FilterJoint: Toward an Understanding of Whole-Body Gesture Articulation

Aishat Aloba
Department of CISE
University of Florida
Gainesville, FL USA
aoaloba@ufl.edu

Julia Woodward
Department of CISE
University of Florida
Gainesville, FL USA
julia.woodward@ufl.edu

Lisa Anthony
Department of CISE
University of Florida
Gainesville, FL USA
lanthony@cise.ufl.edu

ABSTRACT

Classification accuracy of whole-body gestures can be improved by selecting gestures that have few conflicts (i.e., confusions or misclassifications). To identify such gestures, an understanding of the nuances of how users articulate whole-body gestures can help, especially when conflicts may be due to confusion among seemingly dissimilar gestures. To the best of our knowledge, such an understanding is currently missing in the literature. As a first step to enable this understanding, we designed a method that facilitates investigation of variations in how users move their body parts as they perform a motion. This method, which we call *filterJoint*, selects the key body parts that are actively moving during the performance of a motion. The paths along which these body parts move in space over time can then be analyzed to make inferences about how users articulate whole-body gestures. We present two case studies to show how the filterJoint method enables a deeper understanding of whole-body gesture articulation, and we highlight implications for the selection of whole-body gesture sets as a result of these insights.

CCS CONCEPTS

• Human-centered computing~Gestural input • Human-centered computing~Graphics input devices • Social and professional topics~Children

KEYWORDS

Whole-body gestures; motions; template matching; Kinect; gesture recognition; whole-body gesture articulation

ACM Reference format:

Aishat Aloba, Julia Woodward and Lisa Anthony. 2020. FilterJoint: Toward an Understanding of Whole-Body Gesture Articulation. In *Proceedings of 2020 ACM International Conference on Multimodal Interaction (ICMI'20), Oct 25-29, Virtual Event, Netherlands.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3382507.3418822

ICMI'20, October 25–29, 2020, Virtual Event, Netherlands.

ACM ISBN 978-1-4503-7581-8/20/10...$15.00.
DOI: https://doi.org/10.1145/3382507.3418822

1 Introduction

Advances in motion sensor technologies, such as the Microsoft Kinect [16], have increased the popularity of applications that support whole-body gesture interaction (e.g., games [10], human-robot interaction [26]). To support whole-body gesture interaction, applications include recognizers that can accurately classify gestures within the whole-body gesture set that the application supports. Accuracy of classifying these whole-body gestures can be improved by selecting gestures that have few conflicts (i.e., confusions or misclassification). Thus, designers try to select gesture sets to include gestures that are distinct in terms of how the gesture should be performed. For example, the Microsoft Research Cambridge-12 Kinect gesture data set [17], designed to accompany Kinect gaming applications, has six distinct iconic gestures: crouch or hide, shoot a pistol, throw an object, change weapon, kick, and put on night goggles.

Ideally, designers expect that such distinct gestures are not likely to be confused for each other by the user or by the system. However, prior work in 2D stroke gesture research has found that distinct gestures can still be confused for each other by the recognizer. For example, $N-Protractor [4] confused the letters "*A*" and "*K*" unexpectedly due to internal pre-processing that made them look alike to the recognizer [2]. Stroke gesture research has asserted that this unexpected confusion can also result from nuances in how users articulate stroke gestures, defined by the path the finger moves in 2D space over time. Like 2D stroke gestures, whole-body gestures also involve paths moving in space over time; each body part moved by a user creates a path in 3D space over the duration of the motion. Therefore, we posit that nuances in how users articulate whole-body gestures (i.e., how users actually move their body parts through space when making movements) might also make it difficult for designers to predict which whole-body gestures are confusable. If designers understand how users articulate whole-body gestures, then they could select better gesture sets that have fewer conflicts.

To the best of our knowledge, an understanding of these nuances in how users articulate whole-body gestures is currently

missing in the literature. We propose that such understanding can be achieved by identifying the joints that are critical to performing motions and characterizing the features that can describe how the motion is produced. In this work, we focus on the former as a first step to enable this understanding. We designed a method that facilitates investigation of variations in how users move body parts as they perform the gesture. In this method, we operate on 'joints' because our method works on motion that is tracked using skeleton-based motion sensor technology, such as the Microsoft Kinect sensor [16] or the Vicon motion capture sensor (Oxford Metric, Oxford UK), which track the motion of users' joints in space over time. Our method, which we call **filterJoint**, selects the key joints that are actively moving during the performance of the motion. Our focus on actively moving joints stems from the idea that both tracking noise from the sensor and unintentional movements from the user can affect the articulation path (e.g., joints that are not supposed to move will appear to move). Including such articulation paths could introduce noise in the recognition process, thus affecting recognition accuracy, and can also lead to incorrect inferences about how users articulate motions.

To evaluate whether the filterJoint method does indeed select the actively moving joints and is useful for understanding whole-body gesture articulation, we use components from existing template-based gesture recognition algorithms ($3 [13], Protractor [15], and Protractor3D [14]), adapted to account for multiple articulation paths in whole-body gestures. These algorithms compare an articulation of a gesture to other articulations of a set of gestures using point-by-point correspondence and select the articulation that most closely resembles the one being tested based on a distance metric. We use recognition accuracy for the evaluation since tracking noise of the motion sensor and users' unintentional movements are likely to introduce noise in the recognition process, which will negatively impact recognition accuracy. Our evaluation of the filterJoint method using this adaptation on a representative set of adult gestures from the Kinder-Gator dataset [1], a publicly-available dataset of children's and adults' motions, showed that our method (90.7% [SD=6.8%]) achieved a significantly higher recognition accuracy compared to a baseline method involving all joints (81.4% [SD=6.9%]). This finding confirms that our filterJoint method is successful in selecting the key joints that are necessary for articulating whole-body gestures. Thus, the paths along which these joints move in space over time can be analyzed to make inferences about how users articulate whole-body gestures.

We present two case studies to show how the filterJoint method enables a deeper understanding of whole-body gesture articulation, and we highlight implications for the selection of whole-body gesture sets as a result of these insights. For example, we use our recognition results to make intuitive inferences about whole-body gesture articulation. We found that between-user inconsistencies in how users articulate whole-body gestures resulted in unexpected overlaps between dissimilar gestures. For example, we found that the gesture "*Put your hands on your hips and lean to the side (phl)*" was often confused with two seemingly-dissimilar gestures: "*Lift your leg to one side (lyl)*" and "*Kick a ball as hard as you can (kbh)*" based on how users actually articulate these motions (e.g., there was active movement of the foot joint in *phl* when we only expect active movements in the upper limbs). From these findings, designers can use our recognition adaptation approach, which relies on the filterJoint method, as a tool to identify overlaps between whole-body gestures during the selection of whole-body gesture sets.

The contributions of this paper are: (a) an automated approach that filters out tracking noise and unintentional movements to select only the actively moving joints during articulation of a gesture; (b) an adaptation of template-based gesture recognition algorithms to multiple articulation paths, which can be used as a tool to understand whole-body gesture articulation; (c) an investigation of how users articulate whole-body gestures from a specific dataset that shows how useful our approach is for understanding articulation differences; and (d) a set of recommendations to improve selection of whole-body gesture sets for interactive applications. We hope that designers can use our method to understand how users articulate whole-body gestures to select better gestures sets and improve recognition in their applications.

2 Related Work

We review relevant prior work on whole-body gesture articulation and touchscreen stroke gesture articulation.

2.1 Whole-body Gesture Articulation

Supporting whole-body gesture interaction requires accurate recognition of whole-body gestures. To enable accurate recognition, whole-body gesture research has tried various approaches to represent whole-body gestures. For example, Bobick and Davis [6] used Hu moments to compute statistical descriptors (moment-based features) of Motion History Image (MHI) and Motion Energy Image (MEI) representations of video sequences of whole-body gestures. The researchers achieved a recognition accuracy of 83.3% on a whole-body gesture set comprising aerobic exercises. Weinland et al. [27] computed Fourier-based features from Motion History Volume (MHV) representations of whole-body gestures. The researchers achieved accuracies ranging from 73.3% to 93.33% on the IXMAS dataset, a dataset of adults' motions comprising 11 everyday actions (e.g., wave and walk). However, regardless of representation, the selection of whole-body gesture sets is still a challenge because conflicts among gestures within the set can negatively impact recognition.

Whole-body application datasets (e.g., the Microsoft Research Cambridge-12 Kinect gesture dataset [17] and the G3di gaming interaction dataset [5]) aim to include distinct gestures to prevent conflicts. However, the related field of stroke gesture research has found that nuances in how users articulate gestures can result in unexpected conflicts between distinct gestures [3,25], and researchers have used insights from stroke gesture articulation to inform the design of stroke gesture sets. Like stroke gestures, whole-body gestures are defined by paths in

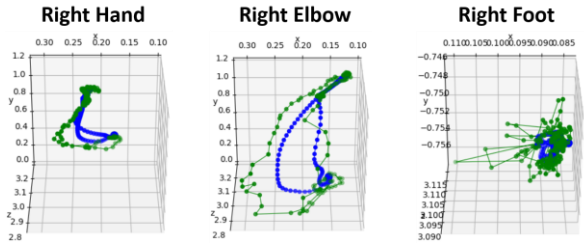

Figure 1. Motion paths when raising the right hand before (green) and after (blue) smoothing using the exponential moving average filter. (Left; Middle) Joints that move and (Right) a joint that does not move.

space over time, so understanding whole-body gesture articulation could inform the selection of whole-body gesture sets. However, the nuances of how users articulate whole-body gestures has been largely overlooked in the literature. To the best of our knowledge, only one prior study considered the idea of understanding whole-body gesture articulation. Vatavu [22] identified features to quantify spatial characteristics (e.g., amount of space required to perform the gesture), kinematic performance (e.g., time it takes to produce the gesture), and body posture appearance (e.g., deviation of the body posture from the centroid posture). Although the author speculates that these features can be used to understand whole-body gesture articulation, the extent to which these features can characterize whole-body gesture articulation is not known. In our work, we created the filterJoint method to enable targeted analysis of the key articulation paths themselves to inform an understanding of whole-body gesture articulation.

2.2 Touchscreen Stroke Gesture Articulation

Touchscreen stroke gestures are defined by the articulation path of a user's finger or stylus on a touch sensitive surface [21]. Research has studied how users articulate touchscreen stroke gestures to inform stroke gesture interaction (e.g., the selection of gesture sets). Anthony et al. [3] identified 10 geometric features (e.g., path length, area of the bounding box) and 2 kinematic features (production time, average speed) that characterize an instance of a gesture to investigate the consistency in how users articulate stroke gestures. For example, the researchers found that consistency in stroke gesture articulation depends on gesture simplicity and users' familiarity with the gesture. From these findings, the authors recommended several guidelines, such as that stroke gesture designers should prefer unistroke gestures over multistroke gestures and familiar gestures (e.g., shapes) over unfamiliar gestures. Vatavu et al. [23] further defined a representative articulation of a gesture as a "gesture task axis" and identified 12 *relative accuracy* features that characterize differences between an articulation path and the gesture task axis. The authors revealed new findings about stroke gesture articulation, such as that gestures articulated at very fast speeds (e.g., when completion time is important) have more shape errors (i.e., absolute deviation of the articulation from the gesture task axis) as opposed to gestures articulated at medium speeds (e.g., when both completion time and accuracy of the gesture are important). Similar to such insights in stroke gesture research, our work aims to inform an understanding of whole-body gesture articulation and use this understanding to aid the selection of whole-body gesture sets.

3 FilterJoint Method

To enable an understanding of whole-body gesture articulation, we identify the *articulation paths* (defined as the 3D positions of joints over the duration of the motion) of joints that users move during the performance of a whole-body gesture. We focus on 'joints' because we assume that users' movements will be tracked using skeleton-based motion sensor technology, such as the Microsoft Kinect [16] or the Vicon motion capture system (Oxford Metric, Oxford UK). Whole-body gestures are characterized by multiple articulation paths but not all joints tracked by the motion sensor are necessary to articulate a gesture. Tracking errors of the motion sensor [19] and unintentional movements from the user could make joints that are not supposed to move appear to move. For example, to raise one's hands, only the joints in the upper limbs (e.g., hand, elbow) should move, while any movement in the joints of the lower limbs (e.g., foot) is likely due either to tracking noise of the sensor or to unintentional movements from the user (see Figure 1). Including these articulation paths during the analysis of whole-body gesture articulation could result in incorrect inferences about how users articulate whole-body gestures.

3.1 Selecting Key Joints

To identify the joints that are necessary to articulate a whole-body gesture, we designed a method that automatically identifies only the joint paths that are due to intentional movements from the user. In this method, which we call the **filterJoint** method, we attempt to select the actively moving joints for a particular whole-body motion gesture instance by computing the variations in joint movement and using a k-means algorithm to group the variations into two clusters. The joints in the cluster with the higher mean (i.e., higher range of motion) are selected as the actively moving joints for that motion. Given a motion instance m for which each joint is defined by N 3D points:

$$m = \begin{bmatrix} j_1^1 & j_1^2 & \dots & j_1^K \\ j_2^1 & j_2^2 & \dots & j_2^K \\ \vdots & \vdots & \vdots & \vdots \\ j_N^1 & j_N^2 & \dots & j_N^K \end{bmatrix}$$

where j_i^k is a 3D point showing the position of j^k at time instance i, N is the number of points in the motion path of j^k, and K is the number of joints tracked by the sensor. We apply the following steps to extract the moving joints for each motion instance:

1. First, we smooth the articulation path of each of the joints in m using an exponential moving average filter with alpha = 0.1. Prior work [20] has found that this filter can remove noise from joint data without introducing any smoothing artifacts to the data. The aim of the smoothing process is to reduce articulation path noise due to tracking issues of the motion sensor.

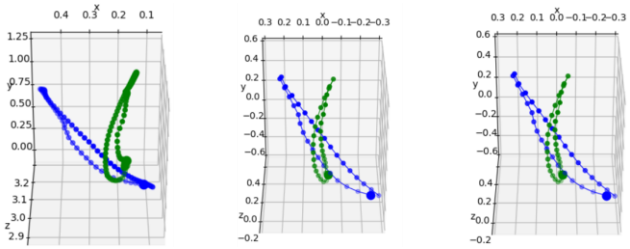

Figure 2. (Left) Motion paths for the gestures "*Raise your arm to one side*" (blue) and "*Raise your hand*" (green) for the same joint. (Middle) Same paths prior to optimal alignment. (Right) Same paths after optimal alignment, where global rotational orientation is maintained.

2. We compute variations for each joint movement j^k in m by computing the standard deviation (SD):

$$SD\ (j^k) = \sqrt{\frac{\sum_{i=1}^{N}\left(j_{ix}^k - \bar{j_x^k}\right)^2 + \left(j_{iy}^k - \bar{j_y^k}\right)^2 + \left(j_{iz}^k - \bar{j_z^k}\right)^2}{N}}$$

where $SD(m) = [SD(j^1), \ldots, SD(j^k)]$. In an ideal scenario, we expect that joints that do not move should have standard deviations close to zero. However, due to tracking noise exhibited by the motion sensor and unconscious joint movements as users articulate motions, what happens is that joints that should not move actually do move.

3. Thus, we attempt to classify intentionally moving joints using k-means [8], a clustering algorithm that partitions a set of values into k disjoint clusters, such that each value within a cluster is closer to the mean for that cluster compared to the mean of any other cluster. We posit that $SD(m)$ can be partitioned into two clusters: joints that move intentionally and joints that should not move. We set the cluster size to two because we expect that joints that users are intentionally moving should have much higher variations compared to joints that move due to noise, such that these variations can be grouped into two separate clusters. Hence, we used k-means to partition $SD(m)$ with the number of clusters set to two (k = 2). The k-means algorithm will define two clusters c_1 and c_2, each having a mean x_1 and x_2, respectively, and guarantees that one of the means is greater than the other. We set the initial means of the clusters as $max(SD(m))$ and $min(SD(m))$ and k-means updates the means iteratively until all the values have been allocated to a cluster. The cluster with the higher mean (e.g., higher range of motion), c_1, we select as containing all the intentionally moving joints.

4. For whole-body gestures that require users to move all of their body parts, for example, when making a "*Jump*", we expect that most of the joint variations will be fairly close to each other. However, with our 2-cluster k-means approach, these joint variations will be forced into two clusters and the cluster c_1 with the higher mean will be selected even though we would not necessarily agree that the joints in c_1 move more than the joints in c_2. To account for such motions, we use a threshold, such that if the difference between the cluster means is below the threshold, then it means that the cluster with the lower mean (i.e., c_2) still includes joints that are actively moving. We compute the threshold as the average of the absolute difference between x_1 and x_2 for all motions being considered. If the absolute difference between x_1 and x_2 for motion m is less than the threshold, we repeat step 3 on c_2 (i.e., the cluster with the lower mean) to further partition the joints. Joints that were initially clustered into c_2 clearly do not move with as much variation in position (i.e., as actively) as the joints previously clustered into c_1. On the other hand, splitting c_2 again should be able to separate joints that have even less variation (i.e., due to tracking noise from the motion sensor and unintentional movements from the user) from more active joints. We add the resulting joints in the new cluster with the higher mean to the previously selected set of actively moving joints (c_1 from step 3).

Table 1. Example motion types in the Kinder-Gator dataset with joints selected by our filterJoint method.

Gesture	Joints Selected
Raise your hand	Right hand, Right wrist, Right elbow
Bend your knee	Right knee, Right ankle, Right foot
Point at the camera	Right hand, Right wrist, Right elbow

The motion m is now defined by the joints selected by the filterJoint method. For example, for a "*Raise your hand*" motion (right hand), our method will select the right hand, right wrist, and right elbow from the full set of 20 joints (Table 1).

3.2 Evaluating the FilterJoint Method

To evaluate that the filterJoint method does indeed select the key actively moving joints, we adapt template-based gesture recognition algorithms. These algorithms compare an articulation of a gesture to other articulations of gestures using point-by-point correspondence in order to select the articulation that most closely resembles the one being tested based on a distance metric. Since the goal of our work is to understand whole-body gesture articulation, we use template-based approaches as opposed to state-of-the-art machine learning approaches because the former can be used to make intuitive inferences about whole-body gesture articulation (e.g., whether users show variations in how they articulate instances of the same gesture type). In template-based gesture methods, changes in the articulation of the path will result in changes in the point-by-point correspondence that affect recognition accuracy in a predictable way. In contrast, state-of-the-art machine learning approaches, such as LSTMs and HMMs, use complex models and a "black-box" approach [8], which makes it less clear how changes in the articulation path affect recognition results.

We compared the recognition accuracy when using our filterJoint method compared to a baseline method (i.e., involving all joints tracked by the motion sensor). In template-based algorithms, recognition accuracy is an indicator of the similarity between articulation of instances of the same gesture type. That is, the higher the recognition accuracy, the higher the consistency between users' articulations of instances of whole-

body gestures of the same type. Joint articulation paths resulting from tracking noise of the sensor and unintentional movements from the user will result in variations in how users articulate similar gestures, which will introduce noise during the recognition process. This noise will detract from a recognizer being able to accurately distinguish between motion types, so exclusion of such joints should improve recognition accuracy. Since our filterJoint method attempts to filter out such joint articulation paths, a higher recognition over the baseline method will indicate that our method was successful in removing noisy or unimportant joints. For our evaluation, we use adults' whole-body gestures from the publicly-available Kinder-Gator dataset [1]. We focus on only adults for this evaluation because stroke gesture research has shown that adults tend to be more consistent than children in how they articulate gestures. Hence, template-based stroke gesture recognizers perform better on adults' gestures compared to children's gestures [2].

3.2.1 Adapting 3D Template Gesture Recognizers to Multiple Articulation Paths. The gesture recognizer we used is the $3 recognizer [13], which is a 3D extension of Wobbrock et al.'s $1 unistroke gesture recognizer [29]. In addition to the $3 recognizer [13], we also used specific components proposed in other gesture recognizers (e.g., protractor3D [14]) to enable the recognition of whole-body gestures. The $3 recognizer [13] uses two steps for the recognition process: a) normalization and b) recognition. We detail these steps and how we adapted them:

3.2.1.1 Normalization. Given a set of M points that define an articulation path of a gesture, $3 selects points so that the articulation path is defined by *N* equidistant points, also known as **resampling** [13]. We resampled the articulation path of a joint to 32 points, which has been shown to be adequate for recognition of stroke gestures [29]. After resampling, $3 **rotates** the articulation path so that its indicative angle (i.e., the angle between the first point and the centroid point of the articulation path [13,29]) is zero. After rotation, $3 **scales** the articulation path non-uniformly to a reference cube of size 100^3 dimension. However, non-uniform scales are not effective when the range of points is close to zero [29]. For example, in Figure 2, the range of points of the green articulation path along the x-axis is close to zero. Hence, we used uniform scaling as proposed in the $P recognizer [24], which applies the same scaling factor to all dimensions. The uniform scaling factor is equal to the maximum range of points from among the ranges of points from each dimension for that gesture instance. After scaling, $3 **translates** the articulation path so that its centroid is at the origin. The above processes ensure that similar gesture instances that differ only by speed, rotation, size, and position respectively can be matched to each other. Lastly, $3 finds the **optimal alignment** between two articulation paths, which is the alignment that gives the minimum average Euclidean distance [13,29]. We use the closed form solution in Protractor3D [14], a 3D extension of the Protractor gesture recognizer [15], to find the optimal alignment. We rotate with respect to the base orientation of the motion using the approach from the original Protractor gesture recognizer [15], so that dissimilar gesture instances that differ only in their orientation can still be distinguished. For example, the gestures "*Raise your arm to one side*" and "*Raise your hand*", differ only in terms of their orientation (horizontal 90° and vertical 90°). The normalization step improves recognition accuracy in the face of minor gesture articulation variations by users. Figure 2 shows example motion paths for one joint in two motion instances after applying all the normalization steps.

3.2.1.2 Recognition. After normalization, $3 consecutively matches the points of the articulation path of the test gesture to be recognized to each gesture in the training set. The gesture in the training set whose articulation path has the lowest Euclidean distance to that of the test gesture is selected. To extend this approach to multiple articulation paths, for each test gesture *C* and a gesture *t* in the training set *T*, we normalize each joint j^i in *C* and *t* (note: if using the filterJoint method, it is first applied to *C*, such that the joints in *C* are the actively moving joints). Thus, *C* becomes *C'* and *t* becomes *t'*. Then, we compute the optimal alignment between joint $j^i_{C'}$ in *C'* and the corresponding joint $j^i_{t'}$ in *t'* and compute the average Euclidean distances e_1 and e_2 after rotating $j^i_{C'}$ in *C'* to match $j^i_{t'}$ in *t'* (r_c) and $j^i_{t'}$ in *t'* to match $j^i_{C'}$ in *C'* (r_t) respectively:

$$e_1 = \frac{\sum_{k=1}^{N} \sqrt{\sum_{q\in\{x,y,z\}} (r_c(k)_q - t'(k)_q)^2}}{N} \quad d_i = \min(e_1, e_2, e_3, e_4)$$

$$e_2 = \frac{\sum_{k=1}^{N} \sqrt{\sum_{q\in\{x,y,z\}} (C'(k)_q - r_t(k)_q)^2}}{N}$$

where *N* is the number of points being considered. To account for directional differences resulting from making the same motions with different limbs (e.g., raising the left hand vs. raising the right hand), we rotated j^i in *C'* 180° *(flip_*j^i*)* by negating the position along the x-dimension. We computed the distances e_3 and e_4 after rotating *flip_*j^i to j^i in *t' (r_flip_*j^i*)* and rotating j^i in *t'* to *flip_*j^i *(r_t_flip)*, respectively. Then, we compute d_i as the minimum of e_1, e_2, e_3, and e_4. The minimum average Euclidean distance *d* defines how close the gesture path between *C* and *t* is, and is calculated as the sum of d_i over all joints *p* in *C*. The gesture from *T* with the lowest Euclidean distance to *C* is the recognition result, that is, *t* for which $d = \sum_{i=1}^{p} d_i$ is minimum.

3.2.2 Motion Selection. To evaluate filterJoint, we use the Kinder-Gator dataset [1]. Motions in this dataset were collected using Kinect v1, which tracks the movements of 20 joints in the body. We chose a representative (distinct) set of gestures from the dataset by removing gestures that are currently out of scope of our work and gestures with obvious conflicts in the dataset. The gestures we considered out of scope of our work are 3D stroke gestures, in which the emphasis is on the shape or symbol being drawn (e.g., "*Draw the letter A in the air*") and periodic gestures: gestures in which the same set of poses occur multiple times (e.g., "*Run in place*"). We excluded periodic gestures because instances of the same gesture type that have different numbers of repetitions might not be matched properly to each other. After exclusion, we grouped gestures that are similar in terms of how they are performed. First, we grouped gestures that have mirrors, since the gestures being performed are the same,

Table 2. Our distinct set of 14 gestures from the Kinder-Gator dataset [1].

Touch your toes (TYT)	Do a forward lunge (DFL)
Point at the camera (PAC)	Lift your leg to one side (LYL)
Raise your hand (RYH)	Jump (J)
Raise your arm to one side (RAS)	Kick a ball as hard as you can (KBH)
Bend your knee (BYK)	Throw a ball as far as you can (TBF)
Put your hands on your hip and lean to one side (PHL)	Swipe across an imaginary screen in front of you (SIF)
Punch (P)	Bow (B)

just with the opposite limb (e.g., "*Raise your other hand*" is a mirror of "*Raise your hand*"). We also grouped gestures that are the same motion differing only in strength (e.g., "*Throw a ball*" vs. "*Throw a ball as far as you can*"; "*Kick a ball*" vs. "*Kick a ball as hard as you can*") as these motions will be articulated the same way. Lastly, we grouped the gestures ("*Point at the camera*", "*Motion someone to stop*", and "*Push an imaginary button in front of you*"). These gestures are difficult to distinguish using the Kinect v1 alone, because their differences are based on the position of the finger, which this sensor cannot track. We selected one gesture from each such group for the representative set. Our final representative set comprises 14 gestures (Table 2).

3.2.3 Results. For testing, we use a leave-one-out cross validation (LOOCV) method [31]. In LOOCV, gestures from one participant is used for testing while all other participants' gestures are used for training. The training/testing process is repeated until the recognizer has been tested on gestures from all participants. Because the Kinder-Gator dataset [1] includes 10 adults, we select motions from 9 participants for training and the remaining participant's motions is left out for testing. The filterJoint method is applied to the test motion to select its actively moving joints prior to matching the motions to all the motions selected for training. We repeat training/testing 10 times so each participant's motions are used in one trial for testing. The accuracy of each trial is the number of correctly classified gestures from among the gestures selected for testing, and the overall accuracy is the sum of the accuracies across all trials divided by the number of trials (i.e., 10 trials).

Our filterJoint method achieved a recognition accuracy of 90.7% [SD = 6.8%] while the baseline method achieved a recognition accuracy of 81.4% [SD = 6.9%] (Figure 3). A pairwise t-test showed that the filterJoint method was significantly more accurate than the baseline ($t(9) = 6.09$, $p < 0.01$). Therefore, we can conclude that the filterJoint method successfully filters out noisy joints that are not important to articulation.

4 Case Studies

We have presented the filterJoint method, which filters out noisy or unimportant joint motion paths in whole-body gesture articulations, because including such joints during the analysis of whole-body gesture articulation could result in incorrect inferences about how users make motions. The articulation paths of joints selected by this method (i.e., key joints that are necessary to articulating the whole-body gestures), can be

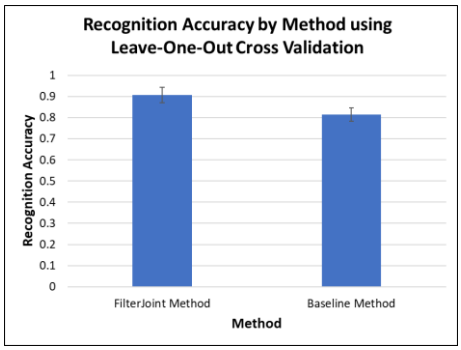

Figure 3. Performance of the gesture recognition algorithm. Error bars indicate 95% confidence interval.

analyzed to investigate nuances in how users articulate whole-body gestures. We present two case studies from the Kinder-Gator dataset [1] that showcase the kinds of new insights about how users articulate whole-body gestures that our filterJoint method enables, and we highlight implications from these insights for selecting whole-body gesture sets. Our aim is not to present general findings about whole-body gesture articulation across all possible gesture types, but rather to demonstrate the applicability of our filterJoint method as a tool that designers can use to understand how users articulate different gesture types, and help them select better gesture sets.

4.1 Identifying Overlaps in Motions

Although we removed gestures with obvious conflicts when selecting our original gesture set, our recognition results still showed recognition errors. We analyzed the *confusion matrix*, which is a tool for exploring data in recognition algorithms, to get a better understanding of the confusions between whole-body gestures (Table 3). The specific confusions the recognizer makes among whole-body gestures may reveal insights about whole-body gesture articulation patterns.

An analysis of the confusion matrix from our LOOCV recognition experiment showed that there are whole-body gestures that are confused for each other even though they do not share any similarities that we would expect in terms of their poses. Since we have identified the actively moving joints using our filterJoint method, we can examine the articulation paths of these joints to understand how participants within the dataset articulate whole-body gestures. Our adaptation allows us to explicitly determine how the articulation of whole-body gestures contributes to confusions between otherwise dissimilar motions. For example, the worst confusions occurred between gestures "*Throw a ball as far as you can (tbf)*" (7) and "*Punch (p)*" (6) (Table 3, row TBF, column P and row P, column TBF). We believe that these confusions are likely because both gestures involve the act of swinging the arm(s) forward, so there is a high chance that users will articulate these gestures in a similar fashion. Supporting our expectations, we found that for 6 out of 10 participants, the actively moving joints selected by our filterJoint method for the *p* gesture overlapped with those selected for the *tbf* gesture for at least one other participant. This finding

Table 3. Confusion matrix for the recognition results with 9 training templates. Rows represent the frequency of times the motion was categorized as the column. Correct recognitions are along the diagonal.

	Bend your knee (BYK)	Bow (B)	Do a forward lunge (DFL)	Jump (J)	Kick a ball as hard as you can (KBH)	Lift your leg to one side (LYL)	Point at the camera (PAC)	Punch (P)	Put your hands on your hips and lean to the side (PHL)	Raise your arm to one side (RAS)	Raise your hand (RYH)	Swipe across an imaginary screen in front of you (SIF)	Throw a ball as far as you can (TBF)	Touch your toes (TYT)
BYK	9	0	1	0	0	0	0	0	0	0	0	0	0	0
B	0	9	0	0	0	0	0	0	0	0	0	0	0	1
DFL	0	0	10	0	0	0	0	0	0	0	0	0	0	0
J	0	0	0	10	0	0	0	0	0	0	0	0	0	0
KBH	0	0	1	0	8	0	0	0	0	0	0	0	1	0
LYL	0	0	0	0	0	10	0	0	0	0	0	0	0	0
PAC	0	0	0	0	0	0	10	0	0	0	0	0	0	0
P	0	0	1	0	1	0	0	6	0	0	0	0	2	0
PHL	0	0	0	0	0	1	0	0	9	0	0	0	0	0
RAS	0	0	0	0	0	0	0	0	0	10	0	0	0	0
RYH	0	0	0	0	0	0	0	0	0	0	10	0	0	0
SIF	0	0	0	1	0	0	0	0	0	0	0	9	0	0
TBF	0	0	0	0	0	0	0	3	0	0	0	0	7	0
TYT	0	0	0	0	0	0	0	0	0	0	0	0	0	10

suggests an overlap between participants' articulation of the *p* and *tbf* gestures. As another example, the gesture "*Kick a ball as hard as you can (kbh)*" (8) was confused as "*Do a forward lunge (dfl)*" (1) (Table 3, row KBH, column DFL) and "*Throw a ball as far as you can*" (1) (Table 3, row KBH, column TBF). From our understanding of how the gestures *kbh* and *dfl* are likely to be articulated, we expect that *kbh* should only involve movement of joints in the lower limb (e.g., knee and foot) while the *dfl* motion should involve movement of both upper and lower limbs. Contrary to our expectation, based on the joints selected by the filterJoint method, we found that 8 out of 10 participants actively moved their upper limbs during the articulation of the *kbh* motion. Although this behavior is not expected, prior work in biomechanics has shown that upper limb movements can help to maintain balance when only one foot is on the ground [12,28], which occurs when articulating the *kbh* motion.

The gesture "*Put your hands on your hips and lean to the side (phl)*" (9) was also confused with one other gesture: "*Lift your leg to one side (lyl)*" (1) even though the *phl* motion does not in and of itself share a similarity with *lyl* (Table 3, row PHL, column LYL). From our understanding, we expect that the *phl* gesture should not actively involve lower limb movements (e.g., foot). Our analysis of the actively moving joints corroborates our expectations; however, we found that the participant whose *phl* gesture was misclassified did actively move their foot during articulation. This foot movement could have resulted in the participant's articulation of the *phl* motion being confused for another participant's articulation of the *lyl* motion because there would have been an overlap between the joints selected by our filterJoint method. We also expected that the *lyl* motion should not involve joints in the upper limb (e.g., hand), but we found that four participants actively moved their hand or shoulder during the articulation of this gesture. Prior work has noted that the farther a user leans to one side, the higher the chance that the user will lose balance due to a shift in their center of mass and gravity [20]. To compensate for the shift, people may raise the leg opposite of the way they are leaning [18]. Hence, the participant may have raised their leg or moved their upper limbs to maintain balance during the articulation of the *phl* and *lyl* gestures respectively, thus, affecting their motion articulations.

The above findings suggest that when motions require balance, users may intentionally move additional joints to maintain balance that we did not initially expect would be critical to the movement. It is a positive outcome that our filterJoint approach is robust enough to ensure that these joints are not filtered out. These findings also suggest that there are between-user inconsistencies in whole-body gesture articulation, so a designer who might have done what we did to select gestures would still see conflicts. Our approach can be used to detect these conflicts, which can make it easier for designers to select a better set of gestures. For example, based on our findings, we might want to exclude the gestures "*Punch (p)*" and "*Kick a ball as hard as you can (kbh)*" from our gesture set to avoid the conflicts arising from users' articulation of these gestures with other gestures like "*Throw a ball as far as you can (tbf)*" and "*Do a forward lunge (dfl)*", respectively. We recommend excluding *p* and *kbh* since these gestures had more variations in how users articulated them compared to the gestures they conflicted with (Figure 3).

We recommend that for applications that require a unique set of whole-body gestures (e.g., exergames), designers should consider applying our filterJoint method after selecting a distinct set of gestures to further exclude gestures that overlap due to nuances in users' whole-body gesture articulation.

4.2 Understanding Motion Articulation in Children and Adults

The Kinder-Gator dataset [1] includes both children's and adults' motions, so we also investigated how the articulation paths selected by our filterJoint method can show differences in children's and adults' whole-body gesture articulations. We investigated children's and adults' *degree of agreement* for each of the gesture types in our representative set (Figure 4). We

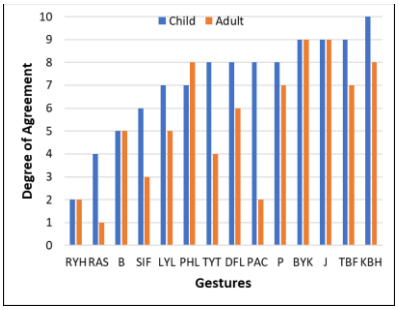

Figure 4: Degree of agreement of children and adults for gestures in our gesture set (lower = higher agreement).

defined the degree of agreement as equal to the total number of unique joint combinations selected within a gesture type. For example, for the gesture "*Raise your arm to one side*", adults in the Kinder-Gator dataset have only one unique combination ("hand right + wrist right" (10 adults)). Children have four unique combinations for the same motion ("hand right + wrist right" (2), "hand right + wrist right + elbow right" (6), "hand right + hand left + wrist right + wrist left + elbow right" (1), and "hand right + hand left + wrist right + wrist left + elbow right + shoulder right + foot left + knee right + head" (1)). We conducted a paired samples t-test to compare degree of agreement in children and adults across gesture types and found a significant difference ($t(13) = 3.38$; $p < 0.01$). Adults had a higher level of agreement (M = 5.43 ± 2.71) than children (M = 7.14 ± 2.21).

These findings suggest that children are less consistent than adults in how they articulate whole-body gestures, which has important implications for the selection of whole-body gesture sets. For example, gesture sets that are suitable for adults are not necessarily suitable for children. Using the Kinder-Gator dataset, we found that children had the highest degree of agreement for gestures that use only arm movements (e.g., "*Raise your hand*", 2 unique combinations and "*Raise your arm to one side*", (4)). On the other hand, children had low degree of agreement for gestures involving the whole-body (e.g., "*Jump*", 9 unique combinations) and gestures involving lower-limb movements (e.g., "*Lift your leg to one side*", 7 unique combinations) (Figure 4). This finding could be due to the increased coordination among many joints required to perform more complex movements. Since children are still developing their motor abilities [7,9], they are less likely to have experience coordinating multiple joints to perform movements compared to adults. In addition, the higher degree of agreement in arm motions, which only involves movement of the upper limbs, can be attributed to balance and postural stability. The lower-limb motions in the Kinder-Gator dataset (e.g., "*Lift your leg to one side*") usually require that the user maintains balance when performing the movement on one leg. Children are more likely than adults to move other joints to maintain balance since they are still developing their postural stability [11] (e.g., the arms play a functionally relevant role in balance among children [11]).

Hence, we recommend that, if possible, designers of whole-body gesture applications for children should prefer gestures that require only upper limb movements, especially arm movements, since children will articulate those motions more consistently, and thus recognition will be more accurate.

5 Conclusion and Future Work

An understanding of the nuances of how users articulate whole-body gestures can help in selecting whole-body gesture sets that run less risk of conflicts due to confusion among seemingly dissimilar gestures. We designed the filterJoint method, which selects the key body parts that users actively move during whole-body motions. The paths along which these body parts move in space over time can then be analyzed to make inferences about how users articulate whole-body gestures. Evaluation of our method using an adaptation of a 3D template-based stroke gesture recognizer showed that our method outperforms the baseline method, meaning we are successfully filtering out noisy or unimportant joints that are not necessary to the articulation of a whole-body gesture. The filterJoint method we developed focuses on interpretability to help designers select whole-body gesture sets. Other feature selection techniques, such as Principal Component Analysis (PCA) [30], could be compared to our method in future work to determine how well they select actively moving joints, and compare their performance and interpretability to our filterJoint method.

A limitation of our approach is that it is currently not robust to datasets that include gestures with obvious conflicts (e.g., same gestures performed with opposite limbs or different degrees of strength), periodic gestures, or gestures which have not been pre-segmented. Future work can consider more challenging gesture sets and segmentation approaches to address these limitations. We showed via two case studies that our filterJoint method enables an understanding of nuances in how users articulate certain gestures. Although our approach to select distinct gestures is not fully automated, we note that the selection of application gesture sets is not an automated process because it relies on the designer's understanding of gestures that are unique (e.g., [3,25]). Future work could expand on our work by examining automated conflict detection to help in the selection of whole-body gesture sets. We hope that designers of whole-body gesture applications can use our method to prune their gesture set to prevent conflicts among seemingly dissimilar whole-body gestures, and ultimately improve the recognition of gestures that their application supports.

ACKNOWLEDGMENTS

This work is partially supported by National Science Foundation Grant Award #IIS-1552598. Opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect these agencies' views. Special thanks to Isaac Wang and Nikita Soni for their input during the writing of initial drafts of this paper.

REFERENCES

[1] Aishat Aloba, Gianne Flores, Julia Woodward, Alex Shaw, Amanda Castonguay, Isabella Cuba, Yuzhu Dong, Eakta Jain, and Lisa Anthony.

2018. Kinder-Gator: The UF kinect database of child and adult motion. In *Eurographics (Short Papers)*, 13–16. https://doi.org/10.2312/egs.20181033

[2] Lisa Anthony, Quincy Brown, Jaye Nias, Berthel Tate, and Shreya Mohan. 2012. Interaction and recognition challenges in interpreting children's touch and gesture input on mobile devices. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces (ITS '12)*, 225. https://doi.org/10.1145/2396636.2396671

[3] Lisa Anthony, Radu-Daniel Vatavu, and Jacob O. Wobbrock. 2013. Understanding the consistency of users' pen and finger stroke gesture articulation. In *Proceedings of the Graphics Interface Conference (GI '13)*, 87–94. https://doi.org/10.5555/2532129.2532145

[4] Lisa Anthony and Jacob O. Wobbrock. 2012. $ N-Protractor: A fast and accurate multistroke recognizer. In *Proceedings of Graphics Interface (GI '12)*, 117–120. https://doi.org/10.5555/2305276.2305296

[5] Victoria Bloom, Vasileios Argyriou, and Dimitrios Makris. 2015. G3di: A gaming interaction dataset with a real time detection and evaluation framework. In *European Conference on Computer Vision (ECCV '15)*, 698–712. https://doi.org/10.1007/978-3-319-16178-5_49

[6] Aaron F. Bobick and James W. Davis. 2001. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*: 257–267. https://doi.org/http://dx.doi.org/10.1109/34.910878

[7] Frances Cleland-Donnelly, Suzanne S Mueller, and David L Gallahue. 2017. *Developmental Physical Education for All Children: Theory Into Practice*.

[8] John A. Hartigan and Manchek A. Wong. 2006. Algorithm AS 136: A K-Means clustering algorithm. *Journal of the Royal Statistical Society Series C (Applied Statistics)* 29, 1: 100–108. https://doi.org/10.2307/2346830

[9] Kathleen Haywood and Nancy Getchell. 2019. *Life span motor development*. Human Kinetics.

[10] Hamilton A. Hernandez, Zi Ye, T.C. Nicholas Graham, Darcy Fehlings, and Lauren Switzer. 2013. Designing action-based exergames for children with cerebral palsy. In *Proceedings of the ACM SIGCHI Annual Conference on Human Factors in Computing Systems (CHI '13)*, 1261–1270. https://doi.org/10.1145/2470654.2466164

[11] Mathew W. Hill, Maximilian M. Wdowski, Adam Pennell, David F. Stodden, and Michael J. Duncan. 2019. Dynamic postural control in children: Do the arms lend the legs a helping hand? *Frontiers in Physiology*. https://doi.org/10.3389/fphys.2018.01932

[12] Justin J. Kavanagh, R. S. Barrett, and Steven Morrison. 2004. Upper body accelerations during walking in healthy young and elderly men. *Gait and Posture* 20, 3: 291–298. https://doi.org/10.1016/j.gaitpost.2003.10.004

[13] Sven Kratz and Michael Rohs. 2010. A $3 gesture recognizer: simple gesture recognition for devices equipped with 3D acceleration sensors. In *Proceeding of the international conference on Intelligent user interfaces (IUI '10)*, 341–344. https://doi.org/10.1145/1719970.1720026

[14] Sven Kratz and Michael Rohs. 2011. Protractor3d: a closed-form solution to rotation-invariant 3d gestures. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI '11)*, 371–374. https://doi.org/10.1145/1943403.1943468

[15] Yang Li. 2010. Protractor: a fast and accurate gesture recognizer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*, 2169–2172. https://doi.org/10.1145/1753326.1753654

[16] Microsoft. Kinect for Windows. Retrieved January 31, 2019 from https://developer.microsoft.com/en-us/windows/kinect

[17] Microsoft. 2019. Kinect Gesture Data Set. Retrieved October 8, 2019 from https://www.microsoft.com/en-us/download/details.aspx?id=52283

[18] Lewis M. Nashner. 1980. Balance adjustments of humans perturbed while walking. *Journal of neurophysiology* 44, 4: 650–664. https://doi.org/10.1152/jn.1980.44.4.650

[19] Chuong V. Nguyen, Shahram Izadi, and David Lovell. 2012. Modeling kinect sensor noise for improved 3D reconstruction and tracking. In *Proceedings of the international conference on 3D imaging, modeling, visualization, & transmission (3DIMPVT '12)*, 524–530. https://doi.org/10.1109/3DIMPVT.2012.84

[20] Yi Chung Pai and James Patton. 1997. Center of mass velocity-position predictions for balance control. *Journal of Biomechanics* 30, 4: 347–354. https://doi.org/10.1016/S0021-9290(96)00165-0

[21] Otto Parra, Sergio España, and Oscar Pastor. 2016. GestUI: A model-driven method and tool for including gesture-based interaction in user interfaces. *Complex Systems Informatics and Modeling Quarterly* 6: 73–92. https://doi.org/10.7250/csimq.2016-6.05

[22] Radu-Daniel Vatavu. 2017. Beyond Features for Recognition: Human-Readable Measures to Understand Users' Whole-Body Gesture Performance. *International Journal of Human Computer Interaction* 33, 9: 713–730. https://doi.org/http://dx.doi.org/10.1080/10447318.2017.1278897

[23] Radu-Daniel Vatavu, Lisa Anthony, and Jacob O. Wobbrock. 2013. Relative accuracy measures for stroke gestures. In *International Conference on Multimodal Interaction (ICMI '13)*, 279–286. https://doi.org/10.1145/2522848.2522875

[24] Radu-Daniel Vatavu, Lisa Anthony, and Jacob O Wobbrock. 2012. Gestures as point clouds: A $P recognizer for user interface prototypes. In *ACM International Conference on Multimedia Interaction (ICMI '12)*, 273–280. https://doi.org/10.1145/2388676.2388732

[25] Radu Daniel Vatavu, Lisa Anthony, and Jacob O. Wobbrock. 2014. Gesture heatmaps: Understanding gesture performance with colorful visualizations. In *Proceedings of the International Conference on Multimodal Interaction (ICMI '14)*, 172–179. https://doi.org/10.1145/2663204.2663256

[26] Stefan Waldherr, Roseli Romero, and Sebastian Thrun. 2000. Gesture based interface for human-robot interaction. *Autonomous Robots* 9, 2: 151–173. https://doi.org/10.1023/A:1008918401478

[27] Daniel Weinland, Remi Ronfard, and Edmond Boyer. 2006. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding* 104, 2–3: 249–257. https://doi.org/10.1016/j.cviu.2006.07.013

[28] David A. Winter. 1995. Human balance and posture control during standing and walking. *Gait and Posture* 3, 4: 193–214. https://doi.org/10.1016/0966-6362(96)82849-9

[29] Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. 2007. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *ACM symposium on User interface software and technology (UIST '07)*, 159–168. https://doi.org/10.1145/1294211.1294238

[30] Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems* 2, 1–3: 37–52. https://doi.org/10.1016/0169-7439(87)80084-9

[31] Tzu Tsung Wong. 2015. Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition* 48, 9: 2839–2846. https://doi.org/10.1016/j.patcog.2015.03.009