

Examining the Feasibility of a Microsoft Kinect™ Based Game Intervention for Individuals with Anterior Cruciate Ligament Injury Risk*

Zhiyu Huo, *Student Member, IEEE*, Joseph Griffin, Ryan Babiuch, Aaron Gray, Bradley Willis, Skubic Marjorie, *Senior Member, IEEE*, and Shining Sun

Abstract— In this paper, we describe a feasibility study in which the Microsoft Kinect is used for a game-based exercise to strengthen posterior chain muscles which are often weak in those at high risk of anterior cruciate ligament (ACL) injury. In the game, subjects perform a single posterior chain strengthening exercise. The game uses a side-scrolling video display driven by a hip abduction exercise while a player lies down on the floor. Leg lifts beyond a predetermined angle trigger the jumping action of an animated tiger. We describe the scene and game control, which uses depth images from the Kinect. Although Kinect-based skeletal data are used for many games, the skeletal model does not yield good estimates for positions on the floor. Our proposed system uses multiple leg angle estimators for different angle regions to recognize the player lying down and capture the angle between two legs. We conducted an experiment that validates our system with marker-based Vicon ground truth data. We also present results of an end-to-end test using the game, showing feasibility.

I. INTRODUCTION

Anterior cruciate ligament (ACL) injury incidence rates have been estimated as high as 38,000 in girls and women alone [1]. This is linked to an approximated annual cost of 650 million dollars for their medical management [1]. Female athletes have been identified as having a 4-6x increased risk of an ACL injury over their male counterparts in similar cutting sports, such as basketball and soccer [2]. The majority of the ACL injuries, 60% to 80%, are non-contact in nature with two common moves, cutting and landing [3]. Previous work by Hewett et al., has identified four neuromuscular imbalances in at risk individuals [4]. These include ligament dominance, quadriceps dominance, leg dominance and trunk dominance. During single leg landing actions, a knee abduction moment with increased trunk displacement and limited knee flexion has been identified as a biomechanical stressor for an individual's ACL. Hewett et al. found that focusing on posterior chain strengthening of muscle groups such as the gluteus maximus, gluteus medius, gluteus minimus and hamstrings reduces the load to the ACL by controlling frontal plane motion and improving neuromuscular control [4].

The purpose of this study was to determine feasibility of the Kinect in assisting subjects to perform a single posterior chain strengthening exercise. This study includes one specific exercise due to its initial stage. Proven ACL prevention programs include multiple exercises to reduce injury risk [4].

To promote posterior chain lower extremity strengthening and improved biomechanical control of the femur acting on the knee joint, we initially target the gluteus medius muscle. The gluteus medius originates from the external surface of the ilium and inserts on the oblique ridge on the lateral surface of the greater trochanter of the femur [5]. It is a primary abductor of the hip joint, with the anterior fibers medially rotating the hip joint and posterior fibers laterally rotating and extending the hip [5].

Several studies have examined the effectiveness of various exercises in isolating the gluteus medius [6][7]. They concluded, while using surface EMG, during a maximum voluntary isometric contraction, that the side-lying hip abduction exercise was superior in targeting the gluteus medius out of 12 common strengthening exercises [7]. Although multiple posterior chain muscles and exercises have been proposed to improve neuromuscular control of the lower extremity [4], we chose the side-lying hip abduction raise due to its specificity in targeting the gluteus medius and to test usability with the Kinect. Potential strength training protocols for a user are proposed based on previous foundations of overload training and desired hypertrophy of the muscle [8]. A patient would perform the activity at 8-12 reps at 2-3 sets with a 1-2 minute rest period, up to 3x/week [8].

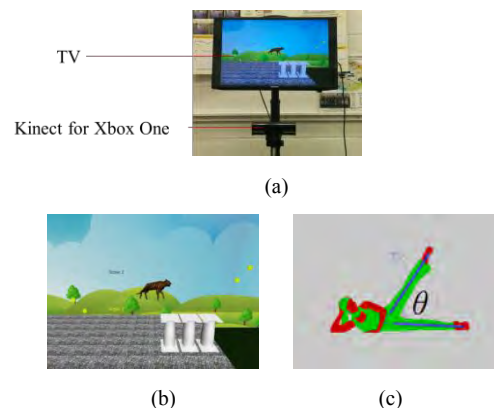


Fig. 1. (a) Game Hardware: TV + Kinect for Xbox One + PC (not shown); (b) Game Scene; (c) Game Input Control Parameter

*Research supported by Mizzou Advantage Program, Univ. of Missouri.

Z. Huo, M. Skubic and S. Sun are with the Dept. of Electrical & Computer Engineering; J. Griffin and R. Babiuch are with the Dept. of Information Science & Learning Technologies; A. Gray and B Willis are with the Dept. of Family Medicine and Orthopedics; University of Missouri, Columbia, MO, USA. (e-mails: zhdy7@mail.missouri.edu, skubicm@missouri.edu)

In previous work, we presented a Kinect-based screening tool for detecting subjects at risk of ACL injury [9]. Here, we describe the “Happy Tiger” intervention, which is a Kinect-based side-scrolling video game developed in the Unity3D game engine [10]. The object is to progress by collecting gold coins. In the game, the player character (a tiger) keeps running forward and jumps onto column arrays to collect coins. If the player misses the columns, it falls to the ground. The jumping action is triggered by the side-lying hip abduction exercise. The Kinect is placed 1m to 1.5m above the floor and tilted down so that the floor is visible. The user lies on one side as shown in Fig. 1, at 2m to 3.5m from the Kinect. The jump is controlled by the angle θ between the two legs, as the user tightens muscles in the front of the top thigh to lift the leg. If the upper leg is lifted such that θ exceeds the threshold angle θ_{th} ($\theta(t) > \theta_{th}$, $\theta(t-1) < \theta_{th}$), the tiger is triggered to jump once. We set θ_{th} to be 30° as a default; it can be changed by the user. Fig. 1 shows the game scene. In Section II, we describe the game methodology. We present a method for estimating the leg angle in the side-lying position using the depth images, due to the limitation of using skeletal data in reclining positions. Section III includes experimental results. We conclude in Section IV.

II. METHODOLOGY

A. Framework of Game Control

The biggest challenge in this game project is to design a fast, accurate and robust game control input system to estimate the angle between two legs when a player is performing the side-lying hip abduction exercise. The control input of a game is typically small. For example, a joystick only sends two float coordinate numbers and a key or throttle sends a single logic or numerical signal. When using the Kinect skeleton model as the control input, the data is only a structure of 25 arrays of 3D joint coordinates. Because the Kinect skeleton model is not accurate for a player lying down, in our game, the input is a 512×424 16-bit depth image stream at 30 frames per second (30 FPS). The image has 217,088 pixels in each frame. Such a large data size requires longer time and more space in processing to get the leg angle as the game control parameter. However, the game engine takes up much of the system resources, so that the time for leg angle estimation is extremely short. Moreover, to make the game work, as a game control approach, our angle estimator should not only estimate leg angle accurately but also cooperate with the game engine and trigger the jumping action during the game. The basic steps to estimate the leg angle θ are:

- **Preprocessing:** Given a raw depth image I , detect the ground plane f and extract the image of the target player I_t from the background.
- **Leg Angle Estimation:** Use one of the three leg angle estimators $\{f_1, f_2, f_3\}$ to get the results $\theta \in \{\theta_1, \theta_2, \theta_3\} = \{f_1(I_t), f_2(I_t), f_3(I_t)\}$.
- **Optimized Selection:** From the training data, we build probability distributions of $P(\theta_m | \theta)$. Select $\theta = \theta_m$ when $m = \text{argmax}_m \{P(\theta_m | \theta)\}$.

Overall, the estimation of leg angle is a selection of the most probable observation value. We do not use Kalman filter or another data fusion approach because, for a single estimator, the performance decreases significantly when it is out of its “working region”. This can lead to poor angle

estimates and false positive or false negative triggers when an estimator is not in its working region. In section C we will discuss three estimators and their respective regions.

B. Background Subtraction

Background subtraction is done first to filter the background and extract the depth image I_t and the corresponding point cloud of the player from the raw image I . In this step, we first detect the floor plane. Using the method in [11], we compute the corresponding camera-coordinate point $p = \{x, y, z\}$ for each pixel of the raw image by using the calibrated intrinsic parameters and generate a point cloud. Then we detect the ground plane using the RANSAC algorithm [12] on the generated point cloud. To speed up the convergence and reduce error, we manually add constraints on the input points. The points to be counted in the RANSAC will only be sampled from the game region $G: \{x, y, z: -1.5m < x < 1.5m, 0m < y < 4m, z < 0m\}$.



Fig. 2. Preprocessing steps: 1) Raw depth image; 2) Extract floor plane and pixels in game region; 3) Extract player and transform points to floor coordinate reference

Let $f: \{x, y, z: ax + by + cz + d = 0\}$ denote the formula of the floor plane. Then we extract the points which are above f and inside G and transform those from camera lens reference to floor reference. Because we only allow one player in the game from the extracted points, we then track the largest point cluster and consider it as the target player. The size of a point cluster is defined by its projection on the floor plane. For efficiency, we generate a 2D binary image from the 3D point cloud (Fig. 2). After getting the target player, the system discriminates the body direction. For estimating the leg angle, we only consider the condition in which the head is to the right and the feet are to the left in the camera view (Fig. 2), while the player may lie on either side when doing the exercise. We build a linear logistic classifier on I_t to decide the direction. If the head is to the left in the input image, the image is horizontally flipped and the x coordinate values of the point cloud as well. This discrimination only runs once at the beginning of the game. The system uses the same direction setting for subsequent frames.

C. Three Leg Angle Estimators

In order to build fast, accurate and robust game control input, we introduce three approaches for estimating leg angle. Each estimator has its own working region. The final result is a combination of these three approaches.

Hough Leg Angle Estimator (HE)

The Hough leg angle estimator (HE) uses the Hough transform method [13] to detect the most likely angles for two legs by voting. We first compute the centroid of the player body which is used as the hip position. Assuming x_n is the x value of point n in all the N points belong to the player, the x coordinate of the body centroid is $x_c = 1/N \sum_{n=1}^N x_n$. Then we only consider the M points which have their x coordinate value $X_m = \{x_m: |x_m - x_c| < 0.05m\}$ so that we have the y and z

coordinate values of the centroid as $y_c = 1/M \sum_{m=1, x \in x_m}^M y_m$ and $z_c = 1/M \sum_{m=1, x \in x_m}^M z_m$. Next, we extract a point set $P_b = \{p_b, x_{p_b} < z_c\}$ which are the points of the lower body. In the voting step, we set the angle values from -90° to $+90^\circ$. The angle α_p and weight w_p for a point p_b vote is given by: $\alpha_p = \tan^{-1}(z_{p_b} - z_c / x_{p_b} - x_c)$ and $w_p = d((x_{p_b}, z_{p_b}), (x_c, z_c))$ where $d(p, q)$ is the Euclidean distance between points p and q . The weight of each angle is the accumulation of weights from points. Fig. 3(a) shows the result of HE and the weights histogram after voting. Then we detect the highest two cones and extract their centroids as angles of each leg with respect to a horizontal line through the centroid. The leg angle is the difference between these two angles.

Body-End Leg Angle Estimator (BE)

In the body-end leg angle estimator (BE), we estimate leg angle by detecting the two ends of the lower body, i.e., the player's feet, and using the body centroid to get the leg angle. Several methods have been tested for detecting the feet; we ultimately chose an iterative approach to search for the end of a human body on the binary image. Starting from the body center $i_c = (u_c, v_c)$, two 3×3 patches W_{up} and W_{down} are directed using two potential matrices M_{up} , M_{down} to detect the upper and lower feet.

$$M_{up} = \begin{bmatrix} 8 & 5 & 3 \\ 7 & 0 & 2 \\ 6 & 4 & 1 \end{bmatrix}, M_{down} = \begin{bmatrix} 5 & 4 & 1 \\ 7 & 0 & 2 \\ 8 & 6 & 3 \end{bmatrix}$$

Let $i_x(t)$ denote the pixel of the current center point of a patch $W_x^{i_x(t)}$. The next step patch center $i_x(t+1)$ is selected from the eight neighbors of $i_x(t)$ by using M_x . The weight of selecting a neighbor is:

$$i_x(t+1) = \prod W_x^{i_x(t+1)} M_x(i_x(t+1) - i_x(t)). \quad (1)$$

We move the patch to the pixel with the highest weight. A potential matrix M_x keeps the patch moving inside the white pixels of I_t and drives it towards the x direction of the body end. The iteration stops when $d(i(t'), i(t'-\varphi)) < \beta$. Here d is the Euclidean distance between two pixels, $\varphi=10$ and $\beta=1$ which means that the iterations stop when the patch cannot move further (Fig. 3(b)). Assuming $p_{i(t')} = (x_{i(t')}, y_{i(t')}, z_{i(t')})$ is the point of the end patch center pixel $i(t')$, the leg angle of the corresponding end detected is:

$$\theta_x = \tan^{-1} \frac{z_{i(t')} - z_c}{x_{i(t')} - x_c} \quad (2)$$

Then the leg angle is $\theta_{BE} = \theta_{up} - \theta_{down}$.

LBP-Foot Leg Angle Estimator (LE)

We estimate the leg angle by detecting each foot in the LBP-Foot leg angle estimator (LE). Local-binary-patterns (LBP) offer an approach for generating features very quickly and are often used for classification in computer vision [14]. Here, we generate LBP features over a 32×32 patch on the binary image I_t . We train a foot detector using 100 patches which are randomly chosen from a 20s Kinect video of the hip abduction exercise. When running the detection, we use the 32×32 foot detector to scan the body image I_t through the region $R_s, \{u < u_c, v < v_c\}$ by a step length of 4 by 4. Then we select two patches which have the highest response as the upper foot and the lower foot (Fig. 3(c)). The overlapping

patches are removed from the result and the final results are the raised foot and the foot on the floor. To locate foot position, we compute the mean of the point coordinates for all the white pixels in a patch as $P_{feet} = \{x_{feet}, y_{feet}, z_{feet}\} = \{\text{mean}(X_{white}), \text{mean}(Y_{white}), \text{mean}(Z_{white})\}$. Then the leg angle is:

$$\theta_{LE} = \tan^{-1} \frac{z_{feet}^{up} - z_c}{x_{feet}^{up} - x_c} - \tan^{-1} \frac{z_{feet}^{down} - z_c}{x_{feet}^{down} - x_c} \quad (3)$$

D. Final Leg Angle Estimate

During testing each angle estimator, we observe that the deviation of an estimator can stay very small inside a region and increase very fast when the leg angle moves out of this region. Fortunately, the combined "working region" of the three estimators covers the entire angle space from 0° to 90° so that there is at least one estimator that works well throughout this range. Therefore, we draw the distributions of leg angle as $\{P(\theta_{HD}|\theta), P(\theta_{BD}|\theta), P(\theta_{FD}|\theta)\}$ for each estimator on different angles. The predicted angle $\theta(t)$ at time t is:

$$\theta(t) = \theta_x(t) = \text{argmax}_{x \in \{HD, BD, FD\}} \{P(\theta_x(t)|\theta)\} \quad (4)$$

In our program, we let $P(\theta_x|\theta) \approx P(\theta_x|\theta_\mu)$ because we do not know $P(\theta_x|\theta)$ so that at time t we use $\theta_\mu(t) = \theta(t-1) + (\theta(t-1) - \theta(t-2))$ which yields a good estimation of the real leg angle.

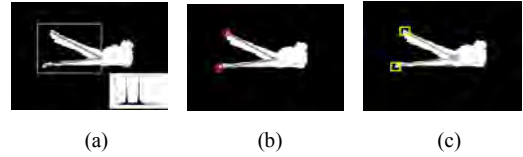


Fig. 3 (a) Hough leg angle estimator (HE); (b) Body-end leg angle estimator (BE); (c) LBP-Foot leg angle estimator (LE)

III. FEASIBILITY STUDY

As a precursor to a human subjects experiment with athletes, a feasibility study was conducted to assess the leg angle estimation and to investigate the leg angle as a control mechanism in the game. To test the leg angle estimation, six athletes performed 18 to 24 leg lifts, yielding a total of 132 lifts following Institutional Review Board. We used a marker-based Vicon system to capture the ground truth of the leg angle. To collect Vicon data, we attached four markers to subjects. Two of them were placed at the anterior aspect of the ankles; the other two were placed at the anterior hip. The four markers draw the shape of two legs on each person. The Kinect data and Vicon data were recorded simultaneously during the test. Because the Vicon system captures data at a much higher frequency than the Kinect, we sample the Vicon data at the Kinect capture moment to better compare with the Kinect data. To evaluate the system, we logged the leg angle result obtained separately by each estimator and the angle captured by the combined estimation. Table I shows the results of the Kinect against the Vicon reference angles.

TABLE I. Performance of Kinect against Vicon

	HD	BD	FD	Combined
Error([0, 90]) (deg)	6.1±3.1	6.5±4.7	6.9±5.6	5.5±3.62
Working Region (WR) (deg)	[0, 20]	[20, 45]	[45, 90]	[0, 90]
Error (WR) (deg)	7.2±2.8	5.7±4.2	3.0±2.4	N/A

There are many factors affecting the error of the leg angle estimation. In addition to estimation error, the time difference between the Kinect and Vicon data captures can introduce errors. Because the Kinect and Vicon systems capture data at different frequencies, it is not possible to consistently compare angles captured at exactly the same time. Fig. 4(a) shows leg angle estimates produced by the two systems. As a result, we show results from another test, in which the jump trigger times are compared using the Kinect and Vicon leg angle estimates. It can be noticed that Fig. 4(a) shows oscillation when the leg angle is approaching zero degrees. This is because the error of the estimators increases at the region around 0° . However, the error will not affect the game control if we set the trigger angle from 15° to 55° which is consider as the working region of the combined leg angle estimator. Fig. 4(b) shows the average difference between the Kinect and Vicon as a function of the trigger angle from 15° to 55° . The average across all trigger angles is about 0.05s which shows that the Kinect-based leg angle game input system provides a real-time and accurate response to the hip abduction exercise motion.

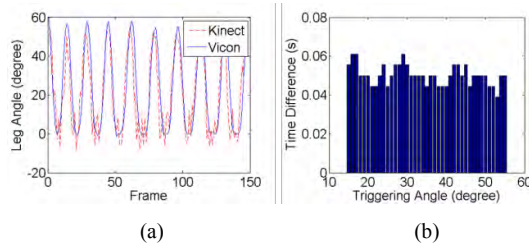


Fig. 4. (a) Time-Leg Angle curves: **dashed line** – Kinect; **solid line** – Vicon. (b) Trigger time difference between Kinect and Vicon.

To further test the performance of the leg angle estimator, we integrated our leg angle estimation control into the “Happy Tiger” game and ran end-to-end tests, using an HP envy laptop (CPU: 2.2GHz, RAM: 4G) with the Windows 8.1 operating system. The leg angle estimator runs in a different thread from the game scene and takes 7ms for image processing. The game scene ran at 30 frames per second which is a standard speed as a video game. Tests were performed using each angle estimator individually and the combined estimator. The number of lifts and triggered jumps by individual estimators and the combined angle estimator are listed in Table II. Using the combined leg angle estimator, the success rate of triggering is 99%.

TABLE II. Performance of Using Leg Angle Estimators in the Game

	# of Lifts	# of Jumps	Success Rate
Combined	203	201	99.0%
HD	40	30	97.5%
BD	40	36	90.0%
FD	40	35	87.5%

IV. CONCLUSION

In this paper, we tested the feasibility of a Kinect-based computer game for a side-lying hip abduction exercise. We focused on the game control, which used Kinect depth images to estimate the angle between two legs. The combination of three estimators remarkably reduced the error and improved the success rate on triggering jumps. Game testing showed that our system is fast, accurate, robust and works well for game control. The results illustrate that it is feasible to use the

Kinect for hip strengthening exercises, targeting individuals at risk for ACL injury. A Kinect-based exercise game has advantages. First, it shares the same game platform as other Kinect-based video games. Second, it is convenient for players to start and play the game, not relying on wearable markers or wearable sensors. This paper shows the potential for the Kinect to be an interactive medium for floor-based exercises, targeting injury prevention. In future work, we will conduct studies with athletes. We plan to extend the system to incorporate other exercises and review feedback from users.

REFERENCES

- [1] A. P. Toth, and F. A. Cordasco, “Anterior cruciate ligament injuries in the female athlete,” *The Journal of Gender-Specific Medicine: JGSM: The Official Journal of the Partnership for Women’s Health at Columbia*, vol. 4, no.4, pp. 25-34, 2001.
- [2] E. A. Arendt, A. Julie, and R. Dick, “Anterior cruciate ligament injury patterns among collegiate men and women,” *Journal of Athletic Training*, vol. 34, no. 2, pp. 86-92, 1999.
- [3] T. Krosshaug, A. Nakamae, B. P. Boden, L. Engebretsen, G. Smith, J. R. Slauterbeck, T. E. Hewett, and R. Bahr, “Mechanisms of anterior cruciate ligament injury in basketball video analysis of 39 cases,” *The American Journal of Sports Medicine*, vol. 35, no. 3, pp. 359-367, 2007.
- [4] T. E. Hewett, K. R. Ford, B. J. Hoogenboom, and G. D. Myer, “Understanding and preventing acl injuries: current biomechanical and epidemiologic considerations-update 2010,” *North American Journal of Sports Physical Therapy: NAJSPT*, vol. 5, no. 4, pp. 234-251, 2010.
- [5] E. K. McCreary, P. G. Provance, M. M. Rodgers, and W. A. Romani, *Muscles: Testing and Function, With Posture and Pain*, 5th ed. Lippincott Williams & Wilkins (LWW), Philadelphia, 2005
- [6] J. M. McBeth, J. E. Earl-Boehm, S. C. Cobb, and W. E. Huddleston, “Hip muscle activity during 3 side-lying hip-strengthening exercises in distance runners,” *Journal of Athletic Training*, vol. 47, no. 1, pp. 15-23, 2012.
- [7] L. J. Distefano, J. T. Blackburn, S. W. Marshall, and D. A. Padua, “Gluteal muscle activation during common therapeutic exercises,” *Journal of Orthopaedic & Sports Physical Therapy*, vol. 39, no. 7, pp. 532-540, 2009.
- [8] L. T. Brody, C. M. Hall, *Therapeutic Exercise: Moving toward Function*. Lippincott Williams & Wilkins (LWW), Philadelphia, 2011.
- [9] E. E. Stone, M. Butler, A. McRuer, A. Gray, J. Marks, and M. Skubic, “Evaluation of the Microsoft Kinect for screening ACL injury,” in *Proc. 35th Engineering in Medicine and Biology Society (EMBC)*, Chicago, IL, 2013, pp. 4152-4155.
- [10] N. H. Khalifa1, Q. V. Nguyen, S. Simoff and D. Catchpoole, “A Visualization System for Analyzing Biomedical and Genomic Data Sets Using Unity3D Platform,” in *Proc. 8th Australasian Workshop on Health Informatics and Knowledge Management*, Sydney, Australia, 2015, pp. 47-53.
- [11] J. Smisek, M. Jancosek, and T. Pajdla, “3D with Kinect,” in *Proc. IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, Barcelona, Spain, 2011, pp. 3-25.
- [12] P. Meer, D. Mintz, A. Rosenfeld, and D. Y. Kim, “Robust regression methods for computer vision: A review,” *International Journal of Computer Vision*, vol. 6, no.1, pp. 59-70, 1991.
- [13] J. Illingworth, and J. Kittler, “A survey of the Hough transform,” *Computer Vision, Graphics, and Image Processing*, vol. 44, no. 1, pp. 87-116, 1988.
- [14] Z. Guo, and D. Zhang, “A completed modeling of local binary pattern operator for texture classification,” *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1657-1663, 2010.