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

Publisher's Statement

Electronic version of an article published as [International Journal of Pattern Recognition and Artificial Intelligence, 29, 5, 2015, 1-48] [10.1142/S0218001415550083] © [2015 World Scientific Publishing Company] [http://www.worldscientific.com/doi/abs/10.1142/S0218001415550083]

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.[34] 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 in door 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.

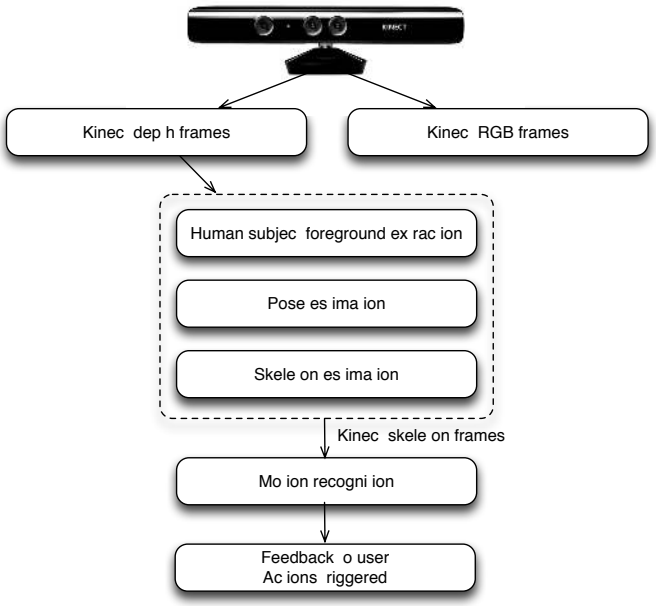

Fig. 1. Typical steps of motion tracking and analysis with Kinect.

- 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 for 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.[143] 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.

Feature	Kinect v1	Kinect v2
Depth Sensing Technology	Triangulation with structured light	Time of flight
Color Image Resolution	640x480 30fps 1280x960 12fps	1920x1080 30fps (12fps low light)
IR Image Resolution	640x480 30fps	512x424 30fps
Depth Sensing Resolution	640x480 30fps 320x240 30fps 80x60 30fps	512x424 30fps
Field of View	43° vertical 57° horizontal	> 43° vertical 70° horizontal [60]
Depth Sensing Range	0.4m - 3m (near mode) 0.8m - 4m (default mode)	0.5m - 4.5m Up to 8m without skeletonization
Skeleton Tracking (with full skeleton)	Up to 2 subjects 20 joints per skeleton	Up to 6 subjects 25 joints per skeleton
Built-in Gestures	None	Hand state (open, close, lasso) Hand pointer controls; lean
Unity Support	Third party	Yes
Face APIs	Basic	Extended massively
Runtime Design	Can run multiple Kinect sensors per computer; One app per Kinect	At most one Kinect per computer; Multiple apps share same Kinect
Windows Store	Cannot publish to	Yes

terns to the objects in the field of view. 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.[64] 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.[29] It appears that the technology is based on that developed by Canesta, which was acquired by Microsoft in late 2010.[62] 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

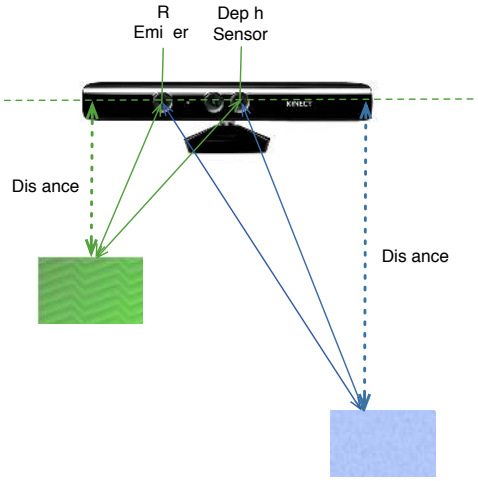

Fig. 2. Kinect v1 uses the structured light triangulation method for depth sensing.

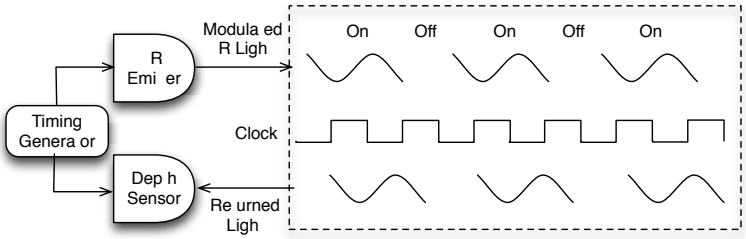

Fig. 3. Kinect v2 uses the time-of-flight method for depth sensing.

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.

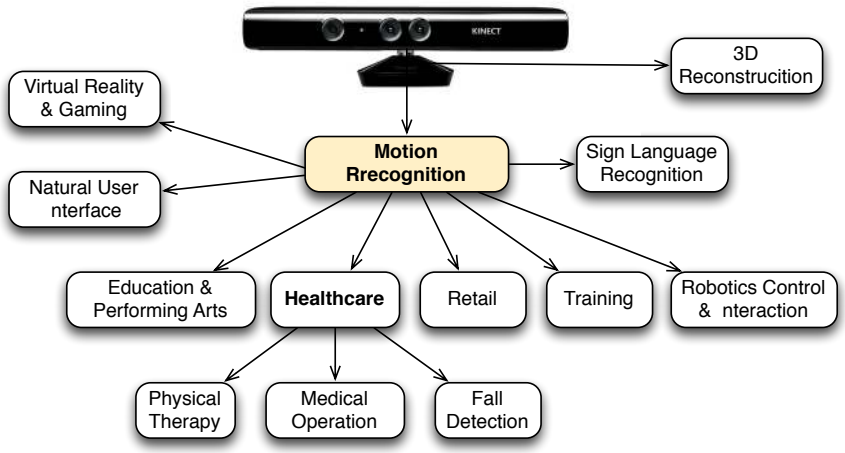

Fig. 5. Primary Kinect application categories.

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.

Applications	Main Contributions	References
Physical Therapy and Rehabilitation	Assessment of Kinect motion tracking accuracy for healthcare applications	19
	An interactive game-based rehabilitation tool for adults with neurological injury	67, 68
	Full-body control in virtual reality applications	118
	Integration of Kinect and a smart glove for patients with upper extremity impairment	53
	An interactive rehabilitation system for disabled children	2
	A rehabilitating program for young adults with motor impairments	20
	Cognitive rehabilitation for Alzheimer's patients using a Kinect-based game	21
	A Kinect-based game for stroke rehabilitation	107
	An exercise rehabilitation program for individuals with spinal cord injury	46
	Integration of Kinect with rehabilitation robotics	94
	Integration of inertial sensors with Kinect	14
	An at-home exercise monitoring system	148, 149
Medical Operating Room Assistance	A Kinect-based intra-operative medical image viewer	13
	A system that enables touchless controlling of medical images with hand and arm gestures	42
Fall Detection and Prevention	A real-time fall monitoring and detection system	81
	Overcoming occlusions for human body fall detection	104
	Fall detection based on Kinect skeletal data	12
	Human fall detection using two Kinect	147
	Capturing variations of stride-to-stride gait for elderly adults	114, 113
	Detecting falls and other abnormal events on stairs	92
	Fall prevention in hospital ward environment	88

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.[46]

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

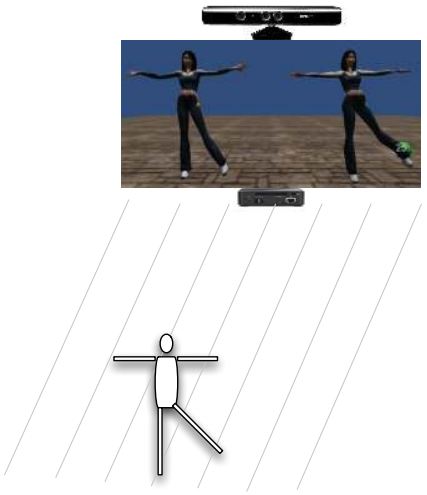

Fig. 6. An at-home exercise monitoring system with Kinect.

both devices may achieve more satisfying results. Bo *et al.* proposed a method to combine Kinect with portable sensors, such as accelerometers, gyrometers, and magnetometers, for measuring human motion for rehabilitation purposes.[14] 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.[127] 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.[117] 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

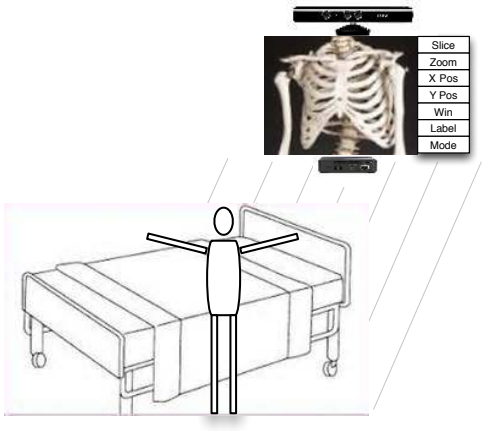

Fig. 7. A medical image viewer based on Kinect.

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.[148] 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.[58] 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.[13,42].

Bigdelou *et al.* developed a Kinect-based intra-operative medical image viewer for use in a surgical environment.[13] 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.[42] 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.[12,81,88,92,104,113,147]

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.[12] 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 predefined particular body parts.[81]

Ni *et al.* developed a Kinect-based system to prevent potential falls in the hospital ward environments.[88] 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.[92] 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.[104] 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.[114] 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.[147] 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.[4] 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 join rotation angles from Kinect for skeleton animation.[123] 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.[39] 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.[122]

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.[86] Hai *et al.* developed an interaction system for treadmill games based on Kinect depth maps.[49] 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.[51] The Wizard-of-Oz is a guessability game to examine child-defined gestures using Kinect.[27] 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.[90]

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.[36] 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.

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

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

Applications	Main Contributions	References
Natural User Interface	Performing desktop tasks via arm gestures	36
	3D object manipulation on a desktop display	97
	3D control method based on Kinect	61
	A 3D navigation user interaction system	38
	A group meeting application based on Kinect	17
	Controlling virtual globes via Kinect	15
	Web browsing via natural user interfaces	71
	Automatic camera control based on Kinect	136

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.[15] 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.[71] 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.[136] The approach enables video cameras to automatically follow a presenter on different premises with different geometries.

3.4. *Education and Performing Arts*

In education, especial 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 5. Summary of Kinect applications in education and performing arts.

Applications	Main Contributions	References
Education and Performing Arts	A classroom teaching system with Kinect	129
	A interactive music conductor generation system with Kinect	23
	A puppetry control application with Kinect	70
	A MotionDraw tool for enhancing art performance	102

Villaroman *et al.* proposed a classroom teaching system that uses Kinect for classroom instruction on natural user interaction.[129] 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.[23]

Leite *et al.* developed a puppetry control application through body motion with Kinect.[70] 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.[102] 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 6. Summary of Kinect applications in robotics control and interaction.

Applications	Main Contributions	References
Robotics Navigation & Control	Robot navigation using Kinect and inertial sensors	33
	Feasibility on using gestures to control industrial robots	52
	A human imitation system	87
Interactively Controlling Robotics	Navigating a robot using hand gestures	140
	A human-robot interactive demonstration system with a gesture recognizer	24
Robotics Remote Control	An athletic training speed skating system using Kinect	16
	A Kinect on-board system that enables the control of velocity and attitude of a mobile robot	76
	Controlling altitude of a quadrotor helicopter via Kinect	115
	A real-time human imitation system for robotics	131
	Tele-operating a humanoid robot using Kinect	150

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.[33] This application integrates 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.[52] 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.[87] The objective of this system is to reproduce imitated human motions during continuous and online observation with a humanoid robot using Kinect. Using straight-forward 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.[140] 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.[24] 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.[16] 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.[76] 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.[115] 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.[131] 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.[150] 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.[41] 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.[3] 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.[7] 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.[8] 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.

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

tron (MLP), which is a feed-forward neural network, is used as a classifier for the recognition task.

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.[44] 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.[57] 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.[69] 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).[80] 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.[83] The spatio-temproal 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.[91] 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.[96] 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.[98] 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.[101] 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.[124] 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.[128] 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.[142] 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.*,[142] 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.[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 *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.[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.

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

3.8. *Workplace Safety Training*

Work place safety training could also be benefited by the Kinect technology. Martin *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.[79] The system can also be used as a training tool due to its capability of recognizing lift skills.

Dutta *et al.* utilized Kinect to record postures and movements for determining the risk of musculoskeletal injury in the workplace.[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. *3D Reconstruction*

It is a challenging task to build geometrically consistent 3D models because of individual pairwise errors. Chatterjee *et al.* proposed to use Kinect for 3D human

body reconstruction.[22] 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.[35] 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.[6] 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.[56] 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 necessary 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).[11] However, other algorithms have been used for direct matching as well.[99,100,137]
- 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.

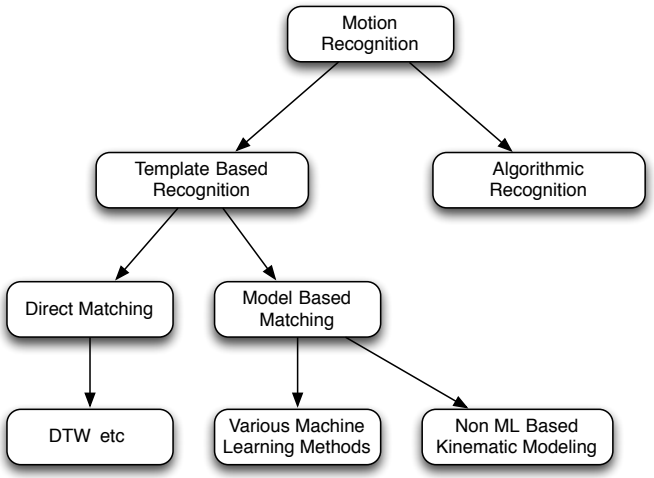

Fig. 8. Classification of gesture recognition methods.

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.

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

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.[47] 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.[149] 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.[85] 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.[137] For static gestures, the distance between two gestures can be calculated using an algorithm called Earth Mover's Distance. [105]

4.2.1. *DTW*

Doliotis *et al.* proposed to use DTW to recognize hand gestures (digits recognition in particular).[30] 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.[130] 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.[117] 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 fussy 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.[121] 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 slop 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.[137]

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.[99,100] 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.[127] 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. *Machining-Learning-Based Motion Recognition*

Machine-learning-based motion recognition typically relies on one or more sophisticated statistical models, such the Hidden Markov Model (HMM),[9] Artificial Neural Networks (ANNs),[82,103] Support Vector Machine (SVM),[28] 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.[13]

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)[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 (*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 *et al.* proposed to use the dynamic features instead of kinematic features for human action recognition using HMM.[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 *et al.* employed HMM to classify hand gesture sequences for real-time navigation of a robot using human hand gestures.[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.[119] 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.[144] 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.[82,103] 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[106] has been used to classify static gestures using Kinect data.[8] 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.[89]

Hafiz *et al.* used a single-layered complex-valued neural network (CVNN) for

static hand gesture recognition.[48] 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.[28] 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).[126] 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.[77] 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.[18] 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 are 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.[116]

Miranda *et al.* used the decision forest algorithm to identify gestures in real-time on Kinect motion data.[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.[12,45] However, due to changes in the orientation of the body during movement, the accuracy of recognition is reduced.

4.4.5. *Adaboost*

Adaboost refers to a meta-algorithm for machine learning called Adaptive Boosting.[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.[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. *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.[13] 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),[59] 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.[1] 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.[120] 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.[72] 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.[110] 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 10. Summary of publicly available datasets.

Dataset	Dataset Webpage
Indoor scenes[109,108]	http://cs.nyu.edu/ silberman/datasets/
Human actions[65,132,133,141]	http://research.microsoft.com/en-us/ um/people/zliu/actionrecorsrc/
Hand gestures[73]	http://lshao.staff.shef.ac.uk/data/ SheffieldKinectGesture.htm
Object tracking[111]	http://tracking.cs.princeton.edu/dataset.html
3D reconstruction[139]	http://sun3d.cs.princeton.edu/
Category modeling[145]	http://shiba.iis.u-tokyo.ac.jp/song/?page_id=343
3D scans[31]	http://vcl.iti.gr/3d-scans/
Gestures[37]	http://research.microsoft.com/en-us/um/ cambridge/projects/msrc12/
Human actions[138]	http://cvrc.ece.utexas.edu/KinectDatasets/HOJ3D.html
Unstructured human activity[63,119]	http://pr.cs.cornell.edu/humanactivities/data.php
Scene in hall way[74,112]	http://www2.informatik.uni-freiburg.de/ ~spinello/RGBD-dataset.html
Various objects[54,55]	http://kinectdata.com/

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.*

Type	Kinect v1	Kinect v2
Types of indoor scenes	7	26
Number of scenes	64	464
Unlabeled frames	108,617	407,024
Densely labeled frames	2347	1449
Number of classes	Over 1000	Over 1000

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.[31] 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.[37]

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.[138]

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.[63,119]

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.[74,112]

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.[54,55]

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 their 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

1. Visual gesture builder: A data-driven solution to gesture detection. https://msdn.microsoft.com/en-us/library/dn785529.aspx, July 2014.
2. M. Abdur Rahman, A. M. Qamar, M. A. Ahmed, M. Ataur Rahman, and S. Basalamah. Multimedia interactive therapy environment for children having physical disabilities. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, ICMR '13, pages 313–314. ACM, New York, NY, USA, 2013. ISBN 978-1-4503-2033-7. URL `http://doi.acm.org/10.1145/2461466.2461522`.
3. A. Agarwal and M. K. Thakur. Sign language recognition using microsoft kinect. In *Contemporary Computing (IC3), 2013 Sixth International Conference on*, pages 181–185. IEEE, 2013.
4. K. Aitpayev and J. Gaber. Collision avatar (ca): Adding collision objects for human body in augmented reality using kinect. In *Application of Information and Communication Technologies (AICT), 2012 6th International Conference on*, pages 1–4, 2012.
5. E. Akdoğan, E. Taçgın, and M. A. Adli. Knee rehabilitation using an intelligent robotic system. *Journal of Intelligent Manufacturing*, 20(2):195–202, 2009.
6. D. Alexiadis, D. Zarpalas, and P. Daras. Real-time, full 3-d reconstruction of moving foreground objects from multiple consumer depth cameras. *Multimedia, IEEE Transactions on*, 15(2):339–358, 2013. ISSN 1520-9210.
7. S. G. M. Almeida, F. G. Guimarães, and J. A. Ramírez. Feature extraction in brazilian sign language recognition based on phonological structure and using rgb-d sensors. *Expert Systems with Applications*, 41(16):7259–7271, 2014.
8. M. d. S. Anjo, E. B. Pizzolato, and S. Feuerstack. A real-time system to recognize static gestures of brazilian sign language (libras) alphabet using kinect. In *Proceedings of the 11th Brazilian Symposium on Human Factors in Computing Systems*,

IHC '12, pages 259–268. Brazilian Computer Society, Porto Alegre, Brazil, Brazil, 2012. ISBN 978-85-7669-262-1.
9. L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, pages 1554–1563, 1966.
10. B. C. Bedregal, A. C. Costa, and G. P. Dimuro. Fuzzy rule-based hand gesture recognition. In *Artificial Intelligence in Theory and Practice*, pages 285–294. Springer, 2006.
11. D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of hte Workshop on Knowledge Discovery in Databases*, volume 10, pages 359–370. Seattle, WA, 1994.
12. Z.-P. Bian, L.-P. Chau, and N. Magnenat-Thalmann. Fall detection based on skeleton extraction. In *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*, VRCAI '12, pages 91–94. ACM, New York, NY, USA, 2012. ISBN 978-1-4503-1825-9.
13. A. Bigdelou, T. Benz, L. Schwarz, and N. Navab. Simultaneous categorical and spatio-temporal 3d gestures using kinect. In *3D User Interfaces (3DUI), 2012 IEEE Symposium on*, pages 53–60, 2012.
14. A. Bo, M. Hayashibe, and P. Poignet. Joint angle estimation in rehabilitation with inertial sensors and its integration with kinect. In *Engineering in Medicine and Biology Society,EMBC, 2011 Annual International Conference of the IEEE*, pages 3479–3483, 2011. ISSN 1557-170X.
15. M. N. K. Boulos, B. J. Blanchard, C. Walker, J. Montero, A. Tripathy, and R. Gutierrez-Osuna. Web gis in practice x: a microsoft kinect natural user- google earth. *International Journal of Health Geographics*, 10(45):2566 – 2570, 2011. ISSN 0891-4222.
16. J. Boyd, A. Godbout, and C. Thornton. In situ motion capture of speed skating: Escaping the treadmill. In *Computer and Robot Vision (CRV), 2012 Ninth Conference on*, pages 460–467, 2012.
17. A. Bragdon, R. DeLine, K. Hinckley, and M. R. Morris. Code space: touch + air gesture hybrid interactions for supporting developer meetings. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '11, pages 212–221. ACM, New York, NY, USA, 2011. ISBN 978-1-4503-0871-7.
18. L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.
19. C.-Y. Chang, B. Lange, M. Zhang, S. Koenig, P. Requejo, N. Somboon, A. Sawchuk, and A. Rizzo. Towards pervasive physical rehabilitation using microsoft kinect. In *Proceedings of the 6th International Conference on Pervasive Computing Technologies for Healthcareon*, pages 159–162, 2012.
20. Y.-J. Chang, S.-F. Chen, and J.-D. Huang. A kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. *Research in Developmental Disabilities*, 32(6):2566 – 2570, 2011. ISSN 0891-4222.
21. J. Chapinal Cervantes, F. L. G. Vela, and P. P. Rodrĺguez. Natural interaction techniques using kinect. In *Proceedings of the 13th International Conference on Interacción Persona-Ordenador*, INTERACCION '12, pages 14:1–14:2. ACM, New York, NY, USA, 2012. ISBN 978-1-4503-1314-8.
22. A. Chatterjee, S. Jain, and V. M. Govindu. A pipeline for building 3d models using depth cameras. In *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing*, ICVGIP '12, pages 38:1–38:8. ACM, New York, NY, USA, 2012. ISBN 978-1-4503-1660-6.
23. S. Chen, Y. Maeda, and Y. Takahashi. Music conductor gesture recognized interac-

tive music generation system. In *Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), 2012 Joint 6th International Conference on*, pages 840–845, 2012.

24. L. Cheng, Q. Sun, H. Su, Y. Cong, and S. Zhao. Design and implementation of human-robot interactive demonstration system based on kinect. In *Control and Decision Conference (CCDC), 2012 24th Chinese*, pages 971–975, 2012.
25. R. A. Clark, Y.-H. Pua, A. L. Bryant, and M. A. Hunt. Validity of the microsoft kinect for providing lateral trunk lean feedback during gait retraining. *Gait & posture*, 38(4):1064–1066, 2013.
26. R. A. Clark, Y.-H. Pua, K. Fortin, C. Ritchie, K. E. Webster, L. Denehy, and A. L. Bryant. Validity of the microsoft kinect for assessment of postural control. *Gait and posture*, 36(3):372–377, 2012. ISSN 1520-9210.
27. S. Connell, P.-Y. Kuo, L. Liu, and A. M. Piper. A wizard-of-oz elicitation study examining child-defined gestures with a whole-body interface. In *Proceedings of the 12th International Conference on Interaction Design and Children*, IDC '13, pages 277–280. ACM, New York, NY, USA, 2013. ISBN 978-1-4503-1918-8.
28. C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
29. C. Demerjian. A long look at microsofts xbox one kinect sensor. http://semiaccurate.com/2013/10/15/long-look-microsofts-xbox-one-kinect-sensor/, October 2013.
30. P. Doliotis, A. Stefan, C. McMurrough, D. Eckhard, and V. Athitsos. Comparing gesture recognition accuracy using color and depth information. In *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '11, pages 20:1–20:7. ACM, New York, NY, USA, 2011. ISBN 978-1-4503-0772-7.
31. A. Doumanoglou, S. Asteriadis, D. Alexiadis, D. Zarpalas, and P. Daras. A dataset of kinect-based 3d scans. In *IVMSP Workshop, 2013 IEEE 11th*, pages 1–4, June 2013.
32. T. Dutta. Evaluation of the kinect sensor for 3-d kinematic measurement in the workplace. *Applied Ergonomics*, 43(4):645 – 649, 2012. ISSN 0003-6870.
33. R. El-laithy, J. Huang, and M. Yeh. Study on the use of microsoft kinect for robotics applications. In *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, pages 1280–1288, 2012. ISSN 2153-358X.
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/`.
35. S. Farag, W. Abdelrahman, D. C. Creighton, and S. Nahavandi. Extracting 3d mesh skeletons using antipodal points locations. In *UKSim*, pages 135–139, 2013.
36. F. Farhadi-Niaki, R. GhasemAghaei, and A. Arya. Empirical study of a vision-based depth-sensitive human-computer interaction system. In *Proceedings of the 10th asia pacific conference on Computer human interaction*, APCHI '12, pages 101–108. ACM, New York, NY, USA, 2012. ISBN 978-1-4503-1496-1.
37. S. Fothergill, H. Mentis, P. Kohli, and S. Nowozin. Instructing people for training gestural interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1737–1746. ACM, 2012.
38. R. Francese, I. Passero, and G. Tortora. Wiimote and kinect: gestural user interfaces add a natural third dimension to hci. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, pages 116–123. ACM, New York, NY, USA, 2012. ISBN 978-1-4503-1287-5.

39. T. Franke, S. Kahn, M. Olbrich, and Y. Jung. Enhancing realism of mixed reality applications through real-time depth-imaging devices in x3d. In *Proceedings of the 16th International Conference on 3D Web Technology*, Web3D '11, pages 71–79. ACM, New York, NY, USA, 2011. ISBN 978-1-4503-0774-1.
40. Y. Freund and R. E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.
41. G. Galatas, G. Potamianos, and F. Makedon. Audio-visual speech recognition using depth information from the kinect in noisy video conditions. In *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '12, pages 2:1–2:4. ACM, New York, NY, USA, 2012. ISBN 978-1-4503-1300-1.
42. L. Gallo, A. Placitelli, and M. Ciampi. Controller-free exploration of medical image data: Experiencing the kinect. In *Computer-Based Medical Systems (CBMS), 2011 24th International Symposium on*, pages 1–6, 2011. ISSN 1063-7125.
43. M. Geetha, C. Manjusha, P. Unnikrishnan, and R. Harikrishnan. A vision based dynamic gesture recognition of indian sign language on kinect based depth images. In *Emerging Trends in Communication, Control, Signal Processing & Computing Applications (C2SPCA), 2013 International Conference on*, pages 1–7. IEEE, 2013.
44. L. Geng, X. Ma, B. Xue, H. Wu, J. Gu, and Y. Li. Combining features for chinese sign language recognition with kinect. In *Control & Automation (ICCA), 11th IEEE International Conference on*, pages 1393–1398. IEEE, 2014.
45. R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 415–422, 2011. ISSN 1550-5499.
46. M. Gotsis, V. Lympouridis, D. Turpin, A. Tasse, I. Poulos, D. Tucker, M. Swider, A. Thin, and M. Jordan-Marsh. Mixed reality game prototypes for upper body exercise and rehabilitation. In *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*, pages 181–182, 2012. ISSN 1087-8270.
47. T. Hachaj and M. R. Ogiela. Rule-based approach to recognizing human body poses and gestures in real time. *Multimedia Systems*, 20(1):81–99, 2014.
48. A. R. Hafiz, M. F. Amin, and K. Murase. Real-time hand gesture recognition using complex-valued neural network (cvnn). In *Neural Information Processing*, pages 541–549. Springer, 2011.
49. H. Hai, L. Bin, H. Benxiong, and C. Yi. Interaction system of treadmill games based on depth maps and cam-shift. In *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, pages 219–222, 2011.
50. J. Han, L. Shao, D. Xu, and J. Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *Cybernetics, IEEE Transactions on*, 43(5):1318–1334, 2013. ISSN 2168-2267.
51. O. Hilliges, D. Kim, S. Izadi, M. Weiss, and A. Wilson. Holodesk: direct 3d interactions with a situated see-through display. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, pages 2421–2430. ACM, New York, NY, USA, 2012. ISBN 978-1-4503-1015-4.
52. C. Hoilund, V. Kruger, and T. Moeslund. Evaluation of human body tracking system for gesture-based programming of industrial robots. In *Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on*, pages 477–480, 2012.
53. M.-C. Huang, W. Xu, Y. Su, B. Lange, C.-Y. Chang, and M. Sarrafzadeh. Smartglove for upper extremities rehabilitative gaming assessment. In *Proceedings of the*

5th International Conference on PErvasive Technologies Related to Assistive Environments, PETRA '12, pages 20:1–20:4. ACM, New York, NY, USA, 2012. ISBN 978-1-4503-1300-1.

54. A. Janoch, S. Karayev, Y. Jia, J. Barron, M. Fritz, K. Saenko, and T. Darrell. A category-level 3-d object dataset: Putting the kinect to work. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1168–1174, Nov 2011.
55. A. Janoch, S. Karayev, Y. Jia, J. Barron, M. Fritz, K. Saenko, and T. Darrell. A category-level 3d object dataset: Putting the kinect to work. In A. Fossati, J. Gall, H. Grabner, X. Ren, and K. Konolige, editors, *Consumer Depth Cameras for Computer Vision*, Advances in Computer Vision and Pattern Recognition, pages 141–165. Springer London, 2013. ISBN 978-1-4471-4639-1.
56. W. Jia, W.-J. Yi, J. Saniie, and E. Oruklu. 3d image reconstruction and human body tracking using stereo vision and kinect technology. In *Electro/Information Technology (EIT), 2012 IEEE International Conference on*, pages 1–4, 2012. ISSN 2154-0357.
57. Y. Jiang, J. Tao, W. Ye, W. Wang, and Z. Ye. An isolated sign language recognition system using rgb-d sensor with sparse coding. In *Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on*, pages 21–26. IEEE, 2014.
58. R. Johnson, K. O'Hara, A. Sellen, C. Cousins, and A. Criminisi. Exploring the potential for touchless interaction in image-guided interventional radiology. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 3323–3332. ACM, New York, NY, USA, 2011. ISBN 978-1-4503-0228-9.
59. I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.
60. A. Kadambi, A. Bhandari, and R. Raskar. 3d depth cameras in vision: Benefits and limitations of the hardware. In *Computer Vision and Machine Learning with RGB-D Sensors*, pages 3–26. Springer, 2014.
61. J.-w. Kang, D.-j. Seo, and D.-s. Jung. A study on the control method of 3-dimensional space application using kinect system. *IJCSNS International Journal of Computer Science and Network Security*, 11(9):55–59, 2011. ISSN 1738-7906.
62. N. Kolakowski. Microsoft acquires Canesta, 3D tech patents. http://www.eweek.com/c/a/Windows/Microsoft-Acquires-Canesta-3D-Tech-Patents-342975/, November 2010.
63. H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32 (8):951–970, 2013.
64. O. Kreylos. The kinect 2.0. http://doc-ok.org/?p=584, May 2013.
65. A. Kurakin, Z. Zhang, and Z. Liu. A real time system for dynamic hand gesture recognition with a depth sensor. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 1975–1979. IEEE, 2012.
66. S. Lang, M. Block, and R. Rojas. Sign language recognition using kinect. In *Artificial Intelligence and Soft Computing*, pages 394–402. Springer, 2012.
67. B. Lange, C.-Y. Chang, E. Suma, B. Newman, A. Rizzo, and M. Bolas. Development and evaluation of low cost game-based balance rehabilitation tool using the microsoft kinect sensor. pages 1831–1834, 2011. ISSN 1557-170X.
68. B. Lange, S. Koenig, E. McConnell, C. Chang, R. Juang, E. Suma, M. Bolas, and A. Rizzo. Interactive game-based rehabilitation using the microsoft kinect. In *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*, pages 171–172, 2012. ISSN 1087-8270.

69. G. C. Lee, F.-H. Yeh, and Y.-H. Hsiao. Kinect-based taiwanese sign-language recognition system. *Multimedia Tools and Applications*, pages 1–19, 2014.
70. L. Leite and V. Orvalho. Shape your body: control a virtual silhouette using body motion. In *Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts*, CHI EA '12, pages 1913–1918. ACM, New York, NY, USA, 2012. ISBN 978-1-4503-1016-1.
71. D. Liebling and M. R. Morris. Kinected browser: depth camera interaction for the web. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces*, ITS '12, pages 105–108. ACM, New York, NY, USA, 2012. ISBN 978-1-4503-1209-7.
72. S.-Y. Lin, C.-K. Shie, S.-C. Chen, M.-S. Lee, and Y.-P. Hung. Human action recognition using action trait code. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3456–3459, 2012. ISSN 1051-4651.
73. L. Liu and L. Shao. Learning discriminative representations from rgb-d video data. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 1493–1500. AAAI Press, 2013. ISBN 978-1-57735-633-2.
74. M. Luber, L. Spinello, and K. O. Arras. People tracking in rgb-d data with on-line boosted target models. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3844–3849. IEEE, 2011.
75. Y. M. Lui. A least squares regression framework on manifolds and its application to gesture recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 13–18, 2012. ISSN 2160-7508.
76. E. Machida, M. Cao, T. Murao, and H. Hashimoto. Human motion tracking of mobile robot with kinect 3d sensor. In *SICE Annual Conference (SICE), 2012 Proceedings of*, pages 2207–2211, 2012. ISSN pending.
77. R. C. B. Madeo, C. A. M. Lima, and S. M. Peres. Gesture unit segmentation using support vector machines: segmenting gestures from rest positions. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, pages 46–52. ACM, New York, NY, USA, 2013. ISBN 978-1-4503-1656-9.
78. A. Mansur, Y. Makihara, and Y. Yagi. Inverse dynamics for action recognition. *Cybernetics, IEEE Transactions on*, 43(4):1226–1236, 2013. ISSN 2168-2267.
79. C. Martin, D. Burkert, K. Choi, N. Wieczorek, P. McGregor, R. Herrmann, and P. Beling. A real-time ergonomic monitoring system using the microsoft kinect. In *Systems and Information Design Symposium (SIEDS), 2012 IEEE*, pages 50–55, 2012.
80. S. Masood, M. P. Qureshi, M. B. Shah, S. Ashraf, Z. Halim, and G. Abbas. Dynamic time wrapping based gesture recognition. In *Robotics and Emerging Allied Technologies in Engineering (iCREATE), 2014 International Conference on*, pages 205–210. IEEE, 2014.
81. G. Mastorakis and D. Makris. Fall detection system using kinects infrared sensor. *Journal of Real-Time Image Processing*, pages 1–12, 2012. ISSN 1861-8200.
82. W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
83. A. Memiş and S. Albayrak. A kinect based sign language recognition system using spatio-temporal features. In *Sixth International Conference on Machine Vision (ICMV 13)*, page 90670X. International Society for Optics and Photonics, 2013.
84. L. Miranda, T. Vieira, D. Martinez, T. Lewiner, A. W. Vieira, and M. F. M. Campos. Real-time gesture recognition from depth data through key poses learning and decision forests. *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images*,

0:268–275, 2012. ISSN 1530-1834.
85. C. Myers, L. Rabiner, and A. E. Rosenberg. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(6):623–635, 1980.
86. I. Nakachi, Y. Takeuchi, and D. Katagami. Perception analysis of motion contributing to individuality using kinect sensor. In *RO-MAN, 2012 IEEE*, pages 308–313, 2012. ISSN 1944-9445.
87. V. V. Nguyen and J.-H. Lee. Full-body imitation of human motions with kinect and heterogeneous kinematic structure of humanoid robot. In *System Integration (SII), 2012 IEEE/SICE International Symposium on*, pages 93–98, 2012.
88. B. Ni, N. C. Dat, and P. Moulin. Rgbd-camera based get-up event detection for hospital fall prevention. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 1405–1408, 2012. ISSN 1520-6149.
89. S. Nomm and K. Buhhalko. Monitoring of the human motor functions rehabilitation by neural networks based system with kinect sensor. In *Analysis, Design, and Evaluation of Human-Machine Systems*, volume 12, pages 249–253, 2013.
90. C. O'Connor, A. Davy, and B. Jennings. Controlling the transfer of kinect data to a cloud-hosted games platform. In *Proceeding of the 23rd ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, NOSSDAV '13, pages 55–60. ACM, New York, NY, USA, 2013. ISBN 978-1-4503-1892-1.
91. M. Oszust and M. Wysocki. Recognition of signed expressions observed by kinect sensor. In *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*, pages 220–225. IEEE, 2013.
92. G. Parra-Dominguez, B. Taati, and A. Mihailidis. 3d human motion analysis to detect abnormal events on stairs. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 97–103, 2012.
93. O. Patsadu, C. Nukoolkit, and B. Watanapa. Human gesture recognition using kinect camera. In *Computer Science and Software Engineering (JCSSE), 2012 International Joint Conference on*, pages 28–32, 2012.
94. L. Pedro and G. de Paula Caurin. Kinect evaluation for human body movement analysis. In *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS EMBS International Conference on*, pages 1856–1861, 2012. ISSN 2155-1774.
95. M. Popa, A. Koc, L. Rothkrantz, C. Shan, and P. Wiggers. Kinect sensing of shopping related actions. In undefined, K. Van Laerhoven, and J. Gelissen, editors, *Constructing Ambient Intelligence: AmI 2011 Workshops*. Amsterdam, Netherlands, 11 2011.
96. N. Pugeault and R. Bowden. Spelling it out: Real-time asl fingerspelling recognition. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1114–1119. IEEE, 2011.
97. M. Raj, S. H. Creem-Regehr, K. M. Rand, J. K. Stefanucci, and W. B. Thompson. Kinect based 3d object manipulation on a desktop display. In *Proceedings of the ACM Symposium on Applied Perception*, SAP '12, pages 99–102. ACM, New York, NY, USA, 2012. ISBN 978-1-4503-1431-2.
98. E. Rakun, M. Andriani, I. W. Wiprayoga, K. Danniswara, and A. Tjandra. Combining depth image and skeleton data from kinect for recognizing words in the sign system for indonesian language (sibi [sistem isyarat bahasa indonesia]). In *Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on*, pages 387–392. IEEE, 2013.
99. Z. Ren, J. Meng, and J. Yuan. Depth camera based hand gesture recognition and its

applications in human-computer-interaction. In *Information, Communications and Signal Processing (ICICS) 2011 8th International Conference on*, pages 1–5, 2011.
100. Z. Ren, J. Yuan, J. Meng, and Z. Zhang. Robust part-based hand gesture recognition using kinect sensor. *Multimedia, IEEE Transactions on*, 15(5):1110–1120, 2013. ISSN 1520-9210.
101. L. Rioux-Maldague and P. Giguere. Sign language fingerspelling classification from depth and color images using a deep belief network. In *Computer and Robot Vision (CRV), 2014 Canadian Conference on*, pages 92–97. IEEE, 2014.
102. D. G. Rodrigues, E. Grenader, F. d. S. Nos, M. d. S. Dall'Agnol, T. E. Hansen, and N. Weibel. Motiondraw: a tool for enhancing art and performance using kinect. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pages 1197–1202. ACM, New York, NY, USA, 2013. ISBN 978-1-4503-1952-2.
103. F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
104. C. Rougier, E. Auvinet, J. Rousseau, M. Mignotte, and J. Meunier. Fall detection from depth map video sequences. In *Proceedings of the 9th international conference on Toward useful services for elderly and people with disabilities: smart homes and health telematics*, ICOST'11, pages 121–128. Springer-Verlag, Berlin, Heidelberg, 2011. ISBN 978-3-642-21534-6.
105. Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.
106. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.
107. S. Saini, D. Rambli, S. Sulaiman, M. Zakaria, and S. Shukri. A low-cost game framework for a home-based stroke rehabilitation system. In *Computer Information Science (ICCIS), 2012 International Conference on*, volume 1, pages 55–60, 2012.
108. N. Silberman, L. Shapira, R. Gal, and P. Kohli. A contour completion model for augmenting surface reconstructions. In *Computer Vision–ECCV 2014*, pages 488–503. Springer, 2014.
109. N. Silberman, D. Sontag, and R. Fergus. Instance segmentation of indoor scenes using a coverage loss. In *Computer Vision–ECCV 2014*, pages 616–631. Springer, 2014.
110. R. Sivalingam, G. Somasundaram, V. Bhatawadekar, V. Morellas, and N. Papanikolopoulos. Sparse representation of point trajectories for action classification. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3601–3606, 2012. ISSN 1050-4729.
111. S. Song and J. Xiao. Tracking revisited using rgbd camera: Unified benchmark and baselines. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 233–240, Dec 2013. ISSN 1550-5499.
112. L. Spinello and K. O. Arras. People detection in rgb-d data. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3838–3843. IEEE, 2011.
113. E. Stone and M. Skubic. Passive in-home measurement of stride-to-stride gait variability comparing vision and kinect sensing. In *Engineering in Medicine and Biology Society,EMBC, 2011 Annual International Conference of the IEEE*, pages 6491–6494, 2011. ISSN 1557-170X.
114. E. Stone and M. Skubic. Passive, in-home gait measurement using an inexpensive depth camera: Initial results. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2012 6th International Conference on*, pages 183–186, 2012.
115. J. Stowers, M. Hayes, and A. Bainbridge-Smith. Altitude control of a quadrotor

helicopter using depth map from microsoft kinect sensor. In *Mechatronics (ICM), 2011 IEEE International Conference on*, pages 358–362, 2011.
116. C. Strobl, J. Malley, and G. Tutz. An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological methods*, 14(4):323, 2009.
117. C.-J. Su. Personal rehabilitation exercise assistant with kinect and dynamic time warping. *International Journal of Information and Education Technology*, pages 448–454, 2013.
118. E. Suma, B. Lange, A. Rizzo, D. M. Krum, and M. Bolas. FAAST: the flexible action and articulated skeleton toolkit. In *IEEE Virtual Reality*, page 245246. Singapore, Mar. 2011.
119. J. Sung, C. Ponce, B. Selman, and A. Saxena. Unstructured human activity detection from rgbd images. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 842–849. IEEE, 2012.
120. T. T. Thanh, F. Chen, K. Kotani, and H.-B. Le. Extraction of discriminative patterns from skeleton sequences for human action recognition. In *Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2012 IEEE RIVF International Conference on*, pages 1–6, 2012.
121. J. Tian, C. Qu, W. Xu, and S. Wang. Kinwrite: Handwriting-based authentication using kinect. In *Proceedings of the 20th Annual Network and Distributed System Security Symposium*, 2013.
122. J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan. Scanning 3d full human bodies using kinects. *Visualization and Computer Graphics, IEEE Transactions on*, 18(4):643–650, 2012. ISSN 1077-2626.
123. X. Tong, P. Xu, and X. Yan. Research on skeleton animation motion data based on kinect. In *Computational Intelligence and Design (ISCID), 2012 Fifth International Symposium on*, volume 2, pages 347–350, 2012.
124. P. Trindade, J. Lobo, and J. P. Barreto. Hand gesture recognition using color and depth images enhanced with hand angular pose data. In *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on*, pages 71–76. IEEE, 2012.
125. P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1473–1488, 2008.
126. V. Vapnik. *The nature of statistical learning theory*. springer, 2000.
127. E. Velloso, A. Bulling, and H. Gellersen. Motionma: motion modelling and analysis by demonstration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1309–1318. ACM, 2013.
128. H. V. Verma, E. Aggarwal, and S. Chandra. Gesture recognition using kinect for sign language translation. In *Image Information Processing (ICIIP), 2013 IEEE Second International Conference on*, pages 96–100. IEEE, 2013.
129. N. Villaroman, D. Rowe, and B. Swan. Teaching natural user interaction using openni and the microsoft kinect sensor. In *Proceedings of the 2011 conference on Information technology education*, SIGITE '11, pages 227–232. ACM, New York, NY, USA, 2011. ISBN 978-1-4503-1017-8.
130. C. Waithayanon and C. Aporntewan. A motion classifier for microsoft kinect. In *Computer Sciences and Convergence Information Technology (ICCIT), 2011 6th International Conference on*, pages 727–731, 2011.
131. F. Wang, C. Tang, Y. Ou, and Y. Xu. A real-time human imitation system. In *Intelligent Control and Automation (WCICA), 2012 10th World Congress on*, pages

3692–3697, 2012.
132. J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. Robust 3d action recognition with random occupancy patterns. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision ECCV 2012*, Lecture Notes in Computer Science, pages 872–885. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-33708-6.
133. J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1290–1297. IEEE, 2012.
134. L. Wang, R. Villamil, S. Samarasekera, and R. Kumar. Magic mirror: A virtual handbag shopping system. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 19–24, 2012. ISSN 2160-7508.
135. Q. Wang, P. Turaga, G. Coleman, and T. Ingalls. Somatech: an exploratory interface for altering movement habits. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, pages 1765–1770. ACM, 2014.
136. M. Winkler, K. Hover, A. Hadjakos, and M. Muhlhauser. Automatic camera control for tracking a presenter during a talk. In *Multimedia (ISM), 2012 IEEE International Symposium on*, pages 471–476, 2012.
137. D. Wu, F. Zhu, and L. Shao. One shot learning gesture recognition from rgbd images. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 7–12, 2012. ISSN 2160-7508.
138. L. Xia, C. Chen, and J. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 20–27. IEEE, 2012.
139. J. Xiao, A. Owens, and A. Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1625–1632, Dec 2013. ISSN 1550-5499.
140. D. Xu, Y.-L. Chen, C. Lin, X. Kong, and X. Wu. Real-time dynamic gesture recognition system based on depth perception for robot navigation. In *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, pages 689–694, 2012.
141. G. Yu, Z. Liu, and J. Yuan. Discriminative orderlet mining for real-time recognition of human-object interaction. In *Proceedings of the 12th Asian Conference on Computer Vision*. Singapore, November 2014.
142. Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti. American sign language recognition with the kinect. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 279–286. ACM, 2011.
143. Z. Zalevsky, A. Shpunt, A. Malzels, and J. Garcia. Method and system for object reconstruction, Mar. 19 2013. US Patent 8,400,494.
144. L. Zhang, J.-C. Hsieh, and J. Wang. A kinect-based golf swing classification system using hmm and neuro-fuzzy. In *Computer Science and Information Processing (CSIP), 2012 International Conference on*, pages 1163–1166, 2012.
145. Q. Zhang, X. Song, X. Shao, R. Shibasaki, and H. Zhao. Category modeling from just a single labeling: Use depth information to guide the learning of 2d models. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 193–200, June 2013. ISSN 1063-6919.
146. Z. Zhang. Microsoft kinect sensor and its effect. *MultiMedia, IEEE*, 19(2):4–10, 2012. ISSN 1070-986X.
147. Z. Zhang, W. Liu, V. Metsis, and V. Athitsos. A viewpoint-independent statistical method for fall detection. In *ICPR*, pages 3626–3630. IEEE, 2012. ISBN 978-1-4673-

2216-4.

148. W. Zhao, H. Feng, R. Lun, D. D. Espy, and M. Reinthal. A kinect-based rehabilitation exercise monitoring and guidance systems. In *Proceedings of the 5th IEEE International Conference on Software Engineering and Service Science*, pages 762–765. IEEE, 2014.
149. W. Zhao, R. Lun, D. D. Espy, and M. A. Reinthal. Rule based realtime motion assessment for rehabilitation exercises. In *Proceedings of the IEEE Symposium on Computational Intelligence in Healthcare and e-Health*, pages 133–140, December 2014.
150. F. Zuher and R. Romero. Recognition of human motions for imitation and control of a humanoid robot. In *Robotics Symposium and Latin American Robotics Symposium (SBR-LARS), 2012 Brazilian*, pages 190–195, 2012.