8-1-2015

A Survey of Applications and Human Motion Recognition with Microsoft Kinect

Roanna Lun
Cleveland State University

Wenbing Zhao
Cleveland State University, w.zhao1@csuohio.edu

Follow this and additional works at: https://engagedscholarship.csuohio.edu/enece_facpub

Part of the Electrical and Computer Engineering Commons

How does access to this work benefit you? Let us know!

Repository Citation
Lun, Roanna and Zhao, Wenbing, "A Survey of Applications and Human Motion Recognition with Microsoft Kinect" (2015). Electrical Engineering and Computer Science Faculty Publications. 408.
https://engagedscholarship.csuohio.edu/enece_facpub/408

This Article is brought to you for free and open access by the Electrical and Computer Engineering Department at EngagedScholarship@CSU. It has been accepted for inclusion in Electrical Engineering and Computer Science Faculty Publications by an authorized administrator of EngagedScholarship@CSU. For more information, please contact library.es@csuohio.edu.
A SURVEY OF APPLICATIONS AND HUMAN MOTION RECOGNITION WITH MICROSOFT KINECT

ROANNA LUN
Department of Electrical and Computer Engineering, Cleveland State University, 2121 Euclid Avenue, Cleveland, Ohio 44115
r.lun@csuohio.edu

WENBING ZHAO
Department of Electrical and Computer Engineering, Cleveland State University, 2121 Euclid Avenue, Cleveland, Ohio 44115
wenbing@ieee.org

Microsoft Kinect, a low-cost motion sensing device, enables users to interact with computers or game consoles naturally through gestures and spoken commands without any other peripheral equipment. As such, it has commanded intense interests in research and development on the Kinect technology. In this article, we present a comprehensive survey on Kinect applications, and the latest research and development on motion recognition using data captured by the Kinect sensor. On the applications front, we review the applications of the Kinect technology in a variety of areas, including healthcare, education and performing arts, robotics, sign language recognition, retail services, workplace safety training, as well as 3D reconstructions. On the technology front, we provide an overview of the main features of both versions of the Kinect sensor together with the depth sensing technologies used, and review literatures on human motion recognition techniques used in Kinect applications. We provide a classification of motion recognition techniques to highlight the different approaches used in human motion recognition. Furthermore, we compile a list of publicly available Kinect datasets. These datasets are valuable resources for researchers to investigate better methods for human motion recognition and lower-level computer vision tasks such as segmentation, object detection, and human pose estimation.

Keywords: Human Motion Recognition; Machine Learning; Microsoft Kinect

1. Introduction

Launched in 2010, Microsoft Kinect is one of the most popular game controllers in recent years, having sold more than 24 million units as of February 2013. Kinect allows users to naturally interact with a computer or game console with gestures and/or voice commands. With such widespread popularity in the market, Microsoft Kinect has attracted many researchers to investigate its applications beyond video gaming, as well as to study fundamentals in computer vision-based human motion tracking and recognition.

In late 2011, Microsoft released a Software Development Kit (SDK) for its Kinect
sensors. The SDK enables users to develop sophisticated computer-based human motion tracking applications in C# and C++ programming languages. The immersive Kinect technology from both hardware design and the SDK makes it possible to detect, track and recognize human motion dynamically in real-time. Applications of Microsoft Kinect have been extended to many fields beyond video games, including healthcare, education, retail, training, virtual reality, robotics, sign languages, and other areas. Moreover, researchers have intensively studied fundamental techniques for human motion tracking and analysis using Microsoft Kinect.

In this article, we present a comprehensive review of the applications of the Microsoft Kinect sensor in various domains and recent studies on human motion recognition that power these applications. The main difference between our study and the existing reviews of the Kinect technology in Ref. 50, 146 lies in the comprehensiveness of our review in two specific areas: (1) Kinect applications, and (2) human motion recognition.

The primary objective of Ref. 146 is to illustrate the technology embedded inside the Kinect device and its SDK, such as the hardware design, sensor calibration, human skeletal tracking techniques, and head pose and facial-expression tracking mechanisms. It also presents a prototype system that utilizes multiple Kinect sensors in an immersive teleconferencing application. The review provided in Ref. 50 covers a broad topics related to the Kinect technology, including object detection, human activities, hand gestures, and indoor 3D mapping. Our work has a number of major differences from Ref. 50:

- We provide a comprehensive review of the application of the Kinect technology in many different areas, which is not included in Ref. 50.
- Our survey focuses on human motion recognition. Even though some overlap is inevitable with Ref. 50, our survey provides a much more in-depth coverage on methods of human motion recognition. For example, in addition to Hidden Markov Model (HMM) and Dynamic Time Warping (DTW), which are briefly covered in Ref. 50, we survey several other algorithms and techniques such as artificial neural networks, randomized decision forests, Adaboost, least squares regression, kernel regression, rule-based realtime gesture and gesture state recognition. Furthermore, we provide a classification of common approaches used in human motion recognition, and review each approach accordingly.
- We intentionally omit the literature on low-level object tracking and detection, and the work on human pose and skeleton estimation, because these topics have been well reviewed in Ref. 50.
- We compile a list of most recent publicly available Kinect datasets with URL to each dataset. Even though a list of datasets are also included in Ref. 50, no efforts were made to ensure that each dataset is actually available publicly. As such, many cited datasets are in fact not available publicly.
The remainder of this article is organized as follows: Section 2 introduces the main features of the two versions of Kinect sensors and the underlying depth sensing technologies. Section 3 reviews various applications of the Microsoft Kinect sensor. Section 4 surveys the models and algorithms used for recognizing human gestures and activities with Kinect. Section 5 compiles a list of publicly available Kinect datasets. Finally in section 6, we conclude the article.

2. The Kinect Technology

The Kinect sensor, together with the Microsoft Kinect SDK, or a third-party software toolkit such as OpenNI (http://structure.io/openni), provides a user with several streams of information. The most common streams include:

- A stream of 2D color image frames.
- A stream of 3D depth image frames.
- A stream of 3D skeletal frames for at least one human subject in the view. A skeletal frame may contain the 3D position information for various number of joints.

The availability of the skeletal frames with extensive joint 3D positions has greatly facilitated Kinect application development because it frees the application developers from dealing with the complicated task of human pose estimation. Nevertheless, the availability of the RGB frames and the 3D depth frames facilitates researchers and software developers to perform their own pose estimation instead of using the built-in skeletal frames provided by the SDK. As shown in Figure 1, in general, to develop a useful Kinect application involving human motion recognition, the following steps are typically needed:

- Human skeleton estimation. A version of skeleton estimation is incorporated in the Kinect SDK and one can obtain a stream of skeleton frames directly. To accomplish realtime human skeleton estimation, a model is developed to represent a full human skeleton. Then, the model is trained with extensive labeled data. The trained model is incorporated in the SDK for realtime skeleton tracking. The following shows the main steps for human skeleton estimation:
  - Retrieve the stream of depth frames containing one or more human subjects.
  - Perform human subject foreground extraction (i.e., background subtraction) and detection from depth frames.
  - Match the extracted human subject against the trained model to estimate the current pose.
  - Infer the skeleton joint positions once the current pose is estimated and subsequently refined.
Motion recognition. In this step, the semantics of the activity or gesture formed by the motion is recognized.

Feedback to users and/or actions triggered by the detection of the particular motion.

Kinect has gone through two versions so far. The first version of the Kinect sensor (referred to as Kinect v1) was released for the Xbox 360 game console in November 2010. A minor revised version, called Kinect for Windows, was released for application development on computers in February 2012. The Kinect for Windows sensor offers a near mode for depth sensing, but all other hardware specification remains the same as the original Kinect sensor. The second version of Kinect (referred to as Kinect v2) was officially released in summer 2014. Kinect v2 uses a completely different depth sensing technology and offers much improved depth sensing accuracy as well as color image resolution. The main features for the two versions of the Kinect sensor are summarized in Table 1.

The depth-sensing technology used in Kinect v1 was developed by PrimeSense. The depth of each pixel is calculated by a form of triangulation. Normally, two cameras are needed to facilitate the triangulation calculation. Instead, in Kinect v1, a structured light method is used to enable the use of a single infrared (IR) emitter and a single depth sensor to calculate the depth of each pixel. As shown in Figure 2, the IR emitter beams structured light with predefined pat-
Table 1. Comparison of main features of the two versions of the Kinect sensor.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Kinect v1</th>
<th>Kinect v2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth Sensing Technology</td>
<td>Triangulation with structured light</td>
<td>Time of flight</td>
</tr>
<tr>
<td>Color Image Resolution</td>
<td>640x480 30fps</td>
<td>1920x1080 30fps (12fps low light)</td>
</tr>
<tr>
<td>IR Image Resolution</td>
<td>640x480 30fps</td>
<td>512x424 30fps</td>
</tr>
<tr>
<td>Depth Sensing Resolution</td>
<td>640x480 30fps</td>
<td>512x424 30fps</td>
</tr>
<tr>
<td></td>
<td>320x240 30fps</td>
<td></td>
</tr>
<tr>
<td></td>
<td>80x60 30fps</td>
<td></td>
</tr>
<tr>
<td>Field of View</td>
<td>43° vertical 57° horizontal</td>
<td>&gt; 43° vertical 70° horizontal</td>
</tr>
<tr>
<td>Depth Sensing Range</td>
<td>0.4m - 3m (near mode)</td>
<td>0.5m - 4.5m</td>
</tr>
<tr>
<td></td>
<td>0.8m - 4m (default mode)</td>
<td>Up to 8m without skeletonization</td>
</tr>
<tr>
<td>Skeleton Tracking (with full skeleton)</td>
<td>Up to 2 subjects</td>
<td>Up to 6 subjects</td>
</tr>
<tr>
<td>Built-in Gestures</td>
<td>None</td>
<td>Hand state (open, close, lasso) Hand pointer controls; lean</td>
</tr>
<tr>
<td>Unity Support</td>
<td>Third party</td>
<td>Yes</td>
</tr>
<tr>
<td>Face APIs</td>
<td>Basic</td>
<td>Extended massively</td>
</tr>
<tr>
<td>Runtime Design</td>
<td>Can run multiple Kinect sensors per computer; One app per Kinect</td>
<td>At most one Kinect per computer; Multiple apps share same Kinect</td>
</tr>
<tr>
<td>Windows Store</td>
<td>Cannot publish to</td>
<td>Yes</td>
</tr>
</tbody>
</table>

By observing the unique pattern, the depth sensor can infer the line from the IR emitter to the pixel with the pattern, hence, the depth sensor can calculate the vertical distance between the IR emitter-depth sensor line to the pixel using trigonometry, which is the depth reading of the pixel. While this is a very clever scheme, the fidelity of the depth measurement is quite low because for the depth sensing to work perfectly, there has to be a visible unique pattern for each pixel. Because there has to be some space between two adjacent dots as part of the structured light and this space has to be wide enough for the depth sensor to distinguish, only about 1 in every 20 pixels has a true depth measurement in typical situations and the depths for other pixels must be interpolated. Hence, the depth sensing resolution is actually significantly below the nominal 640x480 for Kinect v1. Furthermore, due to the use of IR light patterns, the depth sensing fidelity may also be compromised in the presence of strong light.

The depth-sensing technology used in Kinect v2 is completely different and the depth is calculated based on time of flight. It appears that the technology is based on that developed by Canesta, which was acquired by Microsoft in late 2010. As shown in Figure 3, Kinect v2 is also equipped with an IR emitter and a depth sensor. However, the IR emitter consists of a IR laser diode to beam modulated IR light to the field of view. The reflected light will be collected by the depth sensor. A timing generator is used to synchronize the actions of the IR emitter and the depth sensor. The depth of each pixel can be calculated based on the phase shift between
Fig. 2. Kinect v1 uses the structured light triangulation method for depth sensing.

Fig. 3. Kinect v2 uses the time-of-flight method for depth sensing.

the emitted light and the reflected light. A clever design in Kinect v2 is that the IR emitter is periodically turned on and off, and the output from the depth sensor is sent to two different ports when the light is on and off, respectively. Let the output when the light is on be A, and the output when the light is off be B. A contains the light from both the emitted IR light and ambient light (i.e., the light already in the field of view), and B contains only the ambient light. Hence, subtracting B from A (i.e., A-B) gives only the reflected modulated light from the IR emitter, which can be used to calculate the depth accurately. Furthermore, the magnitude of (A-B) gives a high quality IR image without ambient lighting. This design makes Kinect v2 produce much better IR images and depth images, as shown in Figure 4.

Another major change is the Kinect runtime design. For Kinect v1, a Kinect sensor is exclusively used for a single application, and multiple Kinect sensors can be used and controlled by a single application. The application has full control over the Kinect sensors it connects to, including various settings on the resolution in color
Fig. 4. Comparison of the IR and depth image quality between Kinect v1 and Kinect v2. The
two sets of images were taken with the maximum resolutions and with the same Kinect-to-user
distance. As can be seen, not only the quality of Kinect v2 images is obviously higher, the field of
view is wider as well.

and depth frames. However, it is not possible for different applications to share the
same Kinect sensor on the same computer. For Kinect v2, the Kinect runtime is
elevated to a system-level service to facilitate the use of the various data collected
by the Kinect sensor by multiple different applications. As a tradeoff, an application
can no longer choose the resolution settings and at most one Kinect sensor can be
used at a computer (i.e., one cannot connect two or more Kinect v2 sensors to
the same computer). Furthermore, due to the maturity of the Kinect technology,
developers can now publish Kinect v2 applications to the Windows store. Kinect
v2 SDK also provides official support for Unity, which is a development platform
for 3D games.

Finally, Kinect v2 SDK provides a tool to develop gesture recognition based on
machine learning. The details of the algorithms used will be elaborated in Section 4.
The availability of this tool might greatly facilitate the development of gesture-based
Kinect v2 applications.
3. Applications of the Kinect Technology

Microsoft Kinect was originally released exclusively for the Xbox gaming and entertainment consoles. It allows Xbox game players to interactively control the console through body gestures and voice commands without using any other peripheral equipment. With the introduction of the free Microsoft Kinect SDK in 2011, the Kinect technology opened a huge door for developing other applications beyond Xbox games. Figure 5 shows major Kinect application categories, spanning from healthcare, to education, retail, training, gaming, robotics control, natural user interface, sign language recognition, as well as 3D reconstruction, which is a great fit for the 3D printing revolution. Except for 3D reconstruction applications, virtually all other Kinect applications require motion recognition so that the semantics of a human gesture or action can be interpreted automatically. Hence, motion recognition is the fundamental enabling technology, which we will review in the next Section in detail. Among all these applications, healthcare related applications have attracted the most research and development effort.

3.1. Healthcare Applications

In this section we review the applications of the Kinect technology in healthcare. We focus on physical rehabilitation exercises, medical operating room assistance, and fall detection and prevention. The summary of the literatures reviewed is given in Table 2.
Table 2. Summary of Kinect applications in healthcare.

<table>
<thead>
<tr>
<th>Applications</th>
<th>Main Contributions</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Therapy and Rehabilitation</td>
<td>Assessment of Kinect motion tracking accuracy for healthcare applications</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>An interactive game-based rehabilitation tool for adults with neurological injury</td>
<td>67, 68</td>
</tr>
<tr>
<td></td>
<td>Full-body control in virtual reality applications</td>
<td>118</td>
</tr>
<tr>
<td></td>
<td>Integration of Kinect and a smart glove for patients with upper extremity impairment</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>An interactive rehabilitation system for disabled children</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>A rehabilitating program for young adults with motor impairments</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Cognitive rehabilitation for Alzheimer’s patients using a Kinect-based game</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>A Kinect-based game for stroke rehabilitation</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td>An exercise rehabilitation program for individuals with spinal cord injury</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>Integration of Kinect with rehabilitation robotics</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>Integration of inertial sensors with Kinect</td>
<td>14, 149</td>
</tr>
<tr>
<td>Medical Operating Room Assistance</td>
<td>A Kinect-based intra-operative medical image viewer</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>A system that enables touchless controlling of medical images with hand and arm gestures</td>
<td>42</td>
</tr>
<tr>
<td>Fall Detection and Prevention</td>
<td>A real-time fall monitoring and detection system</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>Overcoming occlusions for human body fall detection</td>
<td>104</td>
</tr>
<tr>
<td></td>
<td>Fall detection based on Kinect skeletal data</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Human fall detection using two Kinect</td>
<td>147</td>
</tr>
<tr>
<td></td>
<td>Capturing variations of stride-to-stride gait</td>
<td>114, 113</td>
</tr>
<tr>
<td></td>
<td>for elderly adults</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Detecting falls and other abnormal events on stairs</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>Fall prevention in hospital ward environment</td>
<td>88</td>
</tr>
</tbody>
</table>

3.1.1. Physical Therapy and Rehabilitation

Traditional physical therapy rehabilitation training programs typically involve extensive, repetitive range-of-motion and coordination exercises, and require medical professionals to supervise patients’ movements and assess their progress. In order to meet increasing demands and reduce the cost, physical therapy and rehabilitation providers are looking for computer technology that can assist them to provide services to patients in an affordable, convenient and user-friendly way. An essential requirement for such technology is that it helps patients learn and perform preventive and rehabilitative movement patterns repetitively and correctly.

Human motion recognition technologies have been used to monitor physical rehabilitation exercises and other patients’ activities long before the release of Microsoft Kinect. However, most of them rely on motion tracking tools that are intrusive because patients either have to wear markers or attach inertial sensors. With the introduction of Kinect, it is possible to provide markerless full-body tracking.
A research group at the University of Southern California Institute for Creative Technologies (ICT) has successfully used Kinect to develop a virtual reality simulation technology for clinical purposes. Through extensive evaluation, assessment, and analysis, researchers at ICT have proved that the Kinect technology can make a major contribution to the quality of traditional intervention training programs that specialize in mental health therapy, motor skills rehabilitation, cognitive assessment and clinical skills training.19,53,67,68,118

Chang et al. presented a comprehensive assessment of using Kinect for motion tracking with an OptiTrack optical system as comparison.19 The experimental results show that Kinect can achieve competitive motion tracking performance compared to the OptiTrack system, and it can also provide “pervasive” accessibility to patients so that they can take rehabilitation treatment in clinic and as well as in home environments.

Lange et al. developed and assessed an interactive game-based rehabilitation tool for balance training for adults with neurological injuries.67 Furthermore, Lange developed a video game called “JewelMine” to use in balance training.68

Suma et al. developed the Flexible Action and Articulated Skeleton Toolkit (FAAST) to facilitate the integration of full-body control with virtual reality applications and video games using Kinect.118

Huang et al. designed a motion and angle extraction device for patients with upper extremity impairment by integrating Kinect and a smart glove.53 This approach overcomes the limitation of Kinect when testing subjects who are out of camera range or whose upper extremities are occluded by their body.

Rahman et al. presented an interactive rehabilitation system for disabled children.2 This system utilizes Kinect to record rehabilitation exercises performed by a physiotherapist or a disabled child. The exercise session can be synchronously played in an on-line virtual system, which provides patients with visual guidance for performing correct movements.

Chang et al. described a study that assesses the possibility of rehabilitating two young adults with motor impairments using a Kinect.20

Cervantes et al. presented their work on cognitive rehabilitation for Alzheimer’s patients using a Kinect-based video game.21 The interactive body motion controlled game increases patients’ motivation to participate in exercises. In a similar research work, Saini et al. aimed at increasing patients’ motivation for therapy using a Kinect-based game for stroke rehabilitation.107 They studied the feasibility and effect of new game technology to improve the accuracy of stroke exercises for hand and leg rehabilitation. Gotsis et al. demonstrated a mixed reality game for upper body exercise and rehabilitation using Kinect.36

Pedro et al. proposed to use Kinect in conjunction with rehabilitation robotics.94 The benefit of combining the Kinect device with a robot is the reduction of hardware cost when multiple cameras are needed to overcome occlusions.

Although Kinect has been proven to be a good replacement for the commonly used inertial sensor for tracking human body motion, the combination of using
both devices may achieve more satisfying results. Bo et al. proposed a method to combine Kinect with portable sensors, such as accelerometers, gyroimeters, and magnetometers, for measuring human motion for rehabilitation purposes. In this study, Kinect was used to temporarily correct the overall estimate and to calibrate the inertial sensors for long-term operations.

Providing interactive feedbacks to patients is an important requirement for rehabilitation exercise monitoring applications. During rehabilitation training, patients are required to perform an exercise in a specific manner to meet the objectives of rehabilitation. The output of human motion recognition should be presented as feedbacks to patients in realtime to inform them about any incorrect movement.

Velloso et al. developed a system aimed to facilitate at-home rehabilitation exercise monitoring. The system employed a kinematic model to identify static and dynamic axis in a prescribed exercise. The model parameters are automatically fitted using an exemplar. This finalized model enables the system to continuously monitor violations of static axes in realtime, and to count the repetitions for dynamic joints.

Su developed a similar system. However, fuzzy logic is employed to capture the clinician’s subjective requirements on performing the exercises. After the gesture recognition step on individual features, a single score is given to the patient after combining the fuzzy rules.

In our recent work in Ref. 149, we proposed a set of extensive/adaptable rules (with error bounds to capture the fuzziness of requirements) for each exercise, such that specific feedback on rule violations can be provided directly to the patients. Unlike playing games, which a user normally only expects a total score, a patient

Fig. 6. An at-home exercise monitoring system with Kinect.
who is carrying out a rehabilitation exercise expects to be informed exactly what is not done right to ensure proper recovery. The proposed rules have been incorporated into an at-home exercise monitoring system. The system has a 3D user interface implemented using Unity3D. As shown in Figure 6, the left side of the user interface shows a 3D avatar demonstrating the correct movement, while the right side of the interface mirrors the patient’s movement. A visual target ball for the leg in a hip abduction exercise is provided. The target ball changes color depending on whether or not any correctness rule is violated. If the patient does one iteration correctly, the ball shows the green color. Otherwise, it shows yellow and the specific rule that is violated is displayed in text on the interface. The target ball also shows the repetition count for correct iterations.

3.1.2. Medical Operating Room Assistance

The growing use of advanced imaging devices and image guided procedures in surgical settings has imposed an increasing need for interaction under high sterile conditions between medical professionals and images. Kinect provides an appealing opportunity to control medical images or image-guided devices without touching. Some researchers have developed Kinect-based gesture recognition systems to address the needs in medical surgical rooms. Bigdelou et al. developed a Kinect-based intra-operative medical image viewer for use in a surgical environment. The system incorporates a gesture recognizer based on kernel regression such that both the categorical information and the state of the gesture can be recognized. A doctor could manipulate a medical image without touch using the system during a surgery, such as zooming in, moving the image around, add a label at the specific place in the image, as shown in Figure 7.
Gallo et al. developed a Kinect-based open-source system that allows interactive exploration of medical digital images, such as CT, MRI, or PET in operating rooms. The interface utilizes hand and arm gestures to execute basic tasks such as image selection, zooming, translating, rotating and pointing, and some complex tasks such as the manual selection and extraction of a region of interest as well as interactive modification of the transfer function used to visualize medical images.

3.1.3. Fall Detection and Prevention

Kinect has been used to detect and prevent falls and other dangerous activities for elderly people in a number of studies. Bian et al. presented an approach to detecting falls by extracting skeleton data from Kinect depth images based on the fast randomized decision forest algorithm. This algorithm produces more accurate detection by properly rotating frames to match human orientation.

Mastorakis et al. introduced a Kinect-based real-time fall monitoring and detection system that can automatically detect a range of falls including backward, forward and sideways, without pre-knowledge of the floor plane coordinates or pre-defined particular body parts.

Ni et al. developed a Kinect-based system to prevent potential falls in the hospital ward environments. This system automatically detects the event of patient getting up out of a bed. The nursing staff is alarmed immediately to provide assistance once the getting up event is detected. The detection algorithm combines multiple features from multiple modalities via an MKL framework to achieve high accuracy and efficiency.

Parra-Dominguez et al. proposed a method to detect falls and other abnormal events on stairways instead of at flat level using Kinect. This method automatically estimates walking speed and extracts a set of features that encode human motion during stairway descent.

Rougier et al. introduced an approach to addressing the occlusion issue in detecting human body falls using Kinect. The method is based on human centroid height relative to the ground and body velocity. With the help of computing 3D personal velocity just before occlusion occurs, this method can accurately detect falls by measuring human centroid height, as the vast majority of falls end on the ground or near the ground.

Stone and Skubic developed a Kinect-based system for capturing the variations of stride-to-stride gait in home environments for elderly adults. By measuring the changes in gait, falls can be predicted. If the motion of joints of a specific body subject is detected in an unusual time sequence, a prevention message is generated as a caution.

Zhang et al. proposed a viewpoint-independent statistical method for fall detection. They used 5 features, including the duration of a fall in frames, the total head drop during the fall, the maximum speed of the fall, the smallest head
height (after the fall has happened), and the fraction of frames where head drops. The fall probability was calculated using the Bayes rule.

3.2. Virtual Reality and Gaming

The innovative human motion tracking and recognition technology enabled by Kinect allows users to interact with augmented objects freely in real-time for computer-based virtual reality applications and augmented reality games. The list of work in this area is summarized in Table 3.

Aitpayev et al. applied the Kinect technology to make the human body a physical part of augmented reality for interaction without wearing a special suit with infrared LEDs or attaching special markers. Furthermore, they presented two methods, the Ragdoll method and the rotation angle method, for animating the collision of objects in real-time.

Tong et al. studied computational algorithms to calculate joint rotation angles from Kinect for skeleton animation. The resulting data is saved in the Biovision Hierarchy character animation file format, which can be represented in future study and analysis.

Franke et al. proposed mathematical foundations for using Kinect depth images in mixed reality applications. Tong and his colleagues demonstrated a system that can scan 3D full human body shapes by using multiple Kinect devices in a more convenient way.

Kinect has also been widely used in augmented reality games. A computer game developed by Nakachi et al. can express “individuality” in their proprietary software package using Kinect. Hai et al. developed an interaction system for treadmill games based on Kinect depth maps. The HoloDesk is an interactive augmented reality system combining an optical see-through display and a Kinect to create the illusion that users can directly interact with 3D graphics. The Wizard-of-Oz is a guessability game to examine child-defined gestures using Kinect. This game can simulate on-screen whole-body interaction for prototyping touch-free interactive games for children. There is also research on reducing the volume of data transferred over the network for cloud-based games using Kinect.

3.3. Natural User Interface

A natural user interface refers to a new type of human computer interface where a user can command and interact with a computer naturally using hand gestures or body poses, as well as voice commands, instead of a keyboard or mouse pointing device. Kinect is a critical enabling device for the development of various novel natural user interfaces. The literature on Kinect-based natural user interface development is summarized in Table 4.

Farhadi-Niaki et al. proposed an input system using Kinect for performing typical desktop tasks through arm gestures. The system shows that gestures are more natural and pleasant to use than a mouse and a keyboard.
Table 3. Summary of Kinect applications in virtual reality and gaming.

<table>
<thead>
<tr>
<th>Applications</th>
<th>Main Contributions</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>VR Foundamentals</td>
<td>A markerless virtual reality system</td>
<td>4</td>
</tr>
<tr>
<td>VR Foundamentals</td>
<td>A computational algorithm for skeleton animation</td>
<td>123</td>
</tr>
<tr>
<td>VR Foundamentals</td>
<td>Mixed reality applications based on Kinect depth imaging</td>
<td>39</td>
</tr>
<tr>
<td>VR Foundamentals</td>
<td>3D full human body scan using multiple Kinect</td>
<td>122</td>
</tr>
<tr>
<td>VR Games</td>
<td>A game expressing “individuality”</td>
<td>86</td>
</tr>
<tr>
<td>VR Games</td>
<td>A treadmill game based on Kinect depth maps</td>
<td>49</td>
</tr>
<tr>
<td>VR Games</td>
<td>A HoloDesk game combining an optical display &amp; Kinect</td>
<td>51</td>
</tr>
<tr>
<td>VR Games</td>
<td>A game examining child-defined gestures</td>
<td>27</td>
</tr>
<tr>
<td>VR Games</td>
<td>A cloud-hosted Kinect-based game</td>
<td>90</td>
</tr>
</tbody>
</table>

Table 4. Summary of Kinect applications in natural user interface.

<table>
<thead>
<tr>
<th>Applications</th>
<th>Main Contributions</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural User Interface</td>
<td>Performing desktop tasks via arm gestures</td>
<td>36</td>
</tr>
<tr>
<td>Natural User Interface</td>
<td>3D object manipulation on a desktop display</td>
<td>97</td>
</tr>
<tr>
<td>Natural User Interface</td>
<td>3D control method based on Kinect</td>
<td>61</td>
</tr>
<tr>
<td>Natural User Interface</td>
<td>A group meeting application based on Kinect</td>
<td>17</td>
</tr>
<tr>
<td>Natural User Interface</td>
<td>Controlling virtual globes via Kinect</td>
<td>15</td>
</tr>
<tr>
<td>Natural User Interface</td>
<td>Web browsing via natural user interfaces</td>
<td>71</td>
</tr>
<tr>
<td>Natural User Interface</td>
<td>Automatic camera control based on Kinect</td>
<td>136</td>
</tr>
</tbody>
</table>

Raj et al. presented a different approach to 3D object manipulation on a desktop display.⁹⁷ They examined a number of aspects of this approach: (1) the advantage in response time for the self-avatar versus the generic sphere display as a representation of the user’s rotational device; (2) the differences in the user’s preference to either use an arm gesture or a wrist rotation to manipulate objects; and (3) whether the gender and/or gaming experience would influence task performance.

Kang et al. conducted a study on the control method of 3D applications using Microsoft Kinect.⁶¹ They introduced a control method that naturally regulates the use of distance information and joints location information. They showed that the recognition rate using the natural user interface with Kinect is 27% better than that using a mouse.

Francese et al. presented a 3D navigation user interaction application.³⁸ The proposed system allows users to interact with desktop computers via new forms of natural interfaces and new actions. The system is specifically designed for 3D gestural user interaction on 3D geographical maps.

A Code Space software application developed by Bragdon et al. combines touch and air gesture hybrid interactions to access, control, and share information through different hardware devices for a group meeting.¹⁷ The devices include multi-touch screens, mobile touch devices, and Microsoft Kinect sensors.
Boulos et al. developed an application called “Kinoogle” to control virtual globes, such as Google Earth, Bing Maps 3D, and NASA World Wind using Kinect. The Kinoogle allows the user to control Google Earth through a series of hand and full-body gestures.

Liebling et al. introduced a “Kinected Browser” for Web browsing through touch-free technologies. The developed toolkit can be used to augment web pages with speech input and gesture input via Kinect, it is designed to enable Web interactions for new form-factors such as large display walls, and TV sets.

Winkler et al. introduced a low-cost, non-intrusive solution for automatic camera control for tracking a presenter during a talk using Kinect. The approach enables video cameras to automatically follow a presenter on different premises with different geometries.

3.4. Education and Performing Arts

In education, especially K-12 education, the natural user interface enabled by Kinect offers an opportunity to engage students in a new level. Similarly, it also enables a new powerful way of teaching and assessing the quality of performing arts. The literature reviewed in this section is summarized in Table 5.

<table>
<thead>
<tr>
<th>Applications</th>
<th>Main Contributions</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Education</td>
<td>A classroom teaching system with Kinect</td>
<td>129</td>
</tr>
<tr>
<td>and</td>
<td>A interactive music conductor generation system with Kinect</td>
<td>23</td>
</tr>
<tr>
<td>Performing Arts</td>
<td>A puppetry control application with Kinect</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>A MotionDraw tool for enhancing art performance</td>
<td>102</td>
</tr>
</tbody>
</table>

Villaroman et al. proposed a classroom teaching system that uses Kinect for classroom instruction on natural user interaction. Examples are presented to demonstrate how Kinect-assisted instruction can be utilized to accomplish adequate and beneficial learning results in Human Computer Interaction courses.

Chen et al. proposed an interactive music conductor generation system. It allows the music to be arranged under the human music conductor’s hand gestures in real-time.

Leite et al. developed a puppetry control application through body motion with Kinect. The animating shadow puppets are controlled by the virtual silhouette instead of pulling strings or handling rods.

Rodrigues et al. introduced a tool called “MotionDraw” for enhancing art performance using Kinect. This tool can track live movements of users and enable artists, performers, dancers and the audience to design, create and control hybrid digital performances.
3.5. **Robotics Control and Interaction**

Kinect has also been used to control robots. In recent studies, traditional robotics controlling sensors including laser, ultra-sonic and radar sensors, have been either directly replaced by or integrated with Kinect. In this section we review the applications of the Kinect technology in navigating and controlling mobile devices, interactively controlling humanoid robots, and remotely controlling robotic devices. The summary of the review is given in Table 6.

<table>
<thead>
<tr>
<th>Applications</th>
<th>Main Contributions</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robotics</td>
<td>Robot navigation using Kinect and inertial sensors</td>
<td>33</td>
</tr>
<tr>
<td>Navigation &amp; Control</td>
<td>Feasibility on using gestures to control industrial robots</td>
<td>52</td>
</tr>
<tr>
<td>Interactively</td>
<td>Navigating a robot using hand gestures</td>
<td>140</td>
</tr>
<tr>
<td>Controlling Robotics</td>
<td>A human-robot interactive demonstration system</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>with a gesture recognizer</td>
<td></td>
</tr>
<tr>
<td>Robotics</td>
<td>An athletic training speed skating system using Kinect</td>
<td>16</td>
</tr>
<tr>
<td>Remote Control</td>
<td>A Kinect on-board system that enables the control</td>
<td>76</td>
</tr>
<tr>
<td></td>
<td>of velocity and attitude of a mobile robot</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Controlling altitude of a quadrotor helicopter via Kinect</td>
<td>115</td>
</tr>
<tr>
<td></td>
<td>A real-time human imitation system for robotics</td>
<td>131</td>
</tr>
<tr>
<td></td>
<td>Tele-operating a humanoid robot using Kinect</td>
<td>150</td>
</tr>
</tbody>
</table>

3.5.1. **Navigating and Controlling Robotics**

Humanoid robots have gradually entered our life in many ways, performing house chores, assisting elderly people, providing education, and completing tasks in severe conditions. Researchers now face a challenge of how to naturally navigate this human-body shape device effectively without using wearable devices. The emerging Kinect technology provides an ideal interface to accomplish such a task.

El-Iaithy *et al.* developed an application to navigate an indoor robot. They integrated Kinect with inertial sensors to optimize indoor navigation, particularly for obstacle detection and avoidance. Through the experiments conducted in both indoor and outdoor environments, it shows that Kinect is ideal for indoor robotic applications, but not suitable for outdoor applications or when the robot is under strong lighting sources. In addition, it also shows that Kinect cannot detect glass or transparent plastic well because the IR light from IR emitter is refracted and therefore is not able to enable proper depth estimation.

Hoilund *et al.* evaluated the feasibility of using gestures to control industrial robots via Kinect. Such a technique can enhance a mobile robot with the ability to interpret human actions, so that the robot can be controlled through human actions. The experimental results show that Kinect data is more noisy than more
expensive motion capture systems. However, the authors believe that the quality of the data is sufficient for action recognition using parametric hidden Markov models.

Nguye et al. presented a human imitation system that can map different kinematic structures. The objective of this system is to reproduce imitated human motions during continuous and online observation with a humanoid robot using Kinect. Using straightforward geometry and clavicle, the proposed method requires less time for computation of the kinematics. The experimental results show that this system could feasibly adjust the robot motions to satisfy the mechanical constraints and dynamics consistency.

3.5.2. Interactively Controlling Robotics

Recent advances in the Kinect technology on human computer interaction make it attractive to use Kinect for interactively controlling humanoid robotics.

Xu et al. proposed a system that can navigate a robot using dynamic hand gestures in real-time via Kinect. The system recognizes human hand gestures based on a Hidden Markov Model and converts them to control commands for the robot. Seven hand gestures are used to sufficiently navigate the robot and experiments show that the proposed system can work effectively in the complex environment with an average real-time recognition rate of 98.4%. Furthermore, the robot navigation experiments show a high robustness of human-robot interaction in real-world scenarios.

Cheng et al. developed a human-robot interactive demonstration system. The core of the system is a body gesture recognizer. The recognizer provides a visual interpretation of gestures and sends it to robots to enable natural interaction between a human and a robot. The prototype system was built with a NAO humanoid robot, a Kinect sensor, and a computer. The human gesture modules are loaded into NAO’s behavior manager to simplify the control process.

3.5.3. Remote Control

The Kinect technology has also been used to remotely control robotic devices in a number of applications, such as a mobile robotic motion capture system, a mobile robot tracking system, and a quadrotor helicopter controller.

Boyd et al. developed an athletic training speed skating system by placing Kinect on a mobile robotic platform to capture motion in situ. The system can follow a speed skater on the ice to capture the full-body motion. The moving robotic platform addresses the limited viewing area of Kinect and provides a visual guidance for athletes.

Machida et al. introduced a tracking control system with Kinect on-board of a mobile robot. The human gestures obtained via Kinect are used to control the velocity and altitude of the mobile robot. The Kalman filter algorithm is used to reduce the noise and to estimate the human’s motion state. The experiments
show that the estimation and tracking are effective and the image processing is adequately fast.

Stowers et al. used Kinect’s depth images to control the altitude of a flying quadrotor helicopter. The proposed system is capable of maintaining a steady altitude during flight of the quadrotor helicopter in dynamic environments via a special calibration process. The system demonstrates that Kinect is an attractive motion sensing device for use on real-time robotic platforms due to its low cost, sufficient frame rate and depth sensing accuracy.

Wang et al. developed an application using Kinect and the Aldebaran NAO humanoid robot. The motion data captured via Kinect is transmitted to the NAO robot wirelessly, in which the data is further processed for controlling the joints of the robot. The testing results show that the system is robust and flexible enough to imitate various human motions.

Zuher et al. showed that a humanoid robot can be tele-operated through the recognition of human motions (such as neck, arms, and leg movements) using Kinect. The proposed system focuses on two major tasks: a real-time imitation of human movements and the recognition of such movements to make the robot perform. The evaluation results from teleoperation show that the average accuracy among the four aspects is 80%.

3.6. Speech and Sign Language Recognition

Visual automatic speech recognition has significant impact on our society. Taking advantage of the Kinect technology, researchers extended existing work to use depth information for improving the robustness of speech recognition. Galatas et al. incorporated facial depth data of a speaker as a third data stream in an audio-visual automatic speech recognizer. The results demonstrate that the system performance is improved due to the depth modality, and the accuracy is increased when using both visual and depth modalities over audio-only speech recognition.

Agarwal and Thakur developed a method to recognize sign language gestures using the Kinect depth frames. Depth and motion profiles are extracted from the Kinect depth frames and used to build a feature matrix for each gesture. The support vector machine (SVM) classifier is used for recognition. The Chinese Number Sign Language dataset from the ChaLearn Gesture Dataset was used in their experiments.

Almeida et al. presented a methodology to extract features in Brazilian Sign Language for recognition based on phonological structure. This structure consists of the configuration, movement, and orientation of the hand, as well as the articulation points, which represent the location of the sign. For sign recognition, SVM is used as the classifier.

Anjo et al. developed a system to recognize static gestures representing 10 letters in the Brazilian Sign Language in real-time. The hand shape, which is represented as a 25x25 binary image, is used as the feature vector, and the multi-layer percep-
Table 7. Summary of Kinect applications in speech and sign language recognition. In the table, ASL refers to American Sign Language, Libras refers to the Brazilian Sign Language, CSL refers to Chinese Sign Language, GSL refers to German Sign Language, ISL refers to Indian Sign Language, PSL refers to Pakistani Sign Language, PSL1 refers to Polish Sign Language, PSL2 refers to Portuguese Sign Language, SIBI refers to the Sign System for Indonesian Language, TSL refers to Taiwanese Sign Language, and TSL1 refers to Turkish Sign Language.

<table>
<thead>
<tr>
<th>What is Recognized</th>
<th>Feature Set</th>
<th>Recognition Methods</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speech</td>
<td>Face and mouth</td>
<td>HMM</td>
<td>41</td>
</tr>
<tr>
<td>10 numbers in CSL</td>
<td>Depth and motion profiles</td>
<td>SVM</td>
<td>3</td>
</tr>
<tr>
<td>34 signs in Libras</td>
<td>Phonological structure</td>
<td>SVM</td>
<td>7</td>
</tr>
<tr>
<td>10 words in Libras</td>
<td>hand shape</td>
<td>MLP</td>
<td>8</td>
</tr>
<tr>
<td>10 signs in ISL</td>
<td>hand position and trajectory</td>
<td>Direct comparison</td>
<td>43</td>
</tr>
<tr>
<td>20 words in CSL</td>
<td>Position and trajectory of right hand</td>
<td>ELM and SVM</td>
<td>44</td>
</tr>
<tr>
<td>34 words in CSL</td>
<td>hand trajectories and hand shapes</td>
<td>Sparse coding</td>
<td>57</td>
</tr>
<tr>
<td>25 signs in GSL</td>
<td>9-dimension joints based on Kinect joints</td>
<td>HMM</td>
<td>66</td>
</tr>
<tr>
<td>25 words in TSL</td>
<td>hand positions, movement, and shapes</td>
<td>SVM</td>
<td>69</td>
</tr>
<tr>
<td>4 signs in PSL and 3 generic signs</td>
<td>Kinect joints</td>
<td>DTW</td>
<td>80</td>
</tr>
<tr>
<td>111 words in TSL1</td>
<td>DCT coefficients</td>
<td>KNN clustering</td>
<td>83</td>
</tr>
<tr>
<td>30 words in PSL1 isolated words</td>
<td>Kinect joints and hand shapes from color images</td>
<td>DTW and clustering</td>
<td>91</td>
</tr>
<tr>
<td>ASL alphabet</td>
<td>Hand shapes</td>
<td>Random forests</td>
<td>96</td>
</tr>
<tr>
<td>10 SIBI words</td>
<td>Kinect joint orientation and features from depth images</td>
<td>GLVQ and Random Forest</td>
<td>98</td>
</tr>
<tr>
<td>24 ASL letters</td>
<td>Hand features from color and depth images</td>
<td>Deep believe network</td>
<td>101</td>
</tr>
<tr>
<td>PSL2 alphabet</td>
<td>Hand angular pose</td>
<td>Direct comparison using 3D voxel occupancy</td>
<td>124</td>
</tr>
<tr>
<td>150 gestures</td>
<td>Arm postures and finger-related features</td>
<td>Clustering and HMM</td>
<td>128</td>
</tr>
<tr>
<td>19 ASL words</td>
<td>hand shapes, motion, and pose</td>
<td>HMM</td>
<td>142</td>
</tr>
</tbody>
</table>

Geetha *et al.* reported a method to recognize Indian Sign Language gestures.\(^{43}\) To increase the recognition accuracy, both the local feature, based on a set of 7 key points of the hand, and the global feature, based on the hand trajectory, are used. The recognition of the gestures is based on direct comparison and the similarity between the testing gesture and the template is calculated using the Euclidean distances of feature vectors.

Geng *et al.* proposed an approach to the recognition of 20 Chinese Sign Language gestures based on both the position and trajectory of the right hand using the...
extreme learning machine (ELM) as a classifier. Their experiments show that by combining the features, the recognition accuracy is improved. Furthermore, ELM, which is a variation of neural networks, is shown to perform better than SVM on recognition accuracy.

Jiang et al. developed another approach to the recognition of 34 Chinese Sign Language gestures. The feature vector is based on the combination of hand trajectories and hand shapes. A sparse coding based method is used for gesture recognition. Furthermore, the proposed method was validated with 8 human subjects and the recognition is consistently at close to or better than 90%.

Lang et al. developed a system for recognizing 25 signs of the German Sign Language using Hidden Markov Model (HMM). A 9-dimension feature vector, which is derived from the left/right hand, neck, and right elbow, is used to train the model and for recognition. Their system also allows the definition of new signs.

Lee et al. presented a Kinect-based system to recognize 25 words in the Taiwanese Sign Language. Hand positions, hand moving directions, and hand shapes are used as the feature set, and SVM is used as the classifier for sign recognition.

Massod et al. developed a system for sign language translation based on dynamic time warping (DTW). The system consists of two modes of operations, the recording mode and the translation mode. The recording mode enables the recording of a predefined sign gesture, and during the translation mode, the current gesture is compared with the recorded gesture for recognition using DTW. Eight joint positions from the Kinect skeletal data are used for the comparison. Seven gestures were experimented with, among which, four are from the Pakistani Sign Language, and three are generic signs.

Memes and Albayrak reported a system for the recognition of 111 words in the Turkish Sign Language (with 1002 dynamic signs) using spatio-temporal features and the K-Nearest Neighbor (KNN) clustering classifier. The spatio-temporal features are obtained by applying a 2D discrete cosine transform (DCT) to the accumulated Kinect color and depth images.

Oszust and Wysocki studied the Polish sign gesture recognition problem by experimenting with several clustering algorithms and two different feature sets. One set is entirely based the Kinect joint data and the other set is the combination of Kinect joint data and features extracted from color images. The recognition is accomplished first by calculating DTW matrices, and then applying various clustering methods. Consistent with other research results, the recognition accurate is higher when the combination of features is employed.

Pugeault and Bowden presented an interactive user interface for the recognition of American Sign Language finger-spelling alphabet. Hand shapes are extracted from the kinect color and depth images and used as the feature set for recognition, which is based on random forests.

Rakun et al. described their work on the recognition of the Sign System for Indonesian Language at the individual word level using both the skeleton joint ori-
entation data and hand features extracted from the Kinect depth images. Two different classifiers are used in their experiments, namely, generalized learning vector quantization (GLVQ) and random forest, with the latter producing higher recognition accuracy.

Rioux-Maldague and Giguere proposed an approach to the recognition of static hand poses for 24 letters in the American sign language using a deep learning method called deep belief network. The feature set is based on the combination of hand intensity features (from color images) and hand depth features (from depth frames).

Trindade et al. enhanced the Kinect data with an inertial sensor based pose sensor to help determine the hand angular pose, which is used as the feature set to recognize the Portuguese Sign Language alphabet. The recognition is accomplished via direct comparison with a hand gesture template database. The similarity calculation is based on matching 3D voxel occupancy.

Verma et al. described a two-stage feature extraction scheme for sign language gesture recognition. In the first stage, the arm posture is calculated based on Kinect shoulder, elbow and hand joints. In second stage, features regarding fingers, such as the number of open fingers in the hand, are extracted based on depth images. The HMM is used for recognition. Although the authors stated that 150 gestures were used in their experiment, no details regarding exactly what gestures are used and in the context of which sign language.

Zafrulla et al. pioneered the study on American sign language recognition using Kinect. Furthermore, different from other work reviewed earlier, they aimed to not only perform word-level recognition with HMM, but sentence-level as well based on a pre-defined grammar in the context of an education game. Features used for recognition include hand shapes, hand motion trajectory, and hand poses.

As can be seen from the literatures we have reviewed in this section (which are summarized in Table 7), the sign language recognition research is still in its infancy. Except the work from Zafrulla et al., the current research focuses on the recognition of isolated words and numbers in various sign languages. Furthermore, only a small fraction of the vocabulary has been attempted to be recognized. The limitation of the current research may be partially attributed to the low resolution of depth images and the lack of finger tracking support in the Microsoft SDK for the Kinect v1 sensor. We anticipate that with the much superior depth resolution and finger tracking support of Microsoft Kinect v2, new exciting research on sign language recognition will soon emerge.

### 3.7. Retail Services

The Kinect technology could also be beneficial to retail services. Popa et al. proposed a system for analyzing human behavior patterns related to products interaction, such as browsing through a set of products, examining, picking products, trying products on, interacting with the shopping cart, and looking for support
by waving one hand.\textsuperscript{95} Kinect was used to capture the motions that would help assess customers’ shopping behavior and detect when there is a need for support or a selling opportunity. This application aims to increase customer satisfaction and improve services productivities.

Wang \textit{et al.} proposed an augmented reality system that allows the users to virtually try on different handbags at home in front of a TV screen.\textsuperscript{134} The users can interact with the virtual handbags naturally, such as sliding a handbag to different positions on their arms and rotating a handbag to see it from different angles. Users can also see how the handbags fit them in different virtual environments other than the current real background.

The literatures reviewed in this subsection, together with those in all the remaining subsections are summarized in Table 8.

Table 8. Summary of Kinect applications in retail, training, speech and sign language recognition, and 3D reconstruction.

<table>
<thead>
<tr>
<th>Applications</th>
<th>Main Contributions</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retail</td>
<td>Human behavior pattern recognition</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>on products interaction</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Augmented reality system for virtual handbags</td>
<td>134</td>
</tr>
<tr>
<td>Training</td>
<td>Back injury prevention</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>Musculoskeletal injury prevention</td>
<td>32</td>
</tr>
<tr>
<td>3D Reconstruction</td>
<td>3D human body reconstruction</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>Reconstructing 3D mesh skeleton</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>Realtime 3D reconstructing of moving human body</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Integrate Kinect with high resolution webcam</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>for 3D image reconstruction</td>
<td></td>
</tr>
</tbody>
</table>

3.8. \textit{Workplace Safety Training}

Workplace safety training could also be benefited by the Kinect technology. Martin \textit{et al.} proposed a Kinect-based automated system to aid in the prevention of back injuries by notifying the worker about dangerous movements in real-time at the lift location.\textsuperscript{79} The system can also be used as a training tool due to its capability of recognizing lift skills.

Dutta \textit{et al.} utilized Kinect to record postures and movements for determining the risk of musculoskeletal injury in the workplace.\textsuperscript{32} The Kinect-based system was shown to have comparable accuracy versus existing lab-based systems. It provides a compact, portable motion capture system allowing workplace ergonomic assessments to be done simply and inexpensively.

3.9. \textit{3D Reconstruction}

It is a challenging task to build geometrically consistent 3D models because of individual pairwise errors. Chatterjee \textit{et al.} proposed to use Kinect for 3D human
body reconstruction. A challenge for doing this is that the depth images obtained via Kinect have high noise levels. The proposed approach addresses both the issues of depth image noise as well as the convergence of scan alignment to build accurate 3D models.

Farag et al. proposed an algorithm to efficiently calculate a vertex antipodal point for reconstructing a skeleton of a 3D mesh for mesh animation. The algorithm was successfully tested on different classes of 3D objects and produced efficient results. It is capable of producing high quality skeletons, which makes it suitable for applications where the mesh skeleton mapping is required to be kept as much as possible.

Alexiadis et al. proposed an algorithm to reconstruct an accurate, realistic, full 3D moving human body in real-time. The approach is based on the generation of separate textured meshes from multiple RGB-Depth streams, accurate ICP-based alignment, and a fast zippering algorithm for the creation of a single full 3D mesh.

Jia et al. introduced a novel 3D image reconstruction method using the 2D images from a high resolution webcam combined with Kinect depth images. The proposed system provides a 3D live image without glasses or any other display devices.

4. Human Motion Recognition with Microsoft Kinect

Human motion recognition aims to understand the semantics of the human gestures and activities. A gesture typically involves one or two hands, and possibly body poses, to convey some concrete meaning, such as waving the hand to say goodbye. An activity usually refers to a sequence of full body movements that a person performs, such as walking, running, brushing teeth, etc., which not necessarily conveys a meaning to the computer or other persons. Rehabilitation exercises form a special type of activities.

As shown in Figure 8, the approaches used in gesture and activity recognition can be roughly divided into two categories:

(1) Template based: In this approach, the classification of an unknown gesture or activity is done by comparing with a pre-recorded template motion automatically via pattern recognition.

(2) Algorithmic based: In this approach, a gesture or an activity is recognized based on a set of manually defined rules.

The template based approach can be further divided into two categories:

- Direct matching: In this approach, the template motion is compared directly with the unknown motion to be classified. The most dominating algorithm used for direct matching is dynamic time warping (DTW). However, other algorithms have been used for direct matching as well.

- Modeled based matching: In this approach, a kinematic model or a statistical model is used. The template is used to determine the parameters of the model.
Then, the fitted model is used to classify an unknown gesture or activity. The method used to train the model varies significantly, from simple ones such as obtaining average joint angles as in Ref. 127, to sophisticated machine learning methods such as hidden Markov models as in Ref. 125 and artificial neural networks as in Ref. 89.

The algorithmic-based recognition does not rely on exemplar training data. Instead, it depends on a well-defined specification of the gesture or activity that can be translated into a set of implementable rules for the gesture, and it requires the tuning of the parameters for the rules. On the other hand, the algorithmic based approach can recognize both the type of the gesture or activity and the state of the gesture or activity.

The main benefit of the template-based approach is the automatic classification of unknown gestures or activities. As a tradeoff, most of such model-based matching methods require large amount of training data, and the direct matching method is computational expensive and not suitable for realtime gesture recognition. Furthermore, the feature sets used in the template-based methods usually have to be carefully selected manually to reflect the most distinctive characteristics of the gesture and activity for good classification accuracy. Some machine-learning-based methods require the manual tuning of the model parameters as well.

The various approaches used for motion recognition with Kinect that we survey in this section is summarized in Table 9.
Table 9. Summary of motion recognition techniques.

<table>
<thead>
<tr>
<th>Method</th>
<th>Major Contributions</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithmic Recognition</td>
<td>Rules based on trunk flexion angle and distance traversed</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>Rules based on knee angle</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Rules based on hip angle and smoothness of head movement</td>
<td>135</td>
</tr>
<tr>
<td></td>
<td>Developed a Gesture Description Language</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>Rules for static poses, dynamic movements, and movement invariance</td>
<td>149</td>
</tr>
<tr>
<td>Direct Matching</td>
<td>Hand gesture recognition using DTW with depth image</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Hand gesture recognition using DTW with skeleton data</td>
<td>130</td>
</tr>
<tr>
<td></td>
<td>Quality assessment of rehab exercises using DTW</td>
<td>117</td>
</tr>
<tr>
<td></td>
<td>3D signature recognition for authentication using DTW</td>
<td>121</td>
</tr>
<tr>
<td></td>
<td>Matching using maximum correlation coefficient</td>
<td>137</td>
</tr>
<tr>
<td></td>
<td>Matching using earth mover’s distance</td>
<td>100, 99</td>
</tr>
<tr>
<td>Kinematic Modeling</td>
<td>Programming by demonstration with kinematic modeling</td>
<td>127</td>
</tr>
<tr>
<td></td>
<td>Use dynamic features instead of kinematic ones</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td>Use orientation of hand centroid as feature</td>
<td>140</td>
</tr>
<tr>
<td></td>
<td>Activity recognition using MEMM</td>
<td>119</td>
</tr>
<tr>
<td></td>
<td>Classify individual features using HMM</td>
<td>144</td>
</tr>
<tr>
<td>Neural Networks</td>
<td>Static gesture recognition using MLP</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Quality assessment of rehab exercises using NN-ANARX</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>Static gesture recognition using complex-valued neural network</td>
<td>48</td>
</tr>
<tr>
<td>SVM</td>
<td>Finding best features for SVM classification</td>
<td>77</td>
</tr>
<tr>
<td></td>
<td>Recognition of key poses using SVM</td>
<td>84</td>
</tr>
<tr>
<td>Decision Forest</td>
<td>Modeling sequences of key poses for gesture recognition</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>Human fall detection using Randomized Decision Forest</td>
<td>12, 45</td>
</tr>
<tr>
<td>Adaboost</td>
<td>Used in Kinect v2 SDK for categorical classification of simple gestures</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Kernel regression using skeleton data</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>Least square regression for one shot learning</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>Randomized forest regressor for gesture state estimation</td>
<td>1</td>
</tr>
<tr>
<td>Regression</td>
<td>Formulate as a document classification problem</td>
<td>120</td>
</tr>
<tr>
<td>Others</td>
<td>Recognition based on ATC and the action graph model</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>Trajectory-based classification</td>
<td>110</td>
</tr>
</tbody>
</table>

4.1. Algorithmic-Based Recognition

Algorithmic-based recognition is popular in gaming and healthcare applications because the gestures and/or activities are usually very well defined, relatively simple, and repetitive in nature. Each gesture or activity normally has a pre-defined starting and ending pose that can be used to delineate an iteration of the gesture or activity. Naturally, the algorithmic-based motion recognition approach is a good fit in such application domains.

Furthermore, in some cases, such as rehabilitation exercises, the rules are primarily defined to assess the correctness of movements rather than to classify them because it is assumed that the user already knows or is informed which particular exercise to perform. Hence, it is not necessary for the rules to completely define the exercise as long as they are in line with the therapeutic objectives of the exercise.
and are sufficient to automatically carry out correctness assessment and repetition count. Consequently, most such studies focus on a small set of rules and they are predominately expressed in terms of joint angles.

Compared with other approaches mentioned previously, the algorithmic based approach has a number of limitations:

- The rules for each gesture or activity have to be carefully defined by experts and expressed in an implementable form. This would incur additional financial cost and prevent a regular user from defining his/her own gestures or activities. For rehabilitation exercises and therapeutic games, however, this is largely not an issue because the clinician who prescribes an exercise, or the game designer, is an expert in defining the exercise or the game.
- The gesture has to be simple enough to be defined in terms of a set of implementable rules.
- The parameters used in the rules for the boundary conditions must be manually tuned carefully.

In Ref. 25 and Ref. 26, the rules are expressed in terms of the trunk flexion angle and the distance traversed of a set of joints for postural control, and in terms of the trunk lean angle for gait retraining. In Ref. 14, the knee angle and the ankle angle are used to assess the quality of sit-to-stand and squat, and the shoulder angle is used to assess the shoulder abduction/adduction quality. In Ref. 5, the rules are expressed in terms of the knee angle in a robotic system for knee rehabilitation.

In Ref. 135, two metrics are used to evaluate the quality of the sit-to-stand exercise: (1) the minimum hip angle, in which a younger healthier person would typically have a larger value than an older person; and (2) the smoothness of the head movement, which is quantified as the area of the triangle that is determined by the second highest peak, the valley and lines that are parallel to the axes on the head-speed-versus-time plot.

Far more comprehensive rules have been developed for the purpose of recognizing hand and body gestures. In Ref. 47, a Gesture Description Language (GDL) is introduced, in which a gesture is determined by a set of key frames. A frame contains joint positions reported by the Kinect sensor. All rules are expressed in terms of one or more key frames except the final rule, which defines the gesture in terms of a sequence of basic rules. The rules are written as text files and are parsed with an LALR-1 grammar. During runtime, a gesture is recognized with the following steps executed in a loop:

1. When a new frame arrives, the new motion data is stored in a memory heap. The set of rules that have been satisfied so far are also stored in the heap.
2. Examine the new data to see if any new rule is now satisfied.
3. If a new rule is satisfied, the rule name is placed at the top of heap with a timestamp. If the final rule that defines a gesture is satisfied, then the gesture is recognized.
(4) If a new rule is satisfied in the previous step, go to step 2 to see if any other rule is now satisfied as well. Otherwise, go back to step 1 waiting for the next frame.

Because GDL is designed to be based on a set of key frames, it is resilient to motion sensing errors. However, as a tradeoff, it lacks the support for rules that depend on the entire trajectory of a gesture. It also lacks a guideline as to how to identify the key frames for each gesture.

Recently, we proposed an algorithmic-based approach to assessing the quality of rehabilitation exercises. Our approach is inspired by Ref. 10 in that dynamic movements in each rehabilitation exercise are defined in terms of monotonic segments. However, we also include rules regarding invariance requirements, which may not be important for general purpose motion recognition, but critical for the effectiveness of rehabilitation exercises. For example, for hip abduction, it is important that the abducting leg should remain within the frontal plane the entire time, which deserves a separate invariance rule. We also accommodate rules that define static poses. A finite state machine based approach is used in dynamic rule specification and realtime assessment. In addition to the typical advantages of the algorithmic-based approach, such as realtime motion assessment with specific feedback, our approach has the following advantages: (1) increased reusability of the defined rules as well as the rule assessment engine facilitated by a set of generic rule elements; (2) increased customizability of the rules for each exercise enabled by the use of a set of generic rule elements and the use of extensible rule encoding method; and (3) increased robustness without relying on expensive statistical algorithms to tolerate motion sensing errors and subtle patient errors.

4.2. Direct-Matching-Based Recognition

In this approach, the unknown gesture or activity is directly compared with a set of templates. DTW is perhaps the most well-known technique to analyze the similarity between two temporal sequences that may vary in time and speed by finding an optimal alignment between them. Typically one sequence is an unknown sequence to be classified and the other sequence is a pre-classified reference sequence (i.e., the template, also referred to as the exemplar). The difference between the two sequences is expressed in terms of the distance between the two. In addition to DTW, other direct matching methods have also been used in the literature, for example, the matching can be done via the calculation of the maximum correlation coefficient. For static gestures, the distance between two gestures can be calculated using an algorithm called Earth Mover’s Distance.

4.2.1. DTW

Doliotis et al. proposed to use DTW to recognize hand gestures (digits recognition in particular). Kinect depth frames were used to detect the hand instead of using
the Kinect skeleton frames. The normalized 2D position is used as the feature vector for DTW matching.

Waithayanon et al. presented a study using DTW to recognition 7 hand gestures. The left/right hands and wrists joints obtained directly from Kinect skeleton frames are used as the feature vectors for DTW matching, and the distances between each test run and all the reference gestures are reported. The experimental result shows that 100% accuracy is achieved with the limited set of gesture vocabulary.

In a Kinect-based system for in-home rehabilitation exercises, DTW was used to determine the similarity between the exercise done at the direct supervision of a clinician, and that done at home. The trajectory and speed of individual joints involved in each exercise are used as the feature vector and compared separately using DTW. The joint position information was obtained from the Kinect skeleton frames. The quality of the exercise done at home was evaluated using a set of fuzzy logic rules in terms of the similarity (or dissimilarity) of the trajectory and speed of each joint involved.

Tian et al. proposed a system using 3D signatures for authentication. In the system, DTW was used to compare a test signature with a reference signature. The signatures were recorded using Kinect. Instead of using the hand or wrist joint data reported by Kinect skeleton tracking, the finger tip position of the signing hand was extracted from each depth frame for better accuracy. A 14-dimension feature vector was used for the DTW comparison, including 3D positions, velocity, acceleration of the finger tip, the distance traveled between two consecutive frames, the slope angle, path angle, and the log radius of the curvature of the trajectory of the finger tip. The features are normalized and weighted based on their criticality to correct verification of the signatures. The finger tip positions are also filtered and smoothed using Kalman Filter to reduce the spatial noise of the recorded signatures.

4.2.2. Maximum Correlation Coefficient

In the context of the one-shot learning challenge for Kinect, the classification of an unknown gesture based on a single exemplar gesture set can be done by finding a known gesture that has the maximum correlation coefficient of the corresponding feature vectors, as reported by Wu et al., where motion energy images and motion history images are used as the feature vector.

4.2.3. Earth Mover’s Distance

Ren et al. used an improved version of the Earth Mover’s Distance to calculate the similarity in hand shapes between an unknown static hand gesture and a set of templates. To distinguish hand gestures with slight differences, the finger parts instead of the whole hand, are used for the similarity calculation. The hand shape is detected based on the Kinect depth and color images.
4.3. Non-Machine-Learning-Based Kinematic Modeling

In MotionMA, the assessment of the quality of an exercise is achieved via building a kinematic model using exemplar data, and comparing the observed parameters and the fitted ones. No machine learning method is used to train the model and to classify the observed motion. The kinematic model consists of a collection of joint angles, which are sufficient in the context of rehabilitation exercise monitoring. The training data is first filtered using a low-pass filter to remove noise and feature data is extracted on zero-derivatives (peaks, valleys, and inflexion points). The feature data is merged using k-means clustering. The merged data serves as the model for the gesture and is used to identify static and dynamic axes. This simple model enables the system to monitor violations in static axes continuously in realtime, and to count the repetitions for dynamic joints.

4.4. Machine-Learning-Based Motion Recognition

Machine-learning-based motion recognition typically relies on one or more sophisticated statistical models, such as the Hidden Markov Model (HMM), Artificial Neural Networks (ANNs), Support Vector Machine (SVM), etc. to capture the unique characteristics of a gesture or an activity. Most of such models consist of a large number of parameters, which have to be determined in a training step based on pre-labeled motion data (including both data for the gesture to be recognized, and other motion data that are known not be the specific gesture). In general, the larger of the feature set used for classification, the larger training dataset is required. For some models, such as ANNs, additional modeling parameters have to be manually tuned to achieve good classification accuracy.

Typically, machine-learning-based motion recognition is framed as a classification problem. Hence, the trained model is usually referred to as a classifier. However, motion recognition could also be formulated as the regression problem. In this case, the trained model is referred to as a regressor. Unlike the classifier, which outputs a discrete value (regarding which class the testing gesture or activity belongs to with the highest probability), the output of a regressor is usually a continuous value within some predefined range. To use the regressor as a classifier, a threshold can be used so that when the output of the regressor exceeds the threshold, the class of the testing gesture can be determined. Using a regressor has the advantage of providing not only the classification (by using a heuristic threshold), but may also give the information regarding the state of the gesture or activity, i.e., the progress has made so far in the context of the gesture or activity, which is important for many interactive applications.

In the following, we review machine-learning-based motion recognition work using data collected via a single Kinect sensor. We divide the literatures based on the specific machine learning methods used. Most of methods use a feature set extracted from Kinect skeletal data, which the 3D positions of the joints of interest are readily available, whereas some methods operate directly on depth images,
particularly for research works on the one-shot learning as part of the ChaLearn challenge (http://www.kaggle.com/c/GestureChallenge). The models reported in the literature differ significantly. While some of them require larger training data set than others, it is hard to characterize which model works better than others fundamentally and we do not see any discussion in the surveyed literature regarding why a particular model works better for a particular motion recognition context.

### 4.4.1. The Hidden Markov Model

The Hidden Markov Model (HMM)\textsuperscript{9} is perhaps the most popular model used for motion recognition with Kinect data. HMM is applicable to any dynamic system that is governed by a Markov chain where only its output values are observable and not its states. A system that is modeled by HMM is defined by the following parameters: (1) the number of states; (2) the number of distinct observation symbols per state; (3) the initial state distribution vector; (4) the state transition probability distribution matrix, which defines the probability of the transition from one state to another; (5) the observation probability distribution matrix, which defines the probability of observing each output symbol given each state. The first three parameters must be determined manually and they are application-dependent. To use HMM in the context of machine learning, the last 2 parameters are determined based on training data. It is apparent that the larger the state size and the observation symbol size, the larger amount of training data is required. Once, all parameters are set, one can perform classification by calculating the most likely state sequence given a sequence of output values.

Because a human motion (\textit{i.e.}, a gesture or an activity) consists of a sequence of poses, it is quite natural to use HMM to model a gesture or an activity for the purpose of recognition. The number of states and number of output symbols depends on the motions to be recognized. Typically, large amount of training data is required for HMM to be accurate for human motion recognition.

To reduce the size of the required training dataset, Mansur \textit{et al.} proposed to use the dynamic features instead of kinematic features for human action recognition using HMM.\textsuperscript{78} Dynamic features are derived from the physics-based representation of the human body, such as the torques from some joints. Dynamic features have lower dimension than kinematic features, which is why less pre-labeled motion data are required to train the HMM classifier.

Xu \textit{et al.} employed HMM to classify hand gesture sequences for real-time navigation of a robot using human hand gestures.\textsuperscript{140} The feature vector used in the HMM classification is based on the orientation of the hand centroid extracted directly from the Kinect depth images. Because of the nature of the gestures involved (such as move forward, move back, turn left and turn right), the modeling is restricted to the 2D frontal plane space, which limits the size of the feature set and improves the accuracy of the recognition.

A sophisticated activity may contain several subactivities. For example, the
brushing teeth activity consists of subactivities including “squeezing toothpaste,” “bringing tooth brush close to the head,” and “brushing”. In this case, it is common to use hierarchical HMM to recognize such activities. Sung et al. presented a study that tackles the recognition problem of such activities. The main challenge in hierarchical HMM is to associate subactivities represented by a layer of hidden variables with the activity, which is represented as a node in the higher layer. To accommodate the fact that a single state may connect to different parents only for periods of time, which the traditional hierarchical HMM is incapable of dealing with, a hierarchical maximum entropy Markov model (MEMM) is used instead in the study. Furthermore, various features are compared regarding the final recognition accuracy, including body pose features in the form of joint orientations, hand position with respect to the torso and the head, motion based features in the form of orientation changes across 9 frames within the last three seconds, and finally the image and point-cloud features in the form of Histogram of Oriented Gradients (HOG).

Zhang et al. developed a system to recognize golf swings. HMM is used to classify individual features. The output of HMM of several feature sets are combined using fuzzy logic rules with a single score after a defuzzification step. The main feature set used in the study consists of the joint angles derived from the Kinect skeletal data.

4.4.2. Artificial Neural Networks

Artificial neural networks (ANNs) refer to a collection of statistical learning algorithms inspired by biological neural networks. An ANN models the system as a network of neurons with several layers. The first layer consists of input neurons that send signal to the second layer of neurons. The last layer consists of output neurons, which takes input from other neurons. There could also be intermediate layers.

Prior to training the model, the number of input and output neurons, as well as the activation function must be determined based on the recognition problem. Typically, the number of input neurons depends on the dimension of the feature set, and the number of output neurons depends on the number of classes to be recognized. The total number of neurons needed is typically a tunable parameter. Once the network topology is decided, the weights of the interconnections can be learned with pre-labeled training data.

There are many ANN models. Among them, the multi-layer perceptron (MLP) model has been used to classify static gestures using Kinect data. Another model, referred to as NN-based additive nonlinear auto regressive exogenous (NN-ANARX in short), has been used to determine the quality of a rehabilitation exercise in terms of the difference between the observed motion and the predicted motion with the trained model.

Hafiz et al. used a single-layered complex-valued neural network (CVNN) for
static hand gesture recognition. A hand tree (with key parameters of length and angles of the lines that form the tree) is constructed via both the color and depth images captured from Kinect, and used as the feature set for classification. The feature set is represented using a complex number and is used as the input to the CVNN. The output of the network consists of 26 neurons, which maps to the 26 English characters as the final classification of the testing hand gesture. It shows that CVNN works faster and achieves better accuracy than traditional real-value based neural networks.

4.4.3. Support Vector Machines

The Support Vector Machines (SVMs) are supervised learning models for linear as well as nonlinear classification. For linear classification, the training data is used to determine a plane that separates the data belonging to different classes as further away as possible. This plane then can be used to classify unknown data into one of the two classes. For nonlinear classification, a kernel function is used to make higher dimension classifications (the plane derived from the training data is referred to as hyperplane). The key advantage of SVM is that it guarantees maximum-margin separation using relatively little training data.

Madeo et al. proposed to segment a gesture into a sequence of units and formulate the gesture analysis problem into a classification task using SVM. In addition, they applied several pre-processing methods to extract time-domain and frequency-domain features. The study aims at finding the best parameters for a SVM classifier in order to distinguish the rest positions from a gesture unit. The features used for classification are based on the 3D positions of 6 joints reported by Kinect, including two hands, two wrists, head and spine. First, a normalized vector is derived from the 6 joints, which is followed by the velocity and acceleration information.

SVM was used in Ref. 84 to identify key poses in a sequence of body motion where the joint angles are used as features. The actual gesture recognition was accomplished via a decision forest. Similarly, SVM was used as one of the models in a comparison study in Ref. 93 for a set of static gestures including stand, sit down, and lie down using the 3D positions of the skeletal joints as the feature vector.

4.4.4. Decision Tree, Decision Forest, and Random Decision Forest

A decision tree consists of a collection of nodes connected to a tree structure. Each internal node (often referred to as the split node) in the tree represents a test on one of the features with a threshold, and each branch represents the outcome of the test. A leaf node in the tree represents a class label. A decision can be taken using the decision tree by computing all attributes. The test at the split node is essentially a weak classifier. Hence, a decision tree is an ensemble of weak classifiers on different features, which could lead to a better overall classification than any individual weak classifier. In the context of machine learning, a decision tree is
constructed using pre-classified training data. The constructed decision tree can then be used for the purpose of classification of unknown data or regression.

To implement a multi-class classifier, a collection of decision trees is usually used. The collection of decision trees is referred to as a decision forest. To reduce the correlation among the trees in a decision forest, a random subset of the features is selected at each split during the learning process. This method is referred to as randomized decision forest, or randomized forest for short.\textsuperscript{116}

Miranda \textit{et al.} used the decision forest algorithm to identify gestures in real-time on Kinect motion data.\textsuperscript{84} A gesture is modeled as a sequence of key poses. During training, a decision forest is constructed based on the key poses. Each path from a leaf node to the root represents a gesture (the gesture identifier is stored at the leaf node). Gesture recognition is reduced to a simple searching problem based on the decision forest.

Several groups of researchers have used randomized forest in fall detection, with the aim to recognize skeleton shape deformation caused by the human body falling.\textsuperscript{12,45} However, due to changes in the orientation of the body during movement, the accuracy of recognition is reduced.

4.4.5. \textit{Adaboost}

Adaboost refers to a meta-algorithm for machine learning called Adaptive Boosting.\textsuperscript{40} Unlike previously introduced models and algorithms, Adaboost is a higher-level algorithm that works with a set of lower-level classifiers, and selects the most optimal ones that lead to a more accurate classification. Specifically, the learning step of Adaboost is not to fit unknown parameters for a model, but instead, to find the best lower-level classifiers. Hence, weak classifiers such as decision dumps (i.e., 1-level decision tree) can be used with Adaboost to form a strong classifier that produces highly accurate classification. Another benefit for using Adaboost is that it can be used to facilitate knowledge discovery in that the user can see which lower-level classifiers are most appropriate for each gesture. As a tradeoff, Adaboost requires high quality training data to achieve good classification accuracy.

Adaboost is used to provide categorical gesture recognition in the Microsoft Kinect v2 SDK.\textsuperscript{1} Decision dumps are used as the low-level weak classifiers, which are automatically generated based on the skeleton joint positions (in the form of angles between body segments and angle velocities). A drawback of using decision dumps as the weak classifiers is that a complex gesture or activity must be manually separated into a set of simple actions and the classification has to done on each simple action to be effective.

4.4.6. \textit{Regression-Based Methods}

Kernel regression maps an input variable to an output value by averaging the outputs of a kernel function based on a set of predefined set of data and the input
variable. Typically, the Gaussian kernel is used in the calculation. Bigdelou et al. proposed a gesture recognition method based on kernel regression. All 20 joints that are obtained from Kinect skeleton tracking are considered. The feature set used in the study includes the distances of the joints with respect to the spine joint, the displacements vectors of the joints with respect to the spine, and those with respect to the parent joint. A gesture is defined as a sequence of control poses. Via principal component analysis (PCA), the set of feature vectors together with their classification are mapped to a one-dimensional signal. Given a test feature vector of an unknown gesture, the kernel regression mapping is used to produce a one-dimensional value with the trained data. This value predicts the state of the gesture. The category of the unknown gesture is obtained via an arg max operation on a Gaussian kernel with respect to the unknown gesture and each of the labeled gesture in the training set.

A benefit of using kernel regression is that the method allows simultaneous recognition of the type of a gesture as well as the relative poses within a gesture (i.e., the state of the gesture). In many interactive applications, the state of the gesture is essential for the system to react to the user’s gesture input in realtime.

Least squares regression aims to fit parameters for a linear or nonlinear function with minimum squared errors. This function can then be used for the purpose of classification. This method has been used by Lui in Ref. 75 for the purpose of gesture recognition as part of the one-shot gesture recognition challenge. The idea is to build a product manifold representation based on the Kinect depth data, where a gesture would be located as a point on the product manifold. Least squares regression is used to produce a smooth decision boundary for classification. The reason why this approach is viable is that a gesture has a unique underlying geometry. A main advantage of this approach is that it works for a small training dataset.

Randomized decision forest regression has been used to determine the progress of a gesture as part of the recently released Kinect v2 SDK. The continuous value given by the regressor is meaningful only when the current gesture has already been identified.

4.4.7. Other Approaches

Thanh et al. formulated activity recognition as the problem of classifying documents into the right categories. Each activity consists of a sequence of subactivities. Here, a subactivity assumes the role of a word in a document, and an activity is analogous to a document containing a sequence of words. The first step is to identify key frames, which are representative of subactivities. The second step is to establish patterns formed by the key frames. The final step aims to identify discriminative patterns, which can be used to classify an unknown activity. This is accomplished by using an adapted weighting method, which is based on finding the frequencies of patterns.

Lin et al. proposed to use the action graph model based on Action Trait Code
(ATC) to classify human actions. The ATC uses the average velocity of body parts to yield a code describing the actions. The average velocity of each body part in an action sequence is labeled as action elements. Then an action graph is constructed based on the training data, which is used to classify unknown actions.

Sivalingam et al proposed to classify human actions using their trajectories. Consequently, how to effectively represent the action trajectories is key. In their study, two different representation schemes, one based on raw multivariate time-series data, and the other based on the covariance descriptors of the trajectories. These features are then coded using the orthogonal matching pursuit algorithm. The classification can then be classified by calculating the reconstruction residuals.

5. Publicly Available Kinect Datasets

In this section, we compile a list of Kinect datasets that are publicly available. These datasets may be very valuable resources for other researchers to carry out additional computer vision and motion analysis work. The datasets are introduced in a reverse-chronological order. For each dataset, we briefly elaborate the content of the dataset and the original research on the dataset. We also summarize the datasets in Table 10 with the URL of each dataset Webpage.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Dataset Webpage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indoor scenes(^{109,108})</td>
<td><a href="http://cs.nyu.edu/silberman/datasets/">http://cs.nyu.edu/silberman/datasets/</a></td>
</tr>
<tr>
<td>Human actions(^{65,132,134,141})</td>
<td><a href="http://research.microsoft.com/en-us/um/people/zliu/actionrecorsrc/">http://research.microsoft.com/en-us/um/people/zliu/actionrecorsrc/</a></td>
</tr>
<tr>
<td>Hand gestures(^3)</td>
<td><a href="http://lshao.staff.shef.ac.uk/data/">http://lshao.staff.shef.ac.uk/data/</a> SheffieldKinectGesture.htm</td>
</tr>
<tr>
<td>Object tracking(^{111})</td>
<td><a href="http://tracking.cs.princeton.edu/dataset.html">http://tracking.cs.princeton.edu/dataset.html</a></td>
</tr>
<tr>
<td>3D reconstruction(^{139})</td>
<td><a href="http://sun3d.cs.princeton.edu/">http://sun3d.cs.princeton.edu/</a></td>
</tr>
<tr>
<td>Category modeling(^{714})</td>
<td><a href="http://shiba.iis.u-tokyo.ac.jp/song/?page_id=343">http://shiba.iis.u-tokyo.ac.jp/song/?page_id=343</a></td>
</tr>
<tr>
<td>3D scans(^{71})</td>
<td><a href="http://vcl.iti.gr/3d-scans/">http://vcl.iti.gr/3d-scans/</a></td>
</tr>
<tr>
<td>Human actions(^{138})</td>
<td><a href="http://cvrc.ece.utexas.edu/KinectDatasets/HO3D.html">http://cvrc.ece.utexas.edu/KinectDatasets/HO3D.html</a></td>
</tr>
<tr>
<td>Scene in hallway(^{14,112})</td>
<td><a href="http://www2.informatik.uni-freiburg.de/%22spinello/RGBD-dataset.html">http://www2.informatik.uni-freiburg.de/&quot;spinello/RGBD-dataset.html</a></td>
</tr>
<tr>
<td>Various objects(^{94,95})</td>
<td><a href="http://kinectdata.com/">http://kinectdata.com/</a></td>
</tr>
</tbody>
</table>

The latest publicly available datasets were released by Silberman et al at New York University. Unlike other datasets, they provided data taken using both Kinect v1 and Kinect v2. The content of the datasets is summarized in Table 11. The original research was focused on scene segmentation without the presence of human subjects.\(^{108,109}\)
Table 11. Details of the datasets from Silberman et al.

<table>
<thead>
<tr>
<th>Type</th>
<th>Kinect v1</th>
<th>Kinect v2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Types of indoor scenes</td>
<td>7</td>
<td>26</td>
</tr>
<tr>
<td>Number of scenes</td>
<td>64</td>
<td>464</td>
</tr>
<tr>
<td>Unlabeled frames</td>
<td>108,617</td>
<td>407,024</td>
</tr>
<tr>
<td>Densely labeled frames</td>
<td>2347</td>
<td>1449</td>
</tr>
<tr>
<td>Number of classes</td>
<td>Over 1000</td>
<td>Over 1000</td>
</tr>
</tbody>
</table>

The MSR action recognition datasets were released by Liu at Microsoft Research. Apparently the datasets have been compiled over several years. The datasets contain the following:

- The 3D online action dataset: It includes matching color and depth data for continuous online human action. It was originally used to study realtime recognition of human-object interaction.\(^{141}\)
- The MSRGesture3D dataset: It contains 12 American sign language gesture performed by 10 human subjects.\(^{65,132}\)
- The MSR daily activity 3D dataset: It consists of 16 activities such as drink, eat, read book, etc. It was originally used to study action recognition by mining actionlet ensembles.\(^{133}\)
- Several other datasets recorded using devices other than Kinect.

The Sheffield Kinect gesture dataset was made available by Liu and Song at The University of Sheffield. The dataset contains 2160 hand gesture sequences with 1080 color image frames and 1080 depth image frames collected from 6 human subjects. The original research on the dataset was on automated feature extraction of spatio-temporal features using an adaptive learning method.\(^{73}\)

The Princeton tracking benchmark dataset contains 100 sets of matching color and depth video. Five of them are validation video with ground truth and the remaining 95 are evaluation video. In addition to the dataset, Matlab code is also provided for benchmarking. The original research on the dataset was to establish a uniform benchmark and baseline for object tracking.\(^{111}\)

The SUN3D database contains a large-scale matching color and depth videos with camera poses and object labels. The database is also hosted by Princeton University. The original research aimed at capturing the full 3D extend of the scene by combining object labels and the structure from motion.\(^{139}\)

The dataset from Zhang et al. at The University of Tokyo contains color and depth images with 900 objects. The dataset was collected both indoors and outdoors. It contains objects of 7 categories, including basket, bucket, bicycle, scanner, fridge, notebook PC, sprayer, dustpan, and platform lorry. The original research on the dataset was to exploit the depth information in images to guide the learning of 2D models.\(^{145}\)
The CERTH/ITI dataset was built for 3D scans of small-sized objects. It contains multi-view range scans of 59 objects. For each object, it contains color and depth information for each view with registered point clouds for all views. In addition, the range scans from an accurate laser scanner were included to establish the ground truth.

The MSRC-12 Kinect gesture dataset was released by Microsoft Research Cambridge in 2012. It contains 594 sequences and 719,359 frames collected from 30 people, each performing 12 gestures (6,244 gesture instances total). The original research was on how to instructing people to perform the gestures to be recorded as the training set.

The UTKinect-Action dataset was released by Xia et al. at University of Texas. The dataset contains a set of videos for actions performed by 10 human subjects. The actions performed include 10 action types, such as walk, sit down, stand up, pick up, carry, throw, push, pull, wave hands, clap hands. The color and depth images, as well as skeleton frames are included in each recording. The original research was to investigate view invariant human action recognition.

The Cornell activity dataset contains 180 videos with matching color and depth frames. It contains activities recorded in various environments such as office, kitchen, bedroom, bathroom, and living room. There are two subsets. The first contains 60 videos with 12 activities performed by 4 human subjects. The second contains 120 videos with 10 activities performed also by 4 human subjects, where the activities are divided into subactivities with labels. In addition to the dataset, source code on feature extraction, activity labeling, activity anticipation, and skeleton visualization, is also provided. The original research on the dataset was to investigate unstructured human activity detection.

The RGB-D people dataset was released by Spinello at University of Freiburg in 2011. The dataset contains over 3000 matching color and depth frames recorded in a University hall. The activities recorded include walking and standing with various orientations and different levels of occlusions. The dataset was originally used to study people detection and tracking.

The Berkeley 3D object dataset was released in 2011 by University of California, Berkeley. In 2014, the dataset was updated with annotations of the 3D center points of all objects. The dataset contains large amount of objects with matching color and depth images. The original research on the dataset was to perform category-level object detection.

6. Conclusion

In this article, we presented a comprehensive survey on the applications of the Kinect technology, and the latest research and development on motion recognition using data captured by the Kinect sensor. On the applications front, we reviewed the applications of the Kinect technology in a variety of areas, including healthcare, education and performing arts, robotics, sign language recognition, retail services,
workplace safety training, as well as 3D reconstructions. On the technology front, we provided an overview of the main features of both versions of the Kinect sensor together with the depth sensing technologies used, and reviewed literatures on human motion recognition techniques used in Kinect applications. We provided a classification of motion recognition techniques to highlight the different approaches used in motion recognition. Each approach has its advantages and disadvantages. Nevertheless, the predominate approach is based on machine learning, such as HMM, ANN, SVM, randomized decision forests, and Adaboost. To achieve high recognition accuracy, the feature set and the model parameters must be carefully selected. Furthermore, we compiled a list of publicly available Kinect datasets. These datasets are valuable resources for researchers to investigate better methods for motion recognition and lower-level computer vision tasks such as segmentation, object detection, and human pose estimation.

Acknowledgments

The authors wish to sincerely thank the anonymous reviewers, and the editor, for their invaluable suggestions in improving an earlier version of this article. The work presented in this article was supported in part by an award from the Cleveland State University Graduate Faculty Travel Program.

References


23. S. Chen, Y. Maeda, and Y. Takahashi. Music conductor gesture recognized interac-


34. Z. Epstein. Microsoft says xbox 360 sales have surpassed 76 million units, kinect sales top 24 million, 2013. URL http://bgr.com/2013/02/12/microsoft-xbox-360-sales-2013-325481/.


99. Z. Ren, J. Meng, and J. Yuan. Depth camera based hand gesture recognition and its


115. J. Stowers, M. Hayes, and A. Bainbridge-Smith. Altitude control of a quadrotor.


