Hindawi Publishing Corporation
ISRN Artificial Intelligence
Volume 2013, Article ID 514641, 18 pages
http://dx.doi.org/10.1155/2013/514641

Review Article

3D Gestural Interaction: The State of the Field

Joseph J. LaViola Jr.

Department of EECS, University of Central Florida, Orlando, FL 32816, USA

Correspondence should be addressed to Joseph J. LaViola Jr.; jjl@eecs.ucf.edu

Received 9 September 2013; Accepted 14 October 2013

Academic Editors: O. Castillo, R.-C. Hwang, and P. Kokol

3D gestural interaction provides a powerful and natural way to interact with computers using the hands and body for a variety of different applications including video games, training and simulation, and medicine. However, accurately recognizing 3D gestures so that they can be reliably used in these applications poses many different research challenges. In this paper, we examine the state of the field of 3D gestural interfaces by presenting the latest strategies on how to collect the raw 3D gesture data from the user and how to accurately analyze this raw data to correctly recognize 3D gestures users perform. In addition, we examine the latest in 3D gesture recognition performance in terms of accuracy and gesture set size and discuss how different applications are making use of 3D gestural interaction. Finally, we present ideas for future research in this thriving and active research area.

1. Introduction

Ever since Sutherland's vision of the ultimate display [1], the notion of interacting with computers naturally and intuitively has been a driving force in the field of human computer interaction and interactive computer graphics. Indeed, the notion of the post-WIMP interface (Windows, Icons, Menus, Point and Click) has given researchers the opportunity to explore alternative forms of interaction over the traditional keyboard and mouse [2]. Speech input, brain computer interfaces, and touch and pen-computing are all examples of input modalities that attempt to bring a synergy between user and machine and that provide a more direct and natural method of communication [3, 4].

Once such method of interaction that has received considerable attention in recent years is 3D spatial interaction [5], where users' motions are tracked in some way so as to determine their 3D pose (e.g., position and orientation) in space over time. This tracking can be done with sensors users wear or hold in their hands or unobtrusively with a camera. With this information, users can be immersed in 3D virtual environments and avateer virtual characters in video games and simulations and provide commands to various computer applications. Tracked users can also use these handheld devices or their hands, fingers, and whole bodies to generate specific patterns over time that the computer can recognize to let users issue commands and perform activities. These specific recognized patterns we refer to as 3D gestures.

1.1.3D Gestures. What exactly is a gesture? Put simply, gestures are movements with an intended emphasis and they are often characterized as rather short bursts of activity with an underlying meaning. In more technical terms, a gesture is a pattern that can be extracted from an input data stream. The frequency and size of the data stream are often dependent on the underlying technology used to collect the data and on the intended gesture style and type. For example, x, y coordinates and timing information are often all that is required to support and recognize 2D pen or touch gestures. A thorough survey on 2D gestures can be found in Zhai et al. [6].

Based on this definition, a 3D gesture is a specific pattern that can be extracted from a continuous data stream that contains 3D position, 3D orientation, and/or 3D motion information. In other words, a 3D gesture is a pattern that can be identified in space, whether it be a device moving in the air such as a mobile phone or game controller, or a user's hand or whole body. There are three different types of movements that can fit into the general category of 3D gestures. First, data that represents a static movement, like making and holding a fist or crossing and holding the arms

together, is known as a posture. The key to a posture is that the user is moving to get into a stationary position and then holds that position for some length of time. Second, data that represents a dynamic movement with limited duration, like waving or drawing a circle in the air, is considered to be what we think of as a gesture. Previous surveys [7, 8] have distinguished postures and gestures as separate entities, but they are often used in the same way and the techniques for recognizing them are similar. Third, data that represents dynamic movement with an unlimited duration, like running in place or pretending to climb a rope, is known as an activity. In many cases these types of motions are repetitive, especially in the entertainment domain [9]. The research area known as activity recognition, a subset of computer vision, focuses on recognizing these types of motions [10, 11]. One of the main differences between 3D gestural interfaces and activity recognition is that activity recognition is often focused on detecting human activities where the human is not intending to perform the actions as part of a computer interface, for example, detecting unruly behavior at an airport or train station. For the purposes of this paper, unless otherwise stated, we will group all three movement types into the general category of 3D gestures.

1.2. 3D Gesture Interface Challenges. One of the unique aspects of 3D gestural interfaces is that it crosses many different disciplines in computer science and engineering. Since recognizing a 3D gesture is a question of identifying a pattern in a continuous stream of data, concepts from time series, signal processing and analysis, and control theory can be used. Concepts from machine learning are commonly used since one of the main ideas behind machine learning is to be able to classify data into specific classes and categories, something that is paramount in 3D gesture recognition. In many cases, cameras are used to monitor a user's actions, making computer vision an area that has extensively explored 3D gesture recognition. Given that recognizing 3D gestures is an important component of a 3D gestural user interface, human computer interaction, virtual and augmented reality, and interactive computer graphics all play a role in understanding how to use 3D gestures. Finally, sensor hardware designers also work with 3D gestures because they build the input devices that perform the data collection needed to recognize them.

Regardless of the discipline, from a research perspective, creating and using a 3D gestural interface require the following:

(i) monitoring a continuous input stream to gather data for training and classification,

(ii) analyzing the data to detect a specific pattern from a set of possible patterns,

(iii) evaluating the 3D gesture recognizer,

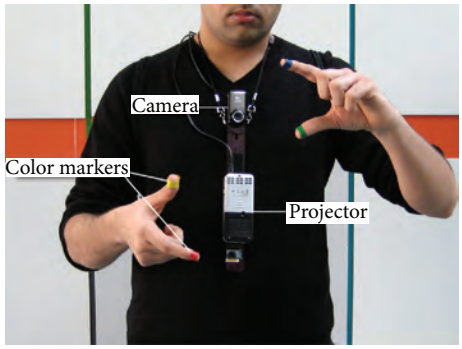
(iv) using the recognizer in an application so commands or operations are performed when specific patterns are detected.

Each one of these components has research challenges that must be solved in order to provide robust, accurate, and intuitive 3D gestural user interaction. For example, devices that collect and monitor input data need to be accurate with high sampling rates, as unobtrusive as possible, and capture as much of the user's body as possible without occlusion. The algorithms that are used to recognize 3D gestures need to be highly accurate, able to handle large gesture sets, and run in real time. Evaluating 3D gesture recognizers is also challenging given that their true accuracies are often masked by the constrained experiments that are used to test them. Evaluating these recognizers in situ is much more difficult because the experimenter cannot know what gestures the user will be performing at any given time. Finally, incorporating 3D gestures recognizers as part of a 3D gestural interface in an application requires gestures that are easy to remember and perform with minimal latency to provide an intuitive and engaging user experience. We will explore these challenges throughout this paper by examining the latest research results in the area.

1.3. Paper Organization. The remainder of this paper is organized in the following manner. In the next section, we will discuss various strategies for collecting 3D gesture data with a focus on the latest research developments in both worn and handheld sensors as well as unobtrusive vision-based sensors. In Section 3, we will explore how to recognize 3D gestures by using heuristic-based methods and machine learning algorithms. Section 4 will present the latest results from experiments conducted to examine recognition accuracy and gesture set size as well as discuss some applications that use 3D gestural interfaces. Section 5 presents some areas for future research that will enable 3D gestural interfaces to become more commonplace. Finally, Section 6 concludes the paper.

2. 3D Gesture Data Collection

Before any 3D gestural interface can be built or any 3D gesture recognizers can be designed, a method is required to collect the data that will be needed for training and classification. Training data is often needed (for heuristic recognition, training data is not required) for the machine learning algorithms that are used to classify one gesture from another. Since we are interested in 3D gestural interaction, information about the user's location in space or how the user moves in space is critical. Depending on what 3D gestures are required in a given interface, the type of device needed to monitor the user will vary. When thinking about what types of 3D gestures users perform, it is often useful to categorize them into hand gestures, full body gestures, or finger gestures. This categorization can help to narrow down the choice of sensing device, since some devices do not handle all types of 3D gestures. Sensing devices can be broken down into active sensors and passive sensors. Active sensors require users to hold a device or devices in their hands or wear the device in some way. Passive sensors are completely unobtrusive and mostly include pure vision sensing. Unfortunately, there is no perfect solution and there are strengths and weaknesses with each technology [12].

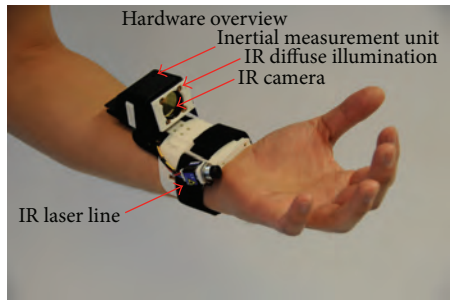


Figur e 1: The SixSense system. A user wears colored fiducial markers for fingertip tracking [14].

2.1. Active Sensors. Active sensors use a variety of different technologies to support the collection and monitoring of 3D gestural data. In many cases, hybrid solutions are used (e.g., combining computer vision with accelerometers and gyroscopes) that combine more than one technology together in an attempt to provide a more robust solution.

2.1.1.Active Finger Tracking. To use the fingers as part of a 3D gestural interface, we need to track their movements and how the various digits move in relation to each other. The most common approach and the one that has the longest history uses some type of instrumented glove that can determine how the fingers bend. Accurate hand models can be created using these gloves and the data used to feed a 3D gesture recognizer. These gloves often do not provide where the hand is in 3D space or its orientation so other tracking systems are needed to complement them. A variety of different technologies are used to perform finger tracking including piezoresistive, fiber optic, and hall-effect sensors. These gloves also vary in the number of sensors they have which determines how detailed the tracking of the fingers can be. In some cases, a glove is worn without any instrumentation at all and used as part of a computer vision-based approach. Dipietro et al. [13] present a thorough survey on data gloves and their applications.

One of the more recent approaches to finger tracking for 3D gestural interfaces is to remove the need to wear an instrumented glove in favor of wearing a vision-based sensor that uses computer vision algorithms to detect the motion of the fingers. One example of such a device is the SixSense system [14]. The SixSense device is worn like a necklace and contains a camera, mirror, and projector. The user also needs to wear colored fiducial markers on the fingertips (see Figure 1). Another approach developed by Kim et al. uses a wrist worn sensing device called Digits [15]. With this system, a wrist worn camera (see Figure 2) is used to optically image the entirety of a user's hand which enables the sampling of fingers. Combined with a kinematic model, Digits can reconstruct the hand and fingers to support 3D gestural interfaces in mobile environments. Similar systems that make use of worn cameras or proximity sensors to track the fingers for 3D gestural interfaces have also been explored [16–19].



Figur e 2: Digits hardware. A wrist worn camera that can optically image a user's hand to support hand and finger tracking [15].

Precise finger tracking is not always a necessity in 3D gestural interfaces. It depends on how sophisticated the 3D gestures need to be. In some cases, the data needs only to provide distinguishing information to support different, simpler gestures. This idea has led to utilizing different sensing systems to support course finger tracking. For example, Saponas et al. have experimented with using forearm electromyography to differentiate fingers presses and finger tapping and lifting [20]. A device that contains EMG sensors is attached to a user's wrist and collects muscle data about fingertip movement and can then detect a variety of different finger gestures [21, 22]. A similar technology supports finger tapping that utilizes the body for acoustic transmission. Skinput, developed by Harrison et al. [23], uses a set of sensors worn as an armband to detect acoustical signals transmitted through the skin [18].

2.1.2. Active Hand Tracking. In some cases, simply knowing the position and orientation of the hand is all the data that is required for a 3D gestural interface. Thus, knowing about the fingers provides too much information and the tracking requirements are simplified. Of course, since the fingers are attached to the hand, many finger tracking algorithms will also be able to track the hand. Thus there is often a close relationship between hand and finger tracking. There are two main flavors of hand tracking in active sensing: the first is to attach a sensing device to the hand and the second is to hold the device in the hand.

Attaching a sensing device to the user's hand or hands is a common approach to hand tracking that has been used for many years [5]. There are several tracking technologies that support the attachment of an input device to the user's hand including electromagnetic, inertial/acoustic, ultrasonic, and others [12]. These devices are often placed on the back of the user's hand and provide single point pose information through time. Other approaches include computer vision techniques where users wear a glove. For example, Wang and Popović [24] designed a colored glove with a known pattern to support a nearest-neighbor approach to tracking hands at interactive rates. Other examples include wearing retroreflective fiducial markers coupled with cameras to track a user's hand.

The second approach to active sensor-based hand tracking is to have a user hold the device. This approach has both strengths and weaknesses. The major weakness is that the

users have to hold something in their hands which can be problematic if they need to do something else with their hands during user interaction. The major strengths are that the devices users hold often have other functionalities such as buttons, dials, or other device tools which can be used in addition to simply tracking the user's hands. This benefit will become clearer when we discuss 3D gesture recognition and the segmentation problem in Section 3. There have been a variety of different handheld tracking devices that have been used in the virtual reality and 3D user interface communities [25–27].

Recently, the game industry has developed several video game motion controllers that can be used for hand tracking. These devices include the Nintendo Wii Remote (Wiimote), Playstation Move, and Razer Hydra. They are inexpensive and massproduced. Both the Wiimote and the Playstation Move use both vision and inertial sensing technology while the Hydra uses a miniaturized electromagnetic tracking system. The Hydra [28] and the Playstation Move [29] both provide position and orientation information (6 DOF) while the Wiimote is more complicated because it provides certain types of data depending on how it is held [30]. However, all three can be used to support 3D gestural user interfaces.

2.1.3. Active Full Body Tracking. Active sensing approaches to tracking a user's full body can provide accurate data used in 3D gestural interfaces but can significantly hinder the user since there are many more sensors the user needs to wear compared with simple hand or finger tracking. In most cases, a user wears a body suit that contains the sensors needed to track the various parts of the body. This body suit may contain several electromagnetic trackers, for example, or a set of retroreflective fiducial markers that can be tracked using several strategically placed cameras. These systems are often used for motion capture for video games and movies but can also be used for 3D gestures. In either case, wearing the suit is not ideal in everyday situations given the amount of time required to put it on and take it off and given other less obtrusive solutions.

A more recent approach for supporting 3D gestural interfaces using the full body is to treat the body as an antenna. Cohn et al. first explored this idea for touch gestures [31] and then found that it could be used to detect 3D full body gestures [32, 33]. Using the body as an antenna does not support exact and precise tracking of full body poses but provides enough information to determine how the body is moving in space. Using a simple device either in a backpack or worn on the body, as long as it makes contact with the skin, this approach picks up how the body affects the electromagnetic noise signals present in an indoor environment stemming from power lines, appliances, and devices. This approach shows great promise for 3D full body gesture recognition because it does not require any cameras to be strategically placed in the environment, making the solution more portable.

2.2. Passive Sensors. In contrast to active sensing, where the user needs to wear a device or other markers, passive sensing makes use of computer vision and other technologies (e.g., light and sound) to provide unobtrusive tracking of the hands, fingers, and full body. In terms of computer vision, 3D gestural interfaces have been constructed using traditional cameras [34–37] (such as a single webcam) as well as depth cameras. The more recent approaches to recognizing 3D gestures make use of depth cameras because they provide more information than a traditional single camera in that they support extraction of a 3D representation of a user, which then enables skeleton tracking of the hands, fingers, and whole body.

There are generally three different technologies used in depth cameras, namely, time of flight, structured light, and stereo vision [38]. Time-of-flight depth cameras (e.g., the depth camera used in the XBox One) determine the depth map of a scene by illuminating it with a beam of pulsed light and calculating the time it takes for the light to be detected on an imaging device after it is reflected off of the scene. Structured-light depth cameras (e.g., Microsoft Kinect) use a known pattern of light, often infrared, that is projected into the scene. An image sensor then is able to capture this deformed light pattern based on the shapes in the scene and finally extracts 3D geometric shapes using the distortion of the projected optical pattern. Finally. stereo based cameras attempt to mimic the human-visual system using two calibrated imaging devices laterally displaced from each. These two cameras capture synchronized images of the scene, and the depth for image pixels is extracted from the binocular disparity. The first two depth camera technologies are becoming more commonplace given their power in extracting 3D depth and low cost.

These different depth camera approaches have been used in a variety of ways to track fingers, hands, and the whole body. For example, Wang et al. used two Sony Eye cameras to detect both the hands and fingers to support a 3D gestural interface for computer aided design [39] while Hackenberg et al. used a time-of-flight camera to support hand and finger tracking for scaling, rotation, and translation tasks [40]. Keskin et al. used structured light-based depth sensing to also track hand and finger poses in real time [41]. Other recent works using depth cameras for hand and finger tracking for 3D gestural interfaces can be found in [42–44]. Similarly, these cameras have also been used to perform whole body tracking that can be used in 3D full body-based gestural interfaces. Most notably is Shotton et al.'s seminal work on using a structured light-based depth camera (i.e., Microsoft Kinect) to track a user's whole body in real time [45]. Other recent approaches that make use of depth cameras to track the whole body can be found in [46–48].

More recent approaches to passive sensing used in 3D gesture recognition are through acoustic and light sensing. In the SoundWave system, a standard speaker and microphone found in most commodity laptops and devices is used to sense user motion [49]. An inaudible tone is sent through the speaker and gets frequency-shifted when it reflects off moving objects like a user's hand. This frequency shift is measured by the microphone to infer various gestures. In the LightWave system, ordinary compact fluorescent light (CFL) bulbs are used as sensors of human proximity [50]. These CFL bulbs

are sensitive proximity transducers when illuminated and the approach can detect variations in electromagnetic noise resulting from the distance from the human to the bulb. Since this electromagnetic noise can be sensed from any point in an electrical wiring system, gestures can be sensed using a simple device plugged into any electrical outlet. Both of these sensing strategies are in their early stages and currently do not support recognizing a large quantity of 3D gestures at any time, but their unobtrusiveness and mobility make them a potential powerful approach to body sensing for 3D gestural user interfaces.

3. 3D Gesture Recognition and Analysis

3D gestural interfaces require the computer to understand the finger, hand, or body movements of users to determine what specific gestures are performed and how they can then be translated into actions as part of the interface. The previous section examined the various strategies for continuously gathering the data needed to recognize 3D gestures. Once we have the ability to gather this data, it must be examined in real time using an algorithm that analyzes the data and determines when a gesture has occurred and what class that gesture belongs to. The focus of this section is to examine some of the most recent techniques for real-time recognition of 3D gestures. Several databases such as the ACM and IEEE Digital Libraries as well as Google Scholar were used to survey these techniques and the majority of those chosen reflect the state of the art. In addition, when possible, techniques that were chosen also had experimental evaluations associated with them. Note that other surveys that have explored earlier work on 3D gesture recognition also provide useful examinations of existing techniques [8, 51–53].

Recognizing 3D gestures is dependent on whether the recognizer first needs to determine if a gesture is present. In cases where there is a continuous stream of data and the users do not indicate that they are performing a gesture (e.g., using a passive vision-based sensor), the recognizer needs to determine when a gesture is performed. This process is known as gesture segmentation. If the user can specify when a gesture begins and ends (e.g., pressing a button on a Sony Move or Nintendo Wii controller), then the data is presegmented and gesture classification is all that is required. Thus, the process of 3D gesture recognition is made easier if a user is holding a tracked device, such as a game controller, but it is more obtrusive and does not support more natural interaction where the human body is the only "device" used. We will examine recognition strategies that do and do not make use of segmentation.

There are, in general, two different approaches to recognizing 3D gestures. The first, and most common, is to make use of the variety of different machine learning techniques in order to classify a given 3D gesture as one of a set of possible gestures [54, 55]. Typically, this approach requires extracting important features from the data and using those features as input to a classification algorithm. Additionally, varying amounts of training data are needed to seed and tune the classifier to make it robust to variability and to maximize accuracy. The second approach, which is somewhat under-utilized, is to use heuristics-based recognition. With heuristic recognizers, no formal machine learning algorithms are used, but features are still extracted and rules are procedurally coded and tuned to recognize the gestures. This approach often makes sense when a small number of gestures are needed (e.g., typically 5 to 7) for a 3D gestural user interface.

3.1. Machine Learning. Using machine learning algorithms as classifiers for 3D gesture recognition represents the most common approach to developing 3D gesture recognition systems. The typical procedure for using a machine learning-based approach is to

(i) pick a particular machine learning algorithm,

(ii) come up with a set of useful features that help to quantify the different gestures in the gesture set,

(iii) use these features as input to the machine learning algorithm,

(iv) collect training and test data by obtaining many samples from a variety of different users,

(v) train the algorithm on the training data,

(vi) test the 3D gesture recognizer with the test data,

(vii) refine the recognizer with different/additional feature or with more training data if needed.

There are many different questions that need to be answered when choosing a machine learning-based approach to 3D gesture recognition. Two of the most important are what machine learning algorithm should be used and how accurate can the recognizer be. We will examine the former question by presenting some of the more recent machine learning-based strategies and discuss the latter question in Section 4.

3.1.1. Hidden Markov Models. Although Hidden Markov Models (HMMs) should not be considered recent technology, they are still a common approach to 3D gesture recognition. HMMs are ideally suited for 3D gesture recognition when the data needs to be segmented because they encode temporal information so a gesture can first be identified before it is recognized [37]. More formally, an HMM is a double stochastic process that has an underlying Markov chain with a finite number of states and a set of random functions, each associated with one state [56]. HMMs have been used in a variety of different ways with a variety of different sensor technologies. For example, Sako and Kitamura used multistream HMMs for recognizing Japanese sign language [57]. Pang and Ding used traditional HMMs for recognizing dynamic hand gesture movements using kinematic features such as divergence, vorticity, and motion direction from optical flow [58]. They also make use of principal component analysis (PCA) to help with feature dimensionality reduction. Bevilacqua et al. developed a 3D gesture recognizer that combines HMMs with stored reference gestures which helps to reduce the training amount required [59]. The method used only one single example for each gesture and the

recognizer was targeted toward music and dance performances. Wan et al. explored better methods to generate efficient observations after feature extraction for HMMs [60]. Sparse coding is used for finding succinct representations of information in comparison to vector quantization for hand gesture recognition. Lee and Cho used hierarchical HMMs to recognize actions using 3D accelerometer data from a smart phone [61]. This hierarchical approach, which breaks up the recognition process into actions and activities, helps to overcome the memory storage and computational power concerns of mobile devices. Other work on 3D gesture recognizers that incorporate HMMs include [62–69].

3.1.2. Conditional Random Fields. Conditional random fields (CRFs) are considered to be a generalization of HMMs and have seen a lot of use in 3D gesture recognition. Like HMMs they are a probabilistic framework for classifying and segmenting sequential data, however, they make use of conditional probabilities which relax any independence assumptions and also avoid the labeling bias problem [70]. As with HMMs, there have been a variety of different recognition methods that use and extend CRFs. For example, Chung and Yang used depth sensor information as input to a CRF with an adaptive threshold for distinguishing between gestures that are in the gesture set and those that are outside the gestures set [71]. This approach, known as T-CRF, was also used for sign language spotting [72]. Yang and Lee also combined a T-CRF and a conventional CRF in a two-layer hierarchical model for recognition of signs and finger spelling [73]. Other 3D gesture recognizers that make use of CRFs include [39, 74, 75].

Hidden conditional random fields (HCRFs) extend the concept of the CRF by adding hidden state variables into the probabilistic model which is used to capture complex dependencies in the observations while still not requiring any independence assumptions and without having to exactly specify dependencies [76]. In other words, HCRFs enable sharing of information between labels with the hidden variables but cannot model dynamics between them. HCRFs have also been utilized in 3D gesture recognition. For example, Sy et al. were one of the first groups to use HCRFs in both arm and head gesture recognition [77]. Song et al. used HCRFs coupled with temporal smoothing for recognizing body and hand gestures for aircraft signal handling [78]. Liu et al. used HCRFs for detecting hand poses in a continuous stream of data for issuing commands to robots [79]. Other works that incorporate HCRFs in 3D gesture recognizers include [80, 81].

Another variant to CRFs is the latent-dynamic hidden CRF (LDCRF). This approach builds upon the HCRF by providing the ability to model the substructure of a gesture label and learn the dynamics between labels, which helps in recognizing gestures from unsegmented data [82]. As with CRFs and HCRFs, LDCRFs have been examined for use as part of 3D gesture recognition systems and received considerable attention. For example, Elmezain and Al-Hamadi use LDCRFs for recognizing hand gestures in American sign language using a stereo camera [83]. Song et al. improved upon their prior HCRF-based approach [78] to recognizing both hand and body gestures by incorporating the LDCRF [84]. Zhang et al. also used LDCRFs for hand gesture recognition but chose to use fuzzy-based latent variables to model hand gesture features with a modification to the LDCRF potential functions [85]. Elmezain et al. also used LDCRFs in hand gesture recognition to specifically explore how they compare with CRFs and HCRFs. They examined different window sizes and used location, orientation, and velocity features as input to the recognizers, with LDCRFs performing the best in terms of recognition accuracy [86].

3.1.3. Support Vector Machines. Support vector machines (SVMs) are another approach that is used in 3D gesture recognition that has received considerable attention in recent years. SVMs are a supervised learning-based probabilistic classification approach that constructs a hyperplane or set of hyperplanes in high dimensional space used to maximize the distance to the nearest training data point in a given class [87]. These hyperplanes are then used for classification of unseen instances. The mappings used by SVMs are designed in terms of a kernel function selected for a particular problem type. Since not all the training data may be linearly separable in a given space, the data can be transformed via nonlinear kernel functions to work with more complex problem domains.

In terms of 3D gestures, there have been many recognition systems that make use of SVMs. For example, recent work has explored different ways of extracting the features used in SVM-based recognition. Huang et al. used SVMs for hand gesture recognition coupled with Gabor filters and PCA for feature extraction [88]. Hsieh et al. took a similar approach for hand gesture recognition but used the discrete Fourier transform (DFT) coupled with the Camshift algorithm and boundary detection to extract the features used as input to the SVM [89]. Hsieh and Liou not only used Haar features for their SVM-based recognizer but also examined the color of the user's face to assist in detecting and extracting the users' hands [90]. Dardas et al. created an SVM-based hand gesture detection and recognition system by using the scale invariance feature transform (SIFT) and vector quantization to create a unified dimensional histogram vector (e.g., bag of words) with K-means clustering. This vector was used as the input to a multiclass SVM [91, 92].

Other ways in which SVMs have been used for 3D gesture recognition have focused on fusing more than one SVM together or using the SVM as part of a larger classification scheme. For example, Chen and Tseng used 3 SVMs from 3 different camera angles to recognize 3D hand gestures by fusing the results from each with majority voting or using recognition performance from each SVM as a weight to the overall gesture classification score [93]. Rashid et al. combined an SVM and HMM together for American sign language where the HMM was used for gestures while the SVM was used for postures. The results from these two classifiers were then combined to provide a more general recognition framework [94]. Song et al. used an SVM for hand shape classification that was combined with a particle filtering estimation framework for 3D body postures and

an LDCRF for overall recognition [84]. Other 3D gesture recognizers that utilize SVMs include [80, 95–101].

3.1.4. Decision Trees and Forests. Decision trees and forests are an important machine learning tool for recognizing 3D gestures. With decision trees, each node of the tree makes a decision about some gesture feature. The path traversed from the root to a leaf in a decision tree specifies the expected classification by making a series of decisions on a number of attributes. There are a variety of different decision tree implementations [102]. One of the most common is the C4.5 algorithm [103] which uses the notion of entropy to identify ranking of features to determine which feature is most informative for classification. This strategy is used in the construction of the decision tree. In the context of 3D gesture recognition, there have been several different strategies explored using decision trees. For example, Nisar et al. used standard image-based features such as area, centroid, and convex hull among others as input to a decision tree for sign language recognition [104]. Jeon et al. used decision trees for recognizing hand gestures for controlling home appliances. They added a fuzzy element to their approach, developing a multivariate decision tree learning and classification algorithm. This approach uses fuzzy membership functions to calculate the information gain in the tree [105]. Zhang et al. combined decision trees with multistream HMMs for Chinese sign language recognition. They used a 3-axis accelerometer and electromyography (EMG) sensors as input to the recognizer [106]. Other examples of using decision trees in 3D gesture recognition include [107, 108].

Decision forests are an extension of the decision tree concept. The main difference is that instead of just one tree used in the recognition process, there is an ensemble of randomly trained decision trees that output the class that is the mode of the classes output by the individual trees [115]. Given the power of GPUs, decision forests are becoming prominent for real-time gesture recognition because the recognition algorithm can be easily parallelized with potentially thousands of trees included in the decision forest [116]. This decision forest approach can be considered a framework that has several different parts that can produce a variety of different models. The shape of the decision to use for each node, the type of predictor used in each leaf, the splitting objective used to optimize each node, and the method for injecting randomness into the trees are all choices that need to be made when constructing a decision forest used in recognition. One of the most notable examples of the use of decision forests was Shotton et al.'s work on skeleton tracking for the Microsoft Kinect [45]. This work led researchers to look at decision forests for 3D gesture recognition. For example, Miranda et al. used decision forests for full body gesture recognition using the skeleton information from the Microsoft Kinect depth camera. Key poses from the skeleton data are extracted using a multiclass SVM and fed as input to the decision forest. Keskin et al. used a depth camera to recognize hand poses using decision forests [41]. A realistic 3D hand model with 21 different parts was used to create synthetic depth images for decision forest training. In another example, Negin et al. used decision forests on kinematic time series for determining the best set of features to use from a depth camera [111]. These features are then fed into a SVM for gesture recognition. Other work that has explored the use of decision forests for 3D gesture recognition include [110, 117, 118].

3.1.5. Other Learning-Based Techniques. There are, of course, a variety of other machine learning-based techniques that have been used for 3D gesture recognition, examples include neural networks [119, 120], template matching [121, 122], finite state machines [121, 123], and using the Adaboost framework [112]. To cover all of them in detail would go beyond the scope of this paper. However, two other 3D gesture recognition algorithms are worth mentioning because they both stem from recognizers used in 2D pen gesture recognition, are fairly easy to implement, and provide good results. These recognizers tend to work for segmented data but can be extended to unsegmented data streams by integrating circular buffers with varying window sizes, depending on the types of 3D gestures in the gesture set and the data collection system. This first one is based on Rubine's linear classifier [124], first published in 1991. This classifier is a linear discriminator where each gesture has an associated linear evaluation function, and each feature has a weight based on the training data. The classifier uses a closed form solution for training which produces optimal classifiers given that the features are normally distributed. However, the approach still produces good results even when there is a drift from normality. This approach also always produces a classification so the false positive rate can be high. However a good rejection rule will remove ambiguous gestures and outliers. The extension of this approach to 3D gestures is relatively straightforward. The features need to be extended to capture 3D information with the main classifier and training algorithm remaining the same. This approach has been used successfully in developing simple, yet effective 3D gesture recognizers [112, 125, 126].

The second approach is based on Wobbrock et al.'s $1 2D recognizer [127]. Kratz and Rohs used the $1 recognizer as a foundation for the $3 recognizer, designed primarily for 3D gestures on mobile devices [113, 128]. In this approach, gesture traces are created using the differences between the current and previous acceleration data values and resampled to have the same number of points as any gesture template. These resampled traces are then corrected for rotational error using the angle between the gesture's first point and its centroid. Average mean square error is then used to determine the given gesture trace's distance to each template in the gesture class library. A heuristic scoring mechanism is used to help reject false positives. Note that a similar approach to constructing a 3D gesture recognizer was done by Li, who adapted the Protractor 2D gesture recognizer [129] and extended it to work with accelerometers and gyroscope data [114, 130].

3.2. Heuristic Recognizers. Heuristic 3D gesture recognizers make sense when there are a small number of easily identifiable gestures in an interface. The advantage of heuristic-based

approaches is that no training data is needed and they are fairly easy to implement. For example, Williamson et al. [131] developed a heuristic recognition method using skeleton data from a Microsoft Kinect focused on jumping, crouching, and turning. An example of a heuristic recognizer for jumping would be to assume a jump was made when the head is at a certain height above its normal position, defined as

$$J = H_y - \overline{H}_y > C, \tag{1}$$

where J is true or false based on if a jump has occurred, H_y is the height of the head position, $\overline{H}_y$ is the calibrated normal height of the head position with the user standing, and C is some constant. C would then be set to a height that a person would only get to by jumping from the ground. Such recognition is very specialized but simple and explainable and can determine in an instant whether a jump has occurred.

Recent work has shown that heuristic 3D recognition works well with devices that primarily make use of accelerometers and/or gyroscopes (e.g., the Nintendo Wiimote, smart phones). For example, One Man Band used a Wiimote to simulate the movements necessary to control the rhythm and pitch of several musical instruments [132]. RealDance explored spatial 3D interaction for dance-based gaming and instruction [133]. By wearing Wiimotes on the wrists and ankles, players followed an on-screen avatar's choreography and had their movements evaluated on the basis of correctness and timing. These explorations led to several heuristic recognition schemes for devices which use accelerometers and gyroscopes.

Poses and Underway Intervals. A pose is a length of time during which the device is not changing position. Poses can be useful for identifying held positions in dance, during games, or possibly even in yoga. An underway interval is a length of time during which the device is moving but not accelerating. Underway intervals can help identify smooth movements and differentiate between, say, strumming on a guitar and beating on a drum.

Because neither poses nor underway intervals have an acceleration component, they cannot be differentiated using accelerometer data alone. To differentiate the two, a gyroscope can provide a frame of reference to identify whether the device has velocity. Alternatively, context can be used, such as tracking acceleration over time to determine whether the device is moving or stopped.

Poses and underway intervals have three components. First, the time span is the duration in which the user maintains a pose or an underway interval. Second, the orientation of gravity from the acceleration vector helps verify that the user is holding the device at the intended orientation. Of course, unless a gyroscope is used, the device's yaw cannot be reliably detected. Third, the allowed variance is the threshold value for the amount of acceleration allowed in the heuristic before rejecting the pose or underway interval. For example, in RealDance [133], poses were important for recognizing certain dance movements. For a pose, the user was supposed to stand still in a specific posture beginning at time t_0 and lasting until $t_0 + N$, where N is a specified number of beats. A player's score could be represented as the percentage of the time interval during which the user successfully maintained the correct posture.

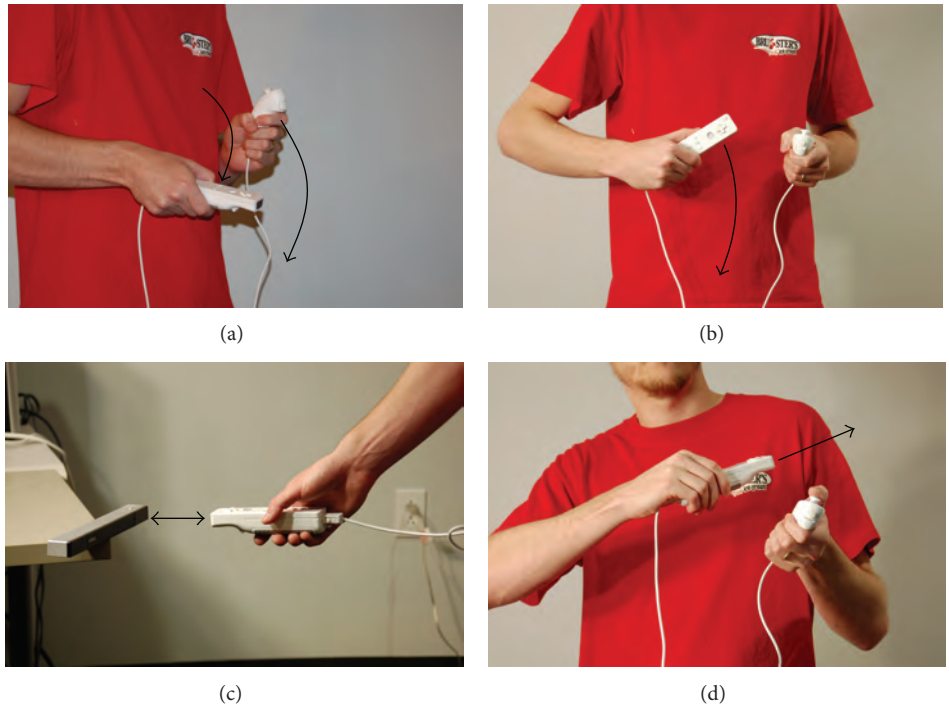
Impulse Motions. An impulse motion is characterized by a rapid change in acceleration, easily measured by an accelerometer. A good example is a tennis or golf club swing in which the device motion accelerates through an arc or a punching motion, which contains a unidirectional acceleration. An impulse motion has two components, which designers can tune for their use. First, the time span of the impulse motion specifies the window over which the impulse is occurring. Shorter time spans increase the interaction speed, but larger time spans are more easily separable from background jitter. The second component is the maximum magnitude reached. This is the acceleration bound that must be reached during the time span in order for the device to recognize the impulse motion.

Impulse motions can also be characterized by their direction. The acceleration into a punch is essentially a straight impulse motion, a tennis swing has an angular acceleration component, and a golf swing has both angular acceleration and even increasing acceleration during the follow-through when the elbow bends. All three of these impulse motions, however, are indistinguishable to an acceleration only device, which does not easily sense these orientation changes. For example, the punch has an acceleration vector along a single axis, as does the tennis swing as it roughly changes its orientation as the swing progresses. These motions can be differentiated by using a gyroscope as part of the device or by assuming that orientation does not change. As an example, RealDance used impulse motions to identify punches. A punch was characterized by a rapid deceleration occurring when the arm was fully extended. In a rhythm-based game environment, this instant should line up with a strong beat in the music. An impulse motion was scored by considering a one-beat interval centered on the expected beat.

Impact Events. An impact event is an immediate halt to the device due to a collision, characterized by an easily identifiable acceleration bursting across all three dimensions. Examples of this event include the user tapping the device on a table or dropping it so it hits the floor. To identify an impact event, the change in acceleration (jerk) vector is required for each pair of adjacent time samples. Here, t_k corresponds to the largest magnitude of jerk:

$$t_k = \arg\max_T \left\| \vec{a}_t - \vec{a}_{t-1} \right\|, \tag{2}$$

where $\vec{a}$ is the acceleration vector at time t. If the magnitude is larger than a threshold value, an impact occurs. As an example, RealDance used impact motions to identify stomps. If the interval surrounding a dance move had a maximal jerk value less than a threshold, no impact occurred. One Man Band also used impact events to identify when a Nintendo Nunchuk controller and Wiimote collided, which is how users played hand cymbals.



Figur e 3: One Man Band differentiated between multiple Wiimote gestures using mostly simple modal differentiations for (a) drums, (b) guitar, (c) violin, and (d) theremin. To the player, changing instruments only required orienting the Wiimote to match how an instrument would be played.

Modal Differentiation. Herustics can also be used as a form of simple segmentation to support the recognition of different gestures. For example, in One Man Band [132], the multi-instrument musical interface (MIMI) differentiated between five different instruments by implementing modal differences based on a Wiimote's orientation. Figure 3 shows four of these. If the user held the Wiimote on its side and to the left, as if playing a guitar, the application interpreted impulse motions as strumming motions. If the user held the Wiimote to the left, as if playing a violin, the application interpreted the impulse motions as violin sounds. To achieve this, the MIMI's modal-differentiation approach used a normalization step on the accelerometer data to identify the most prominent orientation:

$$\vec{a}_{\text{norm}} = \frac{\vec{a}}{\|\vec{a}\|} \tag{3}$$

followed by two exponential smoothing functions

$$\vec{a}_{\text{current}} = \alpha \vec{a}_i + (1 - \alpha)\, \vec{a}_{i-1}. \tag{4}$$

The first function, with an $\alpha = 0.1$, removed jitter and identified drumming and strumming motions. The second function, with an $\alpha = 0.5$, removed jitter and identified short, sharp gestures such as violin strokes.

4. Experimentation and Accuracy

As we have seen in the last section, there have been a variety of different approaches for building 3D gesture recognition systems for use in 3D gestural interfaces. In this section, we focus on understanding how well these approaches work in terms of recognition accuracy and the number of gestures that can be recognized. These two metrics help to provide researchers and developers guidance on what strategies work best. As with Section 3, we do not aim to be an exhaustive reference on the experiments that have been conducted on 3D gesture recognition accuracy. Rather, we present a representative sample that highlights the effectiveness of different 3D gesture recognition strategies.

A summary of the experiments and accuracy of various 3D gesture recognition systems is shown in Table 1. This table shows the authors of the work, the recognition approach or strategy, the number of recognized gestures, and the highest accuracy level reported. As can be seen in the table, there have been a variety of different methods that have been proposed and most of the results reported are able to achieve over 90% accuracy. However, the number of gestures in the gesture sets used in the experiments vary significantly. The number of gestures in the gesture set is often not indicative of performance when comparing techniques. In some cases, postures were used instead of more complex gestures and in some cases, more complex activities were recognized. For example, Lee and Cho recognized only 3 gestures, but these are classified as activities that included shopping, taking a bus, and moving by walking [61]. The gestures used in these actions are more complex than, for example, finger spelling. In other cases, segmentation was not done as part of the recognition process. For example, Hoffman et al. were able to recognize 25 gestures at 99% accuracy, but the data was

Table 1: A table summarizing different 3D gesture recognition approaches, the size of the gesture set, and the stated recognition accuracy.

Author	Recognition approach	Number of gestures	Accuracy
Pang and Ding [58]	HMMs with kinematic features	12	91.2%
Wan et al. [60]	HMMs with sparse coding	4	94.2%
Lee and Cho [61]	Hierarchical HMMs	3	Approx. 80.0%
Whitehead and Fox [68]	Standard HMMs	7	91.4%
Nguyen et al. [66]	Two-stage HMMs	10	95.3%
Chen et al. [63]	HMMs with Fourier descriptors	20	93.5%
Pylvänäinen [67]	HMMs without rotation data	10	99.76%
Chung and Yang [71]	Threshold CRF	12	91.9%
Yang et al. [72]	Two-layer CRF	48	93.5%
Yang and Lee [73]	HCRF with BoostMap embedding	24	87.3%
Song et al. [78]	HCRF with temporal smoothing	10	93.7%
Liu and Jia [80]	HCRF with manifold learning	10	97.8%
Elmezain and Al-Hamadi [83]	LDCRF with depth camera	36	96.1%
Song et al. [84]	LDCRF with filtering framework	24	75.4%
Zhang et al. [85]	Fuzzy LDCRF	5	91.8%
Huang et al. [88]	SVM with Gabor filters	11	95.2%
Hsieh et al. [89]	SVM with Fourier descriptors	5	93.4%
Hsieh and Liou [90]	SVM with Haar features	4	95.6%
Dardas and Georganas [92]	SVM with bag of words	10	96.2%
Chen and Tseng [93]	Fusing multiple SVMs	3	93.3%
Rashid et al. [94]	Combining SVM with HMM	18	98.0%
Yun and Peng [101]	Hu moments with SVM	3	96.2%
Ren and Zhang [99]	SVM with min enclosing ball	10	92.9%
Wu et al. [100]	Frame-based descriptor with SVM	12	95.2%
He et al. [96]	SVM with Wavelet and FFT	17	87.4%
Nisar et al. [104]	Decision trees	26	95.0%
Jeon et al. [105]	Multivariate fuzzy decision trees	10	90.6%
Zhang et al. [106]	Decision trees fused with HMMs	72	96.3%
Fang et al. [107]	Hierarchical Decision trees	14	91.6%
Miranda et al. [109]	Decision forest with key pose learning	10	91.5%
Keskin et al. [41]	Decision forest with SVM	10	99.9%
Keskin et al. [110]	Shape classification forest	24	97.8%
Negin et al. [111]	Feature selection with decision forest	10	98.0%
Ellis et al. [74]	Logistic regression	16	95.9%
Hoffman et al. [112]	Linear classifier	25	99.0%
Kratz and Rohs [113]	$3 gesture recognizer	10	80.0%
Kratz and Rohs [114]	Protractor 3D (rotation invariance)	11	Approx. 91.0%

presegmented using button presses to indicate the start and stop of a gesture [112].

It is often difficult to compare 3D gesture recognition techniques for a variety of reasons including the use of different data sets, parameters, and number of gestures. However, there have been several, more inclusive experiments that have focused on examining several different recognizers in one piece of research. For example, Kelly et al. compared their gesture threshold HMM with HMMs, transition HMMs, CRFs, HCRFs, and LDCRFs [64] and found their approach to be superior, achieving over 97% accuracy on 8 dynamic sign language gestures. Wu et al. compared their frame-based descriptor and multiclass SVM to dynamic time warping, a naive Bayes classifier, C4.5 decision trees, and HMMs and showed their approach has better performance compared to the other methods for both user dependent (95.2%) and user independent cases (89.3%) for 12 gestures [100]. Lech et al. compared a variety of different recognition systems for building a sound mixing gestural interface [118]. They compared the nearest neighbor algorithm with nested generalization, naive Bayes, C4.5 decision trees, random trees, decision forests, neural networks, and SVMs on a set of four gestures and found the SVM to be the best approach for their application. Finally, Cheema et al. compared a linear classifier, decision trees, Bayesian networks, SVM, and AdaBoost using decision trees as weak learners on a gesture set containing 25

Figur e 4: A user performing a gesture in a video game application [9].

gestures [125]. They found that the linear classifier performed the best under different conditions which is interesting given its simplicity compared to the other 3D gesture recognition methods. However, SVM and AdaBoost also performed well under certain user independent recognition conditions when using more training samples per gesture.

Experiments on 3D gesture recognition systems have also been carried out in terms of how they can be used as 3D gestural user interfaces and there have been a variety of different application domains explored [134]. Entertainment and video games are just one example of an application domain where 3D gestural interfaces are becoming more common. This trend is evident since all major video game consoles and the PC support devices that capture 3D motion from a user. In other cases, video games are being used as the research platform for 3D gesture recognition. Figure 4 shows an example of using a video game to explore what the best gesture set should be for a first person navigation game [9], while Figure 5 shows screenshots of the video game used in Cheema et al.'s 3D gesture recognition study [125]. Other 3D gesture recognition research that has focused on the entertainment and video game domain include [132, 135–137].

Medical applications and use in operating rooms are an area where 3D gestures have been explored. Using passive sensing enables the surgeon or doctor to use gestures to gather information about a patient on a computer while still maintaining a sterile environment [138, 139]. 3D gesture recognition has also been explored with robotic applications in the human robot interaction field. For example, Pfeil et al. (shown in Figure 6) used 3D gestures to control unmanned aerial vehicles (UAVs) [140]. They developed and evaluated several 3D gestural metaphors for teleoperating the robot. Other examples of 3D gesture recognition technology used in human robot interaction applications include [141–143]. Other application areas include training and interfacing with vehicles. Williamson et al. developed a full body gestural interface for dismounted soldier training [29] while Riener explored how 3D gestures could be used to control various components of automotive vehicles [144]. Finally, 3D gesture recognition has recently been explored in consumer electronics, specifically for control of large screen smart TVs [145, 146].

5. Future Research Trends

Although there have been great strides in 3D gestural user interfaces from unobtrusive sensing technologies to advanced machine learning algorithms that are capable of robustly recognizing large gesture sets, there still remains a significant amount of future research that needs to be done to make 3D gestural interaction truly robust, provide compelling user experiences, and support interfaces that are natural and seamless to users. In this section, we highlight three areas that need to be explored further to significantly advance 3D gestural interaction.

5.1. Customized 3D Gesture Recognition. Although there has been some work on customizable 3D gestural interfaces [147], customization is still an open problem. Customization can take many forms and in this case, we mean the ability for users to determine the best gestures for themselves for a particular application. Users should be able to define the 3D gestures they want to perform for a given task in an application. This type of customization goes one step further than having user-dependent 3D gesture recognizers (although this is still a challenging problem in cases where many people are using the interface).

There are several problems that need to be addressed to support customized 3D gestural interaction. First, how do users specify what gestures they want to perform for a given task. Second, once these gestures are specified, if using machine learning, how do we get enough data to train the classification algorithms without burdening the user? Ideally, the user should only need to specify a gesture just once. This means that synthetic data needs to be generated based on user profiles or more sophisticated learning algorithms that deal with small training set sized are required. Third, how do we deal with user defined gestures that are very similar to each other? This problem occurs frequently in all kinds of gestures recognition, but the difference in this case is that the users are specifying the 3D gesture and we want them to use whatever gesture they come up with. These are all problems that need to be solved in order to support truly customized 3D gestural interaction.

5.2. Latency. 3D gesture recognition needs to be both fast and accurate to make 3D gestural user interfaces usable and compelling. In fact, the recognition component needs to be somewhat faster than real time because responses based on 3D gestures need to occur at the moment a user finishes a gesture. Thus, the gesture needs to be recognized a little bit before the user finishes it. This speed requirement makes latency an important problem that needs to be addressed to ensure fluid and natural user experiences. In addition, as sensors get better at acquiring a user's position, orientation, and motion in space, the amount of data that must be processed will increase making the latency issue a continuing problem.

Latency can be broken up into computational latency and observational latency [74, 148]. Computational latency is the delay that is based on the amount of computation needed

(a) (b)

Figure 5: Screenshots of a video game used to explore different 3D gesture recognition algorithms [125].

Figure 6: A user controlling a UAV using a 3D gesture [140].

to recognize 3D gestures. Observational latency is the delay based on the minimum amount of data that needs to be observed to recognize a 3D gesture. Both latencies present an important area in terms of how to minimize and mitigate them. Parallel processing can play an important role in reducing computational latency while better understanding the kinematics of the human body is one of many possible ways to assist in reducing observational latency.

5.3. Using Context. Making use of all available information for recognizing 3D gestures in a 3D gestural interface makes intuitive sense because it can assist the recognizer in several ways. First, it can help to reduce the amount of possible 3D gestures that could be recognized at any one time and it can assist in improving the recognition accuracy. Using context is certainly an area that has received considerable attention [149–151], especially in activity recognition [152–154], but there are several questions that need to be answered specifically related to context in 3D gestural interfaces. First, what type of context can be extracted that is most useful to improve recognition. As an example, in a video game, the current state of the player and the surrounding environment could provide useful information to trivially reject certain gestures that do not make sense in a given situation. Second, how can context be directly integrated into 3D gesture recognizers? As we have seen, there are a variety of different approaches to recognize 3D gestures, yet it is unclear how context can be best used in all of these algorithms. Finally, what performance benefits do we gain making use of context both in terms of accuracy and in latency reduction when compared to recognizers that do not make use of context? It is important to know how much more of an improvement we can get in accuracy and latency minimization so we can determine what the best approaches are for a given application.

5.4. Ecological Validity. Perhaps, one of the most important research challenges with 3D gestural interfaces is determining exactly how accurate the 3D gesture recognizer is that makes up the 3D gestural interface from a usability standpoint. In other words, how accurate is the 3D gestural interface when used in its intended setting. Currently most studies that explore a recognizer's accuracy are constrained experiments intended to evaluate the recognizer by having users perform each available gesture n number of times. As seen in Section 4, researchers have been able to get very high accuracy rates. However, we have also seen from Cheema et al. [125, 126] that accuracy can be severely reduced when tests are conducted in more realistic, ecologically valid scenarios. Even in the case of Cheema et al.'s work, their experiments do not come close to the ecological validity required to truly test a 3D gestural interface. Thus, these studies act more as an upper bound on gesture recognition performance than a true indicator of the recognition accuracy in everyday settings.

The open research problem here is how to design an ecologically valid experiment to test a 3D gestural interface. To illustrate the challenge, consider a 3D gestural interface for a video game. To adequately test the 3D gesture recognizer, we need to evaluate how accurately the recognizer can handle each gesture in the gesture set. However, to be ecologically valid, the game player should be able to use any gesture that makes sense at any given time. Thus, we do not know what gestures the user will be doing at any given time nor if they will provide enough test samples to adequately test the recognizer. That presents a difficult challenge. One option is to try to design the game so that each gesture is needed for an integral multiple of times, but this may not be the best user experience, if, for example, a user likes one particular gesture over another. Another option is to have many users

test the system, video tape the sessions, and then watch them to determine which gestures they appear to perform. With enough users, the number of gestures in the test set would approach the appropriate amount. Neither of these two options seem ideal and more research is needed to determine the best way to deal with the ecological validity issue.

6. Conclusion

3D gestural interaction represents a powerful and natural method of communication between humans and computers. The ability to use 3D gestures in a computer application requires a method to capture 3D position, orientation, and/or motion data through sensing technology. 3D gesture recognizers then need to be developed to detect specific pattern in the data from a set of known patterns. These 3D gesture recognizers can be heuristic-based, where a set of rules are encoded based on observation of the types of gestures needed in an application or through machine learning techniques where classifiers are trained using training data. These recognizers must then be evaluated to determine their accuracy and robustness.

In this paper, we have examined 3D gesture interaction research by exploring recent trends in these areas including sensing, recognition, and experimentation. These trends show that there are both strengths and weaknesses with the current state of 3D gesture interface technology. Strengths include powerful sensing technologies that can capture data from a user's whole body unobtrusively as well as new sensing technology directions that go beyond computer vision-based approaches. In addition, we are seeing better and faster 3D gesture recognition algorithms (that can make use of parallel processing) that can reach high recognition accuracies with reasonably sized gesture sets. However, 3D gestural interaction still suffers from several weaknesses. One of the most important is that although accuracy reported for various 3D gesture recognizers is high, these results are often considered a theoretical upper bound given that the accuracy will often degrade in more ecologically valid settings. Performing accuracy tests in ecologically valid settings is also a challenge making it difficult to determine the best ways to improve 3D gesture recognition technology. In addition, latency is still a concern in that delays between a user's gesture and the intended response can hamper the overall user experience. In spite of these issues, the field of 3D gesture interfaces is showing maturity as evidenced by 3D gestural interfaces moving into the commercial sector and becoming more commonplace. However, it is clear that there is still work that needs to be done to make 3D gesture interfaces truly mainstream as a significant part of human computer interfaces.

References

[1] I. E. Sutherland, "The ultimate display," in *Proceedings of the IFIP Congress*, pp. 506–508, 1965.

[2] A. van Dam, "Post-WIMP User Interfaces," *Communications of the ACM*, vol. 40, no. 2, pp. 63–67, 1997.

[3] P. Kortum, "HCI beyond the GUI: design for haptic, speech, olfactory, and other nontraditional interfaces," in *Interactive Technologies*, Elsevier Science, New York, NY, USA, 2008.

[4] D. Wigdor and D. Wixon, *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*, Elsevier Science, New York, NY, USA, 2011.

[5] D. A. Bowman, E. Kruijff, J. J. LaViola Jr., and I. Poupyrev, "An introduction to 3-D user interface design," *Presence*, vol. 10, no. 1, pp. 96–108, 2001.

[6] S. Zhai, P. O. Kristensson, C. Appert, T. H. Andersen, and X. Cao, "Foundational issues in touch-surface stroke gesture design-an integrative review," *Foundations and Trends in Human-Computer Interaction*, vol. 5, no. 2, pp. 97–205, 2012.

[7] J. J. LaViola Jr., "A survey of hand posture and gesture recognition techniques and technology," Technical Report, Brown University, Providence, RI, USA, 1999.

[8] S. Mitra and T. Acharya, "Gesture recognition: a survey," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 37, no. 3, pp. 311–324, 2007.

[9] J. Norton, C. A. Wingrave, and J. J. LaViola Jr., "Exploring strategies and guidelines for developing full body video game interfaces," in *Proceedings of the 5th International Conference on the Foundations of Digital Games (FDG '10)*, pp. 155–162, ACM, New York, NY, USA, June 2010.

[10] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: a review," *ACM Computing Surveys*, vol. 43, no. 3, p. 16, 2011.

[11] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, vol. 28, no. 6, pp. 976–990, 2010.

[12] G. Welch and E. Foxlin, "Motion tracking: no silver bullet, but a respectable arsenal," *IEEE Computer Graphics and Applications*, vol. 22, no. 6, pp. 24–38, 2002.

[13] L. Dipietro, A. M. Sabatini, and P. Dario, "A survey of glove-based systems and their applications," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 38, no. 4, pp. 461–482, 2008.

[14] P. Mistry and P. Maes, "SixthSense: a wearable gestural interface," in *Proceedings of the ACM SIGGRAPH ASIA 2009 Sketches (SIGGRAPH ASIA '09)*, p. 11:1,ACM, New York, NY, USA, December 2009.

[15] D. Kim, O. Hilliges, S. Izadi et al., "Digits: freehand 3d interactions anywhere using a wrist-worn gloveless sensor," in *Proceedings of the 25th annual ACM symposium on User interface software and technology (UIST '12)*, pp. 167–176, ACM, New York, NY, USA, 2012.

[16] G. Bailly, J. Muller, M. Rohs, D. Wigdor, and S. Kratz, "Shoesense: a new perspective on gestural interaction and wearable applications," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*, pp. 1239–1248, ACM, New York, NY, USA, 2012.

[17] T. Starner, J. Auxier, D. Ashbrook, and M. Gandy, "Gesture Pendant: a self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring," in *Proceedings of the 4th Intenational Symposium on Wearable Computers*, pp. 87–94, October 2000.

[18] C. Harrison, H. Benko, and A. D. Wilson, "OmniTouch: wearable multitouch interaction everywhere," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)* pp. 441–450, ACM, New York, NY, USA, October 2011.

[19] J. Kim, J. He, K. Lyons, and T. Starner, "The Gesture Watch: a wireless contact-free Gesture based wrist interface," in *Proceedings of the11th IEEE International Symposium on Wearable*

Computers (ISWC '07), pp. 15–22, IEEE Computer Society, Washington, DC, USA, October 2007.

[20] T. S. Saponas, D. S. Tan, D. Morris, and R. Balakrishnan, "Demonstrating the feasibility of using forearm electromyography for muscle-computer interfaces," in *Proceedings of the 26th Annual CHI Conference on Human Factors in Computing Systems (CHI '08)*, pp. 515–524, ACM, New York, NY, USA, April 2008.

[21] T. S. Saponas, D. S. Tan, D. Morris, R. Balakrishnan, J. Turner, and J. A. Landay, "Enabling always-available input with muscle-computer interfaces," in *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology (UIST '09)*, pp. 167–176, ACM, New York, NY, USA, October 2009.

[22] T. S. Saponas, D. S. Tan, D. Morris, J. Turner, and J. A. Landay, "Making muscle-computer interfaces more practical," in *Proceedings of the 28th Annual CHI Conference on Human Factors in Computing Systems (CHI '10)*, pp. 851–854, ACM, New York, NY, USA, April 2010.

[23] C. Harrison, D. Tan, and D. Morris, "Skinput: appropriating the body as an input surface," in *Proceedings of the 28th Annual CHI Conference on Human Factors in Computing Systems (CHI '10)*, pp. 453–462, ACM, New York, NY, USA, April 2010.

[24] R. Y. Wang and J. Popović, "Real-time hand-tracking with a color glove," *ACM Transactions on Graphics*, vol. 28, no. 3, p. 63, 2009.

[25] D. F. Keefe, D. A. Feliz, T. Moscovich, D. H. Laidlaw, and J. LaViola J.J., "CavePainting: a fully immersive 3D artistic medium and interactive experience," in *Proceedings of the 2001 symposium on Interactive 3D graphics (I3D '01)*, pp. 85–93, ACM, New York, NY, USA, March 2001.

[26] C. Ware and D. R. Jessome, "Using the BAT: a six-dimensional mouse for object placement," *IEEE Computer Graphics and Applications*, vol. 8, no. 6, pp. 65–70, 1988.

[27] R. C. Zeleznik, J. J. La Viola Jr., D. Acevedo Feliz, and D. F. Keefe, "Pop through button devices for VE navigation and interaction," in *Proceedings of the IEEE Virtual Reality 2002*, pp. 127–134, March 2002.

[28] A. Basu, C. Saupe, E. Refour, A. Raij, and K. Johnsen, "Immersive 3dui on one dollar a day," in *Proceedings of the IEEE Symposium on 3D User Interfaces (3DUI '12)*, pp. 97–100, 2012.

[29] B. Williamson, C. Wingrave, and J. LaViola, "Full body locomotion with video game motion controllers," in *Human Walking in Virtual Environments*, F. Steinicke, Y. Visell, J. Campos, and A. Lecuyer, Eds., pp. 351–376, Springer, New York, NY, USA, 2013.

[30] C. A. Wingrave, B. Williamson, P. D. Varcholik et al., "The wiimote and beyond: spatially convenient devices for 3D user interfaces," *IEEE Computer Graphics and Applications*, vol. 30, no. 2, pp. 71–85, 2010.

[31] G. Cohn, D. Morris, S. N. Patel, and D. S. Tan, "Your noise is my command: sensing gestures using the body as an antenna," in *Proceedings of the 29th Annual SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*, pp. 791–800, ACM, New York, NY, USA, May 2011.

[32] G. Cohn, S. Gupta, T.-J. Lee et al., "An ultra-low-power human body motion sensor using static electric field sensing," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp '12)*, pp. 99–102, ACM, New York, NY, USA, 2012.

[33] G. Cohn, D. Morris, S. Patel, and D. Tan, "Humantenna: using the body as an antenna for real-time whole-body interaction," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems (CHI '12)*, pp. 1901–1910, ACM, New York, NY, USA, 2012.

[34] J. H. Hammer and J. Beyerer, "Robust hand tracking in realtime using a single head-mounted rgb camera," in *Human-Computer Interaction. Interaction Modalities and Techniques*, M. Kurosu, Ed., vol. 8007 of *Lecture Notes in Computer Science*, pp. 252–261, Springer, Berlin, Germany, 2013.

[35] S.-H. Choi, J.-H. Han, and J.-H. Kim, "3D-position estimation for hand gesture interface using a single camera," in *Human-Computer Interaction. Interaction Techniques and Environments*, J. A. Jacko, Ed., vol. 6762 of *Lecture Notes in Computer Science*, pp. 231–237, Springer, Berlin, Germany, 2011.

[36] S. Rodriguez, A. Picon, and A. Villodas, "Robust vision-based hand tracking using single camera for ubiquitous 3D gesture interaction," in *Proceedings of the IEEE Symposium on 3D User Interfaces 2010 (3DUI '10)*, pp. 135–136, March 2010.

[37] T. Starner, J. Weaver, and A. Pentland, "Real-time american sign language recognition using desk and wearable computer based video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1371–1375, 1998.

[38] A. K. Bhowmik, "3D computer vision," in *SID Seminar Lecture Notes, M9*, 2012.

[39] R. Y. Wang, S. Paris, and J. Popović, "6D hands: markerless hand tracking for computer aided design," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)* pp. 549–557, ACM, New York, NY, USA, October 2011.

[40] G. Hackenberg, R. McCall, and W. Broll, "Lightweight palm and finger tracking for real-time 3D gesture control," in *Proceedings of the 18th IEEE Virtual Reality Conference (VR '11)* pp. 19–26, March 2011.

[41] C. Keskin, F. Kiraç, Y. E. Kara, and L. Akarun, "Real time hand pose estimation using depth sensors," in *Consumer Depth Cameras for Computer Vision, Advances in Computer Vision and Pattern Recognition*, A. Fossati, J. Gall, H. Grabner, X. Ren, and K. Konolige, Eds., pp. 119–137, Springer, 2013.

[42] Z. Feng, S. Xu, X. Zhang, L. Jin, Z. Ye, and W. Yang, "Real-time fingertip tracking and detection using kinect depth sensor for a new writing-in-the air system," in *Proceedings of the 4th International Conference on Internet Multimedia Computing and Service (ICIMCS '12)*, pp. 70–74, ACM, New York, NY, USA, 2012.

[43] H. Liang, J. Yuan, and D. Thalmann, "3D fingertip and palm tracking in depth image sequences," in *Proceedings of the 20th ACM international conference on Multimedia (MM '12)*, pp. 785–788, ACM, New York, NY, USA, 2012.

[44] Z. Ren, J. Yuan, and Z. Zhang, "Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera," in *Proceedings of the 19th ACM International Conference on Multimedia ACM Multimedia (MM '11)* pp. 1093–1096, ACM, New York, NY, USA, December 2011.

[45] J. Shotton, T. Sharp, A. Kipman et al., "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2011.

[46] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, "Real time motion capture using a single time-of-flight camera," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 755–762, June 2010.

[47] L. A. Schwarz, A. Mkhitaryan, D. Mateus, and N. Navab, "Human skeleton tracking from depth data using geodesic distances and optical flow," *Image and Vision Computing*, vol. 30, no. 3, pp. 217–226, 2012.

[48] X. Wei, P. Zhang, and J. Chai, "Accurate realtime full-body motion capture using a single depth camera," *ACM Transactions on Graphics*, vol. 31, no. 6, pp. 188:1-188:12,2012.

[49] S. Gupta, D. Morris, S. Patel, and D. Tan, "Soundwave: using the doppler effect to sense gestures," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems (CHI '12)*, pp. 1911-1914, ACM, New York, NY, USA, 2012.

[50] S. Gupta, K. Chen, M. S. Reynolds, and S. N. Patel, "LightWave: using compact fluorescent lights as sensors," in *Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp '11)*, pp. 65–74, ACM, New York, NY, USA, September 2011.

[51] J. Suarez and R. R. Murphy, "Hand gesture recognition with depth images: a review," in *Proceedings of the 21st IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN '12)*, pp. 411-417, IEEE, 2012.

[52] A. D. Wilson, "Sensor- and recognition-based input for interaction," in *The Human Computer Interaction Handbook*, chapter 7, pp. 133–156, 2012.

[53] Y. Wu and T. S. Huang, "Vision-based gesture recognition: a review," in *Gesture-Based Communication in Human-Computer Interaction*, A. Braffort, R. Gherbi, S. Gibet, D. Teil, and J. Richardson, Eds., vol. 1739 of *Lecture Notes in Computer Science*, pp. 103–115,Springer, Berlin, Germany, 1999.

[54] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer, New York, NY, USA, 2006.

[55] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*, Wiley-Interscience, New York, NY, USA, 2000.

[56] L. R. Rabiner, "Tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[57] S. Sako and T. Kitamura, "Subunit modeling for japanese sign language recognition based on phonetically depend multi-stream hidden markov models," in *Universal Access in Human-Computer Interaction. Design Methods, Tools, and Interaction Techniques for EInclusion*, C. Stephanidis and M. Antona, Eds., vol. 8009 of *Lecture Notes in Computer Science*, pp. 548–555, Springer, Berlin, Germany, 2013.

[58] H. Pang and Y. Ding, "Dynamic hand gesture recognition using kinematic features based on hidden markov model," in *Proceedings of the 2nd International Conference on Green Communications and Networks (GCN '12)*, Y. Yang and M. Ma, Eds., vol. 5–227 of *Lecture Notes in Electrical Engineering*, pp. 255–262, Springer, Berlin, Germany, 2013.

[59] F. Bevilacqua, B. Zamborlin, A. Sypniewski, N. Schnell, F. Guédy, and N. Rasamimanana, "Continuous realtime gesture following and recognition," in *Gesture in Embodied Communication and Human-Computer Interaction*, S. Kopp and I. Wachsmuth, Eds., vol. 5934 of *Lecture Notes in Computer Science*, pp. 73–84, Springer, Berlin, Germany, 2009.

[60] J. Wan, Q. Ruan, G. An, and W. Li, "Gesture recognition based on hidden markov model from sparse representative observations," in *Proceedings of the IEEE 11th International Conference on Signal Processing (ICSP '12)*, vol. 2, pp. 1180–1183, 2012.

[61] Y. Lee and S. Cho, "Activity recognition using hierarchical hidden markov models on a smartphone with 3D accelerometer," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6678, no. 1, pp. 460–467, 2011.

[62] S. Bilal, R. Akmeliawati, A. A. Shafie, and M. J. E. Salami, "Hidden Markov model for human to computer interaction: a study on human hand gesture recognition," *Artificial Intelligence Review*, 2011.

[63] F. Chen, C. Fu, and C. Huang, "Hand gesture recognition using a real-time tracking method and hidden Markov models," *Image and Vision Computing*, vol. 21, no. 8, pp. 745–758, 2003.

[64] D. Kelly, J. McDonald, C. Markham, and editors, "Recognition of spatiotemporal gestures in sign language using gesture threshold hmms," in *Machine Learning for Vision-Based Motion Analysis, Advances in Pattern Recognition*, L. Wang, G. Zhao, L. Cheng, and M. Pietikainen, Eds., pp. 307–348, Springer, London, UK, 2011.

[65] A. Just and S. Marcel, "A comparative study of two state-of-the-art sequence processing techniques for hand gesture recognition," *Computer Vision and Image Understanding*, vol. 113, no. 4, pp. 532–543, 2009.

[66] N. Nguyen-Duc-Thanh, S. Lee, and D. Kim, "Two-stage hidden markov model in gesture recognition for human robot interaction," *International Journal of Advanced Robotic Systems*, vol. 9, no. 39, 2012.

[67] T. Pylvänäinen, "Accelerometer based gesture recognition using continuous HMMs," in *Proceedings of the 2nd Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA '05)*, J. S. Marques, N. P. de la Blanca, and P. Pina, Eds., vol. 3522 of *Lecture Notes in Computer Science*, pp. 639–646, Springer, Berlin, Germany, June 2005.

[68] A. Whitehead and K. Fox, "Device agnostic 3D gesture recognition using hidden Markov models," in *Proceedings of the GDC Canada International Conference on the Future of Game Design and Technology (FuturePlay '09)*, pp. 29–30, ACM, New York, NY, USA, May 2009.

[69] P. Zappi, B. Milosevic, E. Farella, and L. Benini, "Hidden Markov Model based gesture recognition on low-cost, low-power Tangible User Interfaces," *Entertainment Computing*, vol. 1, no. 2, pp. 75–84, 2009.

[70] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: probabilistic models for segmenting and labeling sequence data," in *Proceedings of the 18th International Conference on Machine Learning (ICML '01)*, pp. 282–289, Morgan Kaufmann, San Francisco, Calif, USA, 2001.

[71] H. Chung and H.-D. Yang, "Conditional random field-based gesture recognition with depth information," *Optical Engineering*, vol. 52, no. 1, p. 017201, 2013.

[72] H. Yang, S. Sclaroff, and S. Lee, "Sign language spotting with a threshold model based on conditional random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 7, pp. 1264–1277, 2009.

[73] H. Yang and S. Lee, "Simultaneous spotting of signs and fingerspellings based on hierarchical conditional random fields and boostmap embeddings," *Pattern Recognition*, vol. 43, no. 8, pp. 2858–2870, 2010.

[74] C. Ellis, S. Z. Masood, M. F. Tappen, J. J. LaViola Jr., and R. Sukthankar, "Exploring the trade-off between accuracy and observational latency in action recognition," *International Journal of Computer Vision*, vol. 101, no. 3, pp. 420–436, 2013.

[75] M. Elmezain, A. Al-Hamadi, S. Sadek, and B. Michaelis, "Robust methods for hand gesture spotting and recognition using Hidden Markov Models and Conditional Random Fields," in *Proceedings of the 10th IEEE International Symposium on Signal Processing and Information Technology (ISSPIT '10)*, pp. 131–136, December 2010.

[76] A. Quattoni, S. Wang, L. Morency, M. Collins, and T. Darrell, "Hidden conditional random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, pp. 1848–1853, 2007.

[77] B. W. Sy, A. Quattoni, L. Morency, D. Demirdjian, and T. Darrell, "Hidden conditional random fields for gesture recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 2, pp. 1521–1527, June 2006.

[78] Y. Song, D. Demirdjian, and R. Davis, "Multi-signal gesture recognition using temporal smoothing hidden conditional random fields," in *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition and Workshops (FG '11)*, pp. 388–393, March 2011.

[79] T. Liu, K. Wang, A. Tsai, and C. Wang, "Hand posture recognition using hidden conditional random fields," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '09)*, pp. 1828–1833, July 2009.

[80] F. Liu and Y. Jia, "Human action recognition using manifold learning and hidden conditional random fields," in *Proceedings of the 9th International Conference for Young Computer Scientists (ICYCS '08)*, pp. 693–698, November 2008.

[81] C. R. Souza, E. B. Pizzolato, M. S. Anjo, and editors, "Fingerspelling recognition with support vector machines and hidden conditional random fields," in *Proceedings of the Ibero-American Conference on Artificial Intelligence (IBERAMIA '12)*, J. Pavon, N. D. Duque-Mendez, and R. Fuentes-Fernandez, Eds., vol. 7637 of *Lecture Notes in Computer Science*, pp. 561–570, Springer, 2012.

[82] L. Morency, A. Quattoni, and T. Darrell, "Latent-dynamic discriminative models for continuous gesture recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, June 2007.

[83] M. Elmezain and A. Al-Hamadi, "Ldcrfs-based hand gesture recognition," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '12)*, pp. 2670–2675, 2012.

[84] Y. Song, D. Demirdjian, and R. Davis, "Continuous body and hand gesture recognition for natural human-computer interaction," *ACM Transactions on Interactive Intelligent Systems*, vol. 2, no. 1, pp. 5:1–5:28, 2012.

[85] Y. Zhang, K. Adl, and J. Glass, "Fast spoken query detection using lower-bound dynamic time warping on graphical processing units," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '12)*, pp. 5173–5176, March 2012.

[86] M. Elmezain, A. Al-Hamadi, and B. Michaelis, "Discriminative models-based hand gesture recognition," in *Proceedings of the 2nd International Conference on Machine Vision (ICMV '09)*, pp. 123–127, December 2009.

[87] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.

[88] D. Huang, W. Hu, and S. Chang, "Vision-based hand gesture recognition using PCA+Gabor filters and SVM," in *Proceedings of the 5th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP '09)*, pp. 1–4, September 2009.

[89] C. T. Hsieh, C. H. Yeh, K. M. Hung, L. M. Chen, and C. Y. Ke, "A real time hand gesture recognition system based on dft and svm," in *Proceedings of the 8th International Conference on Information Science and Digital Content Technology (ICIDT '12)*, vol. 3, pp. 490–494, 2012.

[90] C.-C. Hsieh and D.-H. Liou, "Novel haar features for real-time hand gesture recognition using svm," *Journal of Real-Time Image Processing*, 2012.

[91] N. Dardas, Q. Chen, N. D. Georganas, and E. M. Petriu, "Hand gesture recognition using bag-of-features and multi-class support vector machine," in *Proceedings of the 9th IEEE International Symposium on Haptic Audio-Visual Environments and Games (HAVE '10)*, pp. 163–167, October 2010.

[92] N. H. Dardas and N. D. Georganas, "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 11, pp. 3592–3607, 2011.

[93] Y. Chen and K. Tseng, "Multiple-angle hand gesture recognition by fusing SVM classifiers," in *Proceedings of the 3rd IEEE International Conference on Automation Science and Engineering (IEEE CASE '07)*, pp. 527–530, September 2007.

[94] O. Rashid, A. Al-Hamadi, and B. Michaelis, "A framework for the integration of gesture and posture recognition using HMM and SVM," in *Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS '09)*, vol. 4, pp. 572–577, November 2009.

[95] K. K. Biswas and S. K. Basu, "Gesture recognition using Microsoft Kinect," in *Proceedings of the 5th International Conference on Automation, Robotics and Applications (ICARA '11)*, pp. 100–103, December 2011.

[96] Z. He, L. Jin, L. Zhen, and J. Huang, "Gesture recognition based on 3D accelerometer for cell phones interaction," in *Proceedings of the IEEE Asia Pacific Conference on Circuits and Systems (APCCAS '08)*, pp. 217–220, December 2008.

[97] S. H. Lee, M. K. Sohn, D. J. Kim, B. Kim, and H. Kim, "Smart tv interaction system using face and hand gesture recognition," in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE '13)*, pp. 173–174, 2013.

[98] R. C. B. Madeo, C. A. M. Lima, and S. M. Peres, "Gesture unit segmentation using support vector machines: segmenting gestures from rest positions," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC '13)*, pp. 46–52, ACM, New York, NY, USA, 2013.

[99] Y. Ren and F. Zhang, "Hand gesture recognition based on MEB-SVM," in *Proceedings of the International Conference on Embedded Software and System (ICESS '09)*, pp. 344–349, May 2009.

[100] J. Wu, G. Pan, D. Zhang, G. Qi, and S. Li, "Gesture recognition with a 3-D accelerometer," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5585, pp. 25–38, 2009.

[101] L. Yun and Z. Peng, "An automatic hand gesture recognition system based on Viola-Jones method and SVMs," in *Proceedings of the 2nd International Workshop on Computer Science and Engineering (WCSE '09)*, vol. 2, pp. 72–76, October 2009.

[102] W.-Y. Loh, "Classification and regression trees," *Wiley Interdisciplinary Reviews*, vol. 1, no. 1, pp. 14–23, 2011.

[103] J. R. Quinlan, *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*, Morgan Kaufmann, Boston, Mass, USA, 1st edition, 1993.

[104] S. Nisar, A. A. Khan, and M. Y. Javed, "A statistical feature based decision tree approach for hand gesture recognition," in *Proceedings of the 7th International Conference on Frontiers of Information Technology (FIT '09)*, vol. 27, pp. 1–6, ACM, New York, NY, USA, December 2009.

[105] M. Jeon, S. Yang, and Z. Bien, "User adaptive hand gesture recognition using multivariate fuzzy decision tree and fuzzy garbage model," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 474–479, August 2009.

[106] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang, "A framework for hand gesture recognition based on accelerometer and EMG sensors," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 41, no. 6, pp. 1064–1076, 2011.

[107] G. Fang, W. Gao, and D. Zhao, "Large vocabulary sign language recognition based on hierarchical decision trees," in *Proceedings of the 5th International Conference on Multimodal Interfaces (ICMI '03)*, pp. 125–131, ACM, New York, NY, USA, November 2003.

[108] S. Oprisescu, C. Rasche, and B. Su, "Automatic static hand gesture recognition using tof cameras," in *Proceedings of the 20th European Signal Processing Conference (EUSIPCO '12)*, pp. 2748–2751, 2012.

[109] L. Miranda, T. Vieira, D. Martinez, T. Lewiner, A. W. Vieira, and M. F. M. Campos, "Real-time gesture recognition from depth data through key poses learning and decision forests," in *Proceedings of the 25th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI '12)*, pp. 268–275, 2012.

[110] C. Keskin, F. Kirac, Y. Kara, and L. Akarun, "Hand pose estimation and hand shape classification using multi-layered randomized decision forests," in *Proceedings of the 12th European Conference on Computer Vision (ECCV '12)*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., vol. 7577 of *Lecture Notes in Computer Science*, pp. 852–863, Springer, Berlin, Germany, 2012.

[111] F. Negin, F. Ozdemir, C. Akgul, K. Yuksel, and A. Ercil, "A decision forest based feature selection framework for action recognition from rgb-depth cameras," in *Image Analysis and Recognition*, M. Kamel and A. Campilho, Eds., vol. 7950 of *Lecture Notes in Computer Science*, pp. 648–657, Springer, Berlin, Germany, 2013.

[112] M. Hoffman, P. Varcholik, and J. J. LaViola Jr., "Breaking the status quo: improving 3D gesture recognition with spatially convenient input devices," in *Proceedings of the 2010 IEEE Virtual Reality Conference (VR '10)*, pp. 59–66, IEEE Computer Society, Washington, DC, USA, March 2010.

[113] S. Kratz and M. Rohs, "A % gesture recognizer: simple gesture recognition for devices equipped with 3D acceleration sensors," in *Proceedings of the 14th ACM International Conference on Intelligent User Interfaces (IUI '10)*, pp. 341–344, ACM, New York, NY, USA, February 2010.

[114] S. Kratz and M. Rohs, "Protractor3D: a closed-form solution to rotation-invariant 3D gestures," in *Proceedings of the 15th ACM International Conference on Intelligent User Interfaces (IUI '11)*, pp. 371–374, ACM, New York, NY, USA, February 2011.

[115] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision forests: a unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning," *Foundations and Trends in Computer Graphics and Vision*, vol. 7, no. 2-3, pp. 81–227, 2011.

[116] A. Criminisi and J. Shotton, *Decision Forests for Computer Vision and Medical Image Analysis*, Springer, New York, NY, USA, 2013.

[117] S. Fothergill, H. Mentis, P. Kohli, and S. Nowozin, "Instructing people for training gestural interactive systems," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*, pp. 1737–1746, ACM, New York, NY, USA, 2012.

[118] M. Lech, B. Kostek, A. Czyzewski, and editors, "Examining classifiers applied to static hand gesture recognition in novel sound mixing systemvolume," in *Multimedia and Internet Systems: Theory and Practice*, A. Zgrzywa, K. Choros, and A. Sieminski, Eds., vol. 183 of *Advances in Intelligent Systems and Computing*, pp. 77–86, Springer, Berlin, Germany, 2013.

[119] K. R. Konda, A. Königs, H. Schulz, and D. Schulz, "Real time interaction with mobile robots using hand gestures," in *Proceedings of the 7th Annual ACM/IEEE International Conference on Human-Robot Interaction (HRI '12)*, pp. 177–178, ACM, New York, NY, USA, March 2012.

[120] K. Murakami and H. Taguchi, "Gesture recognition using recurrent neural networks," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '91)*, pp. 237–242, ACM, New York, NY, USA, 1991.

[121] Z. Li and R. Jarvis, "Real time hand gesture recognition using a range camera," in *Proceedings of the Australasian Conference on Robotics and Automation (ACRA '09)*, pp. 21–27, December 2009.

[122] X. Liu and K. Fujimura, "Hand gesture recognition using depth data," in *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition (FGR '04)*, pp. 529–534, May 2004.

[123] P. Hong, M. Turk, and T. S. Huang, "Gesture modeling and recognition using finite state machines," in *Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 410–415, 2000.

[124] D. Rubine, "Specifying gestures by example," in *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '91)*, pp. 329–337, ACM, New York, NY, USA, 1991.

[125] S. Cheema, M. Hoffman, and J. J. LaViola Jr., "3D gesture classification with linear acceleration and angular velocity sensing devices for video games," *Entertainment Computing*, vol. 4, no. 1, pp. 11–24, 2013.

[126] S. Cheema and J. J. LaViola Jr., "Wizard of Wii: toward understanding player experience in first person games with 3D gestures," in *Proceedings of the 6th International Conference on the Foundations of Digital Games (FDG '11)*, pp. 265–267, ACM, New York, NY, USA, July 2011.

[127] J. O. Wobbrock, A. D. Wilson, and Y. Li, "Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes," in *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST '07)*, pp. 159–168, ACM, New York, NY, USA, October 2007.

[128] S. Kratz and M. Rohs, "The % recognizer: simple 3D gesture recognition on mobile devices," in *Proceedings of the 14th ACM International Conference on Intelligent User Interfaces (IUI '10)*, pp. 419–420, ACM, New York, NY, USA, February 2010.

[129] Y. Li, "Protractor: a fast and accurate gesture recognizer," in *Proceedings of the 28th Annual CHI Conference on Human Factors in Computing Systems (CHI '10)*, pp. 2169–2172, ACM, New York, NY, USA, April 2010.

[130] S. Kratz, M. Rohs, and G. Essl, "Combining acceleration and gyroscope data for motion gesture recognition using classifiers with dimensionality constraints," in *Proceedings of the 2013 international conference on Intelligent user interfaces (IUI '13)*, pp. 173–178, ACM, New York, NY, USA, 2013.

[131] B. Williamson, C. Wingrave, J. LaViola, T. Roberts, and P. Garrity, "Natural full body interaction for navigation in dismounted

soldier training," in *Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC '11)*, pp. 2103–2110, 2011.

[132] J. N. Bott, J. G. Crowley, and J. J. LaViola Jr., "Exploring 3D gestural interfaces for music creation in video games," in *Proceedings of the 4th International Conference on the Foundations of Digital Games (ICFDG '09)*, pp. 18–25, ACM, New York, NY, USA, April 2009.

[133] E. Charbonneau, A. Miller, C. Wingrave, and J. J. LaViola Jr., "Understanding visual interfaces for the next generation of dance-based rhythm video games," in *Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games (Sandbox '09)*, pp. 119–126, ACM, New York, NY, USA, August 2009.

[134] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," *Communications of the ACM*, vol. 54, no. 2, pp. 60–71, 2011.

[135] H. Kang, C. W. Lee, and K. Jung, "Recognition-based gesture spotting in video games," *Pattern Recognition Letters*, vol. 25, no. 15, pp. 1701–1714, 2004.

[136] J. Payne, P. Keir, J. Elgoyhen et al., "Gameplay issues in the design of spatial 3D gestures for video games," in *Proceedings of the CHI '06 Extended Abstracts on Human Factors in Computing Systems (CHI EA '06)*, pp. 1217–1222, ACM, New York, NY, USA, 2006.

[137] T. Starner, B. Leibe, B. Singletary, and J. Pair, "MIND-WARPING: towards creating a compelling collaborative augmented reality game," in *Proceedings of the 5th International Conference on Intelligent User Interfaces (IUI '00)*, pp. 256–259, ACM, New York, NY, USA, January 2000.

[138] A. Bigdelou, L. A. Schwarz, and N. Navab, "An adaptive solution for intra-operative gesture-based human-machine interaction," in *Proceedings of the 17th ACM International Conference on Intelligent User Interfaces (IUI '12)*, pp. 75–83, ACM, New York, NY, USA, February 2012.

[139] L. A. Schwarz, A. Bigdelou, and N. Navab, "Learning gestures for customizable human-computer interaction in the operating room," in *Proceedings of the Medical Image Computing and Computer-Assisted Intervention (MICCAI '11)* G. Fichtinger, A. Martel, and T. Peters, Eds., vol. 6891 of *Lecture Notes in Computer Science*, pp. 129–136, Springer, Berlin, Germany, 2011.

[140] K. Pfeil, S. L. Koh, and J. LaViola, "Exploring 3D gesture metaphors for interaction with unmanned aerial vehicles," in *Proceedings of the 2013 International Conference on Intelligent User Interfaces (IUI '13)*, pp. 257–266, ACM, New York, NY, USA, 2013.

[141] B. Burger, I. Ferrané, F. Lerasle, and G. Infantes, "Two-handed gesture recognition and fusion with speech to command a robot," *Autonomous Robots*, pp. 1–19, 2011.

[142] M. Sigalas, H. Baltzakis, and P. Trahanias, "Gesture recognition based on arm tracking for human-robot interaction," in *Proceedings of the 23rd IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 5424–5429, October 2010.

[143] M. Van Den Bergh, D. Carton, R. De Nijs et al., "Real-time 3D hand gesture interaction with a robot for understanding directions from humans," in *Proceedings of the 20th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN '11)*, pp. 357–362, August 2011.

[144] A. Riener, "Gestural interaction in vehicular applications," *Computer*, vol. 45, no. 4, Article ID 6165247, pp. 42–47, 2012.

[145] S. H. Lee, M. K. Sohn, D. J. Kim, B. Kim, and H. Kim, "Smart tv interaction system using face and hand gesture recognition," in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE '13)*, pp. 173–174, IEEE, 2013.

[146] M. Takahashi, M. Fujii, M. Naemura, and S. Satoh, "Human gesture recognition system for TV viewing using time-of-flight camera," *Multimedia Tools and Applications*, vol. 62, no. 3, pp. 761–783, 2013.

[147] H. I. Stern, J. P. Wachs, and Y. Edan, "Designing hand gesture vocabularies for natural interaction by combining psychophysiological and recognition factors," *International Journal of Semantic Computing*, vol. 2, no. 1, pp. 137–160, 2008.

[148] S. Z. Masood, C. Ellis, A. Nagaraja, M. F. Tappen, J. J. Laviola, and R. Sukthankar, "Measuring and reducing observational latency when recognizing actions," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV '11)*, pp. 422–429, November 2011.

[149] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, pp. 263–277, 2007.

[150] W. Liu, X. Li, and D. Huang, "A survey on context awareness," in *Proceedings of the International Conference on Computer Science and Service System (CSSS '11)*, pp. 144–147, June 2011.

[151] A. Saeed and T. Waheed, "An extensive survey of context-aware middleware architectures," in *Proceedings of the IEEE International Conference on Electro/Information Technology (EIT '10)*, pp. 1–6, May 2010.

[152] D. Han, L. Bo, and C. Sminchisescu, "Selection and context for action recognition," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 1933–1940, October 2009.

[153] D. J. Moore, I. A. Essa, and M. H. Hayes, "Object spaces: context management for human activity recognition," in *Proceedings of 2nd International Conference on Audio-Vision-based Person Authentication*, 1998.

[154] D. J. Moore, I. A. Essa, and M. H. Hayes III, "Exploiting human actions and object context for recognition tasks," in *Proceedings of the 1999 7th IEEE International Conference on Computer Vision (ICCV'99)*, pp. 80–86, September 1999.