Depth Sensing for Improved Control of Lower Limb Prostheses

Nili Eliana Krausz*, *Student Member, IEEE*, Tommaso Lenzi, *Member, IEEE*, and Levi J. Hargrove, *Member, IEEE*

***Abstract*—Powered lower limb prostheses have potential to improve the quality of life of individuals with amputations by enabling all daily activities. However, seamless ambulation mode recognition is necessary to achieve this goal and is not yet a clinical reality. Current intent recognition systems use mechanical and EMG sensors to estimate prosthesis and user status. We propose to complement these systems by integrating information about the environment obtained through the depth sensing. This paper presents the design, characterization, and the early validation of a novel stair segmentation system based on Microsoft Kinect. Static and dynamic tests were performed. A first experiment showed how the resolution of the depth camera affects the speed and the accuracy of segmentation. A second test proved the robustness of the algorithm to different staircases. Finally, we performed an online walking test with the stair segmentation and related measures recorded online at >5 frames/s. Experimental results show that the proposed algorithm allows for an accurate estimate of distance, angle of intersection, number of steps, stair height, and stair depth for a set of stairs in the environment. The online test produced an estimate of whether the individual was approaching stairs in real time with approximately 98.8% accuracy.**

***Index Terms*—Artificial Limbs, intent recognition, pattern recognition, prosthetics, robot vision systems, sensor fusion, wearable sensors.**

I. Introduction

THE need for individuals with lower limb amputations to perform activities of daily living with greater comfort and ability has inspired the development of powered lower limb prostheses [1]–[3]. Particularly for individuals with transfemoral amputations, powered lower limb prostheses have potential benefits over the use of passive prostheses, such as enabling reciprocal stair climbing [4], improving walking biomechanics [5], reducing the risks of falling [6], and reducing metabolic costs [7]. Powered prostheses can enable the achievement of most ambulation modes, including level-ground walking, stair ascent and descent, and ramp ascent and descent [8]. However, there is a need for effective intent recognition algorithms to allow for safe seamless transitions between ambulation modes.

Available algorithms for intent recognition generally use wearable sensors to gather data from the user and the prosthesis [9]. Electromyography (EMG) recorded from the residual limb provides information on the user movement [10]. Inertial measurement units (IMUs) and other mechanical sensors embedded in the prosthesis can inform the intent recognition algorithm about the position, orientation, and overall status of the prosthesis [11]–[13]. Force/torque sensors recording the loading on the leg provide information about the user–prosthesis interaction [14]. Statistical decision algorithms then process the data gathered from the wearable sensors to infer the desired ambulation mode. Multiple approaches have been proposed to determine the current ambulation mode, as well as the users' intent to transition to a different mode [15]–[23]. Bayesian learning has shown good results, providing transition accuracies of up to ∼88% for four or five ambulation modes and ∼99% for three modes [24]. Despite the high accuracy already achieved by previous studies [24], there is the need for near perfect classification to make this approach viable in real-life situations. In fact, even rare misclassifications can result in catastrophic falls or other dangerous situations [25].

A possible explanation for the observed accuracy limitations is the intrinsic variability of the data that is used by the intent recognition algorithms. Gait patterns naturally vary between subjects, days, and individual trials [26]. Therefore, the signals used in the intent recognition systems have inherent intra- and intersubject variability. We hypothesize that including information about the environment will improve an intent recognition by providing an additional source of information that is subject independent. We propose that integrating additional sensory modalities, specifically vision and depth sensing [27]–[30], into the mode estimation may improve ambulation mode estimation and classification accuracy. Specifically these additional sensory modalities are chosen with the goal of detecting the state of the environment, which previous approaches have not included. Our long-term goal is to integrate information from EMG, mechanical sensors, and vision in a probabilistic manner to estimate a user's intended ambulation mode with higher accuracy than previously presented.

In this study, we developed and preliminarily validated a computer vision algorithm to segment stairs from the environment and to extract relevant measures for the estimation of the intended locomotion mode. Section II provides background information about the use of vision for detecting stairs and its potential for prosthesis control. Section III describes our vision system and processing methods for detecting stairs and computing the distance from the user, the angle of intersection with the

Manuscript received December 31, 2014; revised March 31, 2015 and June 8, 2015; accepted June 13, 2015. Date of publication June 22, 2015; date of current version October 16, 2015. This work was supported by the US Army's Telemedicine and Advanced Technology Research Center 81XWH-09-2-0020, a Walter P. Murphy Fellowship, and the MSL Renewed Hope Foundation. *Asterisk indicates corresponding author.*

*N. E. Krausz is with the Center for Bionic Medicine, Rehabilitation Institute of Chicago, Chicago, IL 60611 USA, and also with the Department of Bioengineering, Northwestern University, Evanston, IL 60611 USA (e-mail: nili.krausz@u.northwestern.edu).

T. Lenzi and L. J. Hargrove are with the Rehabilitation Institute of Chicago, and also with the Northwestern University.

This paper contains supplemental materials available online at http://ieeexplore.ieee.org (File size: 7 MB).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBME.2015.2448457

user's walking path, as well as the step count, height, and depth. Section IV presents the methods for system offline validation and online testing. Results are presented in Section V. Significance is discussed in Section VI. Finally, we draw conclusions in Section VII.

II. Background

Research on powered lower limb prostheses is evolving rapidly, but computer vision has not previously been investigated for lower limb prosthesis control. Until recently there were no accurate, small, lightweight, and inexpensive depth cameras that could potentially be used in this context. However, the development and release of inexpensive depth sensors, such as the Microsoft Kinect [31], lightweight and small sensors like the Occipital Structure Sensor [32] and the CamBoard pico XS [33], and wearable devices with cameras, such as Google Glass [34], have made the use of vision a more viable option for prosthesis control.

Vision and depth sensing have previously been described for identifying stairs in the environment for a variety of robotic applications. The methods that have been proposed can be broadly separated into those based on monocular vision and those based on 3-D data (either from stereo vision or from a depth sensor). In each of these categories, there are multiple methods for detecting stairs. For monocular vision-based methods, a filtering method, such as a Gabor filter, or an edge detection method, such as a Canny edge detector, is often first used for finding edges in the images [35], [36]. Then, the presence of stairs can be detected using either machine learning techniques [37], [38] or other algorithmic rule-based approaches [39]–[44].

Several issues may affect monocular stair detection, including difficulty in distinguishing lines belonging to stairs from those of crosswalks or patterned floors [45], the challenge of accurately producing a distance map using a single monocular image [46]–[48], and the changing performance with differences in lighting and shadows [49], [50]. Though few publications detail performance of monocular stair detection algorithms, one study [41] reported 83% accuracy for finding stairs in images with stairs, and 81% accuracy of finding no stairs when stairs were absent. This corresponds to having a false negative detection of stairs 17% of the time and false positive detection 19% of the time.

Though there are potentially ways to compensate for these errors, a 3-D-based sensing has been proposed to eliminate them entirely. Simultaneous localization and mapping [51], RANSAC planar modeling [52]–[54], tensor voting [55], machine learning techniques [56], or using the general slope of the environment immediately in front of the user [30] have been previously proposed for the 3-D sensing of stairs (see [41] for a detailed review of methods for stair detection). While many of these methods have been developed to locate or map the position of stairs in the environment, they were not designed to obtain additional information for prosthesis control. Thus, to obtain the location of the stairs and other metrics that can inform a controller and locomotion mode detection system for lower limb prostheses, we proposed a novel stair segmentation algorithm that is based on a combination of standard 2-D approaches with 3-D strategies.

Environmental sensing and sensory fusion have been discussed previously for the improved control of lower limb prostheses. Novak and Riener presented a thorough overview of research using sensory fusion for the control of prostheses and orthoses [57]. The authors detail the benefits of using sensor fusion for control of assistive devices, and briefly discuss the potential gains of adding vision sensors to prostheses and orthoses. Another recent example was Tucker *et al.* in which the authors recommended adding environmental sensing and spatial context to the control of prostheses and orthoses [58]. The authors highlight the importance of environmental sensing for providing users with safe and responsive assistance, and that it is important to study how the environment affects the user, device, and controller to produce the useful assistance. In a third example, Du *et al.* [59] simulated a model of the environment, which was used to modify the prior probabilities of the locomotion mode classifier; this model improved classification accuracies, demonstrating that the knowledge of the environmental state could be used effectively as a means of decreasing intent recognition errors.

In a recent journal paper, Liu *et al.* [60] used a laser distance sensor to estimate the terrain directly in front of the user. This information was then incorporated into an intent recognition system to produce estimates of the desired locomotion mode. Although the terrain data were not actually used for the online control, the work demonstrated the possible benefits of incorporating environmental data, and reported an improved locomotion mode recognition accuracy.

Unlike previous works based on laser distance sensors [61], we propose the use of a vision or depth system. A vision or depth system can provide information about the global scene in the field of view, as opposed to the single distance measure that is obtained by a laser distance sensor. Thus, depth sensing can allow for the perception of different terrains and the extraction of additional measures, such as the number of stairs in the user's path and the step height and width, without any need for integration over time. Finally, a vision or depth sensor potentially could be used to detect obstacles and terrain changes that are not directly in line with the sensor, unlike a laser distance sensor.

III. Segmentation Algorithms

The proposed segmentation algorithm takes the advantage of standard stair geometry to produce stair-specific segmentation. Depth data were acquired from a Microsoft Kinect for Windows version 1 [31]. Processing was performed *online* in MATLAB/Simulink 2014b (Mathworks, Natick, MA, USA).

A. Segmentation Procedure

The proposed stair segmentation algorithm primarily uses combinations of 2-D image processing techniques for performing 3-D segmentation. Throughout the procedure, segmented regions were treated as binary images with the depth data being used only when needed. In general, the segmentation algorithm uses prior knowledge of the basic structure of stairs, with several components significant for producing the final results, including the convex and concave edges, vertical and horizontal faces, and bottom vertical plane and top horizontal plane (see Fig. 1 for a diagram of these stair components). Fig. 2 shows

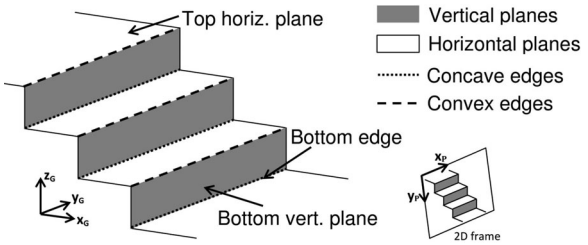

Fig. 1. Stair diagram with stair components that are used in segmentation algorithm is shown, including vertical and horizontal planes, convex and concave edges, etc. Also shown are the global coordinate system (x_G, y_G, z_G) and the 2-D coordinate system (x_P and y_P).

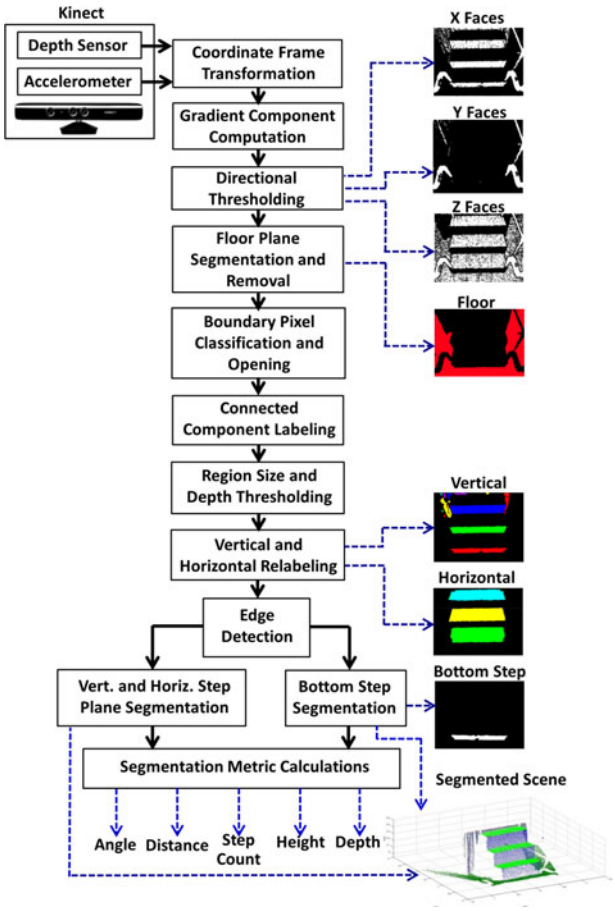

Fig. 2. Diagram of segmentation procedure showing major steps involved in performing the stair segmentation. Arrows shown in blue are the indicative of outputs from the processing blocks. (Best viewed in color).

a diagram of the stair segmentation procedure. The operations encompassed in each block of Fig. 2 are described in the following numbered list.

1) *Coordinate frame transformation:* The acquired depth data were initially in a sensor-based reference frame. Acceleration data from the accelerometer housed in the Kinect were acquired, and an Euler angle rotational transformation [62] was used to estimate the angular rotation of the device with respect to the world, which was then used to rotate the depth data into world coordinates. The yaw, or rotation about the z-axis, was assumed to be zero to allow for the computation of the pitch and roll angles from a single three-axis accelerometer [63]. Using offsets derived from calibration data, the floor plane was then aligned with zero in the z-direction, and the location of the Kinect sensor was aligned with zero in the x- and y-directions.
2) *Gradient component computation*: In the resolution conditions that required up- or downsampling (see Section IV-A), this was performed next. Depth data were next separated into 2-D matrices comprising the x_G, y_G, and z_G components of the depth. Finally, the spatial image gradient operation was performed on each orthogonal 2-D matrix [64].
3) *Directional thresholding:* Each of the three 2-D arrays was processed, and any pixels with values of the gradient magnitude less than a threshold (0.085 for any resolution less than 320×240, 0.025 for the 320×240 resolution) were selected. Thresholds were selected based on the magnitudes of the gradient of the planar regions in each direction. These pixels were assumed to be components of a planar region within the given axial direction; thus, it was possible to produce the segmentation of pixels normal to the x-, y, and z-axes (outputs shown in Fig. 2 as X-, Y-, and Z- faces).
4) *Floor (and wall) plane segmentation and removal:* The floor and any visible back walls were segmented and removed. Since the floor was previously aligned with zero in the z-direction (see step 1), any points below zero or within a threshold of 0.03 m above this minimum were labeled as floor pixels (see Fig. 2). Similarly, for the back wall, any pixels within a threshold of 0.15 m of the maximum value in the x-direction were labeled as a part of the wall. Next, floor pixels and wall pixels were removed from the segmented z-normal and x-normal array, respectively, leaving arrays of segmented horizontal and vertical planes. A morphological dilation operation was then performed using a 3×3 structuring element to smooth and remove holes from the segmented arrays [65]. Though morphological closing would help fill in holes and smooth the face segmentation, dilation not only accomplishes these tasks, but also increases the boundary size of the regions. This helps to ensure that regions "meet" at borders and allows for a good detection of the edges between regions.
5) *Boundary pixel classification and opening:* Any unclassified pixels on the boundary of the previously segmented regions were assigned to the adjacent classes using a nearest neighbor approach [66]. An opening operation [65] was then performed on each segmented class using a 7×7 circular structuring element to remove thin connections between larger regions of each image. Constraints were implemented to ensure that no pixels were defined as belonging to more than one class.
6) *Connected component labeling (CCL):* Next, we performed a CCL on the horizontal and vertical segmented images using four-neighbor connectivity [67].
7) *Region size and depth thresholding:* Labels with fewer than 0.5% of the total number of pixels were filtered

out of the labeled horizontal and vertical images. Since the original CCL only considered the 2-D position of the pixels, pixels with different depths could be labeled as one; thus, the depths were considered as well. For each label, a histogram of depths was produced, with bins of 5 cm in depth for vertical faces and 2.5-cm bins for horizontal faces. Then, for bins with greater than 5 pixels, all pixels within a given bin were assigned a new label.

8) *Vertical and horizontal relabeling:* A second size-based thresholding step was performed by removing labels with vertical pixel heights greater than 35% of the total vertical pixel height of the depth image. This threshold was selected from preliminary offline experiments, and it was defined as a percentage to normalize across different image resolutions. This eliminated objects in the field of view that were too large to be stairs and yielded separated and relabeled vertical and horizontal regions as shown in Fig. 2.
9) *Edge detection:* We detected and extracted a bottom edge, where the floor plane and vertical regions met. Then, convex and concave edges were obtained using the previously segmented regions. Convex edges were obtained by finding points that were either on the bottom edge of a horizontal plane or the top edge of a vertical plane. To be considered as a part of a potential convex stair edge, a point had to belong to both a vertical convex edge and a horizontal convex edge (implying that the point was at the intersection of horizontal and vertical planes). Concave edges were obtained similarly, with vertical concave edges defined as the bottom edge of vertical planes and horizontal concave edges defined as the top edges of horizontal planes.
10) *Bottom step segmentation and centroid calculation:* The bottom step vertical plane was segmented by selecting labels located between the bottom edge and the bottommost convex edge. The centroid of the bottom vertical plane was found by obtaining the halfway point between the maximum and minimum points in the *x*- and *y*-directions of this plane.
11) *Vertical and horizontal step plane segmentation:* The vertical and horizontal step surfaces were obtained by finding labels located between convex and concave step edges, within a vertical band slightly larger than the segmented bottom step. The topmost plane was obtained by finding a horizontal region directly above the topmost convex edge of the vertical plane segmentation. The segmented stair planes were finally combined for an overall 3-D segmentation view as in Fig. 2.

B. Segmentation Measures

The five measures considered were the resultant distance, the angle of intersection, the number of steps, the stair height, and the stair depth.

1) *Resultant distance to bottom stair:* The resultant distance between the Kinect and the centroid of the bottom stair was computed. The resultant distance is the sum of the squares of the *x*- and *y*-distances. This allowed for a single measure that provided an estimate of the combined accuracy of the *x*- and *y*-distances.
2) *Angle of intersection calculation:* The angle of intersection between the stairs and the Kinect was obtained, as described in the flowchart in Fig. 3, using the segmented horizontal and bottom stair faces. Convex lines were extracted from the segmented faces, and if no lines were found then the stairs were defined as not intersecting. The endpoints of the bottommost of these convex lines were obtained, and if the *y*-components of these endpoints had the same sign then again the stairs were defined as not intersecting. Finally, if the stair was found to intersect the Kinect centerline, then the angle of intersection was computed using the equation in Fig. 3.
3) *Number of steps:* The horizontal and topmost planes were combined and used to produce a value for the number of steps in the current view of the Kinect. The segmented horizontal/top planes with more than 30 pixels were relabeled using a CCL method, using a four-neighbor connectivity [67]. The number of labels was then output as the number of steps.

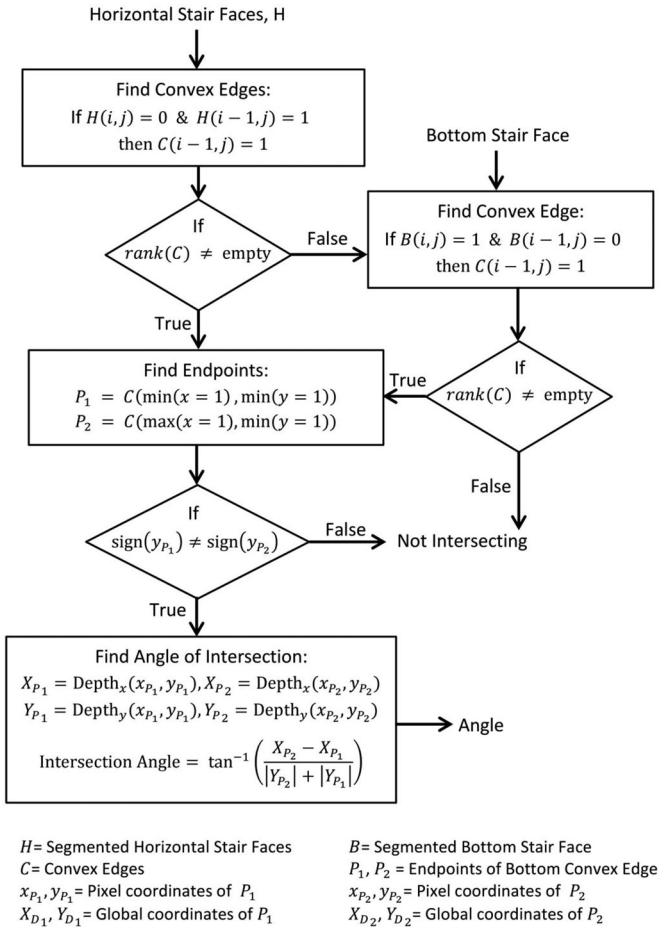

Fig. 3. Method for obtaining the angle of intersection with stairs using segmented horizontal stair faces and bottom stair face, with outputs of either "not intersecting" or an intersection angle (as measured from the *x*-axis). Variables are defined in the figure.

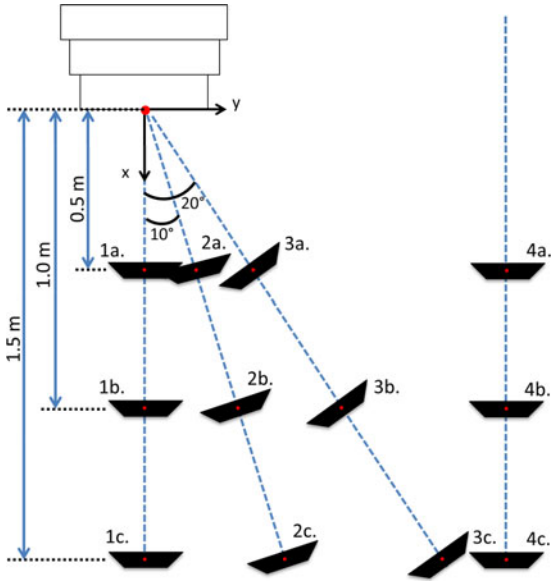

Fig. 4. *Experiment 1*—Setup with four orientations of the Kinect, each tested at three depths from a staircase. In Orientation 1, the Kinect was directly inline with the staircase. In Orientation 2, the Kinect was pointed toward the stairs at a 10° angle from the *x*-axis, and in Orientation 3, this was a 20° angle. In Orientation 4, the Kinect was kept at a constant distance of 1 m in the *y*-direction. In each orientation, the *x*-displacement of the Kinect was 0.5 m at a, 1.0 m at b, and 1.5 m at c.

4) *Stair height:* The stair height was calculated by measuring the difference in global *z*-position of the maximum and minimum points of the bottom vertical face.
5) *Stair depth:* The stair depth was calculated by measuring the difference in the global *x*-position of the maximum and minimum of the bottommost horizontal face at its centroid in the *y*-direction.

IV. Experimental Methods

A. Experiment 1: Resolution, Speed, and Accuracy Test

The goal of the first experiment was to determine how the speed and accuracy of the segmentation algorithm might vary based on the resolution of the depth camera to determine the best resolution for online use. Four different resolutions ($320 \times 240, 160 \times 120, 120 \times 90,$ and 80×60 pixels) were tested at several different orientations and positions as shown in Fig. 4 and described in the next paragraph. The 320×240 and 80×60 resolutions are options native to the Kinect depth stream. The 160×120 resolution was produced by downsampling the 320×240 stream, and the 120×90 resolution was produced by upsampling the 80×60 stream. Though upsampling is not ideal for obtaining the 120×90 resolution, it was chosen because we needed a method of obtaining this intermediate resolution that is not native to the Kinect. The Kinect was attached to a tripod at a constant height of 1.5 m and tilt (or pitch) of $-50°$ from horizontal.

We repositioned the Kinect into four different orientations (shown with numerals 1–4 in Fig. 4), and tested each orientation at three depths (shown with letters a–c in Fig. 4). To evaluate the performance of the angle of intersection computation, for each resolution three angled orientations were considered. Orientation 1 was directly inline with the centroid of the stairs with an approximate angle of intersection of 0° as measured from the centerline of the Kinect to the *x*-axis. Orientation 2 was selected so that the centerline of the Kinect in each position was at a 10° angle from the *x*-axis. Orientation 3 was aligned with the Kinect centerline at a 20° angle from the *x*-axis. Finally, Orientation 4 was parallel to Orientation 1, but with a 1-m offset in the *y*-direction between the Kinect centerline and the stair centroid.

For each orientation, three distances were tested. The distances were selected to simulate how the segmentation would perform during an online walking test. For all orientations, the Kinect had an offset from the stair centroid of 0.5 m in the *x*-direction at point a, 1.0 m at b, and 1.5 m at c. The distances in *y* were varied to maintain a constant angle of intersection.

For each of the four resolutions and 12 positions, 20 frames were acquired and compared. For each frame, three estimates were produced: the resultant distance from the Kinect to the bottom stair centroid, the angle of intersection (with a not intersecting status outputting as infinity), and the speed as a rate of frames per second (frames/s). To evaluate the results of the distance estimate, we computed the mean and standard deviation of the error between the actual distance and the distance estimate for each resolution and orientation. In addition, for each resolution, we computed two R^2 metrics using the data from all frames and orientations. The first of the reported R^2 values is the *Perfect Estimate* R^2 value. This R^2 value was produced to evaluate how close the computed distance was to the actual measured distance. The R^2 *Perfect Fit* is a simple least squares regression with a *y*-intercept equal to 0 and a slope equal to 1. This corresponds to the fit that would be produced if the computed distance perfectly matched the measured distance. High *Perfect Estimate* R^2 values would imply that the computed resultant distance was very close to the actual measured resultant distance. Meanwhile, the *Best Fit* R^2 is a linear fit to the actual data that were recorded. This provides a measure of how linear the data is and how well correlated the computed distance is to the measured distance. High *Best Fit* R^2 values mean that there is a high correlation between the computed and the measured values of distance.

To evaluate the accuracy of angle of intersection estimates, we computed the mean and standard deviation of the angle for each orientation as estimated at different distances (i.e., three distances for each orientation). Then, we computed the difference between the mean orientation estimates and the actual orientation as defined by the experimental setup (i.e., 0° for Orientation 1, 10° for Orientation 2, 20° for Orientation 3, and infinity for Orientation 4 denoting that no intersection is detected between the Kinect centerline and the stairs).

Finally, the speed estimate was evaluated. The algorithm reported speed as an average frames/s rate after every five frames. However, since the frame rate does not reach a steady-state value until after the first few frames, only the last three averaged values were included for each position and resolution. The overall average frame rate was computed and compared for each resolution to evaluate the estimate of the speed.

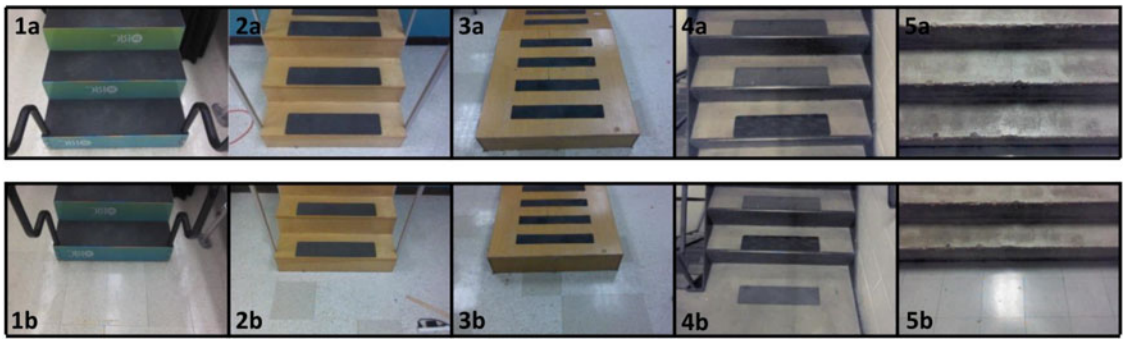

Fig. 5. *Experiment 2* test cases. The distance, angle, number of stairs, step height, and step depth were recorded for each of the five different stair cases at distances of 0.5 m in a and 1.0 m in b. These measures were recorded for 20 frames that were processed online at a resolution of 80×60 pixels.

B. Experiment 2: Stair Height, Depth, and Count Test

The goal of the second experiment was to compare the performance of the proposed algorithm across differing stair conditions. This test included five different staircases (as shown in Fig. 5), for which we evaluated the accuracy of the distance and angle of intersection estimates, using the data analysis approach already reported for *Experiment 1*, as well as the number of steps, the height, and the depth of stairs. These measures were all calculated at two distances (0.5 and 1.0 m) for each of the five different staircases. Mean and standard deviation from 20 frames were computed across all acquired frames for all segmentation measure estimates. Fig. 5 shows RGB images of each of the staircases that were tested. The different staircases were selected to obtain several different conditions, such as the presence of railings, a stairwell, only a single step, and stairs that are wider than the range of view of the Kinect. Additionally, the stair heights and depths differed across the various staircases.

C. Experiment 3: Online Walking Test

The goal of the third experiment was to validate the algorithm in a dynamic real-life scenario. This involved an online walking test, which would allow for demonstration of the performance of the segmentation algorithm during realistic use conditions. In this test, the stair segmentation algorithm was run online at a resolution of 80×60 pixels. Throughout the test, the Kinect was held at approximately the same height and tilt that had been used in the characterization. The subject walked through a hallway for about 10 m before entering into a stairwell, where he ascended stairs for a total of three flights, 27 steps, and 170 frames. Acquisition, processing, and segmentation were performed online. In addition to the segmentation measures, we saved the stream of images of the 2-D stair segmentation and the Kinect RGB camera using a MATLAB/Simulink 2014b.

To help evaluate if the segmentation was effective during the online test, a status of "approaching/not approaching stairs" was produced for each frame. In the case that no stairs were segmented or the angle of intersection between the Kinect and the stairs was infinity, a "not approaching stairs" status was produced. Otherwise, an "approaching stairs" output was recorded. This allowed for evaluation of the performance of the online test, since it would be possible to visually inspect frames to determine what status should be the output (for instance, on landings when not facing stairs it is clear that a not approaching status should be the output).

TABLE I
EXPERIMENT1—SPEED RESULTS

Resolution (pixels)	Frame Rate (frames/s)
80×60	4.7 ± 0.7
120×90	1.8 ± 0.2
160×120	1.0 ± 0.2
320×240	0.1 ± 0.0

V. RESULTS

A. Experiment 1

We reported the mean and standard deviation of the frame rate for each resolution in Table I. The highest frame rate was measured for the smallest resolution of 80×60 pixels, which had a mean frame rate of 4.7 ± 0.7 frames/s, and the lowest mean frame rate of 0.1 ± 0.0 frames/s was measured for the highest resolution of 320×240 pixels.

To assess the accuracy of the proposed algorithm, we compared the estimates of the angle of intersection and distance with actual values as measured in *Experiment 1* for each resolution, orientation, and distance.

Fig. 6 shows the results for the distance estimate as a function of the measured resultant distance. Each panel reports the results for a different resolution. Estimated distances are shown on the *y*-axis, whereas actual distances are shown on the *x*-axis. Estimates for the same distance obtained at different orientations are reported using different colors and markers as shown in the legend of Fig. 6. Each datapoint in Fig. 6 reports the mean and standard deviation for 20 frames. As described in Section IV, for each of the resolutions a *Best Fit* and a *Perfect Estimate* were computed and reported in Fig. 6 with dashed blue lines and dotted black lines, respectively. The highest *Best Fit* R^2 value was produced by the lowest resolution (80×60 pixels), where the value was 0.96. Meanwhile the lowest *Best Fit* R^2 value was computed from the highest resolution (320×240), which had a value of 0.90. Similarly, the highest *Perfect Estimate* R^2 value was produced using the smallest resolution (80×60), where a value of 0.95 was computed, and again the lowest value of 0.87 was obtained from the highest resolution (320×240).

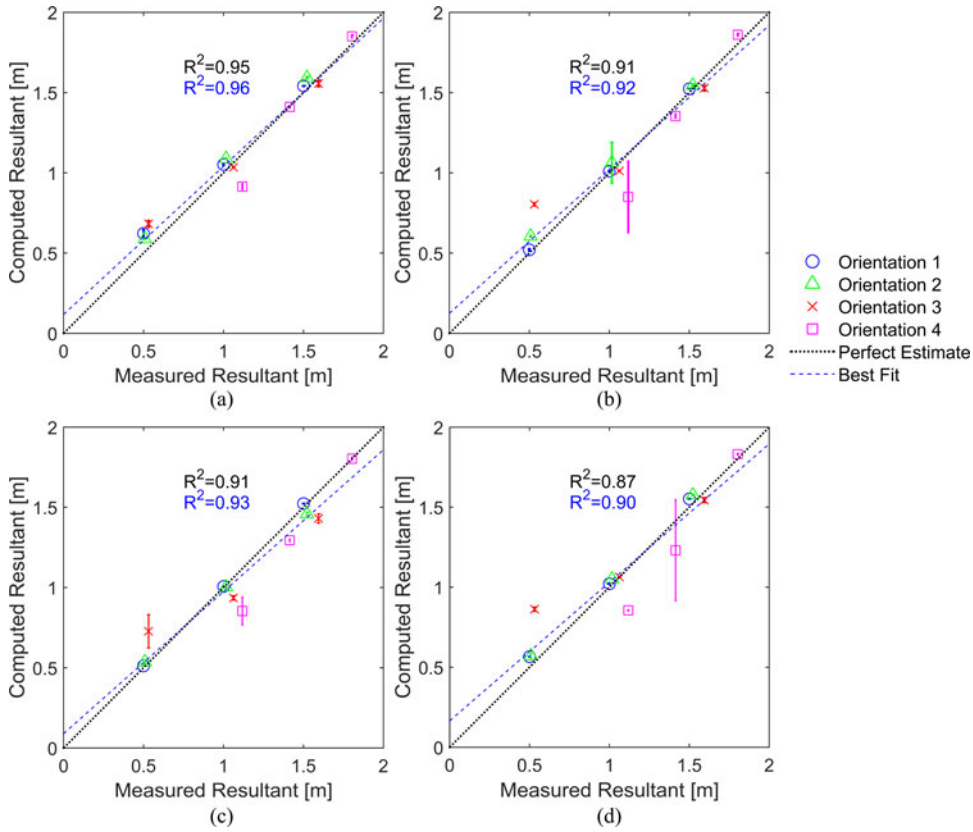

Fig. 6. *Experiment 1*—Distance from sensor to bottom step. Results for each of the four tested resolutions are shown, with a resolution of 80×60 pixels shown in a, 120×90 in b, 160×120 in c, and 320×240 shown in d. The dashed blue line represents the *Best Fit* line, and the dotted black line shows the *Perfect Estimate* line. The *Best Fit* R^2 value shows the linearity/precision of the distance, and the *Perfect Estimate* R^2 value presents the accuracy of the distance estimate. The mean and standard deviation (shown by error bars) for 20 frames is shown for each position, orientation, and resolution. Again, Orientation 1 was at 0°, Orientation 2 was at 10°, Orientation 3 was at 20°, and Orientation 4 was parallel to the stairs at a distance of 1 m.

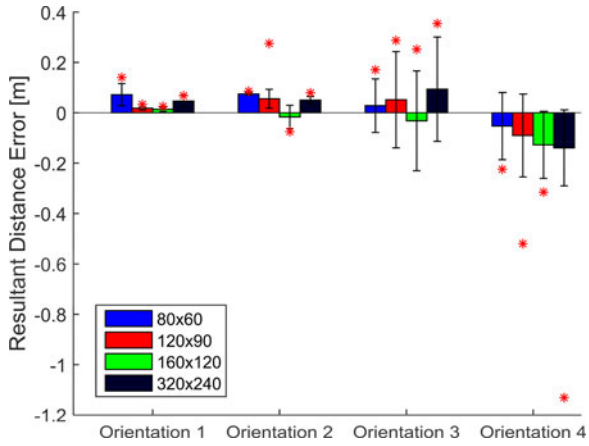

Fig. 7. *Experiment 1*—Resultant distance error. For each orientation, the mean and standard deviation of the resultant distance error is reported. Results for each of the four tested resolutions are shown. The red asterisk is the maximum error for each resolution and orientation. Orientation 1 was at 0°, Orientation 2 was at 10°, Orientation 3 was at 20°, and Orientation 4 was parallel to the stairs at a distance of 1 m.

Fig. 7 presents the mean and standard deviations of the distance error for all orientations (shown in *X*-axis) and resolutions (shown with different colors). The mean and standard deviation of the absolute distance errors across all orientations were 5.7 ± 7.2, 5.4 ± 10.0, 4.7 ± 9.6, and 8.2 ± 9.9 cm for a resolution of 80×60, 120×90, 160×120, and 320×240, respectively, indicating that 80×60 obtained the lowest variance although 160×120 obtained the lowest mean error. The average absolute distance errors across all resolutions were 3.7 ± 2.7, 4.9 ± 2.4, 5.1 ± 3.0, and 10.2 ± 3.9 cm for Orientations 1, 2, 3, and 4, respectively, showing that the distance error increased with the angle of intersection. Overall error averaged 6.0 ± 3.7 cm. The maximum distance error is also shown in Fig. 7 for each resolution and orientation; these errors are in general the smallest for the 80×60 resolution. The max errors also increased with the increasing angle (higher errors for Orientations 3 and 4).

Results for the estimates of the angle of intersection for all tested distances and Orientations 1, 2, 3 are reported in Fig. 8. Different colors indicate different resolutions as shown in the legend. The best results were produced by the 80×60 resolution, which had computed angle mean absolute error of $0.47 \pm 0.38°$ for all orientations. Meanwhile, the 120×90, 160×120, and 320×240 resolutions had estimated the angle mean absolute errors of $1.7 \pm 1.1°$, $1.1 \pm 0.2°$, and $1.3 \pm 0.4°$, respectively, giving an overall accuracy of $1.1 \pm 0.7°$. The angle measurement was essentially independent of the distance in the *x*-direction for all orientations and resolutions, with the exception of Orientation 3 for the 120×90 resolution, where the angle measurement increased linearly with a slope of approximately 5°/m. Orientation 4 was intentionally left out of Fig. 8, as it was supposed to result in no intersection

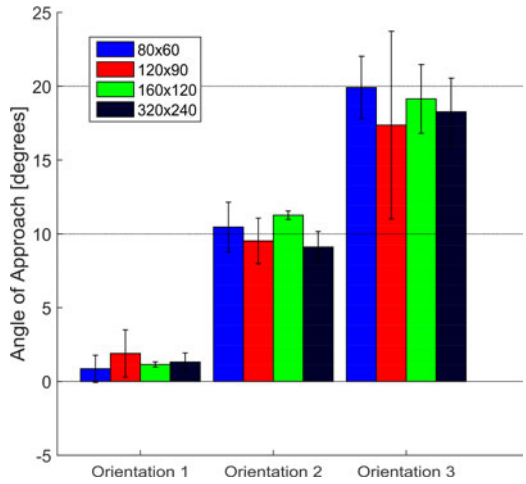

Fig. 8. *Experiment 1*—Angle results. Results for each of the four tested resolutions are shown, with the resolutions shown in the legend. Note that Orientation 4 is not shown because, with the exception of position 4a at the 120×90 resolution, the angle of approach was always given as infinity, since the approach of the Kinect did not intersect with the stairs. Actual angles of intersection are reported in the *Y*-axis, and are also shown with horizontal black dotted lines. Orientation 1 was at 0°, Orientation 2 was at 10°, Orientation 3 was at 20°, and Orientation 4 was parallel to the stairs at a distance of 1 m.

(i.e., infinite angle). Analysis of Orientation 4 estimates indicates that the estimated angles of intersection were indeed infinite for all distances and resolutions except for the shortest distance (i.e., point 4a in Fig. 4) at the resolution of 120×90.

B. Experiment 2

Experiment 2 aimed to test the proposed segmentation algorithm against five different staircases. Table II reports the means and standard deviations for the estimates of distance, angle of intersection, number of steps, stair height, and stair depth in different columns, for all tested stair cases and distances in different rows. In addition to the estimates, we reported the actual measures as a term of comparison. The results from *Experiment 2* show with the exception of two cases (2a and 5b) the step count had zero variance, and the computed step count yielded the correct value. In each of the other two cases, the step count yielded the correct value in a few frames. However, there was some variability in the computed step count. The stair height was within 1 cm of the computed mean stair height for all but two cases—2b and 5b—where the measured stair height was 1.3 and 1.4 cm less than the computed height. Meanwhile, the measured stair depth was always within 3 cm of the computed mean stair depth, with the exception of case 5b, where the mean computed stair depth was 6.2 cm larger than the measured stair depth. However, even in this case, the measured stair depth fell within one standard deviation of the mean computed stair depth. Additionally, for cases 3a and 3b, since there was not a clearly defined measured value of the depth (due to this condition being a single step), the computed values were assumed to be correct as long as they were greater than 1 m.

C. Experiment 3

An online test was performed using the same staircase shown in Fig. 5 Panels 4a and 4b. A single able-bodied subject performed the test in which he walked at a self-selected walking speed. A video showing the RGB stream and segmentation results is included in the supplemental materials. More than 170 frames that were recorded, there was one clear false positive recorded and one clear false negative. This corresponds to an approaching/not approaching accuracy of approximately 98.8%. Fig. 9(a) and (b) shows an example of a true positive frame and a true negative frame, respectively. Additionally, the false positive frame and false negative frame are shown in Fig. 9(c) and (d), respectively. The average stair height and stair depth were 15.6 ± 7.7 and 34.3 ± 15.1 cm, respectively, from all frames, where there was a step count of 1 or more. Additionally, the distance and angles were recorded for each frame of the online test. However, evaluating how closely these values match up with the actual values is challenging, since the global position of the Kinect was not measured during the test. Finally, the average frame rate for the online walking test was 5.2 ± 0.1 frames/s.

VI. Discussion

In *Experiment 1*, the algorithm was tested at 12 different positions for four different depth camera resolutions. Results demonstrate that the proposed segmentation algorithm produced a good estimate of the distance and angle of intersection—within 6.0 cm and 1.1°, respectively, for any distance and resolution (see Figs. 7 and 8). These results are significant because the distance to stairs and the angle of intersection are measures that can estimate user intention to transition to a different ambulation mode. It is worth noticing that the accuracy of both segmentation estimates degraded with the increasing angle of intersection and lateral distance. We believe that this degradation was due to an incomplete segmentation of the bottom step at higher angles and lateral distances, which was caused by the limited range of view of the depth camera in use.

Not surprisingly, characterization results indicate that the smallest resolution (80×60 pixels) obtained the highest framerate—4.7 frames/s (see Table I). Notably, resolution had a minor effect on accuracy, as all resolutions produced absolute distance and angle errors within 5 cm and 2° of each other, though the lowest resolution provided an accuracy that was slightly better than other resolutions for some of the metrics. Therefore, we selected the 80×60 pixel resolution for an online use in Experiments 2 and 3. The frame rate is clearly affected by the resolution, and, in particular, the CCL and relabeling steps of our algorithm are thought to be the most significant in producing this effect. In future work, we intend to consider ways of optimizing the labeling process.

Although mean absolute distance error at the 80×60 resolution was not the best value, this resolution obtained the lowest absolute error in the angle of intersection and the highest *Best Fit* and *Perfect Estimate* R^2 values for the distance tests (see Fig. 6). Thus, the accuracy was generally higher for the 80×60 resolution. We believe this was due to a combination of smoother borders and fewer gaps found between regions in the depth map, as well as slight differences in calibration and variability in the sensor placement. However, the difference in accuracy between resolutions was not high, and, therefore, we do not believe that

TABLE II
EXPERIMENT 2 RESULTS

Case	Measured Distance (cm)	Estimated Distance (cm)	Measured Angle (°)	Estimated Angle (°)	Measured Stair Count	Estimated Stair Count	Measured Stair Height (cm)	Estimated Stair Height (cm)	Measured Stair Depth (cm)	Estimated Stair Depth (cm)
1a	50	50 ± 0.6	0	2.1 ± 0.3	3	3.0 ± 0.0	19.5	20.3 ± 0.2	28	28.5 ± 0.3
1b	100	109 ± 0.4	0	0.7 ± 0.6	2	2.0 ± 0.0	19.5	20.2 ± 0.5	28	29.4 ± 1.0
2a	50	58 ± 17.2	0	0.4 ± 0.2	3	2.0 ± 0.7	18	18.6 ± 0.1	25.5	27.8 ± 11.4
2b	100	101 ± 0.3	0	0.4 ± 0.3	2	2.0 ± 0.0	18	19.3 ± 0.4	25.5	28.2 ± 0.4
3a	50	51 ± 0.1	0	2.1 ± 0.3	1	1.0 ± 0.0	20.5	21.2 ± 0.0	>100*	214.5 ± 1.1
3b	100	102 ± 0.1	0	0.6 ± 0.9	1	1.0 ± 0.0	20.5	19.7 ± 1.2	>100*	148.9 ± 11.4
4a	50	59 ± 2.3	0	0.5 ± 0.1	3	3.0 ± 0.0	18.5	18.6 ± 0.3	28	26.1 ± 1.8
4b	100	108 ± 0.4	0	0.3 ± 0.3	2	2.0 ± 0.0	18.5	19.5 ± 0.4	28	28.8 ± 0.6
5a	50	58 ± 1.3	0	1.7 ± 0.2	3	3.0 ± 0.0	19	19.8 ± 0.3	27.5	27.1 ± 0.6
5b	100	106 ± 0.4	0	1.4 ± 0.4	2	2.2 ± 0.4	19	20.4 ± 0.3	27.5	33.7 ± 8.6

*Cases 3a and 3b had only a single step, and, therefore, the depth was difficult to properly define. However, depth was more than 1 m.

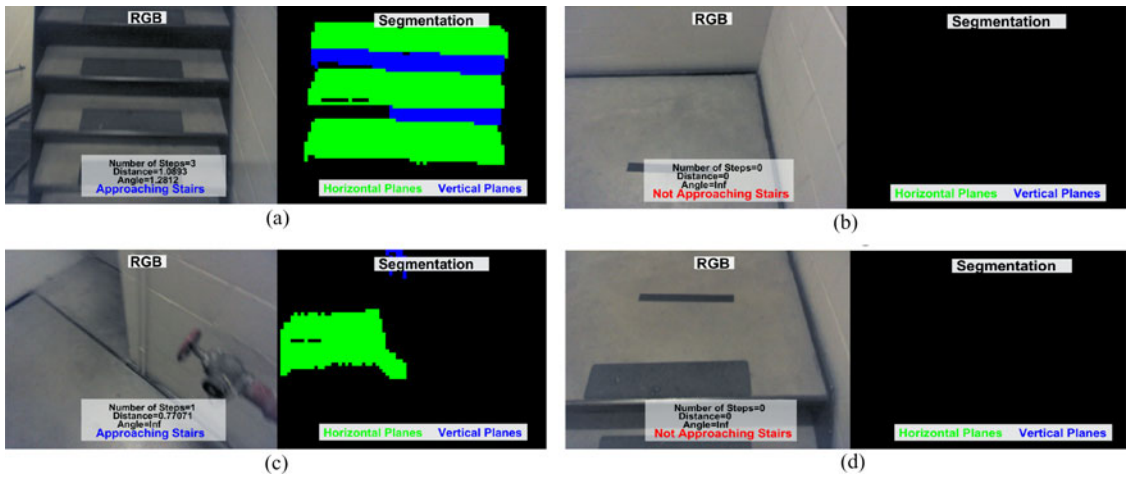

Fig. 9. *Experiment 3*—Online test results. The left half of each frame is the RGB stream, and the right half is a 2-D image of the stair segmentation with horizontal stair faces shown in green and vertical stair faces shown in blue. Clockwise from top left: True positive, true negative, false negative, false Positive. The number of steps, distance, angle, and approaching/not approaching status are displayed above the RGB image. Approaching statuses are shown in blue, and not approaching statuses are shown in red (a) True Positive (b) True Negative (c) False Positive (d) False Negative. (Best viewed in color).

the accuracy should be the main criteria for selecting the preferred resolution. In contrast, there is a large difference in frame rate between resolutions, suggesting that the frame rate played a more meaningful role in the selection of the desired resolution.

Experiment 2 indicated the capability of using the segmentation algorithm for extracting additional information for prosthesis control. Tests on five different staircases showed that we can successfully estimate the number of steps, as well as the height and depth of the stairs, using the proposed algorithm. The computed mean stair heights were generally within 1 cm of the measured stair height, and the computed mean stair depths were generally within 3 cm of the measured stair depth. If the step height and depth are used to enhance prosthesis control, then these margins of error can be factored into the control for the best performance (for instance, due to uncertainty an additional 1 cm or more could be incorporated into the predicted toe clearance). Additionally, we showed that the distance estimate was accurate between different staircases (i.e., average estimated error was within 5 cm for all distances and staircases), also showing minimal variability (i.e., overall standard deviation was 2.3 cm). Similarly, the angle of intersection error estimate was not different from *Experiment 1* (i.e., mean error was 1.0°).

Finally, we validated the segmentation algorithm in a dynamic online walking test. *Experiment 3* demonstrated that we can perform the stair segmentation online in a real-life situation, such as a walking and stair ambulation. In this condition, we recorded an accuracy of the "approaching/not approaching stairs" status of 98.8% of frames, with only one false positive and one false negative produced. This result proved the potential of the proposed segmentation algorithm for improving intent recognition.

While the accuracy we obtained with the proposed algorithm was already satisfactory for an online use, we interpreted each frame separately, rather than combining them to detect the ambulation mode on every step that the individual takes. However, in real-world applications, evaluating how well the system performs per each step taken is more important. Although analyzing step accuracy rate goes beyond the goal of this paper, a simple rule that incorporates prior information to define the step status might be enough to prevent errors due to a single missed frame.

For example, by changing the locomotion mode only after two consecutive frames of matching status, the current two frame errors would be eliminated; thus, obtaining a 100% step accuracy with minimal time delay. Despite the good outcomes of this simple averaging rule, we plan to implement an adaptive filtering, such as the Kalman filter [68], to fuse the segmentation metrics and produce the final step accuracy in our future system.

This study demonstrates the potential of depth sensing to improve intent recognition and control of lower limb prostheses. However, this study has limitations. The errors recorded from the online walking test show two main issues with the use of the proposed algorithm in real-life situations. First, our algorithm relies on the Kinect onboard accelerometers to estimate the absolute Euler rotation angle of the depth camera in respect to the environment. While this approach allows for accurate rotation angle estimates in static conditions, it does not perform well in dynamic conditions, such as in *Experiment 3*. Notably, we use the rotation angle to transform the depth data from the sensor-based coordinate system to the global coordinate system. Therefore, an error in the rotation angle results in an offset in the depth data that necessarily affects the segmentation algorithm and the resulting segmentation measures.

The other critical point in our algorithm is in the method used for locating the floor. In the current implementation of our algorithm, the floor location depends on the offset in the z-direction, defined in this paper as the measured height of the depth sensor. As a consequence, if the sensor height changes (which can likely happen during dynamic conditions), the floor may be slightly offset, causing issues in the segmentation accuracy. For instance, if the floor is offset slightly above zero, a part of the floor may be segmented as a horizontal surface, such as in the false positive result [see Fig. 9(c)]. On the other hand, if the floor is offset below zero, the bottom step may be considered as a part of the floor and removed from the segmentation, as in the false negative result [see Fig. 9(d)]. We believe these two issues account for the false positive and negative segmentation of the online walking test, and can help to explain the discrepancy between the average step height and the depth in the online test and their measured height and depth (mean errors of 2.9 and 6.3 cm, respectively).

Additionally, though the sensor was held at roughly the same height and tilt throughout the walking experiment (height of $\sim$1.5 m and tilt of about -50° from horizontal), it shifted and shook as the subject walked due to the movement and a loose connection. This may have contributed to the two errors that were recorded in the online walking test, by negatively affecting the floor extraction. By adding an IMU and producing a better tilt estimate, we hope to be able to reduce issues due to shaking.

Our experiments proved that the high segmentation accuracy can be obtained. However, better results can be achieved by implementing two minor changes to our system. First, we will use a six degree-of-freedom IMU, rather than the onboard Kinect accelerometer, to determine the orientation of the depth sensor. Using a sensory fusion algorithm, such as a Kalman filter, we aim to improve a tilt angle tracking, and reduce errors and drift, by filtering and optimally fusing the information obtained by a three-axis accelerometer and a three-axis gyroscope [69]–[73]. Second, we will consider other methods to locate the floor that do not depend on the height of the sensor or a defined offset [74]. Better segmentation of the floor should improve the robustness of the segmentation algorithm to angle errors.

Though we have shown that our proposed stair detection algorithm can produce true positive detections of stairs, it is unclear whether our system is vulnerable to false positives due to misperception of boxes on the ground, platforms, etc., as being stairs. More extensive tests will be required to explore this issue. However, we believe that the fusion of the environmental data with the subject-based data sources may help to reduce the possibility of misclassifying a locomotion mode transition due to false positives. For instance, if subjects stand directly in line with a box that is erroneously classified as stairs (i.e., false positive), but their kinetics and kinematics do not correspond to someone moving toward the box or attempting to climb stairs [11], then the false positive can potentially be eliminated. The optimal method for fusing the subject- and environment-data for producing the overall status will be a major focus of the future work.

We aim to investigate methods for fusing segmentation measures obtained from our algorithm with the EMG and mechanical sensors that have been previously used for intent recognition [14]. We will consider how differences in approach speed, occlusions, and the height of camera placement affect the accuracy of our segmentation algorithm. Further, we will investigate the use of stair height, depth, and count to improve the performance of the mode-specific controller of the powered prosthesis, for example, to increase toe clearance during swing phase. In future work, we also plan to test several different locations, where a depth sensor could be worn, such as at the chest level (e.g., Narrative Clip [75], SnapCam [76], or QindredCam [77]), attached to a belt, or at eye-level (e.g., Google Glass [34]), to determine the optimal placement. While the Kinect performed well for this preliminary study, we intend to use a smaller less conspicuous 3-D sensor, such as the occipital structure sensor [32], in a final system that would be tested on individuals with amputations.

We intend to test the proposed intent recognition system on individuals with transfemoral amputations. For this application, the onboard prosthesis processor will receive data from EMG sensors on the subject's residual limb, as well as mechanical sensors from the prosthesis. This data will be fused with the information from our vision and depth sensors to predict the desired ambulation mode, which will then be converted into motor commands using a dedicated controller in the prosthesis [21], [78], [79]. We anticipate the data fusion process will use adaptive filtering methods, with probabilities of desired locomotion mode being updated at every step. Our future work will aim to determine the optimal way of fusing data from different sensor modalities, as well as to test the outcomes of the proposed sensory fusion method. Finally, the proposed segmentation method could potentially be used in the semiautonomous control of a powered exoskeleton [80] and powered wheelchair [81].

VII. Conclusion

We developed an algorithm to segment stairs from depth sensing of the environment and extract significant measures to detect

the user intention, such as the distance and angle from the stairs, the number of steps, stair height, and stair depth. We conducted three experiments to characterize and validate the stair segmentation algorithm online. *Experiment 1* determined the ideal resolution for online stair segmentation to be 80×60 pixels based on accuracy and speed. *Experiment 2* tested the robustness of the stair segmentation algorithm against five different staircases. Finally, *Experiment 3* involved an online walking test to demonstrate the feasibility of the proposed stair segmentation algorithm for online use. This test resulted in 98.8% accuracy. Future work will focus on improving the segmentation algorithm, and integrating depth-based measures with EMG, kinetics, and kinematics for ambulation mode recognition in powered lower limb prostheses. Also, we will extend the depth sensing to detect other environmental constraints, such as ramps or descending stairs.

ACKNOWLEDGMENT

The authors would like to thank S. M. Burt for helping to revise and edit this paper.

REFERENCES

[1] F. Sup *et al.*, "Design and control of a powered transfemoral prosthesis frank," *Int. J. Robot. Res.*, vol. 27, no. 2, pp. 263–273, 2008.
[2] S. K. Au *et al.*, "Biomechanical design of a powered ankle-foot prosthesis," in *Proc. IEEE 10th Int. Conf. Rehabil. Robot.*, 2007, pp. 298–303.
[3] Ossur. Ossur power knee. (2015, Jun.). [Online]. Available: http://www.ossur.com/prosthetic-solutions/products/knees-and-legs/bionic-knees/power-knee
[4] I. C. Narang *et al.*, "Functional capabilities of lower limb amputees," *Prosthet. Orthot. Int.*, vol. 8, no. 1, pp. 43–51, 1984.
[5] T. Schmalz *et al.*, "Energy expenditure and biomechanical characteristics of lower limb amputee gait: The influence of prosthetic alignment and different prosthetic components," *Gait Posture*, vol. 16, no. 3, pp. 255–263, Dec. 2002.
[6] W. C. Miller *et al.*, "The prevalence and risk factors of falling and fear of falling among lower extremity amputees," *Arch. Phys. Med. Rehabil.*, vol. 82, no. 8, pp. 1031–1037, Aug. 2001.
[7] R. L. Waters and S. Mulroy, "The energy expenditure of normal and pathologic gait," *Gait Posture*, vol. 9, no. 3, pp. 207–231, Jul. 1999.
[8] A. M. Simon *et al.*, "Configuring a powered knee and ankle prosthesis for transfemoral amputees within five specific ambulation modes," *PLoS One*, vol. 9, no. 6, p. e99387, Jan. 2014.
[9] H. A. Varol *et al.*, "Multiclass real-time intent recognition of a powered lower limb prosthesis," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 3, pp. 542–551, Mar. 2010.
[10] L. Peeraer *et al.*, "Development of EMG-based mode and intent recognition algorithms for a computer-controlled above-knee prosthesis," *J. Biomed. Eng.*, vol. 12, no. 3, pp. 178–182, 1990.
[11] D. Novak *et al.*, "Automated detection of gait initiation and termination using wearable sensors," *Med. Eng. Phys.*, vol. 35, no. 12, pp. 1713–1720, 2013.
[12] M. Goršič *et al.*, "Online phase detection using wearable sensors for walking with a robotic prosthesis," *Sensors*, vol. 14, no. 2, pp. 2776–2794, Jan. 2014.
[13] I. P. I. Pappas *et al.*, "A reliable gait phase detection system," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 9, no. 2, pp. 113–125, Jun. 2001.
[14] H. Huang *et al.*, "Continuous locomotion-mode identification for prosthetic legs based on neuromuscular—Mechanical fusion," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 10, pp. 2867–2875, Oct. 2011.
[15] L. J. Hargrove *et al.*, "Robotic leg control with EMG decoding in an amputee with nerve transfers," *N. Engl. J. Med.*, vol. 369, no. 13, pp. 1237–1242, Sep. 2013.
[16] M. T. Farrell and H. Herr, "A method to determine the optimal features for control of a powered lower-limb prostheses," in *Proc. IEEE Annu. Int. Conf. Med. Biol. Soc.*, Jan. 2011, vol. 2011, pp. 6041–6046.
[17] E. Zheng *et al.*, "A noncontact capacitive sensing system for recognizing locomotion modes of transtibial amputees," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 12, pp. 2911–2920, Dec. 2014.
[18] X. Guo *et al.*, "A study on control mechanism of above knee robotic prosthesis based on CPG model," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2010, pp. 283–287.
[19] A. Furse *et al.*, "Improving the gait performance of non-fluid-based swing-phase control mechanisms in transfemoral prostheses," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 8, pp. 2352–2359, Aug. 2011.
[20] L. Ambrozic and M. Gorsic, "CYBERLEGs: A user-oriented robotic transfemoral prosthesis with whole-body awareness control," *IEEE Robot. Autom. Mag.*, vol. 21, no. 4, pp. 82–93, Dec. 2014.
[21] C. D. Hoover *et al.*, "Stair ascent with a powered transfemoral prosthesis under direct myoelectric control," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 3, pp. 1191–1200, Jun. 2013.
[22] L. Chen *et al.*, "Above-knee prosthesis control based on posture recognition by support vector machine," in *Proc. IEEE Robot. Autom. Mechatron. Conf.*, 2008, pp. 307–312.
[23] B. B. E. Lawson *et al.*, "A robotic leg prosthesis: Design, control, and implementation," *IEEE Robot. Autom. Mag.*, vol. 21, no. 4, pp. 70–81, Dec. 2014.
[24] A. J. Young *et al.*, "Intent recognition in a powered lower limb prosthesis using time history information," *Ann. Biomed. Eng.*, vol. 42, no. 3, pp. 631–641, Mar. 2014.
[25] F. Zhang *et al.*, "Effects of locomotion mode recognition errors on volitional control of powered above-knee prostheses," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 23, no. 1, pp. 64–72, Jan. 2015.
[26] D. A. Winter, *Biomechanics and Motor Control of Human Gait: Normal, Elderly and Pathological*. Waterloo, ON, Canada: Waterloo Univ. Press, 1991.
[27] J. A. Delmerico *et al.*, "Ascending stairway modeling from dense depth imagery for traversability analysis," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 2283–2290.
[28] R. C. Luo *et al.*, "Multisensor integrated stair recognition and parameters measurement system for dynamic stair climbing robots," in *Proc. IEEE Int. Conf. Autom. Sci. Eng.*, Aug. 2013, pp. 318–323.
[29] J.-S. Gutmann *et al.*, "Stair climbing for humanoid robots using stereo vision," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2004, vol. 2, pp. 1407–1413.
[30] S. Wang *et al.*, "RGB-D image-based detection of stairs, pedestrian crosswalks and traffic signs," *J. Vis. Commun. Image Represent.*, vol. 25, no. 2, pp. 263–272, Feb. 2014.
[31] Microsoft. Microsoft kinect for windows v1. (2014, Mar.). [Online]. Available: http://www.microsoft.com/en-us/kinectforwindows
[32] Occipital Inc. Structure sensor. (2015, Jun.). [Online]. Available: http://structure.io/
[33] PMD. Camboard pico XS. (2015, Jun.). [Online]. Available: http://pmdtec.com/news_media/news/ pico_xs.php
[34] Google. Google glass. (2014, Oct.). [Online]. Available: https://www.google.com/glass/start/
[35] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, Jul. 1986.
[36] N. E. Krausz and L. J. Hargrove, "Recognition of ascending stairs from 2D images for control of powered lower limb prostheses," in *Proc. IEEE Int. Conf. Neural Eng. Rehab.*, 2015, pp. 615–618.
[37] S. Wang and H. Wang, "2D staircase detection using real AdaBoost," Proc. presented at the 7th Int. Conf. Inf., Commun. Signal Process., Macau, 2009.
[38] L. Maohai *et al.*, "A robust vision-based method for staircase detection and localization," *Cogn. Process.*, vol. 15, no. 2, pp. 173–194, 2014.
[39] D. Hernández *et al.*, "Stairway detection based on single camera by motion stereo," *Mod. Approaches Appl. Intell.*, vol. 6703, pp. 338–347, 2011.
[40] C. Zhong *et al.*, "Stairway detection using gabor filter and FFPG," in *Proc. Int. Conf. Soft Comput. Pattern Recog.*, Oct. 2011, pp. 578–582.
[41] S. Shahrabadi *et al.*, "Detection of indoor and outdoor stairs," *Pattern Recog. Image Anal.*, vol. 7887, pp. 847–854, 2013.
[42] D. Hernandez and K. Jo, "Outdoor stairway segmentation using vertical vanishing point and directional filter," in *Proc. Int. Forum Strategic Technol.*, 2010, pp. 82–86.
[43] J. A. Hesch *et al.*, "Descending-stair detection, approach, and traversal with an autonomous tracked vehicle," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Oct. 2010, pp. 5525–5531.
[44] Y. Cong *et al.*, "A stairway detection algorithm based on vision for UGV stair climbing," in *Proc. Int. Conf. Netw., Sens. Control*, 2008, pp. 1806–1811.

[45] S. Se and M. Brady, "Road feature detection and estimation," *Mach. Vis. Appl.*, vol. 14, no. 3, pp. 157–165, 2003.
[46] D. Hoiem *et al.*, "Putting objects in perspective," *Int. J. Comput. Vis.*, vol. 80, no. 1, pp. 3–15, Apr. 2008.
[47] D. Hoiem *et al.*, "Geometric context from a single image," in *Proc. IEEE 10th Int. Conf. Comput. Vis.*, 2005, vol. 1, pp. 654–661.
[48] E. Delage *et al.*, "Automatic single-image 3D reconstructions of indoor manhattan world scenes," *Robot. Res.*, vol. 28, pp. 305–321, 2007.
[49] E. Johns and G. Z. Yang, "Feature co-occurrence maps: Appearance-based localisation throughout the day," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 3212–3218.
[50] W. Maddern *et al.*, "Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles," in *Proc. Vis. Place Recog. Changing Environ. Workshop*, 2014.
[51] L. Qiang and L. Feng, "RGB-D sensor based mobile robot SLAM in indoor environment," in *Proc. Control Decision Conf.*, 2014, pp. 3848–3852.
[52] R. Schnabel *et al.*, "Efficient RANSAC for point-cloud shape detection," *Comput. Graph. Forum*, vol. 26, no. 2, pp. 214–226, Jun. 2007.
[53] X. Qian and C. Ye, "NCC-RANSAC: A fast plane extraction method for navigating a smart cane for the visually impaired," *IEEE Int. Conf. Autom. Sci. Eng.*, 2013, pp. 261–267.
[54] A. Hidalgo-Paniagua *et al.*, "A comparative study of parallel RANSAC implementations in 3D space," *Int. J. Parallel Program.*, available online Jul. 2014.
[55] V. Pradeep *et al.*, "Piecewise planar modeling for step detection using stereo vision," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 1–8.
[56] Y. Lee *et al.*, "Real-time staircase detection from a wearable stereo system," in *Proc. Int. Conf. Pattern Recog.*, 2012, pp. 3770–3773.
[57] D. Novak and R. Riener, "A survey of sensor fusion methods in wearable robotics," *Robot. Auton. Syst.*, 2014, in press.
[58] M. R. Tucker *et al.*, "Control strategies for active lower extremity prosthetics and orthotics: a review," *J. Neuroeng. Rehabil.*, vol. 12, no. 1, pp. 1–29, 2015.
[59] L. Du *et al.*, "Toward design of an environment-aware adaptive locomotion-mode-recognition system," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 10, pp. 2716–2725, Oct. 2012.
[60] M. Liu *et al.*, "Development of an environment-aware locomotion mode recognition system for powered lower limb prostheses," *IEEE Trans. Neural Syst. Rehabil. Eng.*, 2015, in press.
[61] X. Zhang *et al.*, "An automatic and user-driven training method for locomotion mode recognition for artificial leg control," in *Proc. IEEE Annu. Int. Conf. Eng. Med. Biol. Soc.*, 2012, pp. 6116–6119.
[62] R. Pio, "Euler angle transformations," *IEEE Trans. Autom. Control*, vol. AC-11, no. 4, pp. 707–715, Oct. 1966.
[63] J. F. Knight *et al.*, "Uses of accelerometer data collected from a wearable system," *Pers. Ubiquitous Comput.*, vol. 11, no. 2, pp. 117–132, May 2006.
[64] E. Trucco and A. Verri, "Introductory techniques for 3-D computer vision," in *Introductory Techniques for 3-D Computer Vision*. Upper Saddle River, NJ, USA, Prentice-Hall, 1998.
[65] R. Haralick, "Image analysis using mathematical morphology," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. PAMI-9, no. 4, pp. 532–550, Jul. 1987.
[66] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory.*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.
[67] M. Dillencourt *et al.*, "A general approach to connected-component labeling for arbitrary image representations," *J. ACM*, vol. 39, no. 2, pp. 253–280, 1992.
[68] R. E. Kalman, "A new approach to linear filtering and prediction problems[1]," *J. Basic Eng.,* vol. 82, pp. 35–45, 1960.
[69] R. Mahony *et al.*, "Nonlinear complementary filters on the special orthogonal group," *IEEE Trans. Autom. Control*, vol. 53, no. 5, pp. 1203–1218, Jun. 2008.
[70] S. O. H. Madgwick *et al.*, "Estimation of IMU and MARG orientation using a gradient descent algorithm," presented at the IEEE Int. Conf. Rehabilitation Robotics, Zurich, Switzerland, 2011.
[71] S. P. Won *et al.*, "A kalman/particle filter-based position and orientation estimation method using a position sensor/inertial measurement unit hybrid system," *IEEE Trans. Ind. Electron.*, vol. 57, no. 5, pp. 1787–1798, May 2010.
[72] A. Correa *et al.*, "Enhanced inertial-aided indoor tracking system for wireless sensor networks: A review," *IEEE Sensors J.*, vol. 14, no. 9, pp. 2921–2929, Sep. 2014.
[73] Y. Tian *et al.*, "Accurate human navigation using wearable monocular visual and inertial sensors," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 1, pp. 203–213, Jan. 2014.
[74] D. Kircali and F. B. Tek, *Ground Plane Detection Using an RGB-D Sensor*. New York, NY, USA: Springer, 2014.
[75] Narrative. Narrative Clip 2. (2015, Jun.). [Online]. Available: http://getnarrative.com/
[76] I. Camera. SnapCam. (2015, Jun.). [Online]. Available: http://usa.ioncamera.com/snapcam/
[77] Acumulus9. QindredCam. (2015, Jun.). [Online]. Available: http://www.acumulus9.com/qindredcam/
[78] T. Lenzi *et al.*, "Speed-adaptation mechanism: Robotic prostheses can actively regulate joint torque," *IEEE Robot. Autom. Mag.*, vol. 21, no. 4, pp. 94–107, Dec. 2014.
[79] B. E. Lawson *et al.*, "Control of stair ascent and descent with a powered transfemoral prosthesis," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 21, no. 3, pp. 466–473, May 2013.
[80] A. Dollar and H. Herr, "Lower extremity exoskeletons and active orthoses: Challenges and state-of-the-art," *IEEE Trans. Robot.*, vol. 24, no. 1, pp. 144–158, Feb. 2008.
[81] U. Borgolte *et al.*, "Architectural concepts of a semi-autonomous wheelchair," *J. Intell. Robot. Syst.*, vol. 22, pp. 233–253, 1998.

Nili E. Krausz (S'13) received the B.S. and M.S. degrees in mechanical engineering from the University of Colorado Denver, Denver, CO, USA, in 2011 and 2013, respectively. She is currently working toward the Ph.D. degree in biomedical engineering from Northwestern University, Evanston, IL, USA.

She conducts research at the Center for Bionic Medicine at the Rehabilitation Institute of Chicago, IL. Her research interests include robotics, mechatronics, and neurophysiology for rehabilitation and assistive wearable devices, with emphasis on mechanical design, shared and semiautonomous control, and computer vision for improved intent recognition and human machine interaction.

Tommaso Lenzi (S'11–M'13) received the M.Sc. degree in biomedical engineering from the University of Pisa, Pisa, Italy, in 2008, and the Ph.D. degree in biorobotics from Scuola Superiore Sant., (sup size) Anna, Pisa, in 2012.

From August 2011 to February 2012, he was Visiting Research Scientist with the Department of Mechanical Engineering, University of Delaware, Newark, DE, USA. He is currently a Research Scientist with the Center for Bionic Medicine, Rehabilitation Institute of Chicago, IL, USA, and the Department of Physical Medicine and Rehabilitation, Northwestern University, Chicago, IL. He has coauthored 18 ISI journal papers and 31 peer-review conference proceedings papers. His main research interests include robotics, mechatronics, and rehabilitation medicine, with a major emphasis on the design and control of wearable robots for human assistance and rehabilitation.

Levi J. Hargrove (S'05–M'08) received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the University of New Brunswick, Fredericton, NB, Canada, in 2003, 2005, and 2007, respectively.

He joined the Center for Bionic Medicine at the Rehabilitation Institute of Chicago, IL, USA, in 2008. He is also a Research Assistant Professor with the Department of Physical Medicine and Rehabilitation and Biomedical Engineering, Northwestern University, Evanston, IL. His research interests include pattern recognition, biological signal processing, and myoelectric control of powered prostheses.

Dr. Hargrove is a Member of the Association of Professional Engineers and Geoscientists of New Brunswick.