iCoach – Interactive Coaching System on Altera DE2-115 FPGA Architecture

Hsin-Yu Lai*, Guan-Lin Chao*, Meng-Ting Zhong*, and Shao-Yi Chien

Department of Electrical Engineering, National Taiwan University

b99901087@ntu.edu.tw; b99901164@ntu.edu.tw;

b99504066@ntu.edu.tw; sychien@cc.ee.ntu.edu.tw

*These authors contributed equally to this paper

Abstract—**In this paper, we present a new interactive, 2-player coaching system – iCoach, based on Altera DE2-115 FPGA board. The coach can be the 2nd player or any video stream provided by the user. With beautiful virtual background image and different background music choices, he/she can enjoy exercising with the coach and get real-time scores and comments on the performance of their overall motion, thus improving their motor skills. Also, iCoach provides interactive motion games for users to have fun dancing and painting with body parts. In comparison with existing motor learning devises such as Wii Fit, iCoach is more flexible since it enables users to set up their own coaches and virtual environments. Moreover, iCoach is more computationally efficient and uses less memory due to the carefully designed algorithms and the arrangement of FPGA memory.**

Keywords— FPGA; Dance; Coach; Motion-sensing; Interactive game; Rehabilitation

I. Introduction

Motor skills are crucial in all phases of lifespan. People of all ages perform fundamental motor skills such as walking and grabbing, or specific skills such as playing balls and dancing. Therefore, motor learning becomes essential in various areas such as rehabilitation and sports training. In addition, motor learning is found in many studies to have emotional and therapeutic advantages [1], [2]. Since the measurements of motor skills and knowledge of performance are critical variables in motor learning [3]-[5], a device that can measure users' motor skills and will give users feedback to help them improve motor learning is very helpful in daily life. iCoach is aimed at realizing a dance skill learning system.

To realize a motor-learning device, key issues include video processing (for coaches' video), motion detection, tracking, and a good comparison algorithm (for users' images). With respect to video processing, FPGA technology has been proposed as a way of obtaining high performance for computationally intensive DSP application [6], [7]. Moreover, among plenty of approaches for motion detection and object tracking proposed recently [8], [9], many have been implemented by using FPGA technology [10], [11]. Some related work even points out the advantage of structural design techniques over high-level design (e.g. C-like syntax) [12]. As a result, we design our motor learning device with structural design techniques by using FPGA technology, which can make the device much smaller and faster. Additionally, structural design also enables highly efficient use of memory.

One of the most well-known devices for motor learning is Wii Fit. It is a game that features yoga, strength training, aerobics, and balance games. The player is required to stand on a specific platform during exercise. Although it receives some positive reviews [13], Wii Fit lacks the flexibility to design the motion by users themselves and thus the trainings will be limited. Since in iCoach, our design, we enable users to save their coaches' videos and provide users with the performance feedback of each limb, users are able to learn nearly all movements and sports. Besides, iCoach extracts the background, which not only supplies users with a virtual environment that is beneficial to motor learning and rehabilitation [14], but also reduces the noise from the surroundings (which may lead to error motion-sensing). Furthermore, repeat mode, virtual mode, and 2-P mode — these learning modes offered in iCoach give users more choices and more fun, and the last two of these are not provided in Wii Fit.

In addition, we carefully design our algorithms to enhance our system efficiency. To achieve motor skill training, the device has to recognize the positions of the human body parts and then to evaluate the user's performance. The body part recognition can be done by tracking an object throughout video streams or directly recognizing the object in a single frame, and we apply the latter method. Tracking the user's body parts using algorithms like TLD(Tracking-learning detection) [15] requires very high memory usage since the algorithm takes a lot of images of the target object from various angles, let alone we have to track "five" body parts at the same time, which is nearly impossible to be realized on a FPGA board. In contrast, our algorithms identify in every single frame the five important body parts in real time based on the RGB values and coordinate, which is memory-saving with high efficiency of computation.

After identifying the posture of the user, iCoach compares it with that of the coach, who can be another player or a coach from video streams provided by the user himself/herself. SDPAM is suitable for saving video streams due to its capacity and speed to catch up with the playing of the video. iCoach then gives scores to body parts in every frame and an overall feedback comment on the user's whole motion, which identifies the weaknesses of the user's motion and helps to improve the user's skills. In addition to learning mode, which can be applied on exercise and rehabilitation, iCoach also provides a speed movement game and a body art game for the user to move his/her body parts to enjoy interactive motion sensing games and have fun.

The remainder of this paper is organized as follows. In Sec. II, we discuss the interface and each function of iCoach. In Sec. III, we introduce its environment and components. In Sec. IV, we describe our system architecture and in Sec. V we elaborate the algorithms further.

II. Function Description

In this section, we discuss the user interface and functions of iCoach.

A. *Data Pre-process*

1) Images set up: The user can save the coaches' videos by saving videos as a series of images in a SD card and inserting it into the FPGA board which connects to the VGA that shows the coaches' images. When iCoach is turned on, it spends some time processing and saving these images in the SDRAM of that FPGA board. The ratio of completion will be shown on the screen as in Fig. 1.

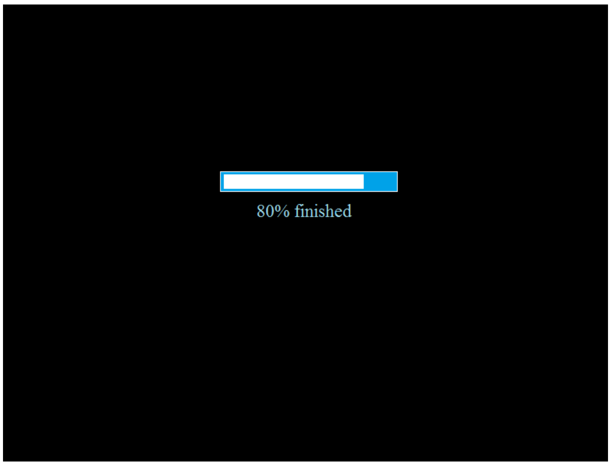


Fig. 1 Data Pre-process Interface

2) Music set up: The user can save the music in the Flash memory before turning on iCoach. As soon as iCoach finishes pre-processing and enters the user interface, it starts to play the music in the Flash memory.

B. *User Interface*

After the pre-processing finishes, user interface displays on the screen, as shown in Fig. 2.

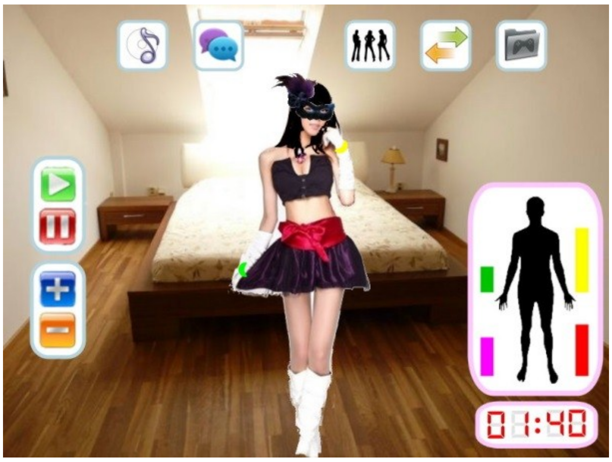


Fig. 2 User Interface

1) Dressing Requirement: The user is required to wear a yellow ring on his/her right wrist and a green ring on the left, as shown in Fig. 3. Since the screen mirrors the user's image, the relative positions of the rings should be the same as those shown in Fig. 2. Moreover, the user should let part of his/her both legs without covered by any cloth. The user also should have black hair, or the user should wear something black on his or her head.

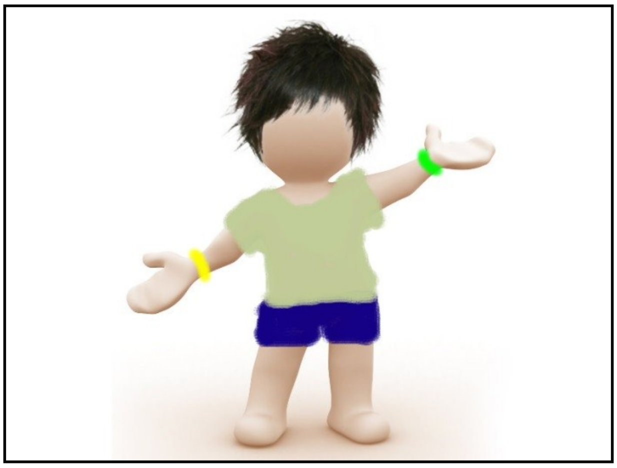


Fig. 3 User Dressing Requirement

2) Comparison Result: The score of each limb is shown in the silhouette of a man on the right side of the user interface. The scores are ranging between 0 to 100 points and are transformed to the length of the bars beside the four limbs. Limbs and their associated score bars are shown as in Table 1.

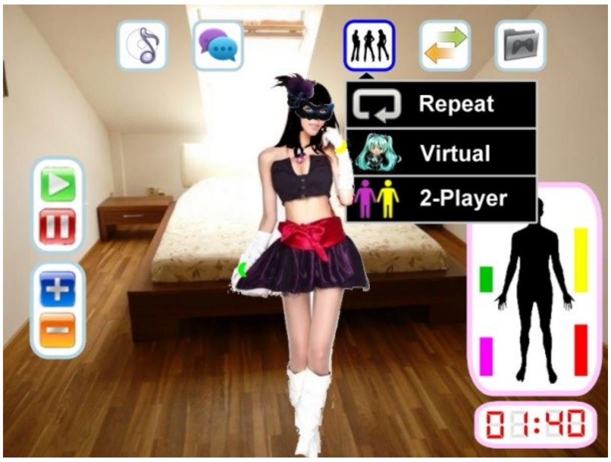
TABLE I

LIMBS AND SCORE BAR ASSOCIATION

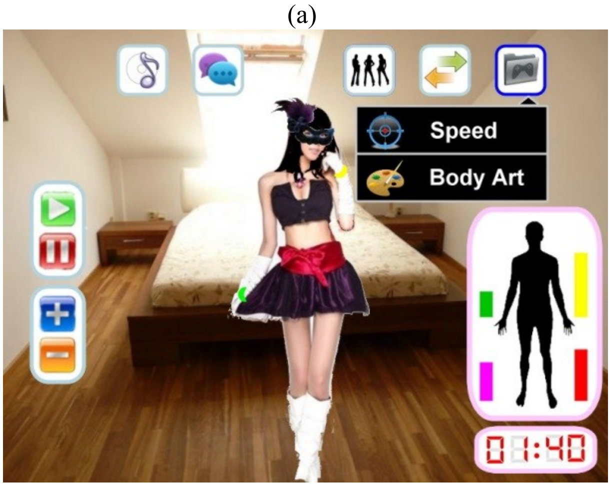
Limb	Associated score bar
right hand	yellow bar
left hand	green bar
right foot	red bar
left foot	magenta bar

3) Mode Button: By pressing the mode button, the user can switch the system's mode between learning mode and game mode. The details of these two modes are mentioned later in this section.

4) Learning and Game Button: In learning mode, the user has three options: repeat mode, virtual mode, and 2-P mode. In game mode, the user has two options: speed game and body art game. The function of these choices is mentioned later in this section.



(a)



(b)

Fig. 4 Learning and Game Button Interface (a) Learning Mode (b) Game Mode

5) Feedback Button: iCoach provides two kinds of feedback to help the user know his/her performance of learning and help the user improve. By pressing feedback button, the user can choose either "stop and start" or "concluded message" as the feedback mode.

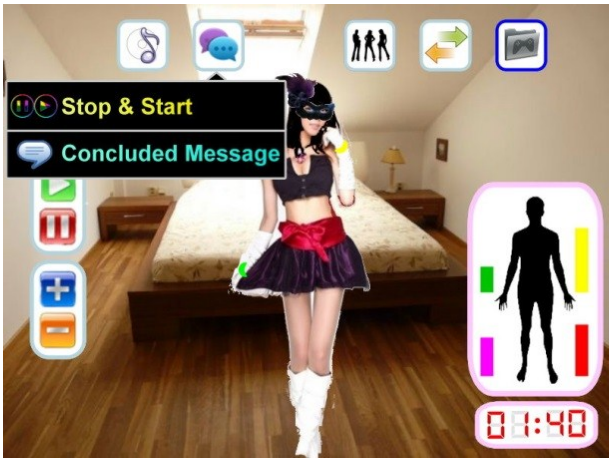


Fig.5 Feedback Button Interface

6) Coach Button: If the user wants to change the current coach, he/she can press this button.

7) Music Button: The user can also change music by pressing this button.

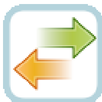
8) Play and Pause Button: The user can stop practicing at any time by pressing the play and pause button. The user can press that button again and the system will be resumed.

9) Clock: A clock is shown on the lower right of the screen to let the user knows how long the user has been learning.

The button icons are listed in Table 2.

TABLE II

BUTTON ICONS

Mode Button	Learning Button	Game Button
		
Feedback Button	Music Button	
		
Coach Button	Play and Pause Button	
		

C. Learning Mode

1) Repeat Mode: The user can replay a part of the video several times to enable his/her to get familiar with a particular movement. In this mode, a segment of the video would be played again once it is finished unless the user changes the current segment by pressing the "coach" button.

Fig. 6 Repeat Mode Screen on 2nd VGA

2) Virtual Mode: The user can also choose virtual learning mode to get trained from a virtual coach. In this mode, iCoach only need to obtain the positions of the limbs of the real coach in advance, the real coach's appearance is not needed. Thus, iCoach could save a longer period of motion. iCoach then uses the positions to enable the virtual coach to act in the same way as the real coach does.

Fig. 7 Virtual Mode Screen on 2nd VGA

3) 2-P Mode: In 2-P mode, the user can compare his/her motion with the second user. Two users are required to stand in front of two separated cameras.

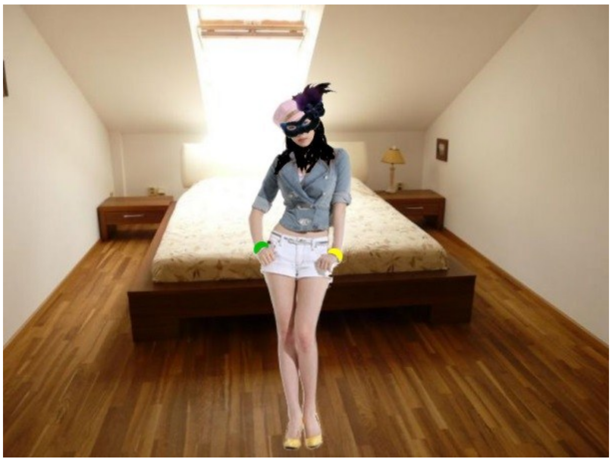


Fig. 8 2-P Mode on 2nd VGA

D. Learning Feedback

iCoach not only shows the current comparison result, it also provides users with two feedback choices.

1) Stop and Start: If the user's motion is far different from that of the coach (similarity is smaller than 60%) and the user chooses the "Stop and Start" feedback, iCoach would stop right away until the user's motion and that of the coach are similar enough.

2) *Concluded Message:* If the user chooses "concluded massage" feedback, iCoach will not stop even when the user could not catch up with the coach. Users can use this feedback mode to get comments on his/her whole motion. Messages of how well the user learns will jump out on the screen only when the whole motion is finished, as shown in Fig. 9.

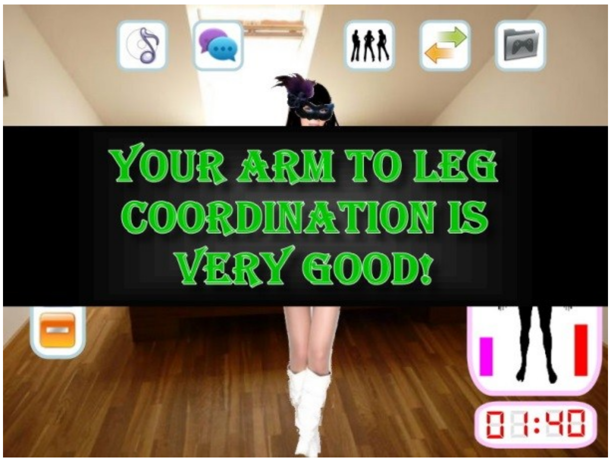


Fig. 9 Concluded Message

E. Game Mode

1) *Speed Game:* In this game, multiple colors of balls will appear in the screen. The balls would change colors and

positions once the user move his/her corresponding limb to touch the colored ball (the association is the same as that mentioned in the user interface subsection) on the screen. Users need to speed up themselves to touch balls as quickly as possible. The number of balls that have appeared would be shown in the upper corner of the screen.

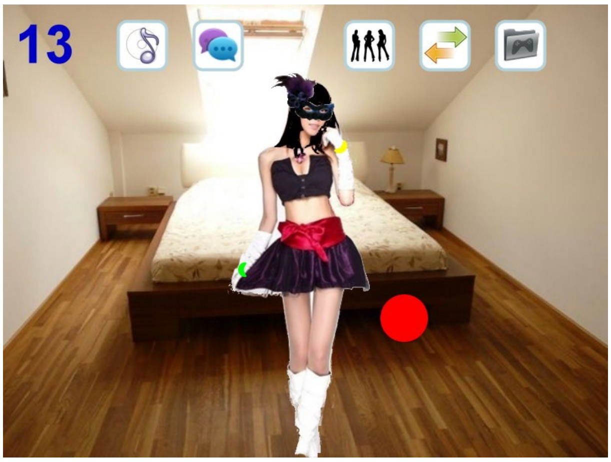


Fig. 10 Speed Game Screen

2) *Body Art Game:* In this game, the user can move his or her limbs to paint a picture. Since different limbs are associated with different colors, the user can use his/her limbs to complete a colorful picture and enjoy painting without paint brushes!

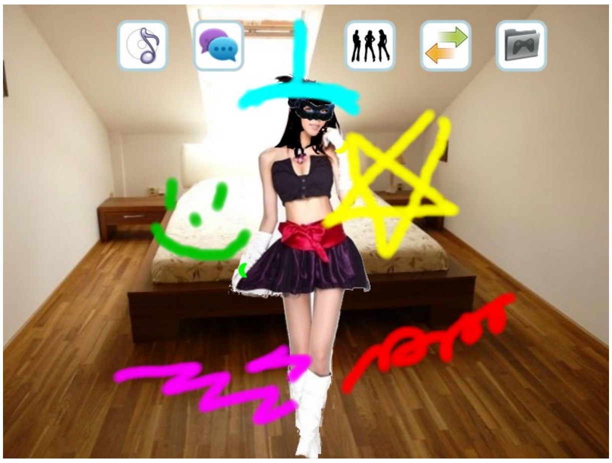


Fig. 11 Body Art Game Screen

III. Environment and Component

iCoach system is designed on EP4CE115F29C7 device in Altera Cyclone IV E family. It runs on FPGA built on two DE2-115 boards and the compiler we use is Quartus II 13.0 (32-bit). Two DE2-115 boards are connected with flexible flat cables inserted into J2 and J3 pins on THDB-HTG boards at each side. Two 500-megapixels D5M cameras with 2560×2160 full-resolution are needed to capture the motions and two VGAs are used to display the coach and the user respectively. Besides, a speaker is connected to one of the DE2-115 board to play music and an SDcard filled with videos is inserted into another DE2-115 board. In order to make our system more robust, sometimes we need to adjust illumination in the environment when illumination is too bright or too dark to detect colors of an object.

IV. System Architecture

In this section, we show data flow diagram, system block diagram and state diagram and discuss the data flow in detail.

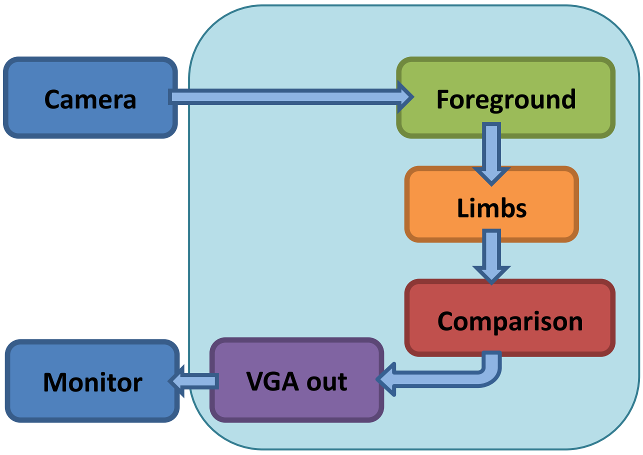


Fig. 12 Data Flow Diagram

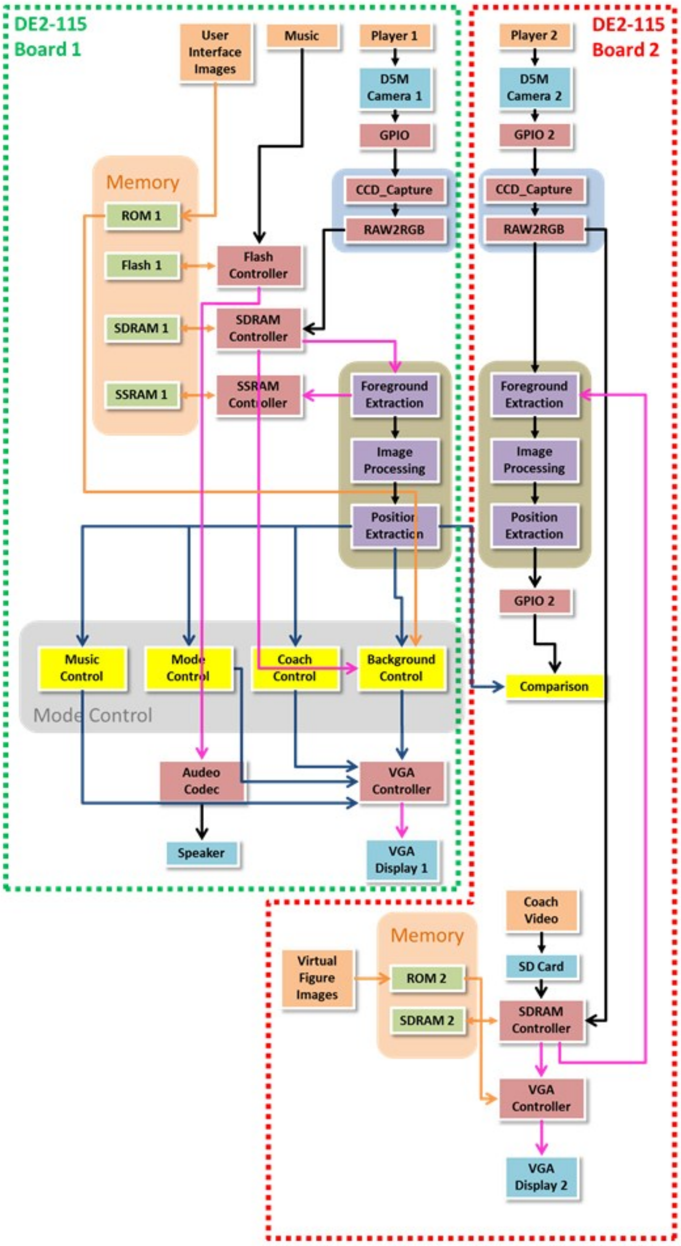


Fig. 13 System Block Diagram

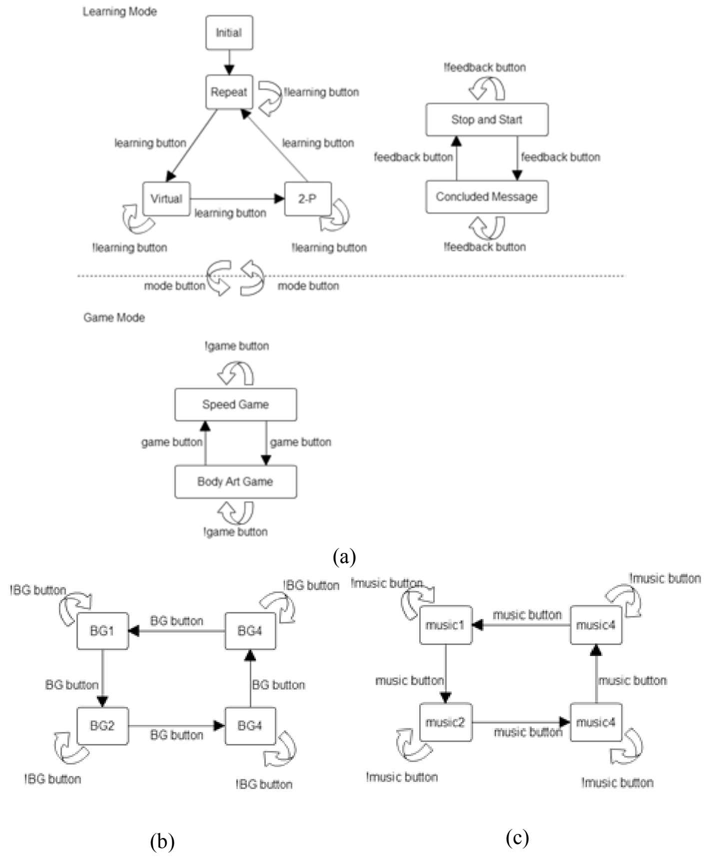


Fig. 14 State Diagram: (a) Mode Switch (b) Background Switch (c) Music Switch

A. Foreground Extraction

To avoid the interference from the background, we first differentiate the user's figure and the background. The differentiation algorithm is described in Section V. After extracting the foreground (the user's figure), we then detect the positions of the user's head and limbs.

B. Limbs Detection

Since we require the user to wear a yellow ring on his/her right wrist and a green one on the other, iCoach can identify the positions of the wrists by detecting the corresponding colors in the foreground. Moreover, iCoach detects the position of head by locating the black color. To obtain the positions of legs, iCoach will detect the skin colors in the foreground and will consider the skin color found in the lower left to be the left leg; the skin color found in the lower right will be considered to be the right leg.

C. Comparison

After knowing all the positions of head and limbs, iCoach compares the obtained positions with the coach's head and limbs positions. In order to help the user know the similarity, iCoach will give the user four scores related to the user's limbs. The comparison criteria will be elaborated in Section V. If the score of any limb is lower than the threshold (60 points), in stop and start mode, the coach video pauses until the user's four limb scores are higher than the threshold. In the "concluded message" feedback mode, the system not only generates a score, but shows comments on the overall performance of the user.

D. VGA out

iCoach will continue doing comparison once it enters the user interface. The system will send the user's image and the coach's image to two separated screens and will also show scores on the screen where the user's image is located. The buttons' images mentioned in Section III will also send to the screen from the ROM memory.

V. Algorithms

A. Camera Image Posture Recognition

1) Camera Image Capture and Transformation from Raw to RGB Image: CCD_Capture module and RAW2RGB module take D5M camera's raw data and transform them to RGB channels respectively.

2) Foreground Extraction: iCoach uses background registration technique [16] to segment a person from the frame. If the difference between current frame and previous frame is below the threshold value for several consecutive frames, background is registered and will be updated over time. After measuring the distance between current frame and background, iCoach regards the pixels with distance above another threshold value as pixels of initial object mask. Finally, post processing is performed and foreground is extracted.

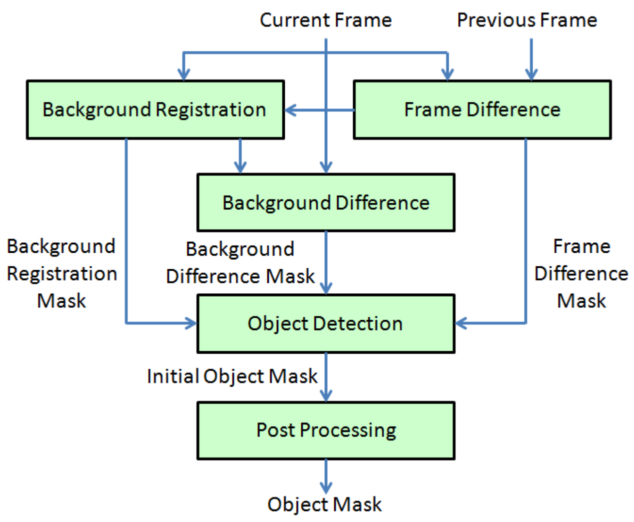


Fig. 15 Block Diagram of Foreground Extraction Algorithm

3) Significant Pixels Extraction: We identify the pixels in some specific color and coordinate ranges as the probable head and limbs pixels and marked them with special tags, and the rest pixels outside the color range are viewed as unimportant, marked with another tag, and are not processed in the following image processing procedures.

Take head for example, we take pixels whose color ranges in {R < 200, G < 200, B < 200} (which is the color range of black hair), coordinate ranges in {200 < x < 400, y < 200} (in 800*600 display) as head pixels and marked them with white tags.

It is noted that we wear a yellow ring on the right wrist and a green ring on the left wrist. We assume that the user wear black hair and shorts since we take black pixels at the top middle part of the screen as head position, and skin color pixels at the bottom part of the screen as foot positions.

The color and coordinate ranges of head and limbs are listed below.

Table. II

Significant Pixels Definition and Associated Tags

body part	color range (R,G, and B ranges from 0 to 1023)	coordinate range (x ranges from 0 to 800, y from 0 to 600)	tag
head	R < 200, G < 200, B < 200	200 < x < 400, y < 200	white
right hand	R - B > 100, G - B > 100	100 < x < 800, y <300	yellow
left hand	R – G > 800, B < 400	x < 500, y < 300	green
right foot	80 < R – G < 350	300 < x < 600, 300 < y	red
left foot	60 < R – G < 450	10 < x < 300, 300 < y	magenta
other			black

4) *Elimination of Noise in Background (Erosion):* Each pixel is compared with its adjacent eight pixels, shown in Fig. 16. If all the nine pixels have the same tag, then this pixel is claimed as its original tag, or it is marked with unimportant (black) tag. The reason why we use this scheme to eliminate the noise in background is that we assume the body part pixels tend to gather, which means the pixels associated with the same body part are located in clusters and have the same tag with their neighborhood. This process is adapted from a morphological image processing skill – erosion. In our implementation, we apply erosion twice and eliminate the most noises from the background.

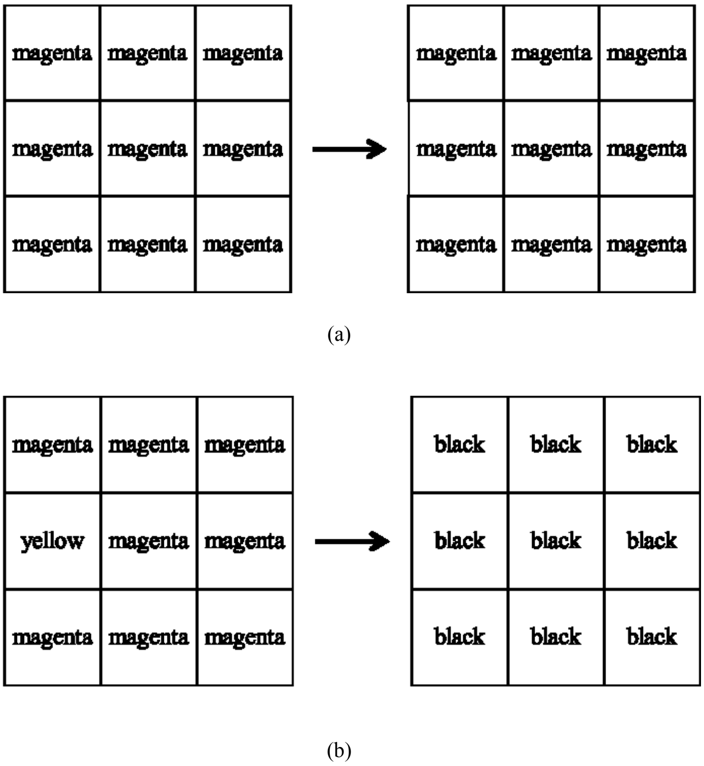


Fig. 16 Erosion Examples

5) *Position Extraction:* After elimination of noise from the original camera image, we can compute the average coordinate of head and limbs more precisely. Take head for example, we identify pixels with white tags as head pixels, compute the average coordinate of these pixels, and get the central coordinate of the head!

To make the center of a specific body part more stable and attenuate the interference of background noise, we limit that the consecutive coordinates of a specific body part should not exceed 120 pixels since the movement of each body part is continuous and that the user should not move his/her limbs too violently.

6) Button: The user can place his/her hand on a button on the user interface. If the hand stays on the range of the button in 60 consecutive frames, the module takes it as a click on the button, thus switching between different modes.

It is noted that staying on the button range for only 60 frames continuously makes the button very sensitive. The user might "click" the button two or three times with only a short while of stay on the button.

B. Mode/States Design

Mode switch and control is done by moving user's hand and clicking the button.

1) Learning: The three learning modes – repeat, virtual, and 2-P modes are switched by clicking the learning button. In the repeat mode, the system plays a specific coach video and replays it repeatedly. After each play, the system displays a score and comments to evaluate the user's latest performance. In the virtual mode, the system transforms the coach's video to a virtual character's posture. We've recorded the significant joint coordinates previously, and generated a virtual figure by connecting the significant joints with its head and limbs images. By doing so, we only have to keep the original coach videos and some static images of virtual figure's body parts in memory, which is more efficient than keeping all these videos. 2-P mode is implemented by connecting two DE2-115 FPGA boards with GPIO interface. One camera captures the images of the user, and the other captures those of the coach (which is the player 2). Then the two boards compute the head and limbs coordinates of two players respectively, and GPIO interface transmits player 2's coordinates to the DE2-115 FPGA board associated to the user, and we can compare the two coordinates and evaluate the user's performance.

2) Game: There are two games for the user to choose from – speed game and body art game. In speed game mode, the system generates the target ball color and its position (in accordance to the coordinate range of that part) randomly and transmits the generated target ball coordinate to VGA controller. If the head or limb moves near the ball, the system considers that it "touched" the ball and then generates the next one. In the body art game, the user can move his/her head and limbs. The RGB value of pixels on the trajectories of a specific body part is transformed from the original value (captured by the camera) to the body part's painting color. The table of painting colors and their associated body parts is as follow.

TABLE III

BODY PARTS AND ASSOCIATED PAINTING COLORS

Body Part	Painting Color
Head	White
Right hand	Yellow
Left hand	Green
Right foot	Red
Left foor	Magenta

3) Music: As the user clicks the music button, the value of music switch plus one. Each music switch value is associated with a flash memory address in which the associated music data stored in flash memory start. And the Audio Codec plays the music according to the address.

4) Coach: We maintain a value to record the clicks from the user on the coach button. If the user pushes the coach button once, the value increases by one. This value represents which coach the user is learning. And *VGA_Controller* module in the DE2-115 FPGA board associated to the coach takes the coach number as input and displays the certain coach on the second screen.

5) User-Defined Background: Since we have done foreground extraction previously, we can replace the real background with other beautiful pictures at the user's preference; it is done by assigning picture RGB values which are saved in ROM memory to the background pixels.

Fig.17 User-Defined Background

C. Comparison

The *compare* module takes 10 position coordinates as input - user's and coach's head and limbs, and outputs the scores of them respectively.

It transforms the positions of the five body parts into relative positions, including distances and angles, normalizes them with the user's and coach's height, compares the relative coordinates.

The scores are graded in 0 ~ 100 points. The angle part takes 80% and the distance part 20%.

It also generates an overall comment on the performance of user's whole motion, which is drawn from a comment library collections defined by ourselves.

D. Multimedia Import and Display

1) Video Import (SD card to SDRAM): Before we can get the image data from the SD card, the host will first stay in the card identification mode until it receives SEND_RCA command (CMD3) [17]. The state diagram of card identification mode is shown in Fig. 18.

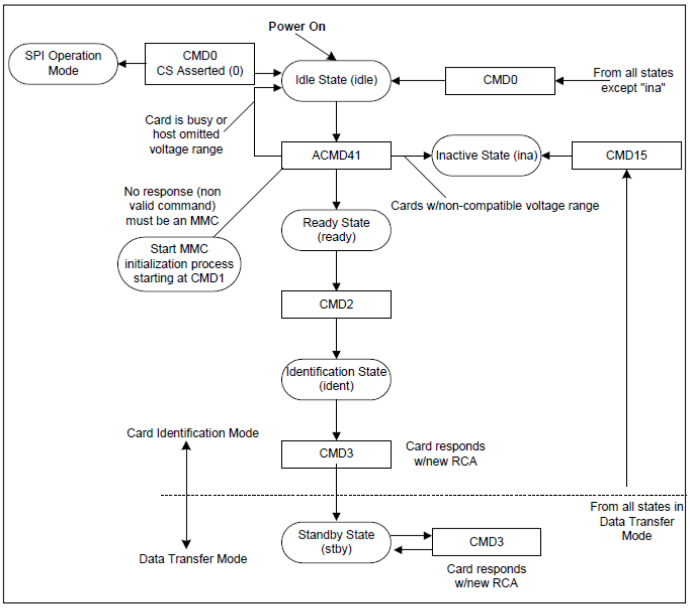


Fig. 18 Card Identification Mode State Diagram

To realize this state diagram, we save CMD sequence in ROM memory and send the MD one by one to the SD card in the same order as shown in Fig. 18. Since SD card sometimes will be busy and will not respond even when we send ACMD41, we repeat sending ACMD41 for 20 times. Then we send CMD2 and CMD3. After SD card receives SEND_RCA command (CMD3), the host will enter data-transfer mode, which is shown in Fig. 19.

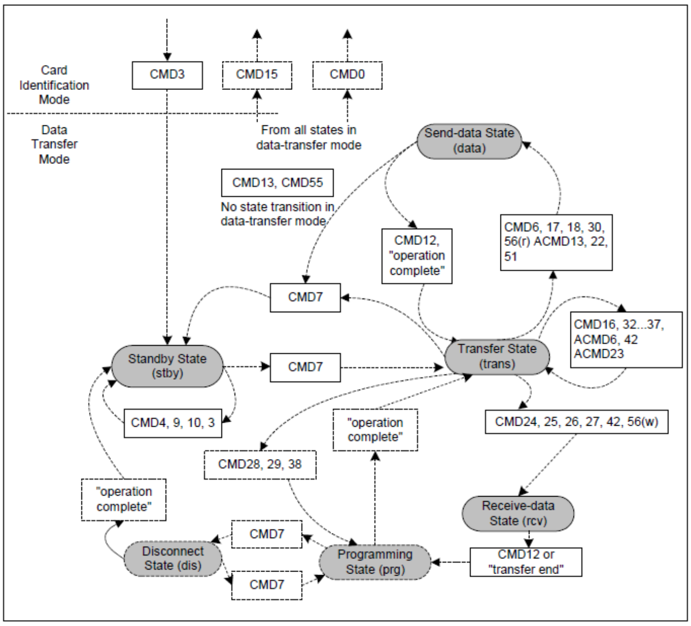


Fig. 19 Data Transfer Mode State Diagram

After the host enters the data transfer mode, it would stay at Stand-by State. We then specify SD card by sending CMD9. We then send CMD10 with SD card's own ID number and send CMD7 to enable the host to enter Transfer State. After sending CMD18, we can then receive the image data from the SD card. These data will be saved in the SDRAM memory.

2) Background Music Import (flash to SRAM): Music should be saved as Windows PCM Wave forms and the header of it should also be removed. The treated music should be saved in the flash memory on the FPGA board using the control panel before turning on iCoach. While iCoach is running, music will be sent from the flash memory to audio codec and will be played by the speaker.

3) VGA Display: After capturing, the RGB data from RAW2RGB module will be sent into *Mirror_col* module. Since the order of the data is contrary to that shown on VGA display, *Mirror_col* module saves the data in the ROM buffer and later the data will be output in a reversed order. On the other side, *VGA_Controller* module displays user interface image data from ROM. This module also takes camera RGB data, coordinates of head and limbs, scores of limbs, current state of each button, and the specific body part associated to the target ball and its coordinate (which is used in the speed game mode) as input. Then it outputs RGB data and coordinate of each pixel. Finally, the VGA display receives and shows the RGB data.

4) Music Display: The size of flash memory is 8MB and iCoach is capable of playing four kinds of music, so we divided flash memory into four 2MB sections. There are idle state, waiting state, starting state in our design and each kind of music has its own starting address and ending address. When music button is pressed, flash_music module receives input signal and sends corresponding DACDAT to audio codec to play the music.

VI. Conclusion

In this paper, a new framework of an interactive coach – iCoach, based on Altera DE2-115 FPGA board is presented. By multiple stages of image processing, the system extracts the body part positions of the user and the coach, compares them by our evaluation algorithm, and generates feedbacks of user's performance on VGA display to help him/her improve. With the use of two DE2-115 FPGA boards, iCoach is a 2-player system, one player can teach the other one to dance. With simple movements of arms, the user can press the buttons shown on the user interface, thus switching several functions: multiple coaches, background music, learning modes and gaming modes at will. With these various choices, iCoach provides users with brand-new and highly flexible exercising experiences. Due to the time- and memory-efficient body part recognition algorithms employed in our system, the computational cost of image processing of iCoach is lower than normal motion sensing system.

References

[1] Krakauer, John W, "Motor learning: its relevance to stroke recovery and neurorehabilitation," Current opinion in neurology 19.1 (2006): 84-90.

[2] Sandel, Susan L., et al, "Dance and movement program improves quality-of-life measures in breast cancer survivors," Cancer nursing 28.4 (2005): 301-309.

[3] Voelcker-Rehage, Claudia, "Motor-skill learning in older adults—a review of studies on age-related differences," European Review of Aging and Physical Activity 5.1 (2008): 5-16.

[4] Newell, Karl Maxim, "Knowledge of results and motor learning," Journal of Motor Behavior, Vol 6(4), Dec 1974, 235-244.

[5] Shea, Charles H., and Gabriele Wulf, "Enhancing motor learning through external-focus instructions and feedback," Human Movement Science 18.4 (1999): 553-571.

[6] Benkrid, K., et al, "High level programming for real time FPGA based video processing," Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on. Vol. 6. IEEE, 2000.

[7] "IEEE Symposium on Field-Programmable Custom Machines," Proceedings of FCCM '98, Mariott at Napa Valley, Napa, CA. 1998.

[8] Cutler, Ross, and Larry S. Davis, "Robust real-time periodic motion detection, analysis, and applications," Pattern Analysis and Machine Intelligence, IEEE Transactions on 22.8 (2000): 781-796.

[9] Yilmaz, Alper, Omar Javed, and Mubarak Shah, "Object tracking: A survey,"Acm Computing Surveys (CSUR) 38.4 (2006): 13.

[10] Shi, Minghua, et al, "An efficient FPGA implementation of Gaussian mixture models-based classifier using distributed arithmetic," Electronics, Circuits and Systems, 2006. ICECS'06. 13th IEEE International Conference on. IEEE, 2006.

[11] Jaward, M., et al, "Multiple object tracking using particle filters," Aerospace Conference, 2006 IEEE. IEEE, 2006.

[12] Benkrid, Khaled, et al, "High level programming for FPGA based image and video processing using hardware skeletons," Field-Programmable Custom Computing Machines, 2001. FCCM'01. The 9th Annual IEEE Symposium on. IEEE, 2001.

[13] Dr. Ivor K. (2008) Nintendo Wii Fit for physiotherapy. [Online]. Available: http://stanford.wellsphere.com/general-medicine-article/nintendo-wii-fit-for-physiotherapy/460917

[14] Holden, M., and Emanuel Todorov, "Use of virtual environments in motor learning and rehabilitation," Handbook of virtual environments: Design, implementation, and applications (2002): 999-1026.

[15] Kalal, Zdenek, Krystian Mikolajczyk, and Jiri Matas, "Tracking-learning-detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34.7 (2012): 1409-1422.

[16] Shao-Yi Chien, Shyh-Yih Ma, and Liang-Gee Chen, "Efficient Moving Object Segmentation Algorithm Using Background Registration Technique," *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, VOL. 12, NO. 7, JULY 2002.

[17] *SanDisk SD Card Product Manual*, SanDisk Corporation, 2004.