S & M 1230

Simulating Feasibility Assessment of the Golf Swing by Kinect Motion-Sensing Photography

Hao-Hsiang Ku, Wen-Cheng Wang,[1*] and Yen-Wu Ti

Department of Computer Science and Information Engineering, Hwa Hsia University of Technology,
No. 111, Gongzhuan Rd., Zhonghe Dist., New Taipei City 235, Taiwan (R.O.C.)
[1]Department of Business Administration, Hwa Hsia University of Technology
No. 111, Gongzhuan Rd., Zhonghe Dist., New Taipei City 235, Taiwan (R.O.C.)

(Received May 28, 2015; accepted April 22, 2016)

Keywords: golf swing, motion-sensing, Kinect

Golf swing analysis is a popular research subject, and is divided into contact sensing and non-contact sensing analysis according to different research methods. Contact sensing research often uses wearable sensors, including triaxial accelerometers, gyroscopes, and gravity accelerators, which are mounted on the golfer's limbs or relevant equipment. This study aims to construct a teaching system for correcting swing action using non-contact sensing Kinect motion-sensing photo equipment. The proposed system is expected to collect a golfer's pose and action during the backswing for data analysis, and detect incorrect backswing actions in order to attain the goal of interactive golf teaching.

1. Introduction

Golf is a popular recreational sport in Europe and North America, and in recent years, prevails throughout Asia. The majority of the golfing population is in the U.S. and Japan, with 26 million and 17 million players, respectively.(1) According to the statistics of the Control Yuan in Taiwan, golf has been listed as a university sports course in Taiwan for 20 years. At present, 112 colleges in Taiwan offer golf courses, and golf will be a title event of the Olympic Games in 2016.(2) However, golf is a challenging game, as it is difficult to strike the ball accurately so that the ball flies high and far in the right direction. Such misplay is likely to frustrate learning.

Cooper and Mather measured the maximum angular rate of golfers' swings as a function of time and constructed a swing mechanics model for different golfers, which could be applied as the standard for a correct swing pose and improve the golfers' swing based on the data.(3) However, swing data differ due to different golfer features; thus, the established swing standard is inapplicable to most people. Reyes and Mittendorf constructed a new golf analysis model, which is a biomechanical model built according to the biological features of specific golfers.(4) The relevant parameters, including length, weight, and area of the club, are adjusted by this model to lower the golf learning threshold. This study aims to construct a teaching system for correcting swing action using non-contact sensing Kinect motion-sensing photo equipment. The proposed system is expected to collect a golfer's pose and action during the backswing for data analysis, and detect incorrect backswing actions to attain the goal of interactive golf teaching.

[*]Corresponding author: e-mail: wcwang@cc.hwh.edu.tw

ISSN 0914-4935 © MYU K.K.

2. Motion Analysis of Golf Swing

Wu *et al*. indicated that the golf swing is a sophisticated multiple-joint involved motion, which requires high stability and accuracy.(5) Ghasemzadeh *et al*. discussed that wrist rotation actions include the forward movement of the left knee, turning of buttocks and spine, and movement of shoulder joints.(6,7) Moreover, the strength received by the feet and wrists is varied with the swing phases. Pink *et al*. divided the golf swing motion into six phases (Fig. 1), including take away, forward swing, acceleration, early follow through, and late follow through.(8) The backswing begins with the feet leading the body to act, namely, the action of the lower body causes the center of body weight to shift to the right foot before the body action of backswing to the top.(9) Faldo, Simmon and Foston reported that the center of gravity gradually shifted to the right during the backswing, and when the top of the backswing was reached, about 90% of body weight was on the right foot, and specifically, it should be on the inside of the right foot.(10,11)

Golf swing analysis is a popular research subject, and it is divided into contact sensing and non-contact sensing according to different research methods.(12,13) Contact sensing often employs wearable sensor devices, including triaxial accelerometers, gyroscopes, and gravity accelerators, which are mounted on golfers' limbs or relevant equipment. The devices obtain an individual golfer's values during the swing, which are analyzed by many researchers.(6,14–16) In contrast, non-contact sensing uses pictures or videos to record a golfer's actions during the swing, and the actions are analyzed.

Blake and Grundy used motion capture to analyze the golf swing.(17) This system builds a complete 3D motion model using data from sensing patches on a golfer's body, and the accuracy of motion is obtained by analyzing the 3D model. However, wearing multiple sensing elements influences a golfer's swing, meaning the golfer cannot swing freely. Somjarod *et al*. recorded 30 male professional and amateur golfers' swing poses in videos.(18) This study analyzed the variances in the knees of the golfers during the period of the swing, using the video data, and observed the appropriate knee angle and direction of displacement. Karliga and Hwang used the video captured by one camera to reconstruct the 3D action of a golf swing and analyzed a 3D model of the body to determine the pose variations.(19) Gehrig *et al*. used the video of swing path to analyze the club head path and constructed a system to determine the result of the golfer's swing.(20)

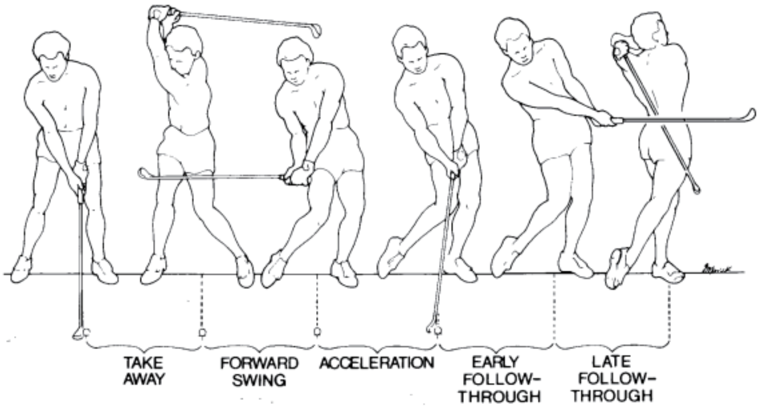

Fig. 1. Golf swing motion as described by Pink (1993).(8)

3. Kinect Action Sensing and Motion-Sensing Photography

The Kinect sensor is required to be placed at a sensible distance from the golfer, which is generally 1.2 to 3.5 m. Figure 2 illustrates a Microsoft Kinect sensor. It has three lenses; the middle lens is an RGB color camera for recording color images, while the left and right lenses are the 3D depth sensors formed from an infrared (IR) transmitter and an IR complementary metal-oxide-semiconductor (CMOS) camera. The Kinect detects the user's action depending on the 3D depth sensor.

As shown in Table 1, the maximum resolution of the Kinect color camera is 1280 × 960, and the maximum resolution of the IR camera is 640 × 480. The Kinect is combined with follow-focus technology, where the base motor rotates as the focused object moves. Kinect can capture two images: a color image, and a depth image. The color image is captured by the middle RGB color camera; the depth image information is captured by the right camera, and the leftmost IR camera emits an invisible laser. Different depths are determined by an algorithm; and the image information is derived from the feedback depth information. As the RGB color camera and IR CMOS camera have horizontal parallax, the color image and depth image captured inside the program have only a small overlapping area that is not completely coincident. Therefore, there must be an additional coordinate mapping function for processing image differences.

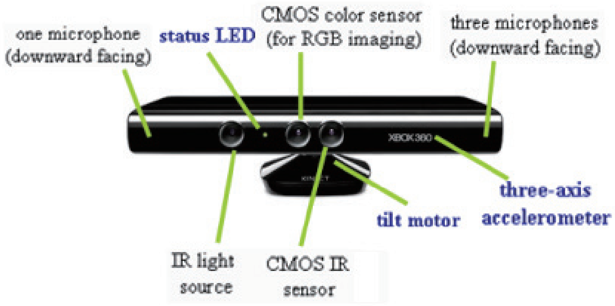

Fig. 2. (Color online) Microsoft Kinect sensor.

Table 1
Specifications for the Kinect.

Field of view	Horizontal field of view	57 degrees
	Vertical field of view	43 degrees
	Physical tilt range	±27 degrees
	Depth sensor range	1.2–3.5m
Data streams	Depth camera: 320 × 240 16-bit @ 30 frames/sec using monochrome video stream Sensing at 16-bit QVGA resolution (320 × 240 pixels with 65,536 levels of sensitivity). 640 × 480 32-bit color@ 30 frames per sec 16-bit audio @ 16 kHz	
Resolution	640 × 480 pixels @ 30 Hz (RGB camera) 640 × 480 pixels @ 30 Hz (IR depth-finding camera)	
Audio format	16-kHz, 24-bit mono pulse code modulation (PCM)	
Audio input characteristics	A four-microphone array with 24-bit analog-to-digital converter (ADC) and Kinect-resident signal processing including acoustic echo cancellation and noise suppression	
Connectivity	USB 2.0	

Kinect uses light coding technology to obtain the image depth. Light coding technology uses IR light to emit an invisible class 1 laser light, and the diffuser in front of the lens uniformly projects the laser light within the measurement space. The laser speckle calibrates the specific space. When the laser light shoots a rough object, random speckles are formed, where the speckles in any two places within a space are of different patterns, and vary with the distance. Therefore, these speckles are equivalent to marks on the overall space. To measure a space, the reference plane must be taken out to a certain distance, then the IR camera records all speckles on each plane. The positioning information for the overall space can be obtained from the laser speckles in random shapes. When the original data are captured, they are calculated by the chip as an image with 3D depth. The light colors in depth image streaming are close to the range sensor, while the deep colors are far from the range sensor (Fig. 3).

The Kinect is the origin of the 3D world coordinate system, and the *z*-axis is the positive direction. A point $p(x, y, z)$ on the surface of an object in the 3D scene is projected to a point $P(X, Y)$ on the 2D focal plane. Their relation can be inferred from reference 18 by Eq. (1):

$$X = f\frac{x}{z},\ Y = f\frac{y}{z}. \tag{1}$$

Since the focal length *f* of a Kinect camera is 22.5–585 mm, it can be determined from Eq. (2):

$$p = (x, y, z) = (z\frac{X}{f}, z\frac{Y}{f}, z). \tag{2}$$

The Python code below illustrates how a 3D point can be computed from the pixel coordinates and the depth value:

```
fx = 525.0  # focal length x
fy = 525.0  # focal length y
cx = 319.5  # optical center x
cy = 239.5  # optical center y
factor = 5000 # for the 16-bit PNG files
```

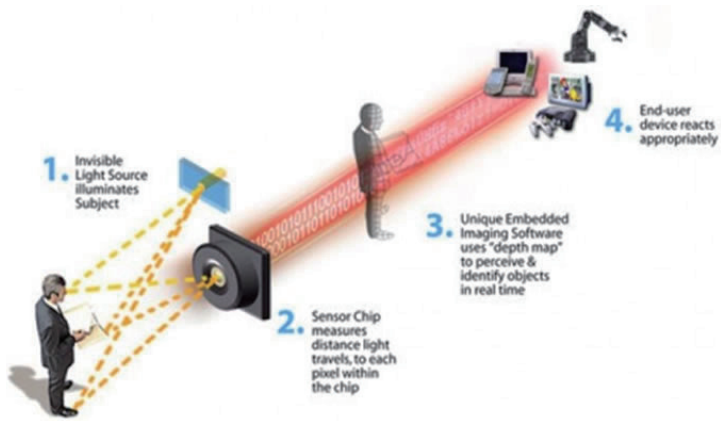
1. Invisible Light Source illuminates Subject
2. Sensor Chip measures distance light travels, to each pixel within the chip
3. Unique Embedded Imaging Software uses "depth map" to perceive & identify objects in real time
4. End-user device reacts appropriately

Fig. 3. (Color online) Kinect 3D and light coding (Figure from http://www3.pcmag.com).

```
# OR: factor = 1 # for the 32-bit float images in the ROS bag files
for v in range(depth_image.height):
    for u in range(depth_image.width):
    Z = depth_image[v,u] / factor;
    X = (u − cx) * Z / fx;
    Y = (v − cy) * Z / fy;
// transform (x,y,z) to (xw,yw,zw)
Eigen::Vector4d w = Tk * Eigen::Vector4d(x, y, z, 1);
    xw = w(0); yw = w(1); zw = w(2);
```

The depth image information obtained by Kinect is converted into a skeleton tracking system. Just as every joint in the human skeleton has a unique name, Software Development Kit (SDK) designates a nodename for access (Fig. 4). The Kinect action sensing side obtains the human skeleton information at a refresh frequency of 30 frames per second, and the information obtained is transmitted via the AUX side to the universal serial bus (USB) connection line of the Kinect processing element, which is located in the central motion information processing side. As the refresh frequency is 30 frames per second, there will be a 33 ms error time between joints at each time. When the user's body information is identified and captured, the system integrates the information into a skeleton table. This skeleton tracking system can simultaneously track 6 human bodies and identify two body actions. For each human body, 20 joints can be recorded, including the trunk and limbs.(20)

4. System Planning and Design

Before the swing motion, the golfer stands within an appropriate range for the Kinect action sensing terminal. The action sensing end is in charge of capturing the color image and depth image from the visual range and transmitting the two data streams, via USB interface to the Kinect processing element of the central motion information processing end. The unit uses Kinect SDK to convert the depth image information into a human skeleton structure, which is sent to the logical processing unit. The motion tracking technology of Kinect is different from the previous 2D

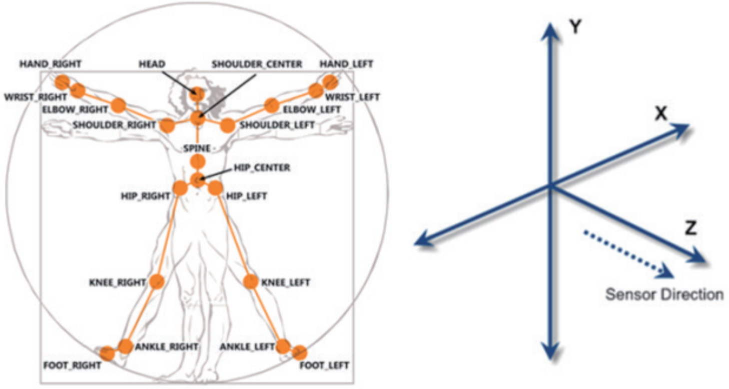

Fig. 4. (Color online) Microsoft Kinect skeleton tracking system (Figure from http://archive.renauddumont.be).

operating mode. The remote control and touch panel operating modes are 2D (they only capture the spatial position of the target, i.e., *x*-axis and *y*-axis), whereas the motion tracking technology not only captures the spatial position (*x*-axis and *y*-axis) and color of the target, but also captures the depth (*z*-axis), range, distance, and the surroundings of the target. Consequently, it is also known as the 3D operating mode.

4.1 Microsoft Kinect starts and sets skeleton detection

The software and hardware for developing a Kinect application are described as follows. The Microsoft Windows 7 operating system must be used, as the software operating environment requires a 32-bit version, although 64-bit is acceptable. Regarding the hardware, there must be a central processing unit (CPU) above a dual-core of 2.66 GHz, random-access memory (RAM) above 2 GB, a video adapter supporting DirectX 9.0c or higher, and a Kinect sensor. The software must be Visual Studio 2010 or Visual C# 2010 Express to compose the program code, the .NET Framework 4.0 environment, and Kinect SDK for Windows. As skeleton detection is used, Microsoft DirectX 9 SDK-June 2010 and Runtime for Microsoft DirectX 9 must be installed.

A simulated evaluation system use built on the C# program of the .Net framework, Kinect for Windows SDK (Fig. 5), and the Bluetooth API link library were used. The Kinect processing element was connected to the Kinect action sensing end, via the AUX end to the USB, and used Kinect for Windows SDK and the Bluetooth API link library. The Kinect for Windows used in this study was Kinect for Windows Developer Toolkit version 1.7. The Bluetooth connection SDK used 32 feet, .NET version 35. Two programs were downloaded from Microsoft's official SDK webpage: Kinect for Windows SDK and Kinect for Windows Developer Toolkit. In the installation, the Kinect for Windows SDK must be installed before the Kinect for Windows Developer Toolkit. Kinect for Windows Developer Toolkit contained a Kinect sensor driver, the Net component, a dynamic linking function library, technological documents, example programs, and auxiliary means.

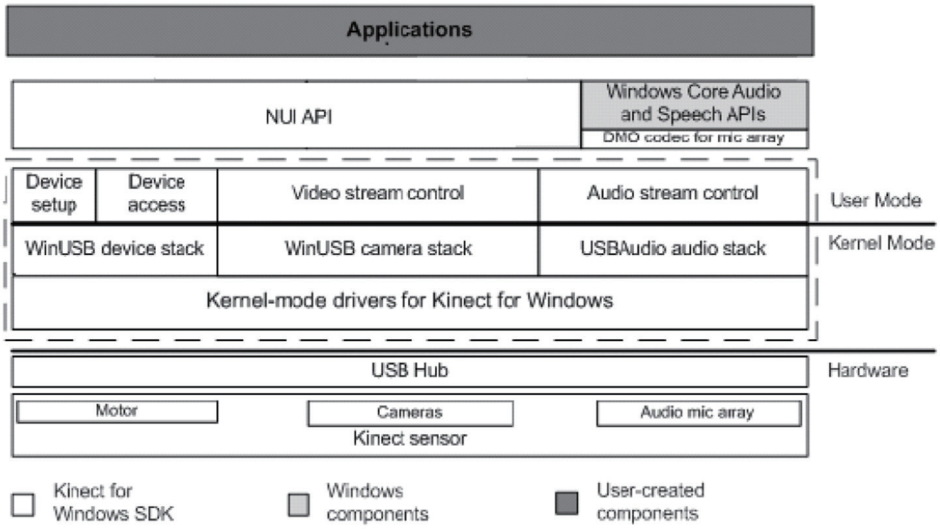

Fig. 5. Microsoft Kinect SDK architecture (Figure from http://msdn.microsoft.com/).

Kinect for Windows SDK provided two development components: the indigenous application database and the.Net component, as repackaged by the bottom dynamic linking function library. The indigenous application database contained the header files (.h file) and related function library (.lib file) for compilation, while the supported development languages included C++, C#, Visual Basic, Ruby, Python, and others. The Kinect application was developed using the NUI Library to receive the color image data, depth image streaming, and voice streaming from Kinect. To obtain Kinect skeleton information, the Skeleton Stream attribute in the Kinect Sensor object, called Enable, was used to open the skeleton information. The detail working item steps are listed as follows:

STEP 1: Download all required software (without Kinect© plugged in).

STEP 2: In order for the Kinect© for Windows SDK v1.5 and the Kinect© SDK with Matlab to work in conjunction, certain changes must be made to their files, to getimagedata.cpp (included in the Kinect© SDK with Matlab program), and to NuiApi.h (included in the Kinect© for Windows SDK v1.5).

STEP 3: Copy the header files from the Kinect© for Windows SDK v1.5 include folder to your current Matlab working folder; Move Kinect10.lib from the Kinect SDK amd64\lib folder to the Microsoft Visual Studio lib folder.

STEP 4: To enter the command:

```
>> mex–setup
```

into the Matlab command window, choose MS Visual Studio 2010 as your compiler, and then enter the command:

```
>> mex getimagedata.cpp -l'Kinect10'.
```

STEP 5: To plug Kinect© into computer and test Kinect© in Matlab, enter the following commands:

```
>> [a b] = getimagedata(1);
>>while(1); [a b] = Kinect(); subplot(1,2,1); imagesc(a); axis image; subplot(1,2,2); imshow(b/255); axis image; drawnow; end;
```

STEP 6: Download the .m files included with this exercise and copy them to the current Matlab working folder.

4.2 Acquisition of joints

When a golfer enters the Kinect action sensing end range, the skeleton joint data are transmitted to the Kinect processing element. The Kinect processing element confirms that the golfer's joints are tracked, and the skeleton joint data are sent to the logical processing unit. The Kinect motion-sensing photo equipment captures the information from the human skeleton in three steps: (1) the human body block is marked by the first 3 bits of depth image; (2) the human body block is separated into 31 parts for calculating skeleton coordinates; and (3) 48 feature points are calculated by the Microsoft human body computing algorithm, and the 48 feature points are exported as 20 human body joints. The program syntax for obtaining joints is described as follows:

```
SkeletonPointHandSkePoint_Hand = jc[JointType. Shoulder_center].Position;
DepthImagePointDepth_hand_centor =
this.sensor.MapSkeletonPointToDepth(HandSkePoint_Center,
DepthImageFormat.Resolution320x240Fps30);
```

When Kinect skeleton information streaming is opened, the skeleton information detected by Kinect can be obtained (at most 6 pieces of skeleton information and 20 human body joints), as defined in NuiSensor.h: #_define NUI_SKELETON_COUNT (6), and stored in NUI_SKELETON_DATA.

```
typedef struct _NUI_SKELETON_DATA {
    NUI_SKELETON_TRACKING_STATE eTrackingState;
    DWORD dwTrackingID;
    DWORD dwEnrollmentIndex;
    DWORDdwUserIndex;
    Vector4 Position;
    Vector4 SkeletonPositions[20];
    NUI_SKELETON_POSITION_TRACKING_STATE eSkeletonPositionTrackingState[20];
    DWORD dwQualityFlags;
  } NUI_SKELETON_DATA;
  eTrackingState：
  typedef enum _NUI_SKELETON_TRACKING_STATE
  {
    NUI_SKELETON_NOT_TRACKED = 0,
    NUI_SKELETON_POSITION_ONLY,
    NUI_SKELETON_TRACKED
  } NUI_SKELETON_TRACKING_STATE;
```

The skeleton joints of Kinect have three states, according to the probable value of SKELETON_TRACKING_STATE, including

(1) NUI_SKELETON_NOT_TRACKED: skeleton not detected,
(2) NUI_SKELETON_POSITION_ONLY: skeleton detected but not tracked,
(3) NUI_SKELETON_TRACKED: skeleton has been tracked.

4.3 Judgment rules

The *x*-axis and *y*-axis of various nodes in the Kinect human joint information are identified. Before the identification, the golfer must revert to a preparatory period. Taking a right-handed golfer as an example, as the golf swing moves the center of gravity to the inside of the left leg, the system establishes the *X* and *Y* thresholds according to the human joints of the left shoulder and left elbow in the preparatory period, where the *X* threshold takes the left shoulder as a reference point, and the *Y* threshold takes the left elbow as a reference point. The golfer's backswing is to the top clockwise from the *X* reference point. The standard stance of the backswing is that the width of the feet is identical to the width of shoulders, and the shoulders drive the backswing. Regarding the club head, both hands and the trunk turn to the upper right during the backswing. In the ideal motion, the waist turns 45°, the shoulders turn 90°, the buttocks turn 45°, and the difference of 45° generates the torque of the upper body. Thus, the body turns to the top of the backswing as the main strength of the downswing. The judgment process is described as follows:

STEP 1: Judge whether Kinect human joint information and triaxial sense information exist or not; if not, the complete data set must be received before startup. The overall judgment process is started when the two data sets are given.

STEP 2: According to the swing period being evaluated, the 20 human joints imported by Kinect are matched with the pose logic data.

STEP 3: If any sense data mismatches the pose logic, a warning message is given; if all sense data are identified, the judgment mechanism returns to the origin, receiving data continuously, and pose judgment is continuously repeated. The judgment rules are described, as follows:

(1) Address

(i) Judgment 1: left shoulder higher than right shoulder

The SHOULDER_LEFT and SHOULDER_RIGHT joints provided by Microsoft Kinect human joints are used for detection. When the *Y* value of SHOULDER_LEFT is smaller than the *Y* value of SHOULDER_RIGHT, it means the golfer's left shoulder is lower than the right shoulder. The discriminant is expressed as

$$\text{If } (\mathrm{Y}_{\mathrm{SHOULDER\_LEFT}} - \mathrm{Y}_{\mathrm{SHOULDER\_RIGHT}}) \geq \mathrm{TH}_1,\ \mathrm{EP}_1 = 1,\ \text{Otherwise } \mathrm{EP}_1 = 0, \tag{3}$$

where TH_1 is the system set threshold.

(ii) Judgment 2: spacing between feet

In terms of golf swing stance, the width between the feet shall be identical with the shoulder width. The SHOULDER_LEFT, SHOULDER_RIGHT, FOOT_LEFT, and FOOT_RIGHT joints provided by Microsoft Kinect human joints are used for detection. When the distance between the *X* value of FOOT_RIGHT and the *X* value of FOOT_LEFT is greater than the distance between the *X* value of SHOULDER_RIGHT and the *X* value of SHOULDER_LEFT, it means the golfer's stance is too wide. The discriminant is expressed as

$$\begin{gathered}\text{If } \mathrm{Distant}_{\mathrm{FL}}^{\mathrm{FR}} - \mathrm{Distant}_{\mathrm{SL}}^{\mathrm{SR}} \geq \mathrm{TH}_2,\ \mathrm{EP}_2 = 0,\ \text{Otherwise } \mathrm{EP}_2 = 1 \\ \mathrm{Distant}_{\mathrm{FL}}^{\mathrm{FR}} = |\mathrm{X}_{\mathrm{FOOT\_RIGHT}} - \mathrm{X}_{\mathrm{FOOT\_LEFT}}| \text{ (feet width)}, \\ \mathrm{Distant}_{\mathrm{SL}}^{\mathrm{SR}} = |\mathrm{X}_{\mathrm{SHOULDER\_RIGHT}} - \mathrm{X}_{\mathrm{SHOULDER\_LEFT}}| \text{ (shoulder width)},\end{gathered} \tag{4}$$

where TH_2 is the system set threshold.

(2) Mid backswing

(i) Judgment 3: right shoulder higher than left shoulder

The correlation among the SHOULDER_LEFT, SHOULDER_RIGHT, SHOULDER_CENTER, and HIP_CENTER joints provided by Kinect human joints is used for judgment. The discriminant is expressed as

$$\begin{gathered}\text{If } |\mathrm{Y}_{\mathrm{SHOULDER\_RIGHT}} - \mathrm{Y}_{\mathrm{SHOULDER\_LEFT}}| \geq \mathrm{TH}_{31} \text{ AND} \\ |\mathrm{X}_{\mathrm{SHOULDER\_CENTER}} - \mathrm{X}_{\mathrm{HIP\_CENTER}}| \geq \mathrm{TH}_{32},\ \mathrm{EP}_4 = 0,\ \text{Otherwise } \mathrm{EP}_3 = 1,\end{gathered} \tag{5}$$

where TH_{31} and TH_{32} are the system set threshold.

(ii) Judgment 4: whether right elbow is raised or not

The correlation among the SHOULDER_RIGHT, ELBOW_RIGHT, and HAND_RIGHT joints provided by Kinect human joints is used for judgment. The discriminant is expressed as

$$\begin{gathered}\text{If } |\mathrm{Y}_{\mathrm{SHOULDER\_RIGHT}} - \mathrm{Y}_{\mathrm{ELBOW\_RIGHT}}| < \mathrm{TH}_{41} \text{ AND} \\ |\mathrm{Y}_{\mathrm{HAND\_RIGHT}} - \mathrm{Y}_{\mathrm{ELBOW\_RIGHT}}| < \mathrm{TH}_{42},\ \mathrm{EP}_4 = 1,\ \text{Otherwise } \mathrm{EP}_4 = 0,\end{gathered} \tag{6}$$

where TH_{42} is the system set threshold.

(3) Top of backswing

(i) Judgment 5: backswing

The *Y*-axis difference value of SHOULDER_CENTER joint provided by Kinect human joints in the swing process from backswing starting point to the top of backswing is used for judgment. The discriminant is expressed as Eq. (7).

$$\text{If } (Y_{\text{start}} - Y_{\text{end}}) \geq TH_{51},\ EP_5 = 1, \text{ Otherwise } EP_5 = 0, \tag{7}$$

where TH_{51} is the system set threshold.

(ii) Judgment 6: backswing amplitude

The body shifts if the backswing amplitude is too large; the *X* value of SHOULDER_CENTER and HIP_CENTER joints provided by Kinect human joints is used for judgment. The discriminant is expressed as Eq. (8).

$$\text{If } |Y_{\text{SHOULDER\_CENTER}} - Y_{\text{HIP\_CENTER}}| < TH_{61},\ EP_6 = 1, \text{ Otherwise } EP_6 = 0, \tag{8}$$

where TH_{61} is the system set threshold.

5. Simulation Results

This section presents the simulation of the golf swing by Kinect. Figure 6 illustrates a golfer's swing as detected by Kinect. A golf swing motion is separated into six phases, including take away, forward swing, acceleration, early follow through, and late follow through. A Kinect sensor can catch and record this action. Figure 7 depicts an implemented golf swing by Kinect motion-sensing

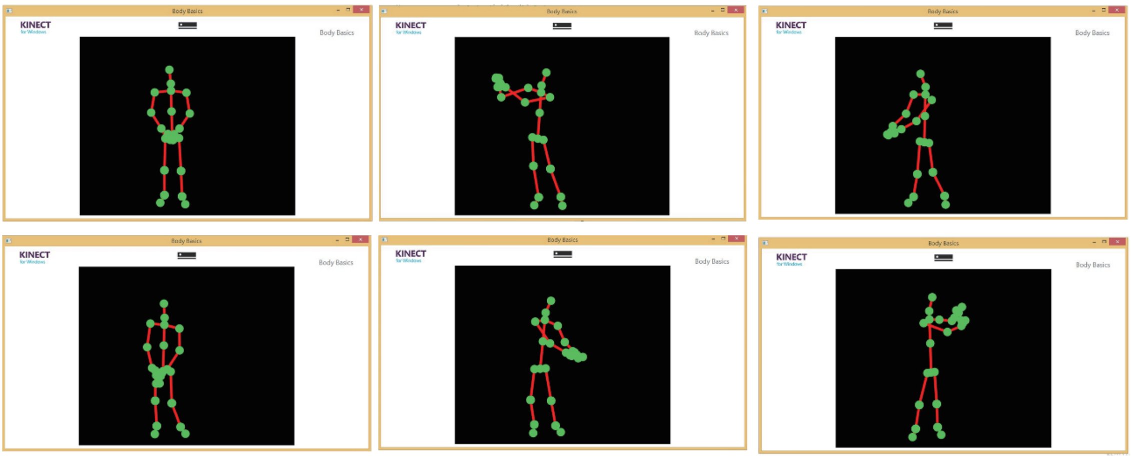

Fig. 6. (Color online) A golf swing depicted by Kinect motion-sensing photography.

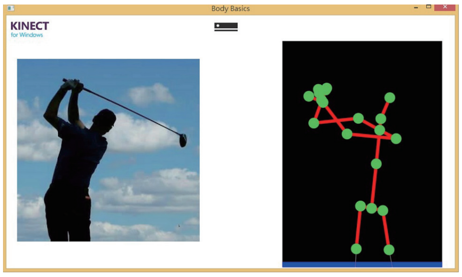

Fig. 7. (Color online) The implementation of a golf swing by Kinect.

photography. Furthermore, the action data detected for a golfer will be packaged in an XML format, which is easy to integrate with other services: the XML format is illustrated in Table 2.

Table 2
XML-based judgements of a golf swing.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Judgment>
  <Judgment JudgmentID="1"> <Judgment Rules="Eq01">
     <Eq>If((Y_shoulder_left-Y_shoulder_right)&gt;=TH01)</Eq>
     <Threshold>TH01</Threshold>
     <Y_shoulder_left>Y value of SHOULDER_LEFT</Y_shoulder_left>
     <Y_shoulder_right>Y value of SHOULDER_RIGHT</Y_shoulder_right>
     </Judgment> </Judgment>
  <Judgment JudgmentID="2"> <Judgment Rules="Eq02">
     <Eq>If((Distant_FR_FL-Distant_SR_SL)&gt;=TH02)</Eq>
     <Threshold>TH02</Threshold>
     <Distant_FR_FL>|X_foot_right-X_foot_left|</Distant_FR_FL>
     <Distant_SR_SL>|X_shoulder_right-X_shoulder_left|</Distant_SR_SL>
     <X_foot_right>The X value of FOOT_RIGHT</X_foot_right>
     <X_foot_left>The X value of FOOT_LEFT</X_foot_left>
     <X_shoulder_right>The X value of SHOULDER_RIGHT</X_shoulder_right>
     <X_shoulder_left>The X value of SHOULDER_LEFT</X_shoulder_left>
    </Judgment> </Judgment>
  <Judgment JudgmentID="3"> <Judgment Rules="Eq03">
     <Eq>If((Y_SR_SL&gt;=TH031)&amp;&amp;(X_SC_HC&gt;=TH032))</Eq>
     <Threshold>TH031</Threshold>
     <Threshold>TH032</Threshold>
     <Y_SR_SL>|Y_shoulder_right-Y_shoulder_left|</Y_SR_SL>
     <X_SC_HC>|X_shoulder_center-X_hip_center|</X_SC_HC>
     <Y_shoulder_right>The Y value of SHOULDER_RIGHT</Y_shoulder_right>
     <Y_shoulder_left>The Y value of SHOULDER_LEFT</Y_shoulder_left>
     <X_shoulder_center>The X value of SHOULDER_CENTER</X_shoulder_center>
     <X_hip_center>The X value of HIP_CENTER</X_hip_center>
    </Judgment> </Judgment>
  <Judgment JudgmentID="4"> <Judgment Rules="Eq04">
     <Eq>If((Y_SR_ER&lt;TH041)&amp;&amp;(Y_HR_ER&lt;TH042))</Eq>
     <Threshold>TH041</Threshold>
     <Threshold>TH042</Threshold>
     <Y_SR_ER>|Y_shoulder_right-Y_elbow_right|</Y_SR_ER>
     <Y_HR_ER>|Y_hand_right-Y_elbow_right|</Y_HR_ER>
     <Y_shoulder_right>The Y value of SHOULDER_RIGHT</Y_shoulder_right>
     <Y_elbow_right>The Y value of ELBOW_RIGHT</Y_elbow_right>
     <Y_hand_right>The Y value of HAND_RIGHT</Y_hand_right>
     <Y_elbow_right>The Y value of ELBOW_RIGHT</Y_elbow_right>
    </Judgment> </Judgment>
  <Judgment JudgmentID="5"> <Judgment Rules="Eq05">
     <Eq>If((Y_start-Y_end)&gt;=TH051)</Eq>
     <Threshold>TH051</Threshold>
     <Y_start>The starting point of backswing</Y_start>
     <Y_end>The top point of backswing</Y_end>
    </Judgment> </Judgment>
  <Judgment JudgmentID="6"> <Judgment Rules="Eq06">
     <Eq>If((Y_SC_HC)&lt;TH061)</Eq>
     <Threshold>TH061</Threshold>
     <Y_SC_HC>|Y_shoulder_center-Y_hip_center|</Y_SC_HC>
     <Y_shoulder_center>The Y value of SHOULDER_CENTER</Y_shoulder_center>
     <Y_hip_center>The Y value of HIP_CENTER</Y_hip_center>
    </Judgment> </Judgment>
</Judgment>
```

6. Conclusion

Golf is a challenging game, as it is difficult to strike the ball accurately so that it flies high and far in the right direction. Such misplay is likely to frustrate learning. Golf-specific exercise drills can be prescribed to a golfer to improve the swing efficiency. Golf swing analysis is a popular research subject, and it is divided into contact sensing and non-contact sensing, according to different research methods. This study aims to construct a teaching system for correcting a swing using non-contact sensing Kinect motion-sensing photo equipment. The proposed system is expected to collect the golfer's pose and action during the backswing for data analysis and to detect incorrect backswing actions to attain the goal of interactive golf teaching.

Before the swing, the golfer stands within an appropriate range of the Kinect action sensing terminal. The action sensing end is in charge of capturing the color image and depth image from the visual range and transmitting the two data streams via USB interface to the Kinect processing element of the central motion information processing end. When the golfer enters the Kinect action sensing end range, the skeleton joint data are transmitted to the Kinect processing element. The Kinect processing element confirms that the golfer's joints are tracked, and the skeleton joint data are sent to the logical processing unit.

Developed to utilize the motion capture advantages of the Kinect, we can use the data coordinates from the Kinect to determine a golfer's exact body measurements and posture. The Kinect motion-sensing system creates an accurate animation of the swing that can be viewed from any angle, as well as exact data on anybody segment throughout the swing. The golfer may use this feedback to enhance the feedback for the movement and learning of the skill is accelerated. This information provides one efficient way for golfers to swing based on what they can do physically. Using a Kinect motion-sensing system can determine a golfer's sequence of movement through a swing.

Acknowledgements

The Ministry of Science and Technology of the R.O.C. under the grant MOST 104-2221-E-146 -002 supports this research.

References

1 Y. C. Chung: A Study of User Interface and Form Design on Golf Dynamic Analysis System, Master's Thesis, Department of Industrial Design, Tatung University Taiwan (2007).

2 C. H. Ay, Y. L. Lai, and H. S. Su: J. Phys. Educ. Fu Jen Catholic University **11** (2006) 65.

3 M. A. J. Cooper and J. S. B. Mather: Proc. World Sci. Congr. Golf (1994) pp. 65–70.

4 M. G. Reyes and A. Mittendorf: Proc. World Scientific Congr. Golf (1999) pp. 13–19.

5 C. H. Wu, Y. F. Pao, and F. C. Chang: Golf Sci. **5** (2008) 68.

6 H. Ghasemzadeh, V. Loseu, and R. Jafari: J. Ambient Intell. Smart Environ. **1** (2009) 173.

7 H. Ghasemzadeh, V. Loseu, E. Guenterberg, and R. Jafari: Proc. 4th Int. Conf. Body Area Netw. (2009) pp. 1–8.

8 M. Pink, P. Jacquelin, and F.W. Jobe: Am. J. Sports Med. **21** (1993) 358.

9 J. M. Cooper, B. T. Bates, J. Bedi, and J. Scheuchenzuber: J. Biomech. **4** (1974) 298.

10 N. Faldo and R. Simmon: A Swing For Life (New York: Penguin Books, 1997).

11 P. Foston: The Encyclopedia of Golf Techniques (London: Courage Books, 1996).

12 Y. Tu , C. L. Zeng , C. H. Yeh , S. Y. Huang, T. X. Cheng, and M. Ouhyoung: 24th IPPR Conf. Comput. Vision Graphics Image Process. (2011) 8.

13 K. K. Wu: Using Kinect Motion-Sensing Camera for Human Action Recognition by Human Skeleton, Master's Thesis, Graduate Institute of Computer Science and Information Engineering, National Taipei University of Technology (2011).
14 S. Chun, D. Kang, H. R. Choi, A. Park, K. K. Lee, and J. Kim: Multimed. Tools Appl. **72** (2014) 253.
15 C. Jung: Measuring Movement of Golfers with an Accelerometer, Master's Thesis, ICT of KTH Stockholm, Sweden (2014).
16 S. Stančin and S. Tomažič: Sensors **13** (2013) 7505.
17 A. Blake and C. Grundy: Proc. 30th Int. Conf. Inf. Technol. Interfaces (2008) pp. 409–414.
18 M. Somjarod, V. Tanawat, and L. Weerawat: Int. Jo. Social Behav. Educ. Econ. Bus. Ind. Eng. **5** (2011) 545.
19 I. Karliga and J. N. Hwang: Proc. 2006 IEEE Int. Conf. Acoustics Speech Signal Process. (2006) 5.
20 N. Gehrig, V. Lepetit, and P. Fua: Br. Mach. Vision Conf. (2003) pp. 1–10.

About the Authors

Hao-Hsiang Ku received the B.S. degree from the Department of Management Information Systems, Chung-Hua University, Hsinchu City, Taiwan, in June 2001, the M.S. degree from the Department of Management Information Systems, National Pingtung University of Science and Technology, Pingtung City, Taiwan, in June 2003, and the Ph.D. degree from the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan City, Taiwan, in June 2009. He is currently an Associate Professor of Computer Science and Information Engineering, Hwa Hsia University of Technology, New Taipei City, Taiwan. His research interests include medical information systems, wireless sensor networks, and embedded multimedia applications.

Wen-Cheng Wang received the B.S. degree in business management from the Oxford College, Taiwan, in 1981 and the B.S. degree in economic form Ming Chuan University, Taipei, Taiwan, in 1984, and the Master degree in political economic from National Cheng Kung University, Tainan, Taiwan, in 1996. From 2005 to 2009, he was an Assistant Professor of Business Administration Department in Hwa Hsia University of Technology. Since 2010. His current research activities include Virtual Reality and Augmented Reality, Human-Computer Interface, Hotel Management and Tourism Management.

Yen-Wu Ti received the B.S. degree from the Department of Mathematics, TamKang University, New Taipei City, Taiwan, in June 1996, the M.S. degree from the Department of Applied Mathematics, National Chiao-Tung University, Hsinchu, Taiwan, in June 1998, and the Ph.D. degree from the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, in June 2009. He is currently an Assistant Professor of Computer Science and Information Engineering, Hwa Hsia University of Technology, New Taipei City, Taiwan. His research interests include medical information systems and image processing.