Fig. 2. Skeletal hand model: (a) Hand anatomy, (b) the kinematic model.

DOF spherical joint overcomes the restrictions [41,42]. Another solution is to have a twist around the bone axis as a linear function of abduction and flexion angles [43]. The angular DOF of fingers, which is often called the *local configuration*, and the six DOF of a frame attached to the wrist, which is often called the *global configuration*, form a configuration vector representing the pose of the hand.

A 27 DOF model that was introduced in [44] and has been used in many studies is shown in Fig. 2b. The CMC joints are assumed to be fixed, which quite unrealistically models the palm as a rigid body. The fingers are modeled as planar serial kinematic chains attached to the palm at anchor points located at MCP joints. The planarity assumption does not hold in general. Standard robotics techniques provide efficient representations and fast algorithms for various calculations related to the kinematics or dynamics of the model. Adding an extra twist motion to MCP joints [45,46], introducing one flexion/extension DOF to CMC joints [47] or using a spherical joint for TM [42] are some examples of the variations of the kinematic model.

The kinematic hand model described above is the most natural choice for parameterizing the 3D hand state but there exist a few exceptions using other types of representations. Sudderth et al. [48] used independent rigid bodies for each component of the hand, leading to a highly redundant model. The kinematic relations between these rigid bodies were enforced using a prior model in their belief propagation network. Heap et al. [49] dropped the kinematic model and modeled the entire surface of the hand using PCA applied on MRI data. Such a representation requires further processing to extract useful higher-level information, such as pointing direction; however, it was shown to be very effective to reliably locate and track the hand in images.

Full DOF hand pose estimation systems extensively rely on a-priori information on the motion and shape of the hand; therefore, the kinematic model is augmented with shape information to generate appearances of the hand in arbitrary configurations, and hand pose or motion constraints to reduce the search space for pose estimation. Although the same motion models could be assumed for arbitrary users, the same assumption cannot hold true for shape models. If precision is a requirement for the application, these models need to go through a calibration procedure to estimate user-specific measurements.

3.2. Modeling natural hand motion

Although active motion of the hand (i.e., motion without external forces) is highly constrained, this is not reflected in the kinematic model. An attempt to capture natural hand motion constraints is by complementing the kinematic model with *static constraints* that reflect the range of each parameter and *dynamic constraints* that reflect the joint angle dependencies. Based on the studies in biomechanics, certain closed-form constraints can be derived [44,42,19]. An important constraint is the relation $\theta_{\mathrm{DIP}} = \frac{2}{3}\theta_{\mathrm{PIP}}$ between the PIP and DIP angles that helps decrease the dimension of the problem by 4. There exist many other constraints that are more complex to be utilized in a pose estimation algorithm. For example, the flexion angle of an MCP joint has an effect on the abduction capability of that joint and neighboring MCP joints.

The very intricate structure of the hand does not allow expressing all the constraints in a closed form. Moreover, the natural motion of the hand may follow more subtle constraints which have nothing to do with structural limitations [50]. These problems have motivated learning-based approaches, which use ground truth data collected using data gloves. The feasible configurations of the hand are expected to lie on a lower dimensional manifold due to biomechanics constraints. Lin et al. [50] applied PCA on a large amount of joint angle data to construct a seven-dimensional space. The data was approximated in the reduced dimensional space as the union of linear manifolds. It is also possible to use the data directly without any further modeling as in [51] to guide the search in the configuration space. Another way to use the glove data is to generate synthetic hand images to build a template database that models the appearance of the hand under all possible poses [52–55].

In addition to modeling the feasible hand configurations, learning the dynamics of hand motion can help tracking algorithms. Zhou et al. [56] presented an EDA (eigen-dynamic analysis) method for modeling the non-linear hand dynamics. First, PCA was used to reduce the dimension of the problem. Then hand motion was modeled in the reduced space, while moving only one of the fingers, using low order linear systems. The resulting five linear models were combined to obtain a high order stochastic linear dynamic system for arbitrary finger motion.

Thayananthan et al. [57] represented the configuration space as a tree, which was constructed using hierarchical clustering techniques or regular partitioning of the eigenspace at multiple resolutions. Each node of the tree corresponds to a cluster of natural hand configurations collected using a data-glove. The tree structure enables fast hierarchical search through Bayesian Filtering. The dynamic model of the system, which is assumed to be a first order Markov process, was built by histogramming state transitions between clusters using large amount of training data.

3.3. Modeling the shape of the hand

Hand shape has both articulated and elastic components; however, computational efficiency reasons do not allow the use of very complex shape models for pose estimation. In many studies, the hand model needs to be projected many times on the input image(s) to obtain features that can be matched against the observed features. Visibility calculations to handle occlusions add extra complexity to the projection calculations. These problems have motivated the use of rough shape models, composed of simple

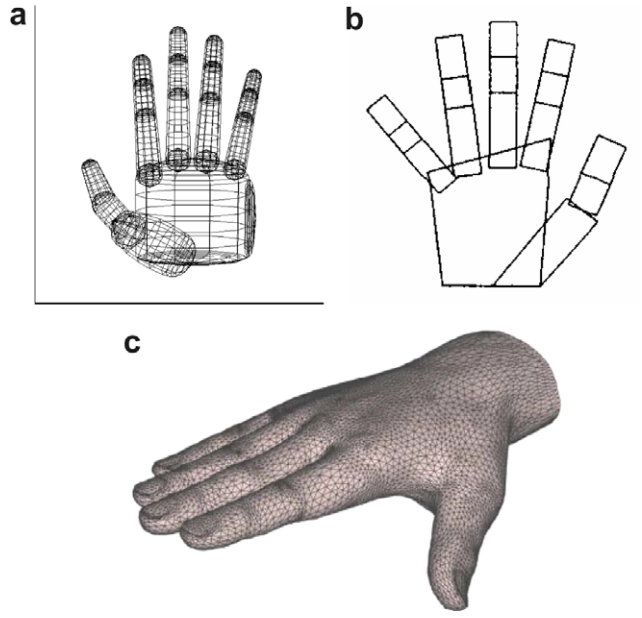

Fig. 3. Hand shape models with different complexity: (a) Quadrics-based hand model taken from [22], (b) cardboard model taken from [58], (c) a realistic hand model taken from [45].

geometric primitives such as cylinders, spheres, ellipsoids attached to each link or joint of the hand skeleton. Figs. 3a and b show two examples having different complexity. Stenger et al. [22] used quadrics as shape primitives, as shown in 3a. Using projective geometry properties of quadrics, fast algorithms were provided for projecting quadrics and calculating their visibility. Wu et al. [58] used an even more economical, view-dependent model called "cardboard model", shown in Fig. 3b. When viewed from a direction orthogonal to the palm, the hand is modeled as the union of rectangles attached to each link and the palm. A visibility map was used to handle visibility calculations.

Although these rough hand models can be processed efficiently, one may also anticipate systems with better performance using more complex models. Therefore, some studies have employed more realistic models. Kuch et al. [42] used a B-spline surface whose control points were attached to the links in the model. In [45] and [59], a deformable skin model was implemented using a skinning technique. Fig. 3c shows the model used by Bray et al. [45]. However, both of these studies make use of 3D features obtained from depth sensors or stereo vision, eliminating complex projection operations.

3.4. Kinematic fitting

User-specific shape parameters of the hand such as the length and width of the links can vary over a wide range, affecting the accuracy of model-based pose estimates. Therefore, there is a need for user-specific calibration of the hand model. This problem is usually solved manually in the literature. One can imagine using 3D sensors [45] or 3D reconstruction from multiple views [42] to capture the shape of the hand in detail; however, kinematic fitting that involves estimating the link lengths or equivalently the locations of each joint is a difficult task.

The kinematic fitting problem is addressed in the area of modeling the human body, or—more general—articulated objects. One method is to estimate the full 3D motion of each component and then process the data to extract the topology and/or joint locations. O'Brien et al. [60] and Taycher et al. [61] used magnetic sensors and multiple rigid motion estimation algorithms, respectively, to calibrate human body models. Another approach is using active or passive point markers as demonstrated in [62]. Using passive markers introduces some difficulties in establishing marker correspondences across frames, compared to active markers. Kakadiaris et al. [63] proposed a complex procedure where the subject performs a set of movements according to a protocol that reveals the structure of the human body. Images from multiple views were then analyzed to build the full human body model.

In the case of the hand, there are only a few studies in literature. In most cases manual calibration is performed. Anthropometric data that establish statistical relations between external measurements and joint locations [64,65] can be valuable for manual calibration. In [42], a semi-automatic hand calibration procedure is described. Several landmarks on the hand are marked manually in images taken from different views and a spline-based model is fit to the landmark points. In [66] and [67], some of the calibration parameters are estimated together with pose parameters. In [68], the image of an open hand and anthropological ratios between finger segments were utilized. In [69], 3D locations of fingertips acquired using a stereo camera and color LED markers were used to calibrate a data glove and a hand model. Very recently, Lien et al. [70] introduced a marker-based calibration method using scalable inverse kinematic solutions that they developed. Each finger in the base model was scaled separately using a grasping motion sequence by the user.

4. Partial hand pose estimation

In this section, we provide a review on estimating partial hand pose, which corresponds to rough models of the hand motion, mainly consisting of position of the fingertips, orientation of the fingers or position and orientation of the palm. Partial hand pose estimation algorithms are used to complement appearance-based systems to provide continuous motion data for manipulation, navigation or pointing tasks. First, we describe the architecture of these systems followed by implementation details.

There are many real-time gesture-based HCI prototype systems implemented using computer vision techniques. Most of these systems employ appearance-based modeling. The application scenarios in these prototypes include 2D VEs or AR applications to manipulate electronic documents, presentations or other applications projected on desks, walls etc. [71–77], and 3D VEs mostly targeting gra-

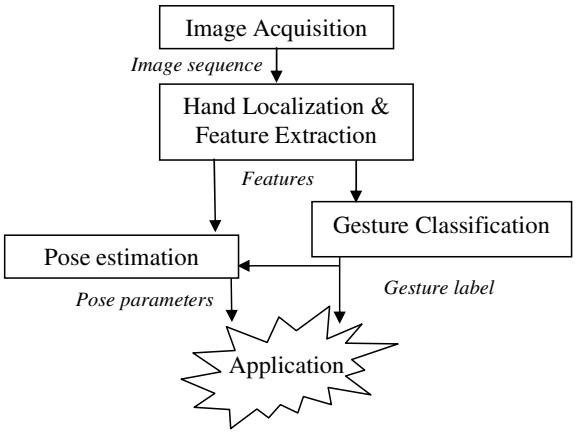

Fig. 4. Block diagram of a generic gesture-based interaction system.

phic design applications with 3D object manipulation tasks [78–86]. The user interface consists of a small gesture vocabulary of static or dynamic gestures. In most cases, static postures are employed in the vocabulary. The use of dynamic gestures is limited to a few studies [72,78,1,87]. In some gestures, hand pose estimation is performed to implement primitive object manipulation tasks, while other gestures have only symbolic interpretations corresponding to commands.

Fig. 4 shows a block-diagram of a generic system. The input images are processed to localize the hand, then some features are extracted to be utilized for pose estimation and/or gesture recognition. The gesture recognition engine is essentially a pattern classification engine that extracts the symbolic content of hand motion. Pose estimation is activated whenever hand shape matches a certain gesture, for example, a pointing gesture. The assumption that the hand has a limited range of appearances enables posture and view-dependent image analysis for estimating the pose.

4.1. Hand localization

Skin color segmentation is a common method for locating the hand because of its fast implementation using lookup tables [74,88–90,82,91,83]. Static background subtraction [80,71], and the use of adaptive background models [75,73] are other common methods. Shadows can be a problem in background subtraction algorithms [73]. A few studies utilize IR cameras that are tuned to human temperature [77,72] to provide fast solutions by simple thresholding operations. Various assumptions are used, such as the hand being the only skin-colored object, uniform ambient lighting, or stationary background. Tracking parametrized deformable templates of the hand or fingers [14,92–94] is a more elaborate and precise method that can handle complex backgrounds. In particular, particle filter-based tracking [14,92] has been shown to be a robust approach. Depending on the algorithm and the users' clothing (e.g., sleeves rolled up), hand-arm segmentation may also be necessary as an extra processing step [77,79] in hand localization.

Another approach to locate the hand involves classification-based object detection methods. In these systems, a large list of hypotheses in the form of subregions in the image are processed by a classifier to decide the presence of the object. Using conventional classifiers for verification however, would not allow an exhaustive list of subregions to be processed in reasonable time. Employing boosted cascades of classifiers improve processing speed drastically [95]. A very fast classifier on top of the hierarchy eliminates a large number of false hypotheses to allow for the use of more accurate but computationally more expensive classifiers at the lower levels of the hierarchy. Training these classifiers requires collecting a large number of hand posture images, possibly from different views. Kolsch et al. [96] introduced a class separability estimation method based on frequency spectrum analysis to reduce the load of training these classifiers. Ong et al. [97] employed clustering methods to cluster the training data into similar shapes and build a tree of classifiers for detection. As one goes down to the branches of the tree, the classifiers are trained to detect more and more specific clusters consisting of similar shapes. It is possible to use such classifiers to recognize gestures as well. However, labeling a large number of samples is a time consuming process. Wu et al. [98] provided a training algorithm that only needs a small portion of the database to be labeled and built a view-independent gesture recognition system by using samples of postures captured from a large number of views. A color segmentation algorithm [99] was employed to reduce the number of hypotheses.

4.2. Gesture classification

Appearance-based gesture classification is an intensive research field involving many machine learning techniques (neural networks, HMM etc.) and a wide range of feature extraction methods (moments, Fourier descriptors, optical flow etc.). In this study, we concentrate only on gestures that are accompanied by pose estimation. Such gestures are limited to a small set of hand postures shown in Fig. 5. Fig. 5a shows an open hand whose rigid motion can provide signals for a navigation task. An open hand also represents a generic model that is used to generate different commands by changing the visibility of fingers [75,79,85,72]. The number of visible fingers and their attributes, such as orientation and fingertip location, are sufficient to differentiate between these commands. The pointing gestures shown in Fig. 5b and c can be seen as states with one and two visible fingers, respectively. The pointing (or selection) operation and the corresponding gesture is common to almost all interfaces. Fig. 5d shows an object manipulation gesture, which has different characteristics and is used in only a few studies. The motion of the index and thumb fingers can provide signals for grabbing [80], resizing [82], translating or rotating [71] objects.

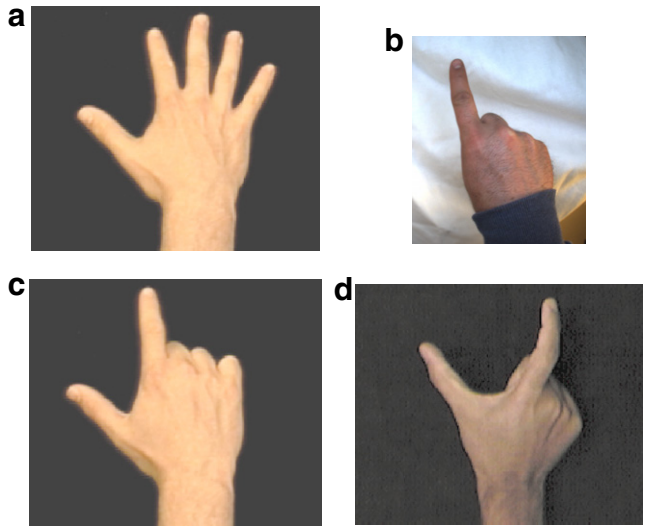

Fig. 5. Gestures used in pose estimation. (a) An open hand (e.g., used for navigation) [78], (b) pointing gesture, (c) another pointing gesture [78], (d) object manipulation gesture [78].

4.3. Feature extraction

Pose estimation based on the gestures described above mainly involves extracting the position and orientation of the hand, fingertip locations, and finger orientation from the images. The hand position is taken to be a unique stable landmark point on its silhouette. By silhouette, we mean the outline of the hand provided by segmentation algorithms, or equivalently the partitioning of the input image into object and background pixels. The center of gravity (or centroid) of the silhouette [83,100] is one choice, but it may not be very stable relative to the silhouette shape due to its dependence on the finger positions. The point having the maximum distance to the closest boundary edge [79,86,77,74] has been argued to be more stable under changes in silhouette shape. Fingertip detection can be handled by correlation techniques using a circular mask [77,72,73], which provides rotation invariance, or fingertip templates extracted from real images [101,76]. Using curvature local maxima on the boundary of the silhouette is another common method to detect the fingertips and the palm-finger intersections [80,71,82]. Sensitivity to noise can be an issue for curvature-based methods in case of noisy silhouette contours. In [100,74], a more reliable algorithm based on the distance of the contour points to the hand position is utilized. The local maxima of the distance between the hand position and farthest boundary point at each direction gives the fingertip locations. Finger or 2D hand orientation can be estimated by calculating the direction of the principal axes of the silhouettes [87,86,80]. All these features can be tracked across frames to increase computation speed and robustness using Kalman filters [74,88] or heuristics that determine search windows in the image [82,85] based on previous feature locations or rough planar hand models. The low computational complexity of these methods enable real-time implementations using conventional hardware. However, their accuracy and robustness are arguable. Since they do not account for perspective distortion, accurate pose estimates are not expected. Moreover, they rely on high quality segmentation lowering their chance of being applied on highly cluttered backgrounds. Failures can be expected in some cases such as two fingers touching each other or out of plane hand rotations.

A more elaborate method is tracking the features directly in 3D using 3D models. Jennings et al. [102] has employed range images, color, and edge features extracted from multiple cameras to track the index finger in a pointing gesture. Very robust tracking results over cluttered and moving backgrounds were obtained. Davis et al. [103] used cylindrical fingertip models to track multiple finger positions in 3D without occlusions, over uniform backgrounds.

Employing markers could be considered intrusive but they have considerable technical advantages in terms of processing speed. Maggioni et al. [85] used elliptical markers to estimate the hand frame in 3D. A similar approach was pursued in [104], using multiple views. In a more recent study, Kim et al. [84] used white fingertip markers under black-light to detect fingertip locations yielding a much richer set of gestures.

4.4. Pose estimation

In 2D applications, the features described above are used directly to avoid 3D pose estimation. However, 3D information can also be useful in those applications as demonstrated by Malik et al. [71]. In that study the disparity of the fingertip was used to decide if the finger was touching the desk to implement a *Visual Touchpad*. In 3D applications, stereo vision comes into use in a straightforward manner to calculate the 3D location of the hand and fingertips, and the orientation of the hand and/or fingers. Using shadows to extract 3D pose information using a single camera was proposed in [81] but this technique requires highly controlled background and lighting. When the hand is in rigid motion, fingertip locations and palm position provides sufficient information to estimate the 3D orientation of the hand [82,83]. In some studies [86,83], multiple cameras are used to allow the user to gesture freely by having the system select by the best view for analysis. In [86], the 3D orientation of the hand plane was used to select the best camera for gesture classification. In [83], the area and aspect ratio of the hand silhouette were used for the same purpose.

4.5. Discussion

The hand model used in these systems consists of a global hand pose, binary states (visible/occluded) of the fingers, fingertip locations, and finger orientations. Extracting these features requires posing to the camera to keep hand appearance at a reasonable range. As a result,

extending a small gesture set, such as the one shown in Fig. 5, seems to be very difficult through appearance-based methods. It would not be possible to implement many other interaction procedures using these systems. On the other hand, all the studies reviewed are real-time systems with sampling frequencies reaching up to 60 Hz [78], which is a critical factor in many applications. There are also environments with limited resources (e.g., wearable computers [105]) that strictly require computationally economical algorithms.

Usability studies of these interfaces are limited and rare. User fatigue is a well-known problem reported in many studies [78,82]. Avoiding too much arm motion is a guideline [2] in the design of these system to avoid user fatigue. Another evaluation is related to the accuracy of the pose estimates especially for pointing gestures. Jitter in pointing gestures becomes crucial when distant objects are pointed at. Except for few studies such as [93], high accuracy is not a virtue of any of these systems. Classification accuracy is another factor that can affect the usability. False positives that occur during transitions from one gesture to another and failures due to user-dependent variations in gesturing were reported in [83]. In [86], some users failed to use some of the gestures in the gesture set. Besides these issues, most studies report encouraging user satisfaction.

5. Model-based tracking

A block diagram of a generic model-based tracking system is shown in Fig. 6. At each frame of the image sequence, a *search* in the configuration space is executed to find the best parameters that minimize a *matching error*, which is a measure of similarity between groups of model features and groups of features extracted from the input images. The search is initiated by a prediction mechanism, based on a model of the system dynamics. In the first frame, a prediction is not available, therefore, a separate initialization procedure is needed. In the search phase, the basic operation is the calculation of the matching error between the features extracted from the input and the features generated by the model. The synthesis of features using the model also enables a selective analysis that focuses on regions or a subset of features instead of the whole input. In the most common implementation, 2D features in the image plane are used. In this case, the projection of the 3D model on the image plane should be performed. If multiple cameras are used, the matching error on all the cameras can be combined without solving any correspondence problem between the images [37]. A less common approach is using 3D features that can be acquired using multiple view systems or depth sensors.

The framework summarized above has been intensively studied in the context of human body tracking research. Two types of systems have emerged. The first one is based on a local search and keeps track of only the best estimate at each frame. We call this type as *single hypothesis tracking* throughout this study. However, this type of tracker is not expected to work well on long sequences, due to some imperfections in the error function. Spurious local minima, singularities and discontinuities originating from background clutter, self-occlusions, and complex motion dynamics lead to tracking failures [106,107]. Morris and Rehg [108,109] have provided an in-depth analysis of the singularity problem and proposed a novel 2D kinematic model called Scaled Prismatic Models (SPMs) in place of 3D models. SPMs can help avoid most of the singularities but require further processing to obtain 3D pose estimates. Employing multiple views is another way to avoid singularities [109,110]. A more general approach that has the ability to address these imperfections is *multiple hypotheses tracking* (MHT). The basic idea in MHT is keeping multiple pose estimates at each frame. If the best estimate fails, the system can still continue tracking using other potential estimates. The idea is best captured by Bayesian filtering, which targets the computation of the posterior probability density function of the hand configurations using the avail-

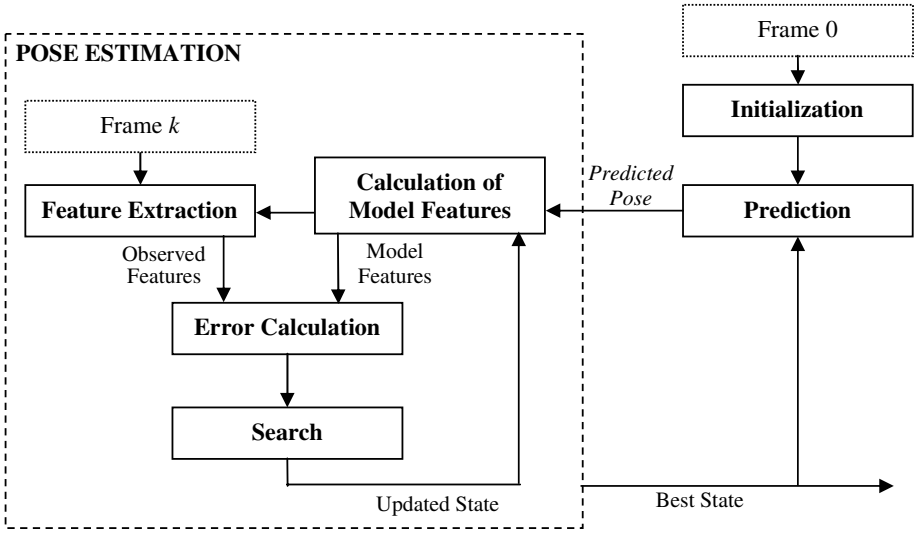

Fig. 6. Model-based tracking.

able observations. An introduction to Bayesian filtering and pointers to other references can be found in [111].

The main modules of the system shown in Fig. 6 are initialization, prediction, feature extraction, and search. The first two tasks are not fully addressed in the literature. Many systems solve the initialization problem manually or by assuming a simple known initial configuration (e.g., a stretched hand with no occlusion). Such an initialization has some undesirable consequences, for example, when the target is lost the user is required to place his hand in the initial configuration. Modeling the non-linear dynamics of hand motion is not an easy problem and limited to a few studies only (i.e., reported in Section 3.2). Instead, weak linear dynamics, that assert smooth state or velocity change, are usually assumed. In the following subsections, we review the implementation of feature extraction and search modules.

5.1. Feature extraction and matching

Feature extraction is a crucial module in any CV system. The implementation of this module has a considerable effect on the robustness of the system to self-occlusions and background clutter. Feature extraction and matching also have a huge impact on the processing speed of the whole system because of the search process (i.e., matching and/or feature extraction is repeated many times during the search).

The hand creates images that are very difficult to analyze in general. High level features such as fingertips, fingers, joint locations, and the links between joints are very desirable, as they provide a very compact representation of the input supporting high processing speeds. However, it is very difficult to extract them robustly without any severe pose restrictions (see Section 4). Therefore, the majority of studies rely on low-level features. Another group of features are 3D features that can be obtained using 3D depth sensors or multiple cameras.

5.1.1. High-level features

Algorithms employing high-level features often rely on colored markers to extract fingertip and joint locations or some anchor points on the palm [44,112,113]. An exception is a fingertip detection system based on Gabor features and a special neural network architecture (LLM-net) reported in [114]. An important problem with point features is their susceptibility to occlusions. Moreover, It may be difficult to track markers on the image plane due to frequent collisions and/or occlusions [113]. Using the predicted marker positions in place of the missing markers can help to avoid failures [113,70].

Non-point features, such as protrusions of hand silhouettes, were utilized in [66] to roughly detect fingers, fingertips, and links. Another example is the DigitEyes system [21] that uses the projection of the links in the skeletal model to guide the extraction of links and fingertips during model-based tracking. Links and fingertips are very compact features allowing very fast similarity computations; however, even with a small amount of occlusion one can expect large fractions of outliers in link extraction. In [115], this system was extended to include appearance templates associated with each link and a layered representation of templates to handle occlusions. Templates associated with each link were registered and masked using a window function that depends on the predicted visibility order. Tracking two fingers in highly occluding poses was used to demonstrate the effectiveness of occlusion handling.

It should be noted that none of the systems mentioned above, even the ones with colored markers, operate in the presence of cluttered backgrounds.

5.1.2. Low-level features

Contours or edges are universal features that can be used in any model-based technique [38]. Often, a volumetric model of the hand is projected on the images to calculate the occluding contours of the projection. Then, point correspondences between model contours and image contours are established based on a certain proximity criterion (e.g., closest point in the normal direction). The edges can be extracted by either processing the whole image, or the model projection can be used as a guide to locate gradient magnitude local maxima (or any other edge detector) in the vicinity (e.g., the first edge in the direction of the contour's normal) of the model contours. The distance between corresponding points gives the matching error [116,51,22,51,49]. Edge-based measures are not expected to be effective under cluttered backgrounds. Thayananthan et al. [67] combined edge orientation and chamfer matching, to demonstrate increased robustness compared to shape context features. This combination was utilized in many studies [57,53,23,54,48]. In [53,57,23,48] skin color models were also employed to increase robustness. The likelihood of the segmentation, asserted by the projection of the model, was calculated using background and skin color models as a measure of similarity. Combinations of multiple features are in general expected to improve robustness. Combining edges, optical flow, and shading was proposed in [117] and successful tracking results, under severe occlusions, were presented.

Silhouette (i.e., the outline of the hand provided by segmentation algorithm) is another common feature, which can mainly help to keep the projection of the model inside the hand region in the images. The overlapping area of the model and hand silhouettes is usually taken to be a measure of similarity [118]. In [47], distance transforms of the silhouettes were correlated instead. Lin et al. [116,51] employed silhouette edge combinations and presented tracking results over cluttered backgrounds in [51]. A drawback of using silhouettes is the necessity for a separate segmentation module.

5.1.3. 3D features

Use of 3D features is limited to a few studies. In [119], a stereo camera was used to obtain a dense 3D reconstruc-

tion of the scene and segment the hand by thresholding the depth map. The depth map enables dealing with cluttered backgrounds as long as the hand is the closest object to the stereo camera. In [45,46], structured light was used to acquire 3D depth data; however, skin color was used for segmenting the hand. In either case, the depth map gives a surface, which is matched against the model surface in 3D. In another study [59], it was proposed to track a large number of points of interest on the surface of the hand using a stereo camera. Motion information obtained from the 3D trajectories of the points was used to augment the range data. One can also imagine a full 3D reconstruction of the hand surface using multiple views; however, an exact, real-time and robust 3D reconstruction is very difficult. Ueda et al. [120] used an approximate but fast reconstruction method based on the visual hull [121] that uses the silhouettes to compute a bounding volume for the object. The visual hull was used for human body pose estimation in some studies and was shown to give satisfactory reconstructions using few cameras; however, the same may not hold true in the case of the hand because of its mostly concave geometry. A drawback of 3D reconstruction is the additional computational cost. Nevertheless, 3D data contains valuable information that can help eliminate problems due to self-occlusions which are inherent in image-based approaches. Markers can lead to 3D information in a much more economical way [122,70,44] using the simple triangulation method. Although colored gloves are expected to be less intrusive than electro-mechanical ones, a markerless solution is much more appealing for widespread use of hand-based HCI.

5.2. Single hypothesis tracking

Single hypothesis tracking corresponds to a best fit search where the matching error is minimized. There are two types of methods for performing the search. The first one is following explicit matching error minimization techniques (i.e., optimization methods). The second one consists of applying physical forces on the model.

5.2.1. Optimization-based methods

The most common approach to fitting a model to the extracted features is using standard optimization techniques. In [21], an error based on joint links and fingertips was minimized using the Gauss–Newton method augmented with a stabilization step [38]. Stabilization was used to deal with kinematic singularities that occur when the change of some parameters does not have any effect on the appearance of the hand. In [113], the same technique was applied using fingertip and joint markers. Silhouette-based error measures were minimized using Nelder Mead Simplex (NMS) in [123], and Genetic Algorithms (GAs) and Simulated Annealing (SA) in [47]. In [123], the NMS algorithm was modified to account for closed-form hand model constraints. In [51], a two-stage model fitting algorithm based on NMS using edge and silhouette features was proposed. First, a coarse stage that forces simplex to pass through sample points collected using a glove-based sensor. Second, a fine tuning stage, where the simplex evolves without any constraints. Lien et al. [122,70] used stereo cameras to extract the 3D locations of a number of markers on the palm and fingertips and applied GAs to estimate the orientation of the palm. The state of the fingers was estimated using inverse kinematics and regression techniques. Bray et al. [46] used Stochastic Gradient Descent (SGD) along with depth features. A small number of points on the model surface were selected randomly at each iteration to reduce computational cost and avoid spurious local minima. Hand model constraints were carefully taken into consideration by using an additional step at each iteration. The resulting algorithm was called Stochastic Meta Descent (SMD).

In [124], a 'divide and conquer' approach was proposed. First, the global motion of the hand was estimated, followed by the estimation of the joint angles. This procedure was applied iteratively until convergence. As it is not possible to accurately segment the palm region from images, outliers are expected. Therefore, robust estimation methods were utilized for estimating the pose of the palm. Later, different global pose estimation algorithms were proposed [116,51,56], all of them modeling the palm with a 2D planar shape.

Kalman filters have also been employed in single hypothesis tracking. In [22], the Unscented Kalman Filter (UKF) was used for tracking. UKF applies a deterministic weighted sampling of the Gaussian posterior to be able to track a non-linear system.

5.2.2. Physical force models

An alternative approach to model fitting utilizes physical force models. In this approach, the matching error is used to create forces to be applied on the surface of the articulated model. Then, the model parameters are updated by solving the dynamic equations of the kinematic model. In [119], the forces were derived using the Iterative Closest Point (ICP) algorithm for registering the model with a 3D reconstruction obtained using a stereo head. Another 3D system described in [120] uses the visual hull of the hand to derive a force model. Using the parts of the model lying outside the visual hull, forces were applied on the link of the skeletal model to push these parts inside the visual hull. In [117], the forces were derived using a combination of multiple features (edges, shading, and optical flow) from the image. Finger motion limits and inter-finger penetrations were considered, and have been reported to significantly improve the robustness of tracking.

5.3. Multiple hypotheses tracking

There are many approaches for multiple hypothesis tracking. Many of them rely on Bayesian filtering or similar formulations. In the following subsections, we describe some representative approaches.

5.3.1. Particle filters

Particle filtering is a well-known technique for implementing recursive Bayesian filters using Monte Carlo simulations. The basic idea of particle filtering is representing an arbitrary probability density using weighted samples drawn from another easy to sample density, called the importance density. The weights represent the probability of occurrence of each sample and the weighted samples are usually called particles. In case of tracking, the particles of the hand configuration distribution should be updated at each frame. In [116], a parametrization of the hand configuration space was used (see Section 3) for generating the samples. It was demonstrated that it is possible to track the fingers by keeping an order of magnitude less samples than that of the more conventional condensation algorithm [125]. A problem with particle filters is the requirement on the number of samples to be kept and tested. The most expensive part of a tracking system is the error calculation, therefore, repeating this operation on a large number of samples (e.g., [116] reports using 100 samples) is not desirable.

To reduce the number of samples, some approximations have been proposed by assuming a semi-parametric posterior [126]. Combinations of semi-parametric particle filters and local search algorithms provide solutions requiring fewer samples. The samples representing the modes of the posterior are kept and used to initiate local search procedures. In [45], the SMD algorithm was employed resulting in an 8 particle tracker, while [51] uses the two-stage NMS algorithm (see Section 5.2.1) with 30 particles.

5.3.2. Tree-based filter

Another approach to implementing Bayesian filtering is grid-based filtering. Such an approach was followed in [53] by partitioning the state space using a regular multi-resolution grid. Bayesian filtering was implemented over the tree by assuming a piecewise constant distribution over its leaves. A large number of images, generated using an artificial hand model and marker-based motion capture, were used to construct representative templates for each node of the tree. During tracking, the tree was traversed to update the probabilities. Skipping the children of the nodes with small probability masses enabled fast traversal of the tree. In a later study [57], alternative tree construction methods, including regular partitioning of the eigenspace and vector quantization, were considered and the hand dynamics were captured by keeping a histogram of the tree node transitions in the training data (see Section 3). The tree-based filter has a distinctive feature that it supports initialization and single frame pose estimation by simply traversing the whole tree (see Section 6).

5.3.3. Bayesian networks

Bayesian networks allow for representing the posterior through lower dimensional distributions. In [48], a NBP (Nonparametric Belief Propagation) network was used to model the kinematic structure of the hand. The network was designed to enforce the kinematic motion and collision constraints on the links that are modeled as independent rigid bodies. In [56], a dynamical model of the hand, constructed using EDA (see Section 3), was used to construct a Dynamic Bayesian network DBN.

5.3.4. Template database search

In [52], an artificially generated template database was employed for tracking. Features providing scale, position and rotation invariance were extracted from the silhouette contour. During tracking, several hypotheses were kept and the neighborhood around each hypothesis was searched using the beam-search technique to find the best matching templates and establish new hypotheses. Once the best match from the database had been found, it was further refined using a local search algorithm.

5.3.5. Other methods

In an older study [66], the Extended Kalman Filter (EKF) was used and the EKF output was modified by using closed-form hand motion constraints. The ambiguities, which appear as singularities (i.e., the Jacobian becomes singular), were detected to generate pose candidates. Then each candidate was tracked separately.

6. Single frame pose estimation

By single frame pose estimation we mean estimating the pose of the hand using a single image or multiple images taken simultaneously from different views. In terms of the model based approach, the solution to this problem corresponds to a global search over the entire configuration space. Especially with a single image and unconstrained hand motion, single frame pose estimation is an ambiguous problem due to occlusions.

One motivation for addressing this more challenging problem is for the purpose of initializing tracking without imposing too many constraints on the user. If a fully unconstrained solution is available, one can also imagine applying the same algorithm at each frame, thus eliminating the need for complex tracking algorithms presented in the previous sections. However, a less intensive form of tracking operating directly over the configuration space might be needed to resolve ambiguities. The fast motion capability of the hand, which is a source of tracking failures, also motivates single frame pose estimation [127].

6.1. Object detection

Single frame pose estimation has a close relationship with object detection methods used in hand localization (see Section 4.1). If the training data is labeled using pose information, it becomes possible to estimate the pose through classification techniques. However, obtaining pose information for all samples is not feasible; therefore, artificial hand models come into use to cheaply provide synthetic but labeled data.

In [23], the tree-based filter (explained in Section 5.3.2), was employed to implement an initialization module for the filter. The multi-resolution tree corresponds to clusters of similar hand poses, each represented using a template. Traversing the tree based on template matching results provides the hand pose estimate. Successful global hand pose estimation results were demonstrated for a fixed posture of the hand.

In [127], real images containing out of plane rotations of a set of finger-spelling gestures were manually labeled and a binary classifier tree was constructed by unsupervised clustering of Fourier-Mellin transform-based features. In case of classification failures, which are expected due to the sparse sampling of hand postures, missing frames in the image sequence were interpolated using successful estimations to drive graphics animation of a moving hand.

6.2. Image database indexing

Another approach to improve searching large databases of templates is using indexing techniques to retrieve the nearest neighbor(s) of a given input. Athitsos et al. [54] introduced indexing methods for chamfer matching and a probabilistic line matching algorithm for this purpose. In a more recent study, Zhou et al. [128] combined a fast text retrieval technique and Chamfer distances. Although none of these studies has reported results that can support reliable frame-by-frame pose estimation with unconstrained motion, it is still possible to employ them as initialization modules in tracking algorithms or in gesture classification.

6.3. 2D–3D mapping

In [55], a general machine learning approach was proposed based on the idea of learning a mapping from a 2D feature space to the parameter space. Rotation and scale invariant moments of the hand silhouette were utilized to implement the mapping by employing a machine learning architecture called Specialized Mapping Architecture (SMA). Unlike searching algorithms used to retrieve the closest matches in a database, SMA provides continuous pose estimates through regression. Some problems with ambiguous appearances were reported. In a later version of that system [129], multiple hypotheses generation was considered. Similar to the database indexing methods in the previous section experimental results that can support initialization and gesture classification were reported. However, it is hard to draw any useful conclusions about the system's performance over real image sequences.

6.4. Inverse kinematics

Calculating the joint angles given the end effector position and orientation corresponds to a classical inverse kinematics problem. In case of the hand, the fingertip positions are used in a similar way; however, the uniqueness of the solution is not guaranteed. Extensive use of hand model constraints help to regularize the problem. For example, the whole finger flexion can be reduced to 1 DOF by relating PIP, DIP, and MCP flexion angles [44,68,114]. In [68], closed-form solutions were derived to calculate the angles from 2D marker positions under orthographic projection. In [114], a neural network architecture, called PSOM, was trained to construct a mapping from 2D fingertip positions to joint angles.

7. Summary and evaluation

The key characteristics of the full DOF hand pose estimation systems reviewed in this study are summarized in Table 1. These studies were chosen on the basis of generality of their solutions and satisfactory experimental results. The first column provides the reference number while the other columns provide the key characteristics of each system. Specifically, we report: (1) the effective number of DOF that the system targets (i.e., the final DOF after possible reduction due constraints), (2) the number and type of cameras used, (3) the ability of the system to operate in a cluttered background, (4) the features used, (5) the approach used to enforce hand model constraints, (6) the type of the system according to the taxonomy used in this study, (7) systems using a database of templates, (8) details of the algorithm, (9) execution speed, and (10) observed pose restrictions. The last field is provided in an extra row for the corresponding study.

In order to evaluate and compare these systems, one would expect a quantitative evaluation in terms of accuracy and robustness. Accuracy evaluation mainly depends on the availability of ground-truth data. However, obtaining ground truth data for hand pose estimation is a difficult problem. A limited number of studies [47,55,116,120,128,54] have reported precision or jitter measurements based on synthetic data generated by the hand model itself. Such results can provide some valuable insight on the operation of the system or can be taken as a proof of correctness, but it is hard to make performance projections on real data. Without ground truth data, it is possible to provide some quantitative results for rough estimates of jitter by plotting the estimates of a fixed DOF. Also, it would be possible to evaluate robustness by reporting the duration of successful tracking over a long sequence of hand motion. Another possibility is a goal oriented evaluation procedure where the algorithms are tested in the context of an application, as in the case of partial pose estimation systems (see Section 4). In general, evaluations of full DOF pose estimation systems are limited to visual, qualitative evaluations, where the hand model is projected on input image(s) to show how well its projection(s) matches the hand image(s) over a short sequence (See Fig. 7). It is not possible to evaluate or compare the quality of a match; furthermore a good match on the images does not guarantee good accuracy of the system in 3D. However, the image

Table 1
Summary of the systems reviewed

Ref.	DOF	Camera	Clut.	Features	Constr.	Method	Templ.	Details	Speed (*f*/*s*)
Bray 04 [46]	20(6)	1 Depth	Y	Depth	C	SH		SMD	1/4.7
Delamarre 01 [119]	21(0)	1 Depth	Y	Depth		SH		Force model	
	Palm faces camera								
Heap 96 [49]	N/A	1	Y	Edge	L	SH		Weighted least squares	10
Holden 97 [113]	15(6)	1		Marker		SH		Gauss–Newton	
	Palm faces camera								
Lien 98 [122]	17(6)	2		Marker	C	SH		Inverse kinematics (GA)	1/1.5
Lien 05 [70]	17(6)	2		Marker	C	SH		Inverse kinematics (GA)	
	Palm faces camera								
Lu 03 [117]	20(6)	1		Edge, opt. flow, silh., fingers		SH		Force model	4
Nirei 96 [47]	27(6)	2		Silh., opt. flow		SH		NMS, GA	
	Minor global motion, moderate occlusion								
Ouhaddi 99 [123]	Unclear(6)	1		Edge, Silh.	C	SH		NMS	
	Limited finger motion, palm faces camera								
Rehg 94 [21]	21(6)	1		Fingertip and link		SH		Gauss–Newton	10
	5(3) DOF tracking								
Rehg 95 [115]	N/A	1		layered templates		SH		Gradient Descent	
	Two-finger tracking at highly occluding poses								
Wu 99 [124]	Unclear	1		Edges, fingertips	C	SH		Inverse kinematics (LMS)	
	Inconclusive								
Kuch 95 [118]	23(6)	1		Silhouette		SH		Model fitting	
	Separate finger and global pose estimation. In case of fingers, palm faces camera								
Stenger 01 [22]	21(6)	1		Edge		SH		UKF	3
	1(6) DOF tracking								
Ueda 03 [120]	21(6)	4		Visual hull		SH		Force model	1/0.340
	Limited finger and global motion								
Dewaele 04 [59]	21(6)	1 Depth		Surface, 3D point trajectory		SH		ICP	
	Palm faces camera								
Bray 04 [45]	20(6)	1 Depth	Y	Depth	C	MH		Particle filter	
Wu 01 [58]	20(Unclear)	1		Edges, silhouette	L	MH		Particle filter	
	Palm faces camera								
Shimada 01 [52]	Unclear(0)	1		Contours, moments	L	MH	Y	Template matching	30
Shimada 98 [66]	20(6)	1		Silh., fingertip, link	C	MH		EKF	
Stenger 03 [53]	21(6)	1	Y	Oriented edge, color	L	MH	Y	Tree-based filter	1/2
Thayananthan 03 [57]	20(0)	1	Y	Oriented edge, color	L	MH	Y	Tree-based filter	1/2
	Palm faces camera								
Lin 02 [116]	20(3)	1		Edge, Silh.	L	MH (SH)		Particle filter (ICP)	
	Palm faces camera but large in plane rotations are allowed								
Lin 04 [51]	21(6)	1	Y	Edge, Silh.	L	MH (SH)		Particle filter (NMS)	
Sudderth 04 [48]	N/A	1	Y	Oriented edge, color		MH		Belief propagation	
	Only moderate occlusions are allowed								
Zhou 03 [56]	21(6)	1	Y	Edge, color	L	MH (SH)		DBN	8
	Separate global and finger motion								
Athitsos 03 [54]	N/A	1	Y	Oriented edge, lines	L	SF	Y	Database indexing	
	Initialization, gesture classification								
Zhou 05 [128]	N/A	1	Y	Silhouette	L	SF	Y	Database indexing	
	Initialization, gesture classification								
Chua 02 [112]	6(6)	1		Marker		SF		Inverse kinematics	
Lee 93 [44]	14(6)	2		Marker	C	SF		Model fitting	
Nolker 99 [114]	10(0)	1		Fingertip	C	SF		Inverse kinematics	
	Palm faces camera								
Rosales 01 [55]	22(2)	1	Y	Moments	L	SF		2D–3D mapping	
	Initialization, gesture classification								
Stenger 04 [23]	N/A	1	Y	Oriented edge, color	L	SF	Y	Object detection	
	Fixed posture initialization								

Abbreviations: C, closed-form constraints; L, learning-based constraints; SH, single hypothesis; MH, multiple hypotheses; SF, single frame; Y, yes. The entries outside the parentheses refer to the pose of the fingers while the entries within parentheses refer to the global pose.

Fig. 7. A representative experimental output demonstrating the performance of pose estimation (from [57]).

sequences presented could be used to draw some useful conclusions on pose and background restrictions imposed.

The lack of quantitative or goal oriented evaluations indicates that the main concern of researchers is finding promising techniques to overcome the basic challenges in this area. In the rest of this section, we discuss existing work in the context of the challenges and difficulties outlined in Section 2.

7.1. High-dimensional problem

The majority of the systems listed in Table 1 use natural hand motion constraints either in closed form or based on learning to explicitly or implicitly reduce the search space (see column 6). Closed-form dynamic constraints help to reduce the dimension directly (see column 2). Learning-based approaches rely on collecting data using data gloves, and using the data directly or applying some dimensionality reduction first. Some of these systems rely on a large database of templates (see column 8), therefore have some deficiencies in terms of computational power and memory requirements as demonstrated by Shimada et al. [52]. On the other hand, these systems perform global or quasi-global search over the configuration space using complex features; off-line computation of the features is a reasonable way to make the search feasible.

Some systems perform low DOF pose estimation by fixing many DOFs in the hand model to support initialization or simply to track low DOF motion primitives such as pointing or navigation. In these systems, the hand is kept almost as a rigid object and mainly the global pose of the hand is estimated (see Fig. 8 for an example).

The ones that perform initialization are single frame pose estimation systems. Except for marker-based systems, one or more fixed postures are assumed during initialization. Stenger et al. [23] presents single fixed posture initialization results, while other markerless approaches [54,128,55] provide some promising results using multiple postures. Ambiguities and classification errors is an important source of failure in these systems.

7.2. Self-occlusions

A common assumption in most systems is having the palm facing the camera and allowing at most in plane rotations with respect to the camera. There are also a few studies that use multiple views of the hand but do not support any type of global motion or limit finger motion (e.g., one finger at a time or adduction/abduction only). Studies without a comment row in Table 1 correspond to the ones that can tolerate certain amount of out of plane rotations. Some of them [53,117,52,45,46] provide remarkable tracking results under severe occlusions with a viewing direction almost parallel to the palm surface. Fig. 9 shows the tracking results of Lu et al. [117] as a representative.

Model-based methods partially handle occlusions through visibility calculation; however, they do not provide a complete solution to the problem especially in the case of

Fig. 8. A demonstration of pointing gesture tracking (from [22]).

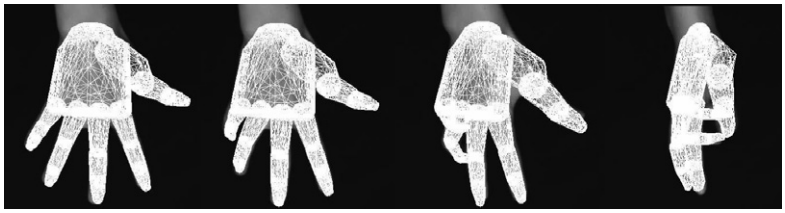

Fig. 9. An example of tracking results containing considerable amount of occlusions and out of plane rotations (from [117]).

single camera systems. Except for the extension of the DigitEyes system [115] and 3D surface or point marker reconstruction systems, there is actually no explicit treatment of occlusions. Therefore, pose restrictions are a part of many full DOF hand pose estimation algorithms in order to avoid large amount of occlusions.

7.3. Processing speed

There are few systems with high processing speed. One of them is the rather old DigitEyes system [21]. It works at 10 Hz on an image processing board and can track three fingers in five DOF motion and the hand in 3 DOF planar motion under uniform background. Another old system is due to Heap et al. [49] running at 10 Hz; however, that system is using a surface model instead of a kinematic hand model. The system of Zhou et al. [56] can operate at 8 Hz on cluttered background. Robust global tracking results (without finger motion) even in case of one hand occluding the other were presented in this study. The fastest system reported in literature is the template-matching system given in [52]. It is implemented using a PC cluster consisting of six PCs and operates at about 30 Hz. Some tracking results under severe occlusion and rapid hand motion were demonstrated.

7.4. Uncontrolled environments

The fourth column of Table 1 shows that there are many systems that can cope with background clutter. By clutter we mean a static background full of many other, mostly non-skin-colored, objects. Most of these systems are multiple hypothesis tracking systems, using edges, oriented edge (i.e., orientation-based chamfer distance) and color combinations. Most other systems should be complemented by strong 2D hand localization algorithms to be able to operate under clutter.

7.5. Rapid hand motion

Restrictions on hand motion speed or, equivalently, desired camera speed can be roughly expressed as the number of frames required to capture certain hand motions. Most of these systems are tested using a hand closing action, where one or more fingers are flexed starting from the stretched pose. In [113,119,45] the testing action was captured in about 6, 20, and 100 frames, respectively. As most of the studies do not report such information, it becomes hard to evaluate the state of the art in terms of hand motion speed.

8. Future research directions

Model-based vision seems to be a promising direction for hand pose estimation. All the studies reviewed here represent important steps taken forward to achieve the ultimate goal but there are also some problems that have not received enough attention. One is the hand model calibration problem, which has received attention only recently [70]. In many applications, precision is important; however, it may not be possible to obtain it with a general manually constructed hand model. Besides, imperfections in the hand model could also result in tracking failures. Automatic calibration of the hand model is not very easy, but without a solution to this problem it may not be possible to use these algorithms in a wide range of applications.

Use of multiple views is mostly limited to marker-based systems. The majority of markerless systems reviewed in this study are designed to operate with a single camera and many of them have a tendency to keep the global hand pose fixed with respect to the camera. If more flexible, non-posed, interaction is required (e.g., for object manipulation tasks), employing multiple cameras would be necessary. Another reason for multiple view systems could be two-handed interaction, which means an increase in occlusions and the dimension of the problem. Possible extensions of existing systems to multiple views would be through best view selection and/or simple combinations of matching error functions. Early combination of multiple view data (i.e., establishing correspondences across cameras) and 3D features have not been explored very well. Such approaches can be more effective to reduce the ambiguities in pose estimation and provide better occlusion handling.

Another important problem is the quantitative evaluation of these systems. Unfortunately it is very difficult to compare different algorithms for many reasons. Design of testing methods, establishing benchmarks and construction of standard databases would give researchers a better idea about the capabilities of each algorithm, in order to take principled steps for the development of better systems.

From the summary and discussion provided in the previous section, we can also conclude that there are a lot of open problems that could lead to quite different, new approaches. In fact, we were able to find some attempts in this direction. One is a hierarchical factorization algorithm presented in [130]. Factorization represents a bottom-up approach, where 2D point features in the images are tracked to reconstruct articulated motion. The algorithm was tested on synthetic hand images and was not completely implemented; therefore, it was not included in our taxonomy. Another one [131] is related to object manipulation tasks in VEs. Instead of tracking the hand all the time, an object-centered approach without any pose estimation was proposed. In this system, the virtual environment was projected on the images, and when the hand came close to the objects in the images, some 3D features were extracted for classifying manipulative and controlling gestures.

It should be noted that our intention is not to argue for the two studies mentioned above or render existing solutions useless, but to support the idea that employing entirely different techniques would always be a part of this research. Probably, appropriate combinations of various techniques will provide more robust solutions to the pose estimation problem.

9. Conclusions

CV has a distinctive role in the development of direct sensing-based HCI. However, various challenges must be addressed in order to satisfy the demands of potential interaction methods. Currently, CV-based pose estimation has some limitations in processing arbitrary hand actions. Incorporating the full functionality of the hand in HCI requires capturing the whole hand motion. However, CV can only provide support for only a small range of hand actions under restrictive conditions. This approach has certain drawbacks in terms of natural interaction requirements (see Appendix A). The hand gestures used in existing real-time systems are limited to a carefully chosen vocabulary of symbolic gestures that mainly serve to issue commands.

Other types of gestures such as gesticulations and manipulation operation, or high DOF interaction require more efficient general pose estimation algorithms. Full DOF hand pose estimation is still a big challenge in CV. The high dimensionality of the problem and self-occlusions are the main obstacles for implementing real-time, robust systems. Nevertheless, various techniques have been explored to engineer full DOF hand pose estimation systems. The existence of an expensive but high speed system is quite encouraging [52]. However, pose restrictions and the lack of an implementation that is part of a real world application indicate that there are still a lot of open problems to be solved in order to obtain robustness, accuracy, and high processing speed.

Acknowledgments

This work was supported by NASA under Grant No. NCC5-583. We acknowledge the editors and reviewers for their constructive comments and pointers to some references that we missed in the first version of this paper.

Appendix A. Hand gesture-based HCI

In this appendix, we provide a brief summary of the potential interaction styles to be employed in advanced HCI systems. In a sense, these interfaces represent the requirements imposed on sensing and interpretation technologies.

There is a growing body of literature on gesture-based HCI. The main objective in this research area is the development of design and evaluation methods to assure usable interaction methods. The ultimate system would be a completely passive system that would interpret all types of hand motions without any constraints; however, current interpretation and sensor technology renders this ultimate interface unfeasible. Researchers should find an equilibrium between limitations of the interpretation technology and natural interaction requirements. Various interaction methods have been proposed as a solution to this very challenging problem. The main difference among these systems is the way that the hand motion is interpreted, which also determines the characteristics of the sensor to be used.

In some applications, the hand serves mainly as a high DOF input device. These applications require high DOF input and suffer from bottlenecks in the classical input devices. Combined with the dexterity and hand–eye coordination skills of humans, the hand, which can technically be considered as a complex input device with more than 20 DOF, can become an easy to use high DOF control device. Sturman [13] presented a design and evaluation method that demonstrates the versatility of the *whole hand input* in a wide range applications with special emphasis on high DOF interaction. Continuous input provided by data-gloves can be mapped to control signals in applications such as complex machinery or manipulator control, computer-based puppetry and musical performance.

In many other applications, maximizing the DOF in the input is not the major concern. Instead, generic interaction styles made up of combinations of low DOF (mostly six or less) motion primitives such as pointing, rotating, and selecting an object are being investigated to provide natural HCI. In the rest of this appendix, we will briefly summarize some gesture-based interaction styles with different characteristics and discuss the importance of pose estimation in their implementation.

A.1. Object manipulation

Object manipulation is a general interaction style that is used in VEs. One application of VEs is to simulate the real-world with a life-like interaction. Surgical simulations [14], immersive training systems [15], diagnosis and therapy in computational neuroscience [132] are some examples. However, a realistic interaction is a complex process that requires full DOF precise hand motion capture, implementation of physics related details of virtual objects, and often employment of force-feedback.

In many VE applications, abstractions of objects and events are employed. In these systems the user *navigates* in the VE, *selects* objects and *manipulates* them by changing their attributes. A wide range of advanced techniques, including eye-gaze or head pose estimation, and human body pose estimation are employed in implementing these tasks. When the hand is used, often low DOF hand pose estimation is utilized [10,11]. Hand position and orientation or pointing direction can provide sufficient input to extend a selection ray, to navigate in VE, and to translate and rotate objects. Some systems support more complex manipulation operations such as resizing and reshaping. On the other hand, lack of force-feedback can still be an issue that affects the naturalness of interaction [12]. Another important component of VEs is the control system, where gesture commands (see next subsection) come into use together with 2D or 3D menus, buttons or tools, which can all be regarded as objects that can also be manipulated in a similar way [133]. These systems, even the ones using data-gloves, look like a 3D extension of

the "direct manipulation" [134] style that is used with conventional input devices and GUI. The use of the hand is often limited to rigid motion in 3D.

A.2. Command and control interfaces

Command and control interaction is common in many applications and gestural communication could be very helpful in these interfaces. There are various types of gestures that people use for communication purposes. Detailed gesture taxonomies in the context of HCI can be found in [89,19,2]. In existing UI designs or systems, the majority of gestures are the ones with a symbolic interpretation leading to simple sign languages tailored for HCI [1–4]. In particular, a predefined set of static hand postures or dynamic gestures form a gesture vocabulary to make up a gesture language that provides structured, unambiguous communication. In these applications, hand motion data is classified into an element of the gesture set and interpreted based only on its symbolic content. One exception to this processing scheme is the pointing gesture, which should always be complemented by a pointing direction estimate. A disadvantage of gesture–language-based communication is the difficulty of learning, since it requires *recall* without any visual guidance as in menu-based interfaces [2].

A.3. Multimodal UI

Multimodal UI is another important trend in HCI. Natural human communication involves multiple modalities. From a technical point of view, multi-modal interfaces can help to resolve ambiguous situations and increase the recognition accuracy of the system. The most promising multimodal interaction style is the combination of speech and hand gestures, which also affects the way that hand gestures are used. Wexelblat [5] criticized the use of gesture commands as an interaction method. He argued that there was little gain in functionality by using the hand that way due to its functional equivalence to pressing a key on the keyboard and the additional cognitive load in recalling the correct gesture. Instead, he proposed the use of *gesticulations*, which are spontaneous movements of the hand and arms that accompany speech, especially in descriptions. It should be mentioned that gesticulations make up 90% of the human gestures [2]. However, modeling and interpreting all gesticulations is an open problem that needs further investigation [6].

Bolt [7] presented a more practical "Put That There" interface that demonstrates the power of speech–gesture combination. The user points at an object or location while issuing voice commands to manipulate them. Pointing belongs to the category of gesticulations called "deictic gestures". Later, the interface was extended by incorporating "iconic gestures" [8]. Iconic gestures are gesticulations depicting some feature of the object, an action or an event. They allow commands such as "Move the teapot like this". After establishing an analogy between hand shape and the object, the transformation derived from hand motion can be applied to the object. Interpretation of iconic gestures, which are free-form gestures without a specific pattern, is a more complex process that raises challenging research problems [135]. This type of use of the hand enables more natural and probably more complex object manipulation. Recent experimental evidence shows that users overwhelmingly prefer speech and speech–gesture combination to pure gestural interfaces [136,9]. A review and some design guidelines for this type interface can be found in [9].

A.4. Discussion

The above interfaces capitalize on either the *manipulative* or the *communicative* skills of humans. In either case, there is a tendency to import the corresponding functionality naturally to allow intuitive interaction. Manipulation corresponds to the main functionality of the hand and natural effective manipulation requires high DOF pose estimation. In the case of communication, there are two approaches lying at opposite ends. The first one is an active approach that asks the users to memorize a gesture vocabulary. The second one is a relatively passive approach in terms of hand usage and utilizes gesticulations and speech. Interpretation of gesticulations again relies on pose estimation.

All of the above interaction methods are experimental and their use is limited to laboratory environments. Their success depends mainly on human factors but the quality of the input has a crucial role too. Broad deployment of such interaction methods would require processing arbitrary hand motions and delivering data that support high recognition accuracy and precision. In that respect, data-gloves provide very general purpose data (i.e., the joint angles) regardless of the application. In CV-based approaches, there is a dilemma between appearance-based or 3D modeling of the hand motion. Appearance-based methods allow high speed processing with a loss of generality while 3D model-based methods provide generality at a higher cost of computational power. However, if we consider high DOF control tasks, object manipulation, and the interpretation of gesticulations, 3D approach becomes a requirement.

References

[1] F.K.H. Quek, Unencumbered gestural interaction, IEEE MultiMedia 3 (4) (1996) 36–47.
[2] M. Turk, Gesture recognition, in: K.M. Stanney (Ed.), Handbook of Virtual Environments: Design, Implementation, and Applications, Lawrence Erlbaum Associates, Hillsdale, N.J, 2002, pp. 223–238.
[3] S. Lenman, L. Bretzner, B. Thuresson, Using marking menus to develop command sets for computer vision based hand gesture interfaces, in: NordiCHI '02: Second Nordic Conference on Human-Computer Interaction, ACM Press, New York, NY, USA, 2002, pp. 239–242.

[4] M. Nielsen, M. Störring, T.B. Moeslund, E. Granum, A procedure for developing intuitive and ergonomic gesture interfaces for HCI, in: 5th International Gesture Workshop, 2003, pp. 409–420.
[5] A. Wexelblat, An approach to natural gesture in virtual environments, ACM Transactions on Computer-Human Interaction 2 (3) (1995) 179–200.
[6] F. Quek, D. McNeill, R. Bryll, S. Duncan, X.-F. Ma, C. Kirbas, K.E. McCullough, R. Ansari, Multimodal human discourse: gesture and speech, ACM Transactions on Computer-Human Interaction 9 (3) (2002) 171–193.
[7] R.A. Bolt, Put-that-there: voice and gesture at the graphics interface, in: SIGGRAPH '80: 7th Annual Conference on Computer Graphics and Interactive Techniques, ACM Press, New York, NY, USA, 1980, pp. 262–270.
[8] D.B. Koons, C.J. Sparrell, Iconic: speech and depictive gestures at the human-machine interface, in: CHI '94: Conference Companion on Human Factors in Computing Systems, ACM Press, New York, NY, USA, 1994, pp. 453–454.
[9] M. Billinghurst, Put that where? Voice and gesture at the graphics interface, SIGGRAPH Computer Graphics 32 (4) (1998) 60–63.
[10] D. Bowman, Principles for the design of performance-oriented interaction techniques, in: K.M. Stanney (Ed.), Handbook of Virtual Environments: Design, Implementation, and Applications, Lawrence Erlbaum Associates, Hillsdale, NJ, 2002, pp. 201–207.
[11] J. Gabbard, A taxonomy of usability characteristics in virtual environments, Master's thesis, Department of Computer Science, University of Western Australia, 1997.
[12] V. Buchmann, S. Violich, M. Billinghurst, A. Cockburn, FingARtips: gesture based direct manipulation in augmented reality, in: GRAPHITE '04: 2nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, ACM Press, New York, NY, USA, 2004, pp. 212–221.
[13] D.J. Sturman, Whole hand input, Ph.D. thesis, MIT, 1992.
[14] A. Liu, F. Tendick, K. Cleary, C. Kaufmann, A survey of surgical simulation: applications, technology, and education, Presence: Teleoperators and Virtual Environments 12 (6) (2003) 599–614.
[15] VGX, Virtualglovebox, http://biovis.arc.nasa.gov/vislab/vgx.htm.
[16] D.J. Sturman, D. Zeltzer, A survey of glove-based input, IEEE Computer Graphics and Applications 14 (1) (1994) 30–39.
[17] E. Foxlin, Motion tracking requirements and technologies, in: K.M. Stanney (Ed.), Handbook of Virtual Environments: Design, Implementation, and Applications, Lawrence Erlbaum Associates, Hillsdale, NJ, 2002, pp. 163–210.
[18] M.W. Krueger, T. Gionfriddo, K. Hinrichsen, Videoplace an artificial reality, in: SIGCHI Conference on Human Factors in Computing Systems, ACM Press, New York, NY, USA, 1985, pp. 35–40.
[19] V.I. Pavlovic, R. Sharma, T.S. Huang, Visual interpretation of hand gestures for human-computer interaction: a review, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 677–695.
[20] J.L. Crowley, J. Coutaz, F. Bérard, Perceptual user interfaces: things that see, Communications of the ACM 43 (3) (2000) 54.
[21] J. Rehg, T. Kanade, Digiteyes: vision-based hand tracking for human-computer interaction, in: Workshop on Motion of Non-Rigid and Articulated Bodies, 1994, pp. 16–24.
[22] B. Stenger, P.R.S. Mendonca, R. Cipolla, Model-based 3D tracking of an articulated hand, IEEE Computer Society Conference on Computer Vision and Pattern Recognition 02 (2001) 310.
[23] B. Stenger, A. Thayananthan, P. Torr, R. Cipolla, Hand pose estimation using hierarchical detection, in: 8th European Conference on Computer Vision Workshop on Human-Computer Interaction, vol. 3058, Springer, Prague, Czech Republic, 2004, pp. 102–112.
[24] Y. Wu, T.S. Huang, Hand modeling, analysis, and recognition, IEEE Signal Processing Magazine 18 (3) (2001) 51–60.
[25] Y. Wu, T.S. Huang, Vision-based gesture recognition: a review, Lecture Notes in Computer Science 1739 (1999) 103+.
[26] Y. Wu, T.S. Huang, Human hand modeling and animation in the context of HCI, in: IEEE International Conference on Image Processing, vol. 3, 1999, pp. 6–10.
[27] M. Kohler, S. Schroter, A survey of video-based gesture recognition—stereo and mono systems, Tech. Rep. Nr. 693/1998, Informatik VII, University of Dortmund, August 1998.
[28] R. Watson, A survey of gesture recognition techniques, Tech. Rep. TCD-CS-93-11, Trinity College, Dublin 2, 1993.
[29] B. Stenger, Model-based hand tracking using a hierarchical bayesian filter, Ph.D. thesis, Department of Engineering, University of Cambridge, 2004.
[30] J.K. Aggarwal, Q. Cai, W. Liao, B. Sabata, Articulated and elastic non-rigid motion: a review, in: IEEE Computer Society Workshop on Motion of Non-rigid and Articulated Objects, Austin, Texas, 1994, pp. 16–22.
[31] J.K. Aggarwal, Q. Cai, Human motion analysis: a review, Computer Vision and Image Understanding 73 (3) (1999) 428–440.
[32] D.M. Gavrila, The visual analysis of human movement: a survey, Computer Vision and Image Understanding 73 (1) (1999) 82–98.
[33] T.B. Moeslund, E. Granum, A survey of computer vision-based human motion capture, Computer Vision and Image Understanding 81 (3) (2001) 231–268.
[34] L. Wang, W. Hu, T. Tan, Recent developments in human motion analysis, Pattern Recognition 36 (3) (2003) 585–601.
[35] J.S.S. Wang, Video analysis of human dynamics-a survey, Real-Time Imaging 9 (5) (2003) 321–346.
[36] J. O'Rourke, N.I. Badler, Model-based image analysis of human motion using constraint propagation, IEEE Transactions on Pattern Analysis and Machine Intelligence 2 (6) (1980) 522–536.
[37] D.M. Gavrila, L.S. Davis, 3D Model-based tracking of humans in action: a multi-view approach, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1996, pp. 73–80.
[38] D.G. Lowe, Fitting parameterized three-dimensional models to images, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (5) (1991) 441–450.
[39] T. Drummond, R. Cipolla, Real-time visual tracking of complex structures, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7) (2002) 932–946.
[40] A. Hollister, W.L. Buford, L.M. Myers, D.J. Giurintano, A. Novick, The axes of rotation of the thumb carpometacarpal joint, Journal of Orthopaedic Research 10 (3) (1992) 454–460.
[41] I. Albrecht, J. Haber, H.-P. Seidel, Construction and animation of anatomically based human hand models, in: ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Eurographics Association, 2003, pp. 98–109.
[42] J.J. Kuch, T.S. Huang, Human computer interaction via the human hand: a hand model, in: Twenty-Eighty Asilomar Conference on Signal, Systems, and Computers, 1994, pp. 1252– 1256.
[43] W. Griffin, R. Findley, M. Turner, M. Cutkosky, Calibration and mapping of a human hand for dexterous telemanipulation, in: ASME IMECE 2000 Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems, 2000, pp. 1–8.
[44] J. Lee, T. Kunii, Constraint-based hand animation, in: Models and Techniques in Computer Animation, Springer, Tokyo, 1993, pp. 110–127.
[45] M. Bray, E. Koller-Meier, L.V. Gool, Smart particle filtering for 3D hand tracking, in: Sixth IEEE International Conference on Automatic Face and Gesture Recognition, IEEE Computer Society, Los Alamitos, CA, USA, 2004, p. 675.
[46] M. Bray, E. Koller-Meier, P. Müller, L.V. Gool, N.N. Schraudolph, 3D Hand tracking by rapid stochastic gradient descent using a skinning model, in: First European Conference on Visual Media Production, London, 2004, pp. 59–68.
[47] K. Nirei, H. Saito, M. Mochimaru, S. Ozawa, Human hand tracking from binocular image sequences, in: 22th International Conference on Industrial Electronics, Control, and Instrumentation, 1996, pp. 297–302.

[48] E.B. Sudderth, M.I. Mandel, W.T. Freeman, A.S. Willsky, Visual hand tracking using nonparametric belief propagation, in: IEEE CVPR Workshop on Generative Model Based Vision, IEEE Computer Society, Washington, DC, USA, 2004, p. 189.
[49] T. Heap, D. Hogg, Toward 3D hand tracking using a deformable model, in: 2nd International Conference on Automatic Face and Gesture Recognition, IEEE Computer Society, Washington, DC, USA, 1996, p. 140.
[50] J. Lin, Y. Wu, T.S. Huang, Modeling the constraints of human hand motion, in: IEEE Human Motion Workshop, 2000, pp. 121–126.
[51] J.Y. Lin, Y. Wu, T.S. Huang, 3D Model-based hand tracking using stochastic direct search method, in: Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Seoul, Korea, 2004, pp. 693+.
[52] N. Shimada, K. Kimura, Y. Shirai, Real-time 3D hand posture estimation based on 2D appearance retrieval using monocular camera, in: ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, IEEE, Vancouver, BC, Canada, 2001, pp. 23–30.
[53] B. Stenger, A. Thayananthan, P.H.S. Torr, R. Cipolla, Filtering using a tree-based estimator, in: Ninth IEEE International Conference on Computer Vision, IEEE Computer Society, Washington, DC, USA, 2003, p. 1063.
[54] V. Athitsos, S. Sclaroff, Estimating 3D hand pose from a cluttered image, IEEE Computer Society Conference on Computer Vision and Pattern Recognition 02 (2003) 432.
[55] R. Rosales, V. Athitsos, L. Sigal, S. Sclaroff, 3D Hand pose reconstruction using specialized mappings, in: Eighth IEEE International Conference on Computer Vision, vol. 1, 2001, pp. 378–385.
[56] H. Zhou, T.S. Huang, Tracking articulated hand motion with eigen dynamics analysis, in: Ninth IEEE International Conference on Computer Vision, IEEE Computer Society, Washington, DC, USA, 2003, p. 1102.
[57] A. Thayananthan, B. Stenger, P.H.S. Torr, R. Cipolla, Learning a kinematic prior for tree-based filtering, in: British Machine Vision Conference, vol. 2, 2003, pp. 589–598.
[58] Y. Wu, Y.L. Lin, T.S. Huang, Capturing natural hand articulation, in: Eighth IEEE International Conference on Computer Vision, 2001, pp. 426–432.
[59] G. Dewaele, F. Devernay, R. Horaud, Hand motion from 3D point trajectories and a smooth surface model, in: T. Pajdla, J. Matas (Eds.), 8th European Conference on Computer Vision, LNCS 3021, vol. I, Springer, Prague, Czech Republic, 2004, pp. 495–507.
[60] J.F. O'Brien, R.E. Bodenheimer, G.J. Brostow, J.K. Hodgins, Automatic joint parameter estimation from magnetic motion capture data, in: Graphics Interface Conference, 2000, pp. 53–60.
[61] L. Taycher, J. Trevor, Recovering articulated model topology from observed motion, in: Neural Information Processing Systems, 2002, pp. 1311–1318.
[62] A.G. Kirk, J.F. O'Brien, D.A. Forsyth, Skeletal parameter estimation from optical motion capture data, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, pp. 782–788.
[63] I.A. Kakadiaris, D. Metaxas, Three-dimensional human body model acquisition from multiple views, International Journal of Computer Vision 30 (3) (1998) 191–218.
[64] B. Buchholz, T.J. Armstrong, S.A. Goldstein, Anthropometric data for describing the kinematics of the human hand, Ergonomics 35 (3) (1992) 261–273.
[65] N.A. Davidoff, A. Freivalds, A graphical model of the human hand using catia, International Journal of Industrial Ergonomics 12 (1993) 255–264.
[66] N. Shimada, Y. Shirai, Y. Kuno, J. Miura, Hand gesture estimation and model refinement using monocular camera—ambiguity limitation by inequality constraints, in: Third IEEE International Conference on Face and Gesture Recognition, 1998, p. 268.
[67] A. Thayananthan, B. Stenger, P.H.S. Torr, R. Cipolla, Shape context and chamfer matching in cluttered scenesIEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. I, IEEE Computer Society, Los Alamitos, CA, USA, 2003, p. 127.
[68] C.S. Chua, H.Y. Guan, Y.K. Ho, Model-based finger posture estimation, in: Fourth Asian Conference on Computer Vision, 2000, pp. 43–48.
[69] H. Hu, X. Gao, J. Li, J. Wang, H. Liu, Calibrating human hand for teleoperating the hit/dlr hand, in: IEEE International Conference on Robotics and Automation, vol. 5, 2004, pp. 4571–4576.
[70] C. Lien, A scalable model-based hand posture analysis system, Machine Vision and Applications 16 (3) (2005) 157–169.
[71] S. Malik, J. Laszlo, Visual touchpad: a two-handed gestural input device, in: ICMI '04: 6th International Conference on Multimodal Interfaces, ACM Press, New York, NY, USA, 2004, pp. 289–296.
[72] K. Oka, Y. Sato, H. Koike, Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems, in: FGR '02: Fifth IEEE International Conference on Automatic Face and Gesture Recognition, IEEE Computer Society, Washington, DC, USA, 2002, p. 429.
[73] J. Letessier, F. Bérard, Visual tracking of bare fingers for interactive surfaces, in: UIST '04: 17th Annual ACM symposium on User Interface Software and Technology, ACM Press, New York, NY, USA, 2004, pp. 119–122.
[74] Z. Mo, J.P. Lewis, U. Neumann, Smartcanvas: a gesture-driven intelligent drawing desk system, in: IUI '05: 10th International Conference on Intelligent User Interfaces, ACM Press, New York, NY, USA, 2005, pp. 239–243.
[75] C. von Hardenberg, F. Bérard, Bare-hand human–computer interaction, in: PUI '01: Workshop on Perceptive User Interfaces, ACM Press, New York, NY, USA, 2001, pp. 1–8.
[76] J. Crowley, F. Bérard, J. Coutaz, Finger tracking as an input device for augmented reality, in: IWAGFR '95: International Workshop on Gesture and Face Recognition, 1995, pp. 195–200.
[77] H. Koike, Y. Sato, Y. Kobayashi, Integrating paper and digital information on enhanceddesk: a method for realtime finger tracking on an augmented desk system, ACM Transactions on Computer-Human Interaction 8 (4) (2001) 307–322.
[78] J. Segen, S. Kumar, Look ma, no mouse! Communications of the ACM 43 (7) (2000) 102–109.
[79] K. Abe, H. Saito, S. Ozawa, 3D drawing system via hand motion recognition from two cameras, in: IEEE International Conference on Systems, Man, and Cybernetics, vol. 2, 2000, pp. 840–845.
[80] J. Segen, S. Kumar, Gesture VR: vision-based 3D hand interface for spatial interaction, in: Sixth ACM International Conference on Multimedia, ACM Press, New York, NY, USA, 1998, pp. 455–464.
[81] Y. Segen, S. Kumar, Shadow gesture: 3D hand pose estimation using a single camera, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999, pp. 479–485.
[82] R.G. O'Hagan, A. Zelinsky, S. Rougeaux, Visual gesture interfaces for virtual environments, Interacting with Computers 14 (2002) 231–250.
[83] Y. Sato, M. Saito, H. Koik, Real-time input of 3D pose and gestures of a user's hand and its applications for HCI, in: IEEE Virtual Reality Conference, IEEE Computer Society, 2001, p. 79.
[84] H. Kim, D.W. Fellner, Interaction with hand gesture for a back-projection wall, in: CGI '04: Computer Graphics International, IEEE Computer Society, Washington, DC, USA, 2004, pp. 395–402.
[85] C. Maggioni, K.B., Gesturecomputer—history, design and applications, in: R. Cipolla, A. Pentland (Eds.), Computer Vision for Human-Machine Interaction, Cambridge, 1998, pp. 312–325.
[86] A. Utsumi, J. Ohya, Multiple-hand-gesture tracking using multiple cameras, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999, pp. 473–478.
[87] V.I. Pavlovic, R. Sharma, T.S. Huang, Invited speech: gestural interface to a visual computing environment for molecular biologists, in: FG '96: 2nd International Conference on Automatic Face and Gesture Recognition, IEEE Computer Society, Washington, DC, USA, 1996, p. 30.

[88] J. Martin, V. Devin, J.L. Crowley, Active hand tracking, in: FG '98: 3rd. International Conference on Face & Gesture Recognition, IEEE Computer Society, Washington, DC, USA, 1998, p. 573.
[89] F. Quek, Eyes in the interface, Image and Vision Computing 13 (6) (1995) 511–525.
[90] R. Kjeldsen, J. Kender, Toward the use of gesture in traditional user interfaces, in: International Conference on Automatic Face and Gesture Recognition, 1996, pp. 151–156.
[91] S. Ahmad, A usable real time 3D hand tracker, in: Twenty-Eighth Asilomar Conference on Signals, Systems and Computers, vol. 2, 1994, pp. 1257–1261.
[92] J. MacCormick, M. Isard, Partitioned sampling, articulated objects, and interface-quality hand tracking, in: 6th European Conference on Computer Vision-Part II, Springer-Verlag, London, UK, 2000, pp. 3–19.
[93] R. Cipolla, N. Hollinghurst, Human–robot interface by pointing with uncalibrated stereo vision, Image and Vision Computing 14 (3) (1996) 171–178.
[94] A. Heap, Real-time hand tracking and gesture recognition using smart snakes, in: Interface to Human and Virtual Worlds, Montpellier, France, 1995.
[95] P.A. Viola, M.J. Jones, Rapid object detection using a boosted cascade of simple features, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001, pp. 511–518.
[96] M. Kolsch, M. Turk, Robust hand detection, Sixth IEEE International Conference on Automatic Face and Gesture Recognition (2004) 614.
[97] E.-J. Ong, R. Bowden, A boosted classifier tree for hand shape detection, in: Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004, pp. 889–894.
[98] Y. Wu, T. Huang, View-independent recognition of hand postures, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. II, 2000, pp. 88–94.
[99] Y. Wu, T. Huang, An adaptive self-organizing color segmentation algorithm with application to robust real time human hand localization, in: Asian Conference on Computer Vision, 2000, pp. 1106–1111.
[100] K.H. Jo, Y. Kuno, Y. Shirai, Manipulative hand gesture recognition using task knowledge for human computer interaction, in: FG '98: 3rd. International Conference on Face & Gesture Recognition, IEEE Computer Society, Washington, DC, USA, 1998, p. 468.
[101] R. O'Hagan, A. Zelinsky, Finger track—a robust and real-time gesture interface, in: AI '97: 10th Australian Joint Conference on Artificial Intelligence, Springer-Verlag, London, UK, 1997, pp. 475–484.
[102] C. Jennings, Robust finger tracking with multiple cameras, in: RATFG-RTS '99: International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, IEEE Computer Society, Washington, DC, USA, 1999, p. 152.
[103] J. Davis, M. Shah, Toward 3D gesture recognition, International Journal of Pattern Recognition and Artificial Intelligence 13 (3) (1999) 381–393.
[104] J.H. Usabiaga, Global hand pose estimation by multiple camera ellipse tracking, Master's thesis, Department of Computer Science, University of Nevada, Reno, 2005.
[105] T.B. Moeslund, L. Norgaard, A brief overview of hand gestures used in wearable human computer interfaces, Tech. rep., Aalborg University, Denmark, 2002.
[106] J. Deutscher, B. North, B. Bascle, A. Blake, Tracking through singularities and discontinuities by random sampling, in: Seventh IEEE International Conference on Computer Vision, 1999, pp. 1144–1149.
[107] T. Heap, D. Hogg, Wormholes in shape space: tracking through discontinuous changes in shape, in: Sixth IEEE International Conference on Computer Vision, Bombay, India, 1998, pp. 344–349.
[108] D.D. Morris, J. Rehg, Singularity analysis for articulated object tracking, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1998, pp. 289–296.
[109] J. Rehg, D.D. Morris, T. Kanade, Ambiguities in visual tracking of articulated objects using two- and three-dimensional models, International Journal of Robotics Research 22 (6) (2003) 393–418.
[110] F. Martin, R. Horaud, Multiple camera tracking of rigid objects, International Journal of Robotics Research 21 (2) (2002) 97–113.
[111] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking, IEEE Transactions on Signal Processing 50 (2) (2002) 173–188.
[112] C. Chua, H. Guan, Y. Ho, Model-based 3D hand posture estimation from a single 2D image, Image and Vision Computing 20 (3) (2002) 191–202.
[113] E. Holden, Visual recognition of hand motion, Ph.D. thesis, Department of Computer Science, University of Western Australia, 1997.
[114] C. Nölker, H. Ritter, Grefit: visual recognition of hand postures, in: A. Braffort, R. Gherbi, S. Gibet, J. Richardson, D. Teil (Eds.), Gesture-Based Communication in Human-Computer Interaction: Proc. International Gesture Workshop, France, Springer Verlag, LNAI 1739, 1999, pp. 61–72.
[115] J.M. Rehg, T. Kanade, Model-based tracking of self-occluding articulated objects, in: Fifth IEEE International Conference on Computer Vision, IEEE Computer Society, Washington, DC, USA, 1995, pp. 612–617.
[116] J. Lin, Y. Wu, T.S. Huang, Capturing human hand motion in image sequences, in: Workshop on Motion and Video Computing, 2002, pp. 99–104.
[117] S. Lu, D. Metaxas, D. Samaras, J. Oliensis, Using multiple cues for hand tracking and model refinement, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003, pp. II: 443–450.
[118] J.J. Kuch, T.S. Huang, Vision based hand modeling and tracking for virtual teleconferencing and telecollaboration, in: Fifth IEEE International Conference on Computer Vision, 1995, pp. 666–671.
[119] Q. Delamarre, O. Faugeras, 3D Articulated models and multiview tracking with physical forces, Computer Vision and Image Understanding 81 (3) (2001) 328–357.
[120] E. Ueda, Y. Matsumoto, M. Imai, T. Ogasawara, A hand-pose estimation for vision-based human interfaces, IEEE Transactions on Industrial Electronics 50 (4).
[121] A. Laurentini, The visual hull concept for silhouette-based image understanding, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (2) (1994) 150–162.
[122] C.C. Lien, C.L. Huang, Model based articulated hand motion tracking for gesture recognition, Image and Vision Computing 16 (1998) 121–134.
[123] H. Ouhaddi, P. Horain, 3D Hand gesture tracking by model registration, in: International Workshop on Synthetic—Natural Hybrid Coding and Three Dimensional Imaging, 1999.
[124] Y. Wu, T.S. Huang, Capturing articulated human hand motion: A divide and conquer approach, in: Seventh IEEE International Conference on Computer Vision, 1999, pp. 606–611.
[125] M. Isard, A. Blake, Contour tracking by stochastic propagation of conditional density, in: European Conference on Computer Vision, vol. 1, Cambridge UK, 1996, pp. 343–356.
[126] T.J. Cham, J.M. Rehg, A multiple hypothesis approach to figure tracking, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999, pp. 239–245.
[127] C. Tomasi, S. Petrov, A. Sastry, 3D Tracking = Classification + Interpolation, in: Ninth IEEE International Conference on Computer Vision, vol. 2, 2003, pp. 1441–1448.
[128] H. Zhou, T. Huang, Okapi-chamfer matching for articulated object recognition, in: Tenth IEEE International Conference on Computer Vision, IEEE Computer Society, Washington, DC, USA, 2005, pp. 1026–1033.
[129] R. Rosales, S. Sclaroff, Algorithms for inference in specialized maps for recovering 3D hand pose, in: Fifth IEEE International Conference on Automatic Face and Gesture Recognition, IEEE Computer Society, Los Alamitos, CA, USA, 2002, p. 0143.

[130] H. Zhou, T.S. Huang, Recovering articulated motion with a hierarchical factorization method, in: Gesture Workshop, 2003, pp. 140–151.
[131] G. Ye, J.J. Corso, G.D. Hager, Visual modeling of dynamic gestures using 3D appearance and motion features, in: B. Kisacanin, V. Pavlovic, T. Huang (Eds.), Real-Time Vision for Human-Computer Interaction, Springer-Verlag, New York Inc., Secaucus, NJ, USA, 2005, pp. 103–120, to appear.
[132] J. Leigh, C.A. Vasilakis, T.A. DeFanti, R. Grossman, C. Assad, B. Rasnow, A. Protopappas, E.D. Schutter, J.M. Bower, Virtual reality in computational neuroscience (1995) 293–306.
[133] Y. Boussemart, F. Rioux, F. Rudzicz, M. Wozniewski, J.R. Cooperstock, A framework for 3D visualization and manipulation in an immsersive space using an untethered bimanual gestural interface, in: ACM Symposium on Virtual Reality Software and Technology, ACM Press, New York, NY, USA, 2004.
[134] B. Shneiderman, Designing the User Interface: Strategies for Effective Human-Computer Interaction, Addison-Wesley Publishing Co, USA, 1987.
[135] T. Sowa, I. Wachsmuth, Interpretation of shape-related iconic gestures in virtual environments, in: GW '01: Revised Papers from the International Gesture Workshop on Gesture and Sign Languages in Human-Computer Interaction, Springer-Verlag, London, UK, 2001, pp. 21–33.
[136] A. Corradini, P.R. Cohen, On the relationships among speech, gestures, and object manipulation in virtual environments: Initial evidence, in: International CLASS Workshop on Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems, Copenhagen, Denmark, 2002.