World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:9, No:1, 2015

Interactive Shadow Play Animation System

Bo Wan, Xiu Wen, Lingling An, Xiaoling Ding

***Abstract*—**The paper describes a Chinese shadow play animation system based on Kinect. Users, without any professional training, can personally manipulate the shadow characters to finish a shadow play performance by their body actions and get a shadow play video through giving the record command to our system if they want. In our system, Kinect is responsible for capturing human movement and voice commands data. Gesture recognition module is used to control the change of the shadow play scenes. After packaging the data from Kinect and the recognition result from gesture recognition module, VRPN transmits them to the server-side. At last, the server-side uses the information to control the motion of shadow characters and video recording. This system not only achieves human-computer interaction, but also realizes the interaction between people. It brings an entertaining experience to users and easy to operate for all ages. Even more important is that the application background of Chinese shadow play embodies the protection of the art of shadow play animation.

***Keywords*—**Gesture recognition, Kinect, shadow play animation, VRPN.

I. Introduction

CHINESE shadow play is a traditional art with a long history and it widely spread in the folk ever. But now, the development of shadow play is extremely slow, because of the impaction of the modern culture and its own limitation such as the complicated and time-consuming process of making a traditional shadow play character, the high requirement of shadow play scene, performers' professional skills and so on. In consequence, fewer and fewer people are willing to throw themselves into this traditional art. It faces a great dilemma that people cannot handed it down smoothly, although the State Council has approved of shadow play being listed in the first batch of national intangible heritage list as a performing art.

In the modern society, science and technology are advancing by leaps and bounds, especially in the areas of computer science. Digitalized shadow play animation has become the new way to protect shadow play. It has its own advantages, for instance, simple process of making a shadow play character, low requirement of devices, convenient operations and no limit of performers' skills. Digitalized shadow play animation is concerned by more and more people and some relevant applications have already emerged.

In the paper, we propose a novel digital and interactive shadow play system, which consists of computer, Kinect and Internet. It provides an interactive interface, through which users can manipulate shadow play character by gestures. We use tree-structure to model the components of shadow character, Kinect to capture the performer's skeleton data, gesture recognition module to control changing of the preform scenes, VRPN to transmit the data from Kinect and display application to complete the edition, generation and exhibition of shadow play animation. Users also can record a video through our system.

Bo Wan is with the School of Computer Science and Technology, Xidian University, Xi'an, 710071, China (corresponding author to provide phone: +86 13571890190; fax: +86 029 88204472; e-mail: wanbo@xidian.edu.cn).

Xiu Wen, Lingling An, and Xiaoling Ding are with the School of Computer Science and Technology, Xidian University, Xi'an, 710071, China (e-mail: wenxiu1991@163.com, an.lingling@gmail.com, zgdingxiaoling@sina.com).

II. Related Work

A. Chinese Shadow Animation Based On Kinect

In [1], the authors developed an animation system made of hierarchical modeling, texture mapping and key-frames interpolating technology, which was used to edit shadow characters and play shadow animation. In [2], shadow characters were created in a semiautomatic process and rendered by a rendering technique based on the photon mapping method. This shadow animation is generated by interpolating between the key-frames. In [3], by analyzing the motions of traditional shadow characters and utilizing two-dimensional and three-dimensional software, the authors built the models of characters and hierarchical structure and generated the animation by key-frames interpolating technology. In [4] and [5], the authors applied motion planning algorithms to generate the motions of a character from an animator's high-level inputs automatically.

These systems mentioned above focus on how to edit shadow characters and generate shadow play animation by analyzing the motion of character. There is not a natural and interactive method to manipulate shadow play characters until the appearance of Kinect. On June 1, 2009, Microsoft released a motion sensing input device for the Xbox 360 video game console and Windows PCs named Kinect which enables users to control and interact with computers through a natural user interface using gestures and voice commands instead of touching game controllers. The device include an RGB camera, infrared laser projector, CMOS sensor, and multi-array microphone running proprietary software [6], which provide 3D motion capture, human recognition and voice recognition capabilities. The working distances of Kinect sensor range from 1.2m to 3.5m, while the motorized pivot enables the sensor track automatically. Microsoft released a noncommercial Kinect SDK for Windows 7 on June, 2011 in 12 countries. As the release of SDK and technical advantages of Kinect, lots of novel applications, so far, have been developed in the field of HCI (Human Computer Interaction), such as digital objects control. Luís ¬¬Leite et al. [7] found the potential of Kinect and proposed to use people's body as a puppetry controller, giving life to a virtual silhouette through acting in front of Kinect. Their method achieved a human-computer interactive, but not supply interactive between humans and not generate a

fully-functional system. Seth Hunter et al. [8] present a digital puppetry project that people give a puppet show in front of Kinect to control the digital puppetry in the screen. Robert et al. [9] proposed a 3D animation system that allows puppeteers load 3D models of some rigid objects, such as toys, into their system and create an animation. Bradley Wesson et al. [10] investigated how to facilitate artistic design by using body gestures instead complex craftsmanship in realty. Kam Lai et al. [11] proposed to apply the Kinect camera to close-range gesture recognition, which used feature vectors derived from the skeleton model provided by the Kinect SDK in real-time.

To this day, Kinect has not applied to the field of cultural heritage.

B. Gesture Recognition

In the field of virtual reality, developers tend to add the function of gesture recognition to their application for the purpose of increasing the interestingness and interactivity of the application. To date, the existing classic methods in gesture recognition are Hidden Markov Models (HMM) [14] and Dynamic Time Warping (DTW) identification method [15] [16]. However, both of them are complicated to process, difficult for developers to program and not too convenient to use. These two algorithms are usually used for recognition of complex dynamic gesture. For example, Karthik Nandakumar et al. [12] write a paper described multi-modal gesture recognition based on Kinect, chosing HMM as recognition algorithm and Paul Doliotis et al. [22] achieve an accurate gesture recognition system under the help of DTW. There, of course, are some other researchers studying other algorithm to satisfy the needs of their systems. Jaemin Soh et al. [23] manipulated 3D object by attracking the change of two hands positions. Tsai-Te Chu et al. [13] apply the least square method for the four direction recognition and a support vector machine library for the handwritten digit recognition.

For our system, we need a simply efficient recognizer to recognize single stroke dynamic gesture. Therefore, our system has adopted the directional correlated singe-stroke gesture recognition algorithm which is quite simple achievement, high accuracy, fast, convenient and strong to resist interference.

III. System Overview

Fig. 1 describes the physical architecture of our system. Kinect is used to track the gesture, motion and speech of performers and deliver the formatted data to computer through VRPN. In the network, another computer receives the data and uses these data to generate shadow play animation. The audience can see the shadow play animation through the projector connected to computer c3. This system, of course, can also just run at the local host and there is only one character on the scene as a consequence. When multiplayer involved in the case, each of them need a Kinect and a computer connected to network. Performers in their own computer can see perform scenes in real time.

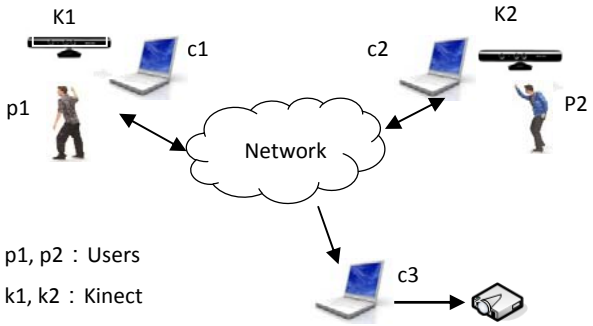

Fig. 1 Architecture of our shadow animation system

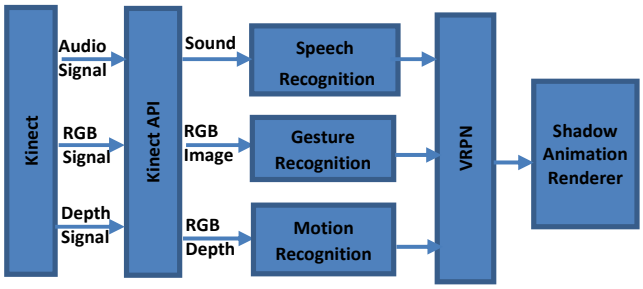

Fig. 2 Data flow between software components

The software of our system mainly consists of Kinect library, speech recognition module, gesture recognition module, motion recognition module, VRPN and shadow animation renderer. Fig. 2 illustrates the relationship between the components and the data flow in the system. When the client-side is running, Kinect start acquiring human-skeleton data and voice commands. In the meantime, gesture module is called when users' right hands get into the area that gestures can be recognized. The recognized gesture and human-skeleton data from Kinect is packaged by vrpn_Analog in VRPN. Speech commands are packaged by vrpn_Text_Sender. These data packaged are received by the server-side with vrpn_Analog_Remote and vrpn_Text_Receiver through the uniform interface, after the two sides connecting successfully. After all these done, renderer gets these data and renders shadow play character by OpenGL. Speech command is used to control the start and end of shadow play animation. As client-side updates data and offers these to server-side constantly, the system generates a coherent shadow play animation.

IV. Data Structure of Shadow Play Characters

Most of Chinese shadow play characters consist of 11 parts, including head, chest, hip, left upper arm, left lower arm, left hand, left leg, right upper arm, right lower arm, right hand, and right leg. According to the analysis of the structure of traditional shadow character, we define a tree-structure to describe the relationship of 11 parts, which divided into child object and parent object. Fig. 3 shows the hierarchy of shadow character and the relationship between hierarchy and characters. We add "Neck" in the tree-structure. It is a virtual node and the root of the tree. It determines the position of the

whole character, just as traditional shadow play.

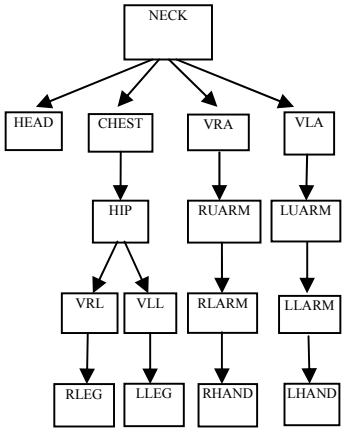

Fig. 3 Tree-structure of shadow character's components

According to the tree-structure, we built a script file to store the information of every part. Fig. 4 describes the file. This file uses the nested structure to describe the hierarchical relationship between each part. Each part consists of four keywords, including OFFSET, DOFS, FILE and JOINT. OFFSET represents child object's offset relative to its parent. DOFS represents the degree of freedom. FILE stores the name of texture file. JOINT is the name of component. OFFSET is a constant value and we can use it to locate the position of every part. This script file is an independent and editable file. By editing the script file, we can add other characters we need, for example, animals or/and other people.

```
ROOT Neck
{
    OFFSET 0.0 0.0 0.0
    DOFS 4
    FILE NULL
    JOINT Vra
    {
        OFFSET 0 0 0
        DOFS 0
        FILE NULL
        JOINT RightUpperArm
        {
            OFFSET 0.68 -0.75 0
            DOFS 1
            FILE right_upper_arm.3ds
            JOINT RightLowerArm
            {
                OFFSET 3.58 0 0
                DOFS 1
                FILE right_lower_arm.3ds
                JOINT RightHand
                {
                    ……
                }
            }
        }
    }
        ……
}
```

Fig 4 The script file use d to describe shadow character

V. Position Data Acquisition and Transmission

In traditional shadow play, one shadow character is controlled by three sticks. One stick is fixed at the neck and determines the movement of the whole body. The other two are fixed at hand respectively and control the motion of arms. Other parts, such as hip and legs, can move through controlling the three sticks skillfully. In order to inherit these features that shadow characters is flexible operation and optimize the operating approach, in our system, we can manipulate all parts of shadow character by using Kinect. Kinect can detect twenty joints per player and return these joints' position. By using the twenty positions, shadow character can imitate the motion of performers accurately on the screen.

A. Data Acquisition

In this paper, Kinect is a core part of the whole control system, including physical control section and speech control section. The following will describe the two parts sequentially.

Physical control section: Users control the shadow characters doing the same action by moving in front of the Kinect sensor. This function worked based on the skeleton-tracking system provided by Microsoft. The system transmitted 3D depth image to the skeletal tracking system. This system can simultaneously detect six individuals, of which two skeletons data would be tracked, which called active tracking. It also can provide location information of the other users at the same time, which called passive tracking. When the characters are standing mode, it can track 20 skeleton joints data [17], while sitting mode can be tracked 10 skeleton joints data.

In order to get the skeleton data, it's necessary to use NUI (Natural User Interface) API to obtain data flow. NUI API is the kernel API for Kinect provided by Microsoft. It supports basic image processing and device management capabilities. For programmers, we need to call the function from NUI API to get the corresponding image data and achieve control of Kinect equipment (Kinect official documentation).

Voice Control section: Activating or terminating the shadow animation system by identifying the corresponding speech commands. Kinect SDK provides Microsoft Speech library to support voice command recognition. What's more, the Kinect microphone array worked over some speech recognition libraries which include DirectX Media Objects (DMO). Developers can access the audio signal processor with DMO.

B. Data Transmission

VRPN (Virtual-Reality Peripheral Network) is a series of class libraries and a set of servers which realized a network-transparent interface between application programs and peripheral devices [18]. We use it to transmit data from the client to the server.

VRPN provides drivers for many peripheral devices. However, that's not the point. The importance of VRPN is providing interface to a set of functions. Particular devices are of one or more device types. Each device type specifies a consistent interface and semantics across devices implementing that function [19]. There are some common devices: Tracker, Button, Analog, Dial and ForceDevice.

By using the appropriate class-of-service and sharing the link, VRPN can implement connection between the applications and

the devices. In our system, we use the type of vrpn_Analog and vrpn_Text. At client-side, vrpn_Analog reports and updates the data of human-skeleton from Kinect and hand signals from gesture recognition module, and vrpn_Text_Sender reports voice commands. When server-side application runs and builds connection with client-side, vrpn_Analog_Remote and vrpn_Text_Receiver get corresponding data from client-side continuously. System uses these data to render shadow character or/and control the system. Fig. 5 describes the 20 skeleton joints data and gesture recognition result the server successfully received from the client through VRPN.

Position[NECK].x = 177.286 Position[S NECK].y = 146.781
Position[HEAD].x = 161.711 Position[HEAD].y = 138.27
Position[CHEST].x = 189.646 Position[CHEST].y = 183.761
Position[VLA].x = 125.737 Position[VLA].y = 209.822
Position[VRA].x = 191.445 Position[VRA].y = 157.226
Position[HIP].x = 191.305 Position[VRA].y = 191.443
Position[LUARM].x = 150.395 Position[LUARM].y = 164.567
Position[RUARM].x = 174.91 Position[RUARM].y = 89.570
Position[LLARM].x = 107.327 Position[LLARM].y = 93.876
Position[RLARM].x = 163.471 Position[RLARM].y = 169.857
Position[LHANE].x =93.005 Position[LHANE].y = 74.611
Position[RHAND]x = 161.222 Position[RHAND].y = 197.256
Position[VLL].x = 142.446 Position[VLL].y = 330.281
Position[VRL].x = 130.109 Position[VRL].y = 437.259
Position[LLEG].x = 160.624 Position[LLEG].y = 145.843
Position[RLEG].x = 165.592 Position[RLEG].y = 248.821
Voice command is "Begin"

Fig 5 The position data and voice command

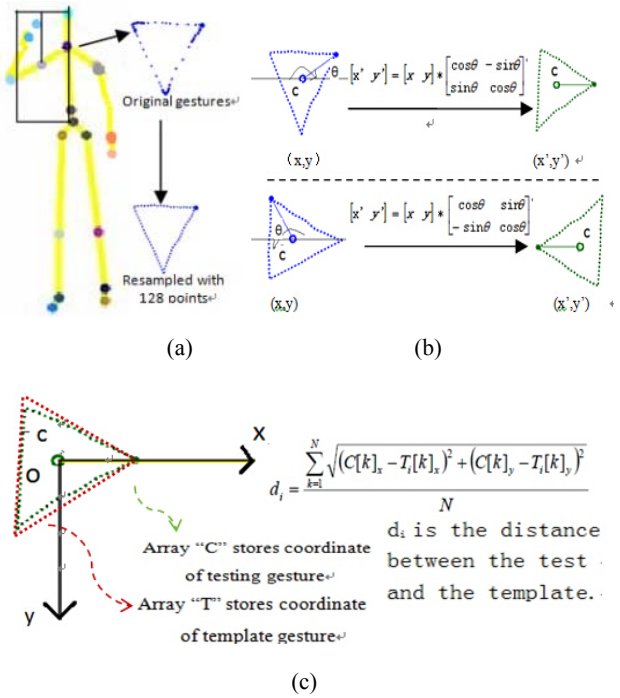

Fig. 6 The process of directional correlated single-step gesture recognition algorithm

VI. Gesture Recognition Module Realization

A. Directional Correlated Single-Step Gesture Recognition Algorithm

Single-step gestures which means can be gestured by one stroke satisfy people's requirements in many applications. Therefore, our system designed a method named directional correlated single-step gesture recognition algorithm aiming at the gestures which are also able to be divided by up, down, left and right. The algorithm was researched on the foundation of $1 unistroke recognizer [20] proposed by Jacob O.Wobbrock and Andres D.Wilson [21] to attune to our needs. $1 Recognizer is rotation invariant so that users can make gestures in arbitrary direction. As a result, the method, obviously, is incapable of identifying the direction the gestures are drawn. However, it is exactly necessary for our system, which users need to control the perform scenes of shadow play rolling forward or backward by waving their hands from left to right or reverse, to divide direction of hand signal.

As a consequence, we designed an effective algorithm could distinguish the gesture directions on the premise of rotation invariant for our system, Fig. 6 shows the process of resample, rotation, scale, translation and calculation in our algorithm. The following is the four steps of our algorithm.

Step 1. Resample gestures to a settled number of points which is set as 128 in this paper, as illustrated in Fig. 6 (a). This step guarantees there is no effect to the recognition result from speed of dynamic gestures.

Step 2. Rotate gestures at a proper angle, as illustrated in Fig. 6 (b), to make sure all gestures keep the same direction for remaining rotation invariant as far as possible. In our algorithm, we first calculate center point of gestures, C, which depends on (1):

$$Cx = \frac{\sum_{i=0}^{N} point[\,i].x}{N} \qquad Cy = \frac{\sum_{i=0}^{N} point[\,i].y}{N} \tag{1}$$

The array "Point" stores the coordinates of all sample points in gestures. Next we establish the vector from the center to the beginning and rotate the whole gestures to make sure the angle α formed by the vector and the horizontal line is 0°. The rotate direction should be chosen according the following four cases.

CASE A. If the starting point on the left side of the center and angle α ([-π ~ π]) is positive, then rotate gestures anticlockwise at angle α.

CASE B. If the starting point on the left side of the center and angle α is negative, then rotate gestures clockwise at angle α.

CASE C. If the starting point on the left side of the center and angle α is positive, then rotate gestures clockwise at angle α.

CASE D. If the starting point on the left side of the center and angle α is negative, then rotate gestures anticlockwise at angle α.

Step 3. Scale gestures to ensure all gestures in the same size so that users may draw gestures in arbitrary size. We scale gestures through the method of defining a standard square.

Step 4. Translate gestures to the origin for keeping the origin and the center point coincide. This step simplifies the further matching calculation (Fig. 6 (c)) between the gestures need to be identified and the templates.

B. Algorithm Testing

We divided templates into five groups in our test, which every group consists of the templates have the same shape but opposite direction, as Fig. 7 shows. In Fig 7, the black dots are initial points and the gestures in the first row were drawn in forward direction and the gestures in the second row were converse. Table I displays the recognition rate of the algorithm. One of the heads of the form named "reversing relative accuracy" means that the rate is got in the condition that the positive gesture in the same team identified correct. According the data from Form 1, we calculated the average accuracy in 96.35% for our algorithm, which is a huge improvement compare with 57.81% for $1 recognizer when applying for direction correlated gestures.

In our system, we chose "Wave" and "WaveReverse" as the scene switching commands and had the desired effect. Given that mature gesture recognition algorithm, it is easier for our system to extend gesture functions.

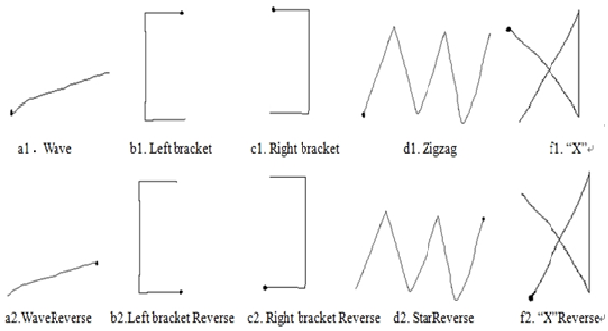

Fig. 7 The gestures used to test

TABLE I
THE RECOGNITION RATE OF OUR ALGORITHM V.S. $1 ALGORITHM

GESTURE	Accuracy of our Algorithm (%)	Accuracy of $1 Algorithm (%)
Wave	98.34	49.50
Left bracket	95.35	53.49
Right bracket	94.02	55.48
Zigzag	96.01	55.81
X	98.01	74.75
Average	**96.35**	**58.81**

VII. EXPERIMENT RESULT

Fig. 8 shows the actual running effect of the shadow animation system. When users enter the effective detection region of Kinect and do some actions, which include walk forward, walk backward, stepping, bending down and waving, etc., the shadow characters shown by the program do the same actions. Pressing the relevant function keys can switch background pictures and subtitles of the current performance casually. The performance subtitles are stored in an editable text file, which can be altered as required. The voice command "Begin" and "Stop" also correspond to a function key.

For example, when the user waving the right hand, the application calculated new start drawing positions of shadow character's the upper right arm, the lower right arm and the right hand, according to the received data of the user's right shoulder, right elbow and the right hand. Based on these new positions, draw and render the corresponding part of the shadow character. The other parts are drawn in the similar way. Fig. 9 shows the waving effect of the shadow character.

Fig. 8 Actual effect of our system

Fig. 9 Effect of waving hand

We tested 30 people including 10 teenagers, 10 youths and 10 elderly people, to know the practicability of our system. According the result displayed in Fig. 10, people in all ages can easily operate the system and almost all people investigated expressed that their attention was totally attracted. Most respondents thought our work had a positive effect on the culture protection of Chinese shadow play.

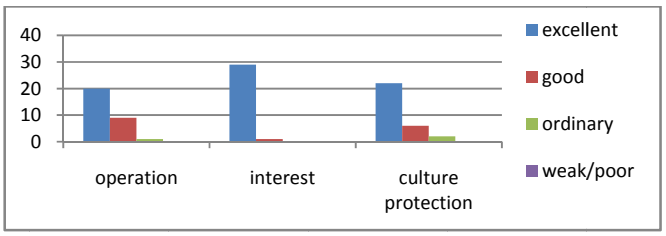

Fig. 10 Experiment result from 30 volunteers

VIII. CONCLUSIONS AND FUTURE WORK

This paper presents an interaction shadow animation system based on the Kinect sensor. The system runs smoothly and the shadow characters can accurately reflect the real-time actions of the users. The generated animation video is satisfactory. In the practical application, the whole system can be constructed conveniently. As an inexpensive device, Kinect has powerful features. What's more, the application founded on the traditional Chinese shadow play, digitizing the traditional

shadow play and makes the performance available to ordinary people who have no professional training and shadow performance experience, so that the form of shadow animation can spread further and wider. It protected one of the precious cultural heritages for our country.

In order to improve the system, a few suggestions have been proposed: Introduce the GPU rendering technology, make the shadow characters more life-like. Increase the function of turn around as the existing system don't support turn-back, so that the action of the character can be more authentic. Add more gesture commands to enhance human-computer interaction. Enable the system read the subtitles aloud when playing the recorded video in order that the finished video can be more impressive.

ACKNOWLEDGMENT

This work is supported by the projects of Xi'an science and technology plans (grant no. CXY1440(2)).

REFERENCES

[1] Ugur Güdükbay, Fatih Erol, Nezih Erdoğan. Tradition offers artistic possibilities for new media technologies:An animation system for shadow theatre, actes / proceedings isea2000 – 10/12/2000 – salle/room 300.
[2] Yi-Bo Zhu, Chen-Jia Li, I-Fan Shen, Kwan-Liu Ma, Aleksander Stompel, A New Form of Traditional Art – Visual Simulation of Chinese Shadow Play. Proceeding SIGGRAPH '03, ACM SIGGRAPH 2003 Sketches & Applications. 2003:1-1.
[3] GAO Lu-jing, CAI jian-ping, Analysis and Implementation of Features of Character's Action in Shadowgraph Animation. Computer Engineering and Design, 2010. 2335-2342
[4] S.W. Hsu, T.Y. Li. Generating Secondary Motions in Shadow Play Animations with Motion Planning Techniques. Proceedings of SIGGRAPH 2005 conference on Sketches & applications, 2005, Article No. 69,1 pages.
[5] S.W. Hsu, T.Y. Li. Planning Character Motions for Shadow Play Animations. Proceedings of Computer Animation and Social Agents, 2005:1-6.
[6] Totilo, Stephen (January 7, 2010). "Natal Recognizes 31 Body Parts, Uses Tenth of Xbox 360 "Computing Resources"". Kotaku, Gawker Media. Retrieved November 25, 2010. http://en.wikipedia.org/wiki/Kinect#Technology
[7] L. Leite and V. Orvalho, "Shape your body: control a virtual silhouette using body motion" CHI '12 Extended Abstracts on Human Factors in Computing Systems: 1913-1918
[8] Seth Hunter, Pattie Maes. Designing digital puppetry systems: guidelines and best practices. Extended Abstracts on Human Factors in Computing Systems. 2013
[9] Robert Held, Ankit Gupta, Brian Curless, Maneesh Agrawala. 3D puppetry: a kinect-based interface for 3D animation. Proceedings of the 25th annual ACM symposium on User interface software and technology, Oct., 2012.
[10] Bradley Wesson, Brett Wilkinson. Evaluating organic 3D sculpting using natural user interfaces with the Kinect, Proceedings of the 25th Australian Computer-Human Interaction Conference, Nov., 2013.
[11] Kam Lai, Konrad, J., Ishwar, P. A gesture-driven computer interface using Kinect, published byImage Analysis and Interpretation (SSIAI), 2012 IEEE. Pages: 185 – 188.
[12] Karthik Nandakumar, Kong Wah Wan, Siu Man Alice Chan, Wen Zheng Terence Ng, Jian Gang Wang, Wei Yun Yau. A multi-modal gesture recognition system using audio, video, and skeletal joint data, Proceedings of the 15th ACM on International conference on multimodal interaction, Dec.,2013
[13] Tsai-Te Chu, Chung-Yen Su. A Kinect-based Handwritten Digit Recognition for TV Remote Controller, published byIntelligent Signal Processing and Communications, 4-7 Nov. 2012.Pages: 414 – 419
[14] Anderson, D., Bailey, C. and Skubic, M. (2004) Hidden Markov Model symbol recognition for sketch-based interfaces.AAAI Fall Symposium. Menlo Park, CA: AAAI Press, 15-21.
[15] M. Black, A. Jepson. A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. in Proc. European Conf. on Computer Vision(ECCV'98), 1998
[16] M. J. Black, A. D. Jepson. Recognition temporal trajectories using the Condensation algorithm. in Proc. IEEE Int'1 Conf. on Automated Face and Gesture Recognition(FG'98), Japan, 1998: 16~21
[17] Asma Ben Hadj Mohamed, Thierry VAL, Laurent Andrieux, Abdennaceur KACHOUR. Assisting people with disabilities through Kinect sensors into a smart house, International Conference on Computer Medical Application,Jan.2013.Pages:1-5
[18] Foley, J., V.L. Wallace, and P. Chan,The Human Factors of Computer Graphics Interaction Techniques. IEEE Computer Graphics and Application, 1984.4 (11): p. 13-48.
[19] Russell M. Taylor II, Thomas C. Hudson, Adam Seeger, Hans Weber, Jeffrey Juliano, Aron T. Helser. VRPN: A Device-Independent, Network-Transparent VR Peripheral System. Proceedings of VRST 2001, ACM Virtual Reality Software and Technology. 2001.
[20] http://depts.washington.edu/aimgroup/proj/dollar/
[21] Jacob O.Wobbrock, Andrew D.Wilson and Yang Li.(2007) Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. Proc. UIST '07. New York: ACM Press, 159-168.
[22] Paul Doliotis, Alexandra Stefan, Christopher McMurrough, David Eckhard, Vassilis Athitsos. Comparing gesture recognition accuracy using color and depth information, Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments, May, 2011.
[23] Jaemin Soh, Yeongjae Choi, Youngmin Park, Hyun S. Yang. User-friendly 3D object manipulation gesture using Kinect, Nov., 2013.