2012 IEEE RO-MAN: The 21st IEEE International Symposium on
Robot and Human Interactive Communication.
September 9-13, 2012. Paris, France.

Hand Gesture Recognition with Depth Images: A Review

Jesus Suarez* and Robin R. Murphy, *Member, IEEE*

***Abstract*— This paper presents a literature review on the use of depth for hand tracking and gesture recognition. The survey examines 37 papers describing depth-based gesture recognition systems in terms of (1) the hand localization and gesture classification methods developed and used, (2) the applications where gesture recognition has been tested, and (3) the effects of the low-cost Kinect and OpenNI software libraries on gesture recognition research. The survey is organized around a novel model of the hand gesture recognition process. In the reviewed literature, 13 methods were found for hand localization and 11 were found for gesture classification. 24 of the papers included real-world applications to test a gesture recognition system, but only 8 application categories were found (and three applications accounted for 18 of the papers). The papers that use the Kinect and the OpenNI libraries for hand tracking tend to focus more on applications than on localization and classification methods, and show that the OpenNI hand tracking method is good enough for the applications tested thus far. However, the limitations of the Kinect and other depth sensors for gesture recognition have yet to be tested in challenging applications and environments.**

I. Introduction

Hand gesture recognition is used in human-robot interaction (HRI) to create user interfaces that are natural to use and easy to learn [1]. Sensors used for hand gesture recognition include wearable sensors such as data gloves and external sensors such as video cameras. Data gloves can provide accurate measurements of hand pose and movement, but they require extensive calibration, restrict natural hand movement, and are often very expensive. Video-based gesture recognition addresses these issues, but presents a new problem: locating the hands and segmenting them from the background in an image sequence is a non-trivial task, in particular when there are occlusions, lighting changes, rapid motion, or other skin-colored objects in a scene (see Mitra [2] and Erol [3] for reviews of video-based methods).

Depth images, either sensed directly with depth cameras such as the Microsoft Kinect, ASUS Xtion, or Mesa SwissRanger, or extracted from stereo video cameras, provide an alternative as they can function in several situations where video cameras cannot, such as in low or unpredictable lighting, and in environments with skin-colored objects other than the hands (such as a face). However, there is no comprehensive study of the state of practice, such as reported in Wachs [1] for applications of video-based hand gestures.

*Jesus Suarez is supported by the NSF Bridge to Doctorate Fellowship.

Jesus Suarez (jsuarez@cse.tamu.edu) and Robin R. Murphy (murphy@cse.tamu.edu) are with the Center for Robot-Assisted Search and Rescue, Texas A&M University, College Station, TX 77840 USA.

This paper surveys 37 papers that describe gesture types and classification using depth in order to answer three questions:

- What methods are being used to achieve hand localization and gesture recognition with depth cameras? Note that hand localization is in this context a computer vision problem (albeit with depth instead of video cameras) and gesture recognition is a machine learning problem.
- What applications and environments are researchers testing their methods in? Do they test them in situations where their depth-based methods have supposed advantages over video-based methods? Are the limitations of depth-based systems tested?
- How has the release of the Kinect and associated libraries affected research in gesture recognition? By far the most popular depth sensor used is the Microsoft Kinect (it is used by 22 of the papers reviewed) and it comes with software libraries to perform hand and body tracking. However, it is not clear whether these libraries are being used or replaced by custom algorithms, and if so, why.

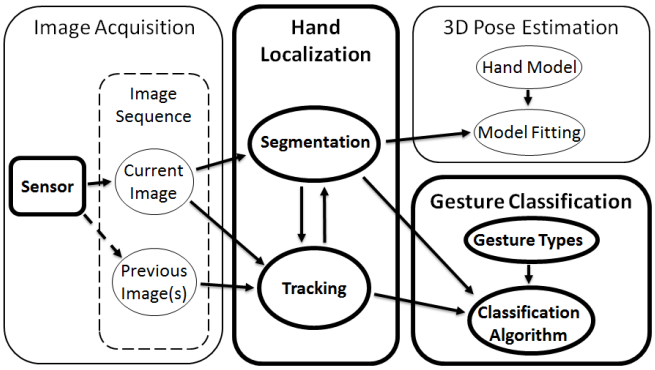

Figure 1: The components of a video- or depth-based hand gesture recognition and pose estimation system. This paper describes depth sensor types, hand localization methods, and gesture classification methods.

This survey is organized according to our novel conceptual model of the major components of a hand gesture recognition system, shown in Figure 1. Depth-based hand gesture recognition begins with depth-image acquisition, which is dependent on the sensor being used (Sec. II). Then hand localization is performed on the acquired image sequence using tracking and segmentation methods (Sec. III). Finally, the segmented hand images and/or their tracked trajectories are classified as a particular gesture or pose, using McNeill's gesture type taxonomy taken from the communication literature [4] (Sec. IV). Note that this review focuses on the components, shown in boldface, involved in gesture classification, which is the process that recognizes a

set of poses and gestures from a predetermined gesture set. It does not explore 3D hand pose estimation, shown in gray, which aims to reconstruct a full Degree-of-Freedom 3D model of the hand's posture. Sec. IV discusses the applications reported to date, and Sec. V details the conclusions.

II. Depth Sensors

Five types of depth cameras and systems are used for hand tracking and gesture recognition in the literature reviewed, but the most prominently used sensor is the Kinect from Microsoft – due primarily to its low cost. All of the sensors and camera systems produce a sequence of depth images that are then used for hand localization.

A. The Microsoft Kinect

The Microsoft Kinect includes a QVGA (320x240) depth camera and a VGA (640x480) video camera, both of which produce image streams at 30 frames per second (fps). The depth camera (developed by PrimeSense[1], and also used in the ASUS Xtion Pro[2]) works on the principle of structured light. An infrared (IR) light emitter projects a dense, non-uniform array of dots onto a scene, which are then detected by an IR camera. The spacing of these dots appears different for objects at different distances from the sensor. Since the pattern and spacing of the projected dots is known, internal processors can compare the spacing measured in the IR image to the known reference values, and compute the distance of each pixel in the scene. The sensor is limited by near and far thresholds for depth estimation, so the effective range is approximately 1.2 to 3.5 meters. Additionally, the sensor does not function well in bright sunlight because the overabundant light drowns out the projected IR dot pattern.

Microsoft developed the Kinect for full-body tracking to be used for interacting with games, videos and menus on the Xbox 360 game console. The proprietary body-tracking methods developed for the Kinect, as well as access to the depth and video streams, were made available by Microsoft in a closed-course Kinect SDK[3] (software development kit). An open-source alternative called the OpenNI[4] framework also provides access to the sensor streams. OpenNI was developed by a consortium of companies lead by PrimeSense, who also published the closed source NITE middleware module for OpenNI to perform body tracking and hand tracking similarly to Microsoft's SDK.

The availability of complete hand- and body-tracking solutions for the Kinect (and ASUS Xtion Pro) has meant that many gesture recognition researchers have focused on classification methods and applications rather than hand localization techniques. Of the papers reviewed, 22 use the Kinect as the depth sensor for gesture recognition [5-26], and of these 9 use OpenNI and the NITE middleware for hand tracking [10, 14-16, 18, 21, 22, 25, 26], as opposed to implementing a method of their own.

B. Other Depth Sensors and Systems

Apart from structured light cameras like the Kinect and Xtion Pro, the most popular depth sensor types used are Time of Flight (ToF) cameras and stereoscopic cameras. ToF cameras (such as the SwissRanger[5] from Mesa Imaging) determine pixel depths in one of two ways: by measuring the round-trip flight-time of light projected onto the scene and reflected back to the sensor, or by measuring the phase-shift of the reflected light. ToF cameras produce accurate depth images at a high frame rate (50 fps), but at a relatively low resolution (144x176). Nine of the papers reviewed use a ToF camera [27-35].

Stereoscopic camera systems capture two simultaneous images from a pair of calibrated video cameras, and use image registration methods to create a disparity map that approximates per-pixel depth. Stereo systems produce lower-fidelity depth images than ToF cameras, and they require computational overhead to solve the image registration problem for each image pair. However, stereo cameras work well in bright light and can be built with standard video cameras. Commercial stereoscopic camera systems, such as the Bumblebee[6] from Point Grey, are also available. Four of the papers reviewed use stereo camera systems [36-39].

Two papers used less common approaches for depth image acquisition. Malassiotis [40] used a coded light approach, which is a precursor to structured light methods. The method projects bands of light of different thicknesses onto an object and measures differences in the observed image. And Feris [41] and used a type of shadow analysis to infer depth information in an image. The method uses multiple light sources and a single camera to detect depth discontinuities, which appear as shadows in the collected images. Both of these methods rely on carefully controlled lighting for the scene.

III. Hand-Localization Methods

The problem of locating the hands in an image is essential to gesture recognition and is split into two sub-problems: hand segmentation, and hand tracking. Hand segmentation is the problem of determining which pixels in an image belong to a hand, and hand tracking is the problem of determining the trajectory of a hand in a sequence of images. The papers reviewed describe a variety of approaches that take advantage of depth information to solve these problems.

A. Hand segmentation

The advantage of depth cameras over color cameras for gesture recognition is perhaps most evident when performing hand segmentation. In applications where the user is expected to face the depth camera and hold their hands out in front of themselves for gesturing, it is common to simply use a depth threshold to isolate the hands. This method is used directly by Zhenyao [27], Breuer [28], Xia [30], Uebersax [34], Yoo [35], Du [7], Frati [17], Ren [11] and Trigueiros [24]. Depth thresholding determines the hands to be those points between some near and far distance thresholds around the Z (depth) value of the expected centroid of the hand – which can be either predetermined and instructed to the user, or determined

[1] http://www.primesense.com
[2] http://www.asus.com/Multimedia/Motion_Sensor/Xtion_PRO
[3] http://www.microsoft.com/en-us/kinectforwindows/
[4] http://www.openni.org
[5] http://www.mesa-imaging.ch/prodview4k.php
[6] http://www.ptgrey.com/products/stereo.asp

as the nearest point in the scene. An effective method to reduce susceptibility to noise is to also place bounds on the area of the detected hand – that is, place limits on the number of pixels expected in the blob segmented by depth thresholding. Li [29] and Klompmaker [20] both use this modification. Another modification to depth thresholding is to determine the predicted hand depth according to the location of other body parts, rather than assuming the hand is necessarily the closest object in the scene. Cerlinca [36] and Van den Bergh [31] both use the robust face detector available in OpenCV[7] (an implementation of the very popular Viola-Jones object-detection method [42]) to determine the head location and then estimate candidate hand locations. Fujimura [33] used body detection based on Karhunen-Loeve Decomposition to estimate hand locations. Biswas [5] used depth thresholding to segment bodies from the background in depth images, but used depth histograms rather than hard cutoff values to ensure continuous regions and reduce noise.

Without the use of depth information the most common methods for hand segmentation are skin-color maps [2, 3] and cascaded classifiers on Haar-like features [43, 44]. Skin-color-based segmentation suffers most significantly from a weakness against lighting changes, even when using a illumination-invariant color scheme. However, Oikonomidis [8] and Tang [12] combined skin color and depth threshold to achieve better hand segmentation.

Another set of hand segmentation methods seen in the literature are clustering and region growing. Clustering works by combining near points into contiguous regions, and region growing works by seeding a point inside the desired area and looking for connected points to grow and fill the region. Region growing works well for segmentation in depth images because a free-moving hand in a depth images can be expected to have depth discontinuities at its border, which would tightly bound the growing region to the hand only. Droeschel [32] uses region growing seeded by a face detector, then segments the arms from the body by estimating the diameter of the torso and removing it, and finally identifying the hands in the arm regions as the points reached latest from the head by the region growing process. Chen [6] used estimated position of the hand according to the previous frame to seed a region growing method for hand segmentation. And Malassiotis [40] used a hierarchical clustering method to segment the hand and arms together, then used statistical modeling to separate the hand from the arm.

Less general approaches to hand segmentation can take advantage of certain conditions in an application. Jojic [38] exploits a special case where the appearance of the background is known, and so uses static background subtraction for segmentation. Park [9], Raheja [10] and Yang [13] ask the user to wave their hand at the beginning of an interaction and use motion images (the accumulated difference between successive frames) to determine the hand's location as an area of high motion then use depth thresholding at that point for segmentation. And Wang [37] and Feris [41] use shadow analysis to find depth discontinuities and thus create a silhouette of the hand; however, this method only works well in very controlled lighting.

B. Hand Tracking

Hand tracking captures temporal as well as spatial information and so is well suited as a precursor to dynamic gesture recognition. The Kinect in particular was developed for full-body motion tracking, and both the OpenNI framework (via the NITE middleware) and the Microsoft Kinect SDK provide fully articulated 20 point body tracking with nodes for each hand. Perhaps because it was available earlier, the NITE body tracking solution is more popular in the research community. NITE also provides waist-up body tracking (useful for gesturing while sitting down) and hand detection via a wave gesture (similar to [9, 10, 13]). Unlike the official Kinect SDK though, NITE does require a calibration pose to initialize body tracking[8]. Avancini [14], Bellmore [15], Chang [16], Frati [17], Hassani [18], Lai [21], Ramey [22], and Zafrulla [26] all directly use the NITE body tracking solution to track the position of the hands in a sequence.

Six papers reviewed implemented their own hand tracking method. The tracking algorithms used were the Kalman filter (used by Park [9] and Trigueiros [24]), CAMSHIFT (used by Yoo [35] and Yang [13]), and mean shift (used by Chen [6] and Keskin [19]). The Kalman filter is a recursive least squares estimate of the state of a dynamic system. For gesture recognition, the state being estimated is the position and orientation of the hand in subsequent frames. Mean shift is an iterative mode-finding algorithm that uses gradient descent to estimate the direction and velocity an area's movement (a hand in this situation). Mean shift works well for deformable objects, making it well suited for hand tracking. CAMSHIFT (continuous adaptive mean shift) is an extension of mean shift that dynamically adapts the region size being compared, thus making it robust to scaling changes.

IV. Gesture Recognition Methods

Once a hand's position and shape are known, its motion or shape deformations can be used as inputs to a classification algorithm that generates predictions of a gesture being performed or a posture being held. This section describes the types of gestures and postures classified by papers reviewed, and the classification methods used.

A. Gesture Types

Gestures, according to McNeill [4], fall broadly into four categories: gesticulations (which accompany speech and are often used for emphasis), emblems (which form part of a mutually understood gesture code, e.g. the "OK" sign), pantomimes (which are used in the absence of speech but are not part of a "code"), and sign language (which replaces speech altogether). In this context, gesture recognition for HCI and HRI is primarily split between using gesticulations, emblems and pantomimes to create new forms of interaction, and attempting to achieve automated sign language

[7] http://opencv.willowgarage.com/wiki/FaceDetection

[8] This restriction is no longer present as of NITE version 1.5.0.1 (released December, 2011). The reviewed papers that used NITE were published before this point.

recognition for the deaf and thus enable natural communication for a set of users.

This split between communication and interaction is evident in the papers reviewed. Accounting for the communication side of gesture recognition, six papers applied classification on sign language. Malassiotis [40] classified letters from the Greek sign language alphabet, and Feris [41] and Keskin [19] did the same for American Sign Language (ASL). Fujimura [33] and Zafrulla [26] implemented gesture recognition of sign language words in Japanese and English, respectively. And Uebersax [34] used both letters and words in ASL. On the interaction side, the most prominent gesture type recognized in the reviewed literature is pointing. Jojic [38], Droeschel [32] and Van den Bergh [25] all feature pointing gestures in their work. Pointing is a gesticulation-type gesture because it can accompany speech and be used to convey information (about direction in this case) better than the verbal equivalent. Pointing gestures are, in fact, common enough that they have their own gesture category: deictic gestures. For HRI in particular, deictic gestures can be intuitively used to direct a robot's movement, or guide its attention – much as we do with other people.

Other gesture types in the reviewed papers were more general-purpose. Finger counting gestures appeared in the works of Zhenyao [27], Li [29], Liu [30], Du [7], and Ren [11], and gestures for menu navigation were seen in papers by Van den Bergh [31], Konda [39], Biswas [5], Tang [12], Yang [13], Lai [21], Bellmore [15], Hassani [18], and Riener [23]. These two gesture types can be applied to a variety of applications, and it is clear this was a motivation for their selection by the authors. In McNeill's gesture taxonomy, finger counting gestures are classified as emblems, because they have a common verbal equivalent to everyone, and the spoken and gestural versions are generally interchangeable in common use, while gestures for menu navigation are pantomimes because they are used in favor of speech rather than as an equal alternative to a verbal counterpart. For example, using a swiping gesture to navigate up and down through a list could conceivably be replaced by the vocal commands "up" and "down," but it is not a reasonable expectation that users would consider the two interface types equal. Some interface tasks are better suited for gestures.

The remaining gesture types observed in the literature review were very application-specific, such as Wang's [37] set of boxing moves for a video game, and Chang's [16] set of arm movements for exercise. Additionally, though it was not the focus of this survey, four reviewed papers applied model-based pose estimation in lieu of or in addition to classifying gestures. Breuer [28] performed a 7 DOF model estimation using Principle Components Analysis (PCA) and an iterative optimization method for model fitting based on a Hausdorff distance objective function. Oikonomidis [8] produced a full 26 DOF hand model on depth data using Particle Swarm Analysis (PSO). And Keskin [19] adapted Shotton's highly robust pose estimation method (which is what is used by the official Kinect SDK and the Xbox 360) for hands, then used the extracted skeleton joints as features for recognizing the set of ten numeric digits in ASL.

B. Gesture Classification

Once the appropriate hand features have been extracted from the depth image (such as the location of the hand centroid or fingertips, or the segmented hand silhouette or depth sub-image), and once a gesture set has been selected, gesture classification can be accomplished by standard machine learning classifiers or a special-purpose classifier that takes advantage of the features selected.

Hidden Markov Models (HMMs) are used for data containing temporal information and they are known to have high classification rates, and so are quite popular for classifying dynamic gestures. Wang [37], Tang [12], Yang [13], Hassani [18], and Zafrulla [26] all use HMMs in a straightforward manner for gesture classification. Droeschel [32] uses HMMs to detect pointing gestures, but then uses Gaussian Process Regression to estimate pointing direction. Similarly, k-Nearest Neighbors (k-NN) classifiers are popular for static poses because of their high classification rates despite being very simple to implement. Malassiotis [40] uses a pure k-NN pose classifier, but Van den Berge [25, 31] and Feris [41] opted to apply some preprocessing first. Van den Berge uses Average Neighborhood Margin Maximization (ANMM) for dimensionality reduction, and Feris uses a histogram-based shape descriptor to extract good features.

Neural Networks [19, 39] and Support Vector Machines (SVMs) [5, 12, 19] are also commonly used for pose and gesture recognition, and both require minimal preprocessing of data (though SVMs are notoriously difficult to implement). Template matching methods are on the other hand a more involved process. Li [29] uses Template Matching for static poses, but a Finite State Machine classifier for dynamic gestures. Xia [30] applies a Chamfer Distance transform before using Template Matching on static poses, then uses a Least Squares regression method for trajectory estimation and an unspecified ensemble classifier to match complete gesture patterns. Ren [11] uses Template Matching on features transformed by a Finger-Earth Mover's Distance. And Ramey [22] uses Finite State Machines to codify hand motion, a search tree for gesture representation, and Template Matching on paths through the search tree for classification.

Less common classification methods include table-based classifiers (used by Fujimura [33]) and Expectation Maximization (used over Gaussian Mixture Models by Jojic [38] to detect pointing gestures, followed by PCA to determine pointing direction). Du [7] created a custom classifier that simply counts the number of convex points in a hand silhouette for classification over a small set of static poses. Bellmore's gesture classifier uses thresholded Euclidean distance between two skeletal points to detect a small gesture set. And Uebersax [34] uses three different classifiers for labeling sign letters – one based on ANMM, one based on pixel-wise depth difference between observed hands and hypothetical models, and one based on estimated hand rotation – and then takes a weighted sum of letter confidences to compute a spelled word score.

V. APPLICATIONS AND ENVIRONMENTS

The variety of the applications for gesture recognition methods presented in the papers was more limited that the

variety of methods themselves. The majority of the applications (75%) fell into one of three categories: interactive displays/tabletops/whiteboards [14, 15, 20, 35, 38], robot motion control [24, 25, 32, 39], and sign language recognition [19, 26, 33, 34, 40, 41]. The remaining papers addressed exercise for physical rehabilitation (Chang [16]) and the elderly (Hassani [18]), novel interfaces for mobile devices (Lai [21]), interaction with social robots (Ramey [22]), in-car computer interfaces (Riener [23]), and enhancing wearable haptics with better position sensing (Frati [17]). It is notable that all of the applications outside the three main categories were seen in papers that used the Kinect as the depth sensor, and in all but one of these (Frati [17]) the hand localization was accomplished via the OpenNI/NITE hand tracking methods.

In 23 papers, the users of the gesture recognition systems were directed to stand or sit in front of the depth sensor at a reasonable range for the sensor and with the hands placed so as to be easily located by the system. The only exception was Fujimura [33], who testing his body detection algorithm under different orientations and distances. All but two systems were also used indoors under controlled lighting. The two exceptions were Konda [39], who used his robot motion control system in an outdoors setting; and Riener [23], whose Driver-Vehicle Interface is ostensibly operated within a car. In effect, very little work has been done to use depth cameras for gesture recognition in settings that are challenging to other systems, and similarly the limits of the new depth-based systems have not been explored.

Although several papers use the Kinect (22 papers) and the OpenNI/NITE hand tracking solution (9 papers) for low-cost and easy hand localization that can then be used for gesture recognition, this approach suffers from some limitations. As Lai [21] points out, the skeleton tracking tends to fail in bright sunlight, presumably because the Kinect's IR structured light pattern becomes washed out, leaving the sensor unable to detect distance. Additionally, using the NITE skeletal tracking method (prior to version 1.5.0.1) requires that the user stand in a calibration pose for approximately 10 seconds before body tracking is initialized. This means that the Kinect/ OpenNI/NITE approach is really only suited to applications with a face-forward interaction in an indoor setting and with the user sitting or standing.

Many applications for depth-based gesture recognition have not yet been explored. These include using such systems in very low illumination or complete darkness, and in environments with a lot of debris which may appear as noise in a depth image. The location and orientation of users in the scene has seen very little variety, so these systems have not yet been used for settings where the user is not in a sitting or standing position (such as lying down), or is not actively facing the sensor, or there are multiple users in the scene. In particular, a user lying down (or otherwise in contact with objects in their environment) may pose a technical challenge to the hand localization methods used with depth cameras. Other missing applications include situations where the user is unfamiliar with the system and gesture set, such as systems that provide feedback to the user or learn a gesture set from the user.

VI. Conclusion

Among the 37 papers reviewed, 13 methods were used for hand localization, and 11 more were used for gesture classification. 24 of the papers included real-world applications to test a gesture recognition system and a total of 8 categories of applications were used. However, 3 of the applications account for 75% of those papers. Though five different types of depth sensors were used, the Kinect was by far the most popular (used by 21 of the papers). The Kinect also has available hand-tracking software libraries that were used by 8 of the papers, and the papers that used the Kinect tended to focus more on applications than on localization and classification techniques.

Returning to the three questions posed in the Introduction: *What methods are being used to achieve hand localization and gesture recognition with depth cameras?* A total of 10 methods are commonly used for hand tracking and gesture recognition in the papers reviewed (2 for segmentation, 3 for tracking, and 5 for classification). Hand segmentation is most commonly accomplished using depth thresholding or region growing techniques. Hand tracking is done using Kalman filters and mean shift, though a significant number of researchers use the NITE body- and hand-tracking module for OpenNI rather than implement their own. And gesture classification is done with a variety of classification algorithms, including Hidden Markov Models, k-Nearest Neighbors, Artificial Neural Networks, Support Vector Machines, and Finite State Machines. The significance of the methods used is that while standard machine learning algorithms are used for gesture classification, hand localization – in particular hand segmentation – has more specialized approaches. Also, custom hand detection and tracking methods are being replaced by off-the-shelf solutions such as PrimeSense's NITE module for the OpenNI framework.

What applications and environments are researchers testing their methods in? Do they test them in situations where their depth-based methods have supposed advantages over video-based methods? Are the limitations of depth-based systems tested? The range of applications presented is still quite limited. Very little work has been done that capitalizes on the advantages of depth cameras over color/intensity cameras for gesture recognition, and the applications developed do not yet test the limitations of depth-based gesture recognition. Also, with the exception of sign language recognition, the majority of applications seen are novelty interfaces for consumer products. Still missing are utility applications of gesture recognition that can take advantage of depth information (perhaps together with color information) in challenging environments.

How has the release of the Kinect and associated libraries affected research in gesture recognition? The release of the Kinect has not only lead to an increase in the number of researchers investigating depth-based gesture recognition, but it has also seemed to shift the focus of the research from

localization and classification methods to applications. Among the 18 reviewed papers not using the Kinect, 13 different methods are used for hand segmentation and tracking, 11 different methods are used for gesture classification, but all described applications fall into one of only three groups (as listed in Section V). By comparison, among the 22 reviewed papers that do use the Kinect, only 7 methods are used for hand segmentation and tracking, 7 methods are used for gesture classification, and 8 categories of applications are presented. In particular, 6 of the 8 applications are presented in the 8 papers that use the NITE skeletal and hand tracking module.

This survey summarizes the techniques that have been used for hand localization and gesture classification in the gesture recognition literature, but shows that very little variety has been seen in the real-world applications used to test these techniques. Applications that take advantage of depth information in challenging environments (such as hand detection and gesture recognition low lighting, or gesture recognition with occlusions) are still missing, as are applications that test the limitations of depth sensors (such as tolerance to noise in depth images, and detecting hands with limited range of motion or in close contact with objects).

REFERENCES

[1] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," *Communications of the ACM,* vol. 54, pp. 60-71, 2011.

[2] S. Mitra and T. Acharya, "Gesture Recognition: A Survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews,* vol. 37, pp. 311-324, 2007.

[3] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: A review," *Computer Vision and Image Understanding,* vol. 108, pp. 52-73, 2007.

[4] D. McNeill, *Language and gesture* vol. 2: Cambridge Univ Pr, 2000.

[5] K. K. Biswas and S. K. Basu, "Gesture recognition using Microsoft Kinect," in *Automation, Robotics and Applications (ICARA)*, pp. 100-103, 2011.

[6] C.-P. Chen, C. Yu-Ting, L. Ping-Han, T. Yu-Pao, and L. Shawmin, "Real-time hand tracking on depth images," in *Visual Communications and Image Processing (VCIP)*, pp. 1-4, 2011.

[7] H. Du and T. To, "Hand Gesture Recognition Using Kinect," Boston University, 2011.

[8] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Efficient model-based 3d tracking of hand articulations using kinect," FORTH Institute of Computer Science, 2011.

[9] S. Park, S. Yu, J. Kim, S. Kim, and S. Lee, "3D hand tracking using Kalman filter in depth space," *EURASIP Journal on Advances in Signal Processing,* vol. 2012, p. 36, 2012.

[10] J. L. Raheja, A. Chaudhary, and K. Singal, "Tracking of Fingertips and Centers of Palm Using KINECT," in *Computational Intelligence, Modelling and Simulation (CIMSiM)*, pp. 248-252, 2011.

[11] Z. Ren, J. Yuan, and Z. Zhang, "Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera," in *ACM international conference on Multimedia*, pp. 1093-1096, 2011.

[12] M. Tang, "Recognizing Hand Gestures with Microsoft's Kinect," Stanford University, 2011.

[13] C. Yang, J. Yujeong, J. Beh, D. Han, and H. Ko, "Gesture recognition using depth-based hand tracking for contactless controller application," in *Consumer Electronics (ICCE)*, pp. 297-298, 2012.

[14] M. Ronchetti and M. Avancini, "Using Kinect to emulate an Interactive Whiteboard," M.S. in Computer Science, University of Trento, 2011.

[15] C. Bellmore, R. Ptucha, and A. Savakis, "Interactive display using depth and RGB sensors for face and gesture control," in *Western New York Image Processing Workshop (WNYIPW)*, pp. 1-4, 2011.

[16] 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,* vol. 32, pp. 2566-2570, 2011.

[17] V. Frati and D. Prattichizzo, "Using Kinect for hand tracking and rendering in wearable haptics," in *World Haptics Conference (WHC)*, pp. 317-321, 2011.

[18] A. Hassani, "Touch versus in-air Hand Gestures: Evaluating the acceptance by seniors of Human-Robot Interaction using Microsoft Kinect," M.S. in Electrical Engineering, Mathematics and Computer Sciences, University of Twente, 2011.

[19] C. Keskin, F. Kirac, Y. E. Kara, and L. Akarun, "Real time hand pose estimation using depth sensors," in *Computer Vision Workshops (ICCV Workshops)*, pp. 1228-1234, 2011.

[20] F. Klompmaker, K. Nebe, and A. Fast, "dSensingNI: a framework for advanced tangible interaction using a depth camera," in *Tangible, Embedded and Embodied Interaction*, pp. 217-224, 2012.

[21] H. Lai, "Using Commodity Visual Gesture Recognition Technology to Replace or to Augment Touch Interfaces," presented at the Twente Student Conference on IT, Enschede, The Netherlands, 2011.

[22] A. Ramey, V. González-Pacheco, and M. A. Salichs, "Integration of a low-cost RGB-D sensor in a social robot for gesture recognition," in *Human-robot interaction*, pp. 229-230, 2011.

[23] A. Riener, M. Rossbory, and A. Ferscha, "Natural DVI based on intuitive hand gestures," in *INTERACT Workshop User Experience in Cars*, Lisbon, Portugal, pp. pp. 62-66, 2011.

[24] P. Trigueiros, A. F. Ribeiro, and G. Lopes, "Vision-based hand segmentation techniques for human-robot interaction for real-time applications," in *CAlg - Artigos em Livros de Actas*, 2011.

[25] M. Van den Bergh, D. Carton, R. De Nijs, N. Mitsou, C. Landsiedel, K. Kuehnlenz, D. Wollherr, L. Van Gool, and M. Buss, "Real-time 3D hand gesture interaction with a robot for understanding directions from humans," in *RO-MAN*, pp. 357-362, 2011.

[26] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti, "American sign language recognition with the kinect," in *International Conference on Multimodal Interfaces*, Alicante, Spain, pp. 279-286, 2011.
[27] M. Zhenyao and U. Neumann, "Real-time Hand Pose Recognition Using Low-Resolution Depth Images," in *Computer Vision and Pattern Recognition*, pp. 1499-1505, 2006.
[28] P. Breuer, C. Eckes, and S. Müller, "Hand Gesture Recognition with a Novel IR Time-of-Flight Range Camera–A Pilot Study," in *Mirage (Computer Vision/Computer Graphics Collaboration Techniques)*, pp. 247-260, 2007.
[29] Z. Li and R. Jarvis, "Real time hand gesture recognition using a range camera," in *Australasian Conference on Robotics and Automation*, 2009.
[30] L. Xia and K. Fujimura, "Hand gesture recognition using depth data," in *Automatic Face and Gesture Recognition*, pp. 529-534, 2004.
[31] M. Van den Bergh and L. Van Gool, "Combining RGB and ToF cameras for real-time 3D hand gesture interaction," in *Workshop on Applications of Computer Vision (WACV)*, pp. 66-72, 2011.
[32] D. Droeschel, J. Stückler, and S. Behnke, "Learning to interpret pointing gestures with a time-of-flight camera," in *Human-robot interaction*, Lausanne, Switzerland, pp. 481-488, 2011.
[33] K. Fujimura and L. Xia, "Sign recognition using depth image streams," in *Automatic Face and Gesture Recognition*, pp. 381-386, 2006.
[34] D. Uebersax, J. Gall, M. Van den Bergh, and L. Van Gool, "Real-time sign language letter and word recognition from depth data," in *International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 383-390, 2011.
[35] B. Yoo, J.-J. Han, C. Choi, K. Yi, S. Suh, D. Park, and C. Kim, "3D user interface combining gaze and hand gestures for large-scale display," in *Human factors in computing systems*, Atlanta, Georgia, USA, pp. 3709-3714, 2010.
[36] T. I. Cerlinca and S. P. Pentiuc, "Robust 3D Hand Detection for Gestures Recognition," in *Intelligent Distributed Computing V*. vol. 382, ed: Springer Berlin / Heidelberg, pp. 259-264, 2012.
[37] W. Yong, Y. Tianli, L. Shi, and L. Zhu, "Using human body gestures as inputs for gaming via depth analysis," in *Multimedia and Expo*, pp. 993-996, 2008.
[38] N. Jojic, B. Brumitt, B. Meyers, S. Harris, and T. Huang, "Detection and estimation of pointing gestures in dense disparity maps," in *Automatic Face and Gesture Recognition*, pp. 468-475, 2000.
[39] K. R. Konda, A. Königs, H. Schulz, and D. Schulz, "Real time interaction with mobile robots using hand gestures," presented at the Human-Robot Interaction, Boston, Massachusetts, USA, 2012.
[40] S. Malassiotis, N. Aifanti, and M. G. Strintzis, "A gesture recognition system using 3D data," in *3D Data Processing Visualization and Transmission*, pp. 190-193, 2002.
[41] R. Feris, M. Turk, R. Raskar, K.-H. Tan, and G. Ohashi, "Recognition of Isolated Fingerspelling Gestures Using Depth Edges," in *Real-Time Vision for Human-Computer Interaction*, ed: Springer US, pp. 43-56, 2005.
[42] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition (CVPR)*, pp. I-511 - I-518, 2001.
[43] M. Kolsch and M. Turk, "Robust hand detection," in *Automatic Face and Gesture Recognition*, pp. 614-619, 2004.
[44] E.-J. Ong and R. Bowden, "A boosted classifier tree for hand shape detection," in *Automatic Face and Gesture Recognition*, pp. 889-894, 2004.