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,

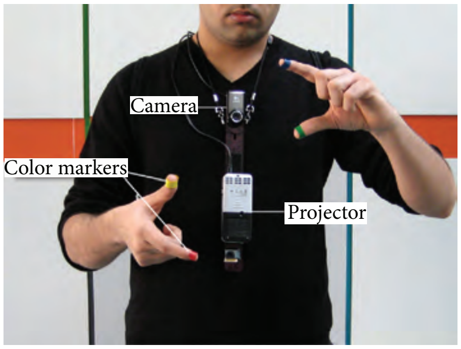
(iv) using the recognizer in an application so commands or operations are performed when specific patterns are detected.

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].

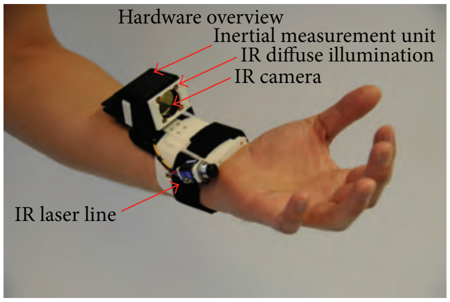


Figur e 1: The SixSense system. A user wears colored fiducial markers for fingertip tracking [14].

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 underutilized, 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

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you’re on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm’s cloud-native docket research platform finds what other services can’t. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**