Moving Kinect-Based Gait Analysis with Increased Range

Madhura P. Pathegama*, Dileepa M. Marasinghe†, Kanishka Wijayasekara‡, Ishan Karunanayake§, Chamira U. S. Edussooriya ¶, Pujitha Silva ‖, and Ranga Rodrigo**
Department of Electronic and Telecommunication Engineering, University of Moratuwa, Moratuwa, Sri Lanka
Email: *madhurapankaja@gmail.com, †dileepamadhubhashana@gmail.com, ‡kanishkawijayasekara@gmail.com, §ishankarunanayake@gmail.com, ¶chamira@ent.mrt.ac.lk, ‖psilva@uom.lk, **ranga@uom.lk

***Abstract*—There are several systems that use one or several Kinect sensors for human gait analysis, particularly for diagnosis of patients. However, due to the limited depth sensing range of the Kinect—a sensor manufactured for video gaming—the depth measurement accuracy reduces with distance from the Kinect. In addition, self-occlusion of the subject limits the accuracy and utility of such systems. We overcome these limitations by using a two-Kinect gait analysis system and mechanically moving the Kinects in synchronization with the test subject and each other. This increases the practical measurement range of the Kinect-based system whilst maintaining the measurement accuracy. Results of the comparison of knee flexion, step length, and stride length with a software based method show that our moving Kinect system can accurately analyse these gait parameters.**

I. Introduction

Gait analysis, the study of human motion, is useful in diagnosing and rehabilitation patients with neurological disorders, and analyzing patterns of sportsmen. Commercially available gait analysis systems are mostly vision based. They need to be generally setup within controlled environments. In addition, they are usually expensive. Most of them employ tracking markers placed carefully on the subject using multiple cameras. The use of markerless motion capture is also possible. Using the Microsoft Kinect, a popular RGBD sensor, for gait analysis usually falls in to the markerless category [1], although there are approaches that use markers [2]. The Kinect Software Development Kit (SDK) [3] provides the option of human skeleton tracking, which gives the 3-D coordinates of the skeleton joints of a person in front of the sensor by processing the depth data captured in each frame. This creates a reasonably accurate easily-setup human gait analysis system. These systems can analyse spatio-temporal gait parameters [4] such as step length, and stride length with kinematic gait parameters, such as knee flexion, and hip flexion for diagnosis applications.

Unfortunately, Kinect-based gait analysis systems have the deficiency imposed by the limited range of depth the Kinect is able to sense [5], typically between 0.5 m and 4 m. This makes the Kinect not so useful in gait analysis, as typical gait analysis of an adult requires a longer walkway. This is to extract natural gait by removing the effect of acceleration and deceleration which occur at the beginning and the end of the trial. In addition, the error of depth measurements increases as the distance from the sensor to the object increases [6] which affects the skeleton tracking accuracy. Moreover, the skeleton tracking in Kinect SDK often suffers from occlusions leading to inferred data for the positions of the occluded parts of the body. Sensing range enhancement and removing occlusions are important to make the Kinect useful for gait analysis.

Kinect has, nevertheless, been used for gait analysis due to its non-invasive nature and inexpensiveness. Gabel *et al*. [1] proposed to use Kinect for full body gait analysis as a markerless, economical and non-intrusive method. This was after the release of Kinect SDK. Obdrzalek *et al*. [7] have concluded that the Kinect pose estimation works with comparable accuracy to commercial level motion capture systems for general postures like standing and sitting. They indicate occlusion as the factor that adversely affect the accuracy of the pose estimation. Fernndez-Baena *et al*. [2] conclude that the Kinects pose estimation accuracy is sufficient for rehabilitation purposes although it is less accurate than a commercial-grade motion capture system. A detailed technical review [5] also concludes that with the tradeoff between accuracy and cost, the Kinect is a suitable technology for rehabilitation purposes. The aforementioned literature use the 1st generation Kinect namely the Kinect for Xbox 360 and its SDK. They all suffer from the range and occlusion deficiencies in gait analysis as mentioned above.

The second-generation Kinect sensor—Kinect for Xbox One—(released in 2013, SDK released in 2014) uses a different technology compared to the first-generation Kinect, namely, the time of flight which improves the depth accuracy. The SDK 2.0 also had improvements in the number of joints tracked and the accuracy. Kinect for Xbox One and the SDK 2.0 provides better results in pose estimation or skeleton tracking [8] with an increased number of joints tracked. However, the accuracy drops in the newer version around lower limb areas due to the change in technology of depth capturing.

With these improvements, gait analysis systems based on second-generation Kinect came into being. Latorre *et al*. [9] have evaluated the spatio-temporal gait parameters such as stride time, step length, velocity, and kinematic parameters such as hip and knee flexions. Again, the range of the Kinect sensor has limited the study to 0.5–4 m. The results mostly reflect the acceleration and deceleration phases than the required gait [9]. However, it has been concluded that Kinect v2 has the comparable accuracy for assessing gait [8]. However, the use of the second-generation Kinect—although improves the accuracy—does not solve the the range and occlusion problems.

In this paper, we present a system of mechanically moving two second-generation Kinects in synchronization with the test subject and each other to increase the practical measurement range and prevent occlusion errors of Kinect based gait analysis systems. In our system the two Kinects travel on two parallel camera sliders with adaptive speed. This increases the range of the gait analysis system. The central application hosted on one computer receives Kinect data from both sensors through a LAN for data fusion. This results in more accurate measurements for joint coordinates than using just one Kinect. The special alignment of the Kinects at an angle to the railings and facing the subject reduces occlusion. The Kinect driving mechanism maintains an approximately constant distance from the Kinects to the subject in the range of 2 m to 3.5 m. This solves problem of Kinect's accuracy change over distance affecting our measurements. Our moving Kinect system thus improves the range and accuracy of Kinect-based gait analysis.

The use of multiple Kinects for gait analysis is not new [10] [11] [12]. Use of data fusion by the employment of Kalman filtering of data from multiple Kinects has given better results for increased accuracy in skeleton tracking in [10]. Geerse *et al*. [11] use multiple Kinects to increase the range and provide a 10-meter walkway for analysis of gait. This study employs four second-generation Kinects. Their system must have used four computers to connect the Kinect sensors since Microsoft recommends only a single second-generation Kinect v2 to be connected to a single computer. These multiple-Kinect systems suffer from the reduction of accuracy of depth with the distance. As our two-Kinect system moves with the subject, this problem does not affect ours. Müller *et al*. [12] use six Kinects with two-sided arrangement have reported to generate results that fairly agree with a Vicon system though it has employed six computers. However in our method, we need only two computers.

There are some systems with moving Kinects as well, mostly mounted on mobile robots. Machida *et al*. [13] reports a human motion tracking system for monitoring elderly people with a Kinect mounted on a mobile robot. They use the skeleton tracking available in the Kinect to control the robots velocity and attitude. Some low-cost mobile gait analysis platforms track the subject using a marker attached to the trunk and data from the Kinect [14] and record the robot local odometry for off-line gait parameters calculation. These systems do not solve the occlusion problem and have the additional burden of self localization.

Some have reported Kinect-based gait analysis systems by mounting Kinects on a treadmill and asking the subject to run on the treadmill [15] [16]. This solves the range-limitation problem of the Kinect. However, it is not clear whether this approach does not alter the regular walking pattern of the subject. In addition, the treadmill determines the speed at which the subject must walk. Therefore, the usability of these systems for diagnosis is questionable for some types of diseases.

In summary, single stationary Kinect-based, multiple stationary Kinect-based, and moving robot-based Kinect platforms for gait analysis still fail to solve the range, depth accuracy variation, and occlusion problems simultaneously. Our moving two-Kinect system solves these problems. In particular, our contributions are:

- Increasing the practical measurement range of the gait analysis system by mechanically moving the two Kinects in synchronization with test subject and each other.
- Using angularly oriented Kinects to in the moving Kinect system to handle occlusion.
- Calculation of cadence, step length, and stride length based on the autocorrelation function.

Section II of the paper describes the full architecture and algorithms used for moving the Kinects on railings. We also present a detailed description of the sub-systems in the same section. Section III presents the experimental results with a discussion and section IV concludes our work

II. Method

Our gait analysis system uses two Microsoft Kinect sensors that move along two linear sliders together and follow the subject. The body tracking data (25 joint coordinates) from one Kinect goes to the primary computer and the data from the other goes to the secondary computer. We use the network time protocol (NTP) to synchronize the two computers to a common timeline and time stamp the data received from each Kinect. This enables us to fuse the joint coordinates as seen from the two Kinects using Kalman filters.

A. Moving Kinect System

We used two motorized linear sliders fixed to height adjustable stands to move the Kinects. The key property of the moving Kinect system is its ability to move in synchronization with the subject maintaining an approximately constant distance between the Kinect and the subject. For controlling the Kinects in this manner, we use the distance to the spine base joint of the fused body tracking data. Based on this, the system determines the speed of the Kinects and communicates to the microcontroller on the moving Kinect system. This, in turn, controls the pulse width that drive the motors. Algorithm 1 describes the moving Kinect system.

Kinect body tracking reduces accuracy in tracking when the subject is turned away from the sensor as described by Wang *et al.* [8].Therefore to cover a maximum area of the subject without compromising accuracy, the Kinects are placed on the railings such that the z-axis of the Kinect lies at an angle between 30–40 degrees to the railing facing the subject as depicted in Fig.1.

B. Data Fusion

We use the subject's joint coordinates extracted from each of the Kinects to generate a better estimate using 25 Kalman

Algorithm 1 Kinects' Velocity Control

Require: $spineBaseDistance, optimumDistance$
Require: $velocityLimit, stepVelocityLimit, beta, \epsilon$
$residual = spineBaseDistance - optimumDistance$
if abs($residual$) $\geq \epsilon$ **then**
  $velocityChange \leftarrow \max(\min(beta \times residual, stepVelocityLimit), -stepVelocityLimit)$
  $kinectVelocity \leftarrow \max(\min(kinectVelocity + velocityChange, velocityLimit), -velocityLimit)$
else
  $kinectVelocity \leftarrow 0$
end if

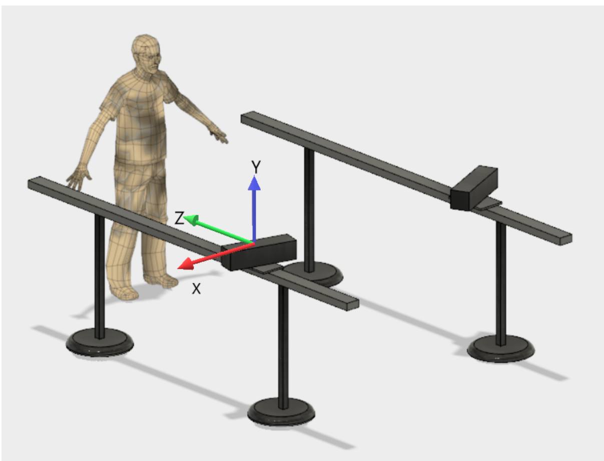


Fig. 1. System setup

filters, corresponding to the number of joints. We define a reference frame that moves with two Kinects and represent the coordinates of the joints with respect to it. As we compute the joint angles using observed joint coordinates, the angles become invariant to the reference frame. Therefore, we directly use the fused body coordinates to measure joint angles.

The first step of data fusion is to pair the corresponding data frames sent from two Kinects, as two frames from the two Kinects may not correspond to the same instance of time. Once the data frames are extracted from Kinects, we put a time stamp on each data frames. These data frames are then sent to two queues in the primary computer. We use algorithm 2 to pair corresponding data frames.

In algorithm 2, $Q1$ and $Q2$ refer to the two queues which store Kinect data frames. $K1_{prev}$ and $K2_{prev}$ variables are used to indicate the data frames extracted in the previous iteration of the algorithm. At first, $K1_{prev}$ and $K2_{prev}$ are initialized as the first frames of corresponding Kinect feeds. Then they are updated accordingly. The fuse$(.,.)$ procedure in Algorithm 2 refers to the fusion done through these 25 filters. One Kalman filter for the whole body more accurately captures the correlated movements among the joints, but we used multiple Kalman filtering approach because it reduces

Algorithm 2 Pairing and Data Fusion Algorithm

Require: $Q1, Q2, K1_{prev}, K2_{prev}$
if $Q1$ is not empty and $Q2$ is not empty **then**
  $K1 \leftarrow Q1$.dequeue()
  $K2 \leftarrow Q2$.dequeue()
  if $K1.TimeStamp \geq K2.TimeStamp$ **then**
    if abs($K1.TimeStamp - K2.TimeStamp$) $\leq$ abs($K2.TimeStamp - K1_{prev}.TimeStamp$) **then**
      fuse($K1, K2$)
    else
      fuse($K1_{prev}, K2$)
    end if
  else
    if abs($K2.TimeStamp - K1.TimeStamp$) $\leq$ abs($K1.TimeStamp - K2_{prev}.TimeStamp$) **then**
      fuse($K1, K2$)
    else
      fuse($K1, K2_{prev}$)
    end if
  end if
  $K1_{prev} \leftarrow K1$
  $K2_{prev} \leftarrow K2$
else if $Q1$ is not empty and $Q2$ is empty **then**
  $K1 \leftarrow Q1.dequeue()$
  fuse($K1, K2_{prev}$)
  $K1_{prev} \leftarrow K1$
else if $Q1$ is empty and $Q2$ is not empty **then**
  $K2 \leftarrow Q2$.dequeue()
  fuse($K1_{prev}, K2$)
  $K2_{prev} \leftarrow K2$
else
  fuse($K1_{prev}, K2_{prev}$)
end if

memory complexity while improving the numerical stability of the outputs. For each Kalman filter we defined the state of the joint to be $[x, \dot{x}, y, \dot{y}, z, \dot{z}]^T$ where $[x, y, z]^T$ indicates the position coordinates of the joint along each axis with respect to a common reference frame that moves with Kinects. Our measurement vector is $[x_1, y_1, z_1, x_2, y_2, z_2]^T$ where x_i, y_i, z_i ; $i \in \{1, 2\}$ are the joint coordinates measured by Kinect K_i relative to the common moving frame.

C. Gait Parameter Calculation

In our work, we estimate two types of gait parameters; kinematic parameters and spatio-temporal parameters. We analyze knee, ankle, and hip flexions, and distance between left and right ankle joints for kinematic parameters. Flexions are the angles measured in the sagittal plane of the body [4]. These parameters can be interpreted as functions of time. We also have added instantaneous velocity to this category since it can also be represented as a function of time. In contrast to kinematic gait parameters, spatio-temporal gait parameters convey global information about the gait pattern as a whole. We considered the following spatio-temporal gait parameters: cadence, average step length, average stride length, and average velocity. Note that we have considered instantaneous velocity as a kinematic parameter and average velocity as a spatio-temporal gait parameter.

As mentioned earlier, Kinect SDK provides coordinates for 25 joints. We represent these joint coordinates as position vectors in 3-D space. Using this representation, we estimate the distance between the two ankles. Finding a joint angle is equivalent to finding an angle between two vectors which are determined by known joint coordinates. However, rather than joint angles, we are interested in flexions. Therefore, we first projected all the joint coordinates to sagittal plane and then calculate the respective joint angles.

Instantaneous velocity(V_{SE}) consists of two parts: velocity of the subject with respect to Kinect (V_{SK}) and the velocity of Kinect with respect to world coordinate system (V_{KE}), i.e., $V_{SE} = V_{SK} + V_{KE}$. We define V_{SK} to be the velocity of spine-base along the z-axis with respect to the moving Kinects and it is directly estimated by the Kalman filter corresponding to the spine base. We derive linear velocity of the moving Kinects from the motor-control inputs. However, we update this velocity only once a second. Since this sampling rate is much lower than the Kinect frame rate, we upsample and smoothen V_{KE}

Cadence is the number of steps per unit time. We calculate this using the autocorrelation function of ankle to ankle distance. Since the time series is approximately periodic, we get local maxima at the multiples of the period. We used x-coordinate of the first maximum in the positive values as our estimation for the period. Reciprocating this value, we obtain an estimation for cadence.

Statistical mean of instantaneous velocity does not provide a good estimation for the natural speed of a test subject due to the acceleration and deceleration at the beginning and the end of a trial. Our intention was to capture the steady state velocity at the middle part of the trial. Therefore we used the 80th percentile of the magnitude of the velocity as the estimate for the average velocity. As mentioned above, only the middle part of the trial depicts the natural gait pattern. We estimate the step length as average velocity divided by cadence. Stride length is defined to be as twice as the step length.

III. RESULTS

We tested our moving two-Kinect system for gait analysis by getting a subject to walk in front of the system, and analysing the gait parameters using our system. We recorded these trials with a camera having the optical axis perpendicular to the direction of motion of the Kinects (and the subject). We analyzed left knee flexion, average step length, and average stride length. We compared the results of our system and results obtained by analyzing the video records using the Kinovea software [17] which is a considerably accurate software employed for clinical assessments [18]. It is an open source software which can be used to track given markers and calculate the associated angle throughout a video. We compared our results with [15] and [16]

Our main intention is to justify that the walking range extension due to the introduction of the moving Kinect system produces reasonable results. Surpassing the accuracy in existing system is not our goal, although our system does in many aspects.

Table I shows a comparison among different gait analysis systems based on Kinects. It verifies that our system posses an increased walking range while eliminating occlusion and preserving natural gait patterns. Under the kinematic parameters we estimated knee flexion, hip flexion, ankle flexion, and ankle to ankle distance. We also measured the velocity variation with time. Fig. 2 shows that these parameters are in general consistent with typical graphs, except of the ankle flexion. We are not able to accurately measure ankle flexion as the Microsoft Kinect SDK does not accurately identify the ankle [8].

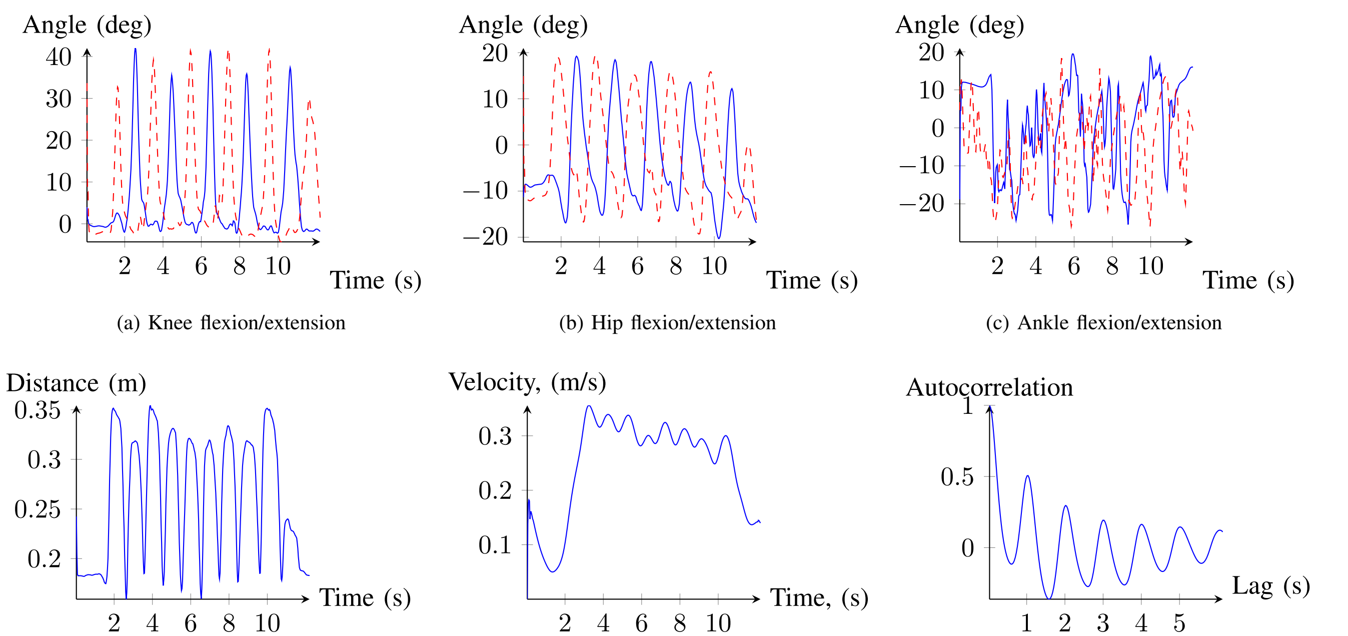
As mentioned before, we used Kinovea software to verify left knee flexion. We obtained videos for each trial and measured the left knee flexion using Kinovea software. Then we compared the values given by our system with those given by Kinovea. However, before comparing the graphs from two systems, we scaled the time axis of Kinovea output to equalize the two sampling rates. Fig. 3 shows the knee flexion obtained through Kinovea and our system for the same trial. For the sample shown in Fig. 3a, the Pearsons correlation coefficient is 0.971 (pooled correlation coefficient for whole data set is 0.947) which strongly suggests they are linearly correlated. Fig. 3b shows that Kinovea output can be roughly approximated by shifting our systems output up. The resulting graphs almost coincides indicating that the main difference between these two time series is a shift. The reason for this shift might be not putting markers precisely at joints for Kinovea to lock on to. The slight difference of these patterns from the normal curves used in clinical studies is due to the slow walking pattern of the test subject needed to accommodate the limited speed of our motors.

To numerically quantify the performance of our system we used two metrics: Pearson's correlation coefficient ρ and root mean square error (RMSE). Before evaluating these metrics, we first identified the corresponding gait cycles in knee flexion graphs generated by Kinovea and our system. After matching full gait cycles, we shifted up our system's estimation so that the mean values of two time series are equal. Considering all the trials, we estimated ρ to be 0.947. We defined error to be the standard deviation of the difference of the two original time series rather than the root mean square of them. We are interested in standard deviation because it gives an idea about the variability of the difference between two time series neglecting the possible shift which is present when measurements are taken from two systems. However for the mean-shifted case, both definitions are equivalent. Error over all trials was estimated to be the pooled standard deviation of the differences of corresponding time series.

We have compared knee flexion extracted from our system with [15] and [16]. Since both of these are a treadmill based approach, performance measures are given for different

TABLE I
COMPARISON OF KINECT BASED GAIT ANALYSIS SYSTEMS

System	Main features	Walking range	Main disadvantages
Lattorre *et al*. [9]	One Kinect	<3 m	Fixed, low range, occlusion
Geerse *et al* [11]	4 Kinects	10 m	Does not maintain constant distance with test subject, occlusion
Machida *et al* [13]	Single Kinect, mounted on a robot	Unlimited	Occlusion
Xu *et al* [15]	One Kinect to analyze treadmill walking	Limited to treadmill	Treadmill alters natural walking pattern
Our system	Pair of synchronized moving Kinects which maintain constant distance with test subjects	Approx 4.5 m	Additional railings are needed to extend the system



(a) Knee flexion/extension
(b) Hip flexion/extension
(c) Ankle flexion/extension
(d) Distance between ankles
(e) Velocity
(f) Auto-correlation function of distance between ankles

Fig. 2. Kinematic gait parameters and autocorrelation function of distance between ankles (in the top row: solid = left, dashed = right.)

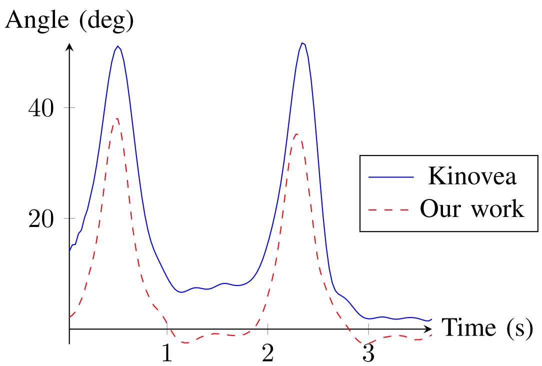
TABLE II
COMPARISON OF KNEE FLEXION (STANDARD DEVIATION IS GIVEN INSIDE THE PARENTHESES)

	Xu *et al*. [15]			Pfister *et al*. [16]			Our system
	0.85 m/s	1.07 m/s	1.3 m/s	3 mph	4.5 mph	5.5 mph	
Difference of max angle (deg)	-38.6 (12.2)	-38.1 (12.8)	-38.1 (12.2)	NA	NA	NA	-1.53 (3.60)
RMSE error (deg)	27.9 (10.0)	28.6 (10.8)	29.0 (10.3)	NA	NA	NA	4.47 (0.85)
ρ	NA	NA	NA	0.43	0.79	0.55	0.947

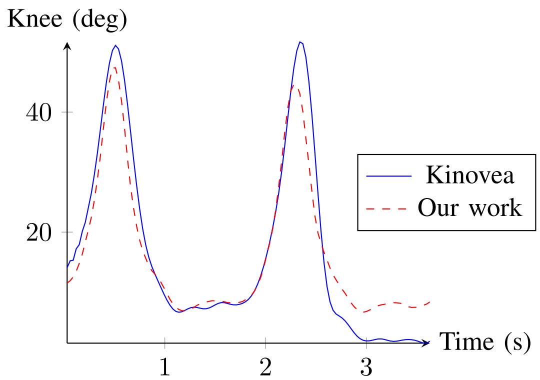
speeds. Table II summarizes these results. Table II clearly shows for knee flexion values, that our system significantly outperforms the systems referred in [15] and [16].

We have compared the estimates given by the system for average step and stride length with the ground truth. The comparison procedure is as follows: First, we put a ruler along the walking path. We measured the step and stride length of each trial avoiding the very start and end to discount acceleration and deceleration phases using video records of the trials. From each recording, we extracted 3 steps to calculate the average step length and 2 strides to calculate the average stride length. Averages of these values were considered as the ground truth for the average step length and the average stride length. For average step length, RMSE was 2.7 cm and for average stride length it was 5.5 cm.

Mean difference comparison shown in Table III suggests that our system does not perform well for average step length and stride length. This may be due to two main reasons. First one is the low update rate of the position of the Kinects. Even though Kinects send data at a rate of 30 fps, position of the Kinects are updated once per second. We have used smoothing and interpolating techniques to overcome this issue but still



(a) Knee flexion from two systems without any shift



(b) After shifting our system's output

Fig. 3. Knee flexion of a trial. Our measurements are consistent with that obtained with Kinovea when shifted.

TABLE III
MEAN DIFFERENCE COMPARISON OF STEP LENGTH AND STRIDE LENGTH

	Xu *et al.* [15]	Müller *et al.* [12]	Our system
For step length (cm)	0.1	0.4	0.5
For stride length (cm)	0.1	0.04	1.4

this may have contributed to erroneous results. The second reason is that the real system may have deviated from the assumption that says average velocity in the middle range can be approximated by the 80^{th} percentile of the velocity profile.

IV. Conclusion

In this paper, we presented a gait analysis system that comprises of two Kinect sensors that move along sliders relative to the subject under test. Following the subject overcomes the inherent limitation of the Kinect sensor: the limited accuracy of depth sensing with increased distance from the sensor. Orienting the Kinects with a little rotation away from the sliders toward the subject overcomes the limitation of the subject self-occluding some parts of his or her body. Increased accuracy is also due to fusing the 3-D joint coordinates through a set of Kalman filters. The gait parameters such as knee flexion were consistent with typical parameters. Verification with the software-based Kinovea was successful. Although our intentions was to show the walking range extension due to our moving Kinect systems, our results surpassed existing results in many cases, except for ankle flexion.

Acknowledgment

The authors would like to thank the Senate Research Committee of the University of Moratuwa for financial support through the grant SRC/LT/2016/04 and Peshala G. Jayasekara for helpful discussions.

References

[1] M. Gabel, R. Gilad-Bachrach, E. Renshaw, and A. Schuster, "Full body gait analysis with Kinect," in *Int. Conf. of the IEEE Eng. in Medicine and Biology Society*, August 2012, pp. 1964–1967.

[2] A. Fern'ndez-Baena, A. Susín, and X. Lligadas, "Biomechanical validation of upper-body and lower-body joint movements of Kinect motion capture data for rehabilitation treatments," in *Int. Conf. on Intelligent Networking and Collaborative Syst.*, September 2012, pp. 656–661.

[3] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, June 2011, pp. 1297–1304.

[4] J. Perry and J. M. Burnfield, *Gait Analysis: Normal and Pathological Function*, 1st ed. SLACK Incorporated, 1992.

[5] H. M. Hondori and M. Khademi, "A review on technical and clinical impact of Microsoft Kinect on physical therapy and rehabilitation," *J. of Medical Eng.*, vol. 2014, 2014.

[6] L. Yang, L. Zhang, H. Dong, A. Alelaiwi, and A. E. Saddik, "Evaluating and improving the depth accuracy of Kinect for Windows v2," *IEEE Sensors J.*, vol. 15, no. 8, pp. 4275–4285, August 2015.

[7] Š. Obdržálek, G. Kurillo, F. Ofli, R. Bajcsy, E. Seto, H. Jimison, and M. Pavel, "Accuracy and robustness of Kinect pose estimation in the context of coaching of elderly population," in *Int. Conf. of the IEEE Eng. in Medicine and Biology Society*, August 2012, pp. 1188–1193.

[8] Q. Wang, G. Kurillo, F. Ofli, and R. Bajcsy, "Evaluation of pose tracking accuracy in the first and second generations of Microsoft Kinect," in *Int. Conf. on Healthcare Informatics*, October 2015, pp. 380–389.

[9] J. Latorre, R. Llorens, A. Borrego, M. Alcañiz, C. Colomer, and E. Noé, "A low-cost Kinect for Windows v2-based gait analysis system," in *Int. Conf. Virtual Rehabilitation*, June 2017, pp. 1–2.

[10] S. Moon, Y. Park, D. W. Ko, and I. H. Suh, "Multiple Kinect sensor fusion for human skeleton tracking using Kalman filtering," *Int. J. of Advanced Robotic Syst.*, vol. 13, no. 2, p. 65, March 2016.

[11] D. J. Geerse, B. H. Coolen, and M. Roerdink, "Kinematic validation of a multi-Kinect v2 instrumented 10-meter walkway for quantitative gait assessments," *PLoS one*, vol. 10, no. 10, p. e0139913, October 2015.

[12] B. Müller, W. Ilg, M. A. Giese, and N. Ludolph, "Validation of enhanced Kinect sensor based motion capturing for gait assessment," *PloS one*, vol. 12, no. 4, p. e0175813, April 2017.

[13] E. Machida, M. Cao, T. Murao, and H. Hashimoto, "Human motion tracking of mobile robot with Kinect 3D sensor," in *Proc. of SICE Annual Conf. (SICE)*, August 2012, pp. 2207–2211.

[14] V. Bonnet, C. A. Coste, L. Lapierre, J. Cadic, P. Fraisse, R. Zapata, G. Venture, and C. Geny, "Towards an affordable mobile analysis platform for pathological walking assessment," *Robotics and Autonomous Systems*, vol. 66, pp. 116–128, April 2015.

[15] X. Xu, R. W. McGorry, L.-S. Chou, J. hua Lin, and C. chi Chang, "Accuracy of the Microsoft Kinect for measuring gait parameters during treadmill walking," *Gait & Posture*, vol. 42, no. 2, pp. 145–151, July 2015.

[16] A. Pfister, A. M. West, S. Bronner, and J. A. Noah, "Comparative abilities of microsoft Kinect and Vicon 3D motion capture for gait analysis," *J. of Medical Eng. & Technology*, vol. 38, no. 5, pp. 274–280, May 2014.

[17] "Kinovea." [Online]. Available: https://www.kinovea.org

[18] C. Damsted, R. O. Nielsen, and L. H. Larsen, "Reliability of video-based quantification of the knee and hip angle at foot strike during running," *Int. J. of sports physical therapy*, vol. 10, no. 2, p. 147, April 2015.