Multimed Tools Appl
DOI 10.1007/s11042-015-2739-6

Implementation of an interactive TV interface via gesture and handwritten numeral recognition

Jia-Shing Sheu[1] · Ya-Ling Huang[1]

Received: 12 January 2015 / Revised: 24 May 2015 / Accepted: 8 June 2015

Abstract In this study, a Kinect controller was used to develop control software for interactive television (ITV) and interactive multimedia, thus enabling users to intuitively and conveniently play videos and perform interactive operations. Because it lacks a button controller, the proposed design can achieve a human–machine interaction effect. The interactive control system is divided into two parts: dynamic gesture and handwriting recognition. The Kinect sensor is used as an input device to recognize the dynamic gestures of users to achieve real-time interactive control. TV channels can also be selected automatically through the recognition of handwritten digits. Furthermore, a back-propagation neural network was used to complete handwriting recognition in space to achieve the optimal recognition rate.

Keywords Back-propagation neural network (BPNN) · Feature extraction · Gesture recognition · Handwriting recognition interactive television (TV) · Principal curves

1 Introduction

The established game interface Kinect was launched in 2010. It captures computer-user action through an infrared sensor and lens, and collects data on the gestures and facial expression of the player. Games can be directly controlled using actions, voice, and intuitive gestures (feelings). Kinect negates the traditional operation method whereby players must sit at a desk and use a keyboard and monitor. Thus, it can make the human–machine interface a reality. Simply by raising a hand, a user can cause Kinect to capture user action immediately and complete an operation; this gives users the impression that they are not operating a computer. If it is applied to the imaginary interface of interactive digital home video [3], users can operate the interface through gestures. This makes operation easier for users and achieves an

✉ Jia-Shing Sheu
jiashing@tea.ntue.edu.tw

Ya-Ling Huang
elisahuang.taipei@gmail.com

[1] Department of Computer Science, National Taipei University of Education, No. 134, Sec. 2, He-Ping Rd., 10671 Taipei, Taiwan

Published online: 23 June 2015

interactive effect. Users are not required to hold a controller or any other input device. Based on these features, this paper proposes an interactive mode in which television (TV) can be operated through handwriting. TV channels can be selected using handwritten digits. To improve performance, a back-propagation neural network (BPNN) was used in a handwriting recognition system to achieve more accurate handwriting recognition. New fonts can be added or learnt as additional control instructions. The purposes of this research are listed as follows: 1) using a Kinect sensor for obtaining image input, without the necessity of a recognizer or gloves to track user gestures; 2) using handwritten digits to select TV channels to achieve interactive operation; 3) adding handwritten fonts for other applications; and 4) integrating dynamic recognition with handwriting recognition to develop an easy-to-use interactive TV operation interface.

The remainder of this paper is organized as follows. Section 2 presents a literature review on related technologies. Section 3 presents the system design and research method, including the system architecture, an introduction to the system hardware, interactive interface design and operation, Kinect skeleton tracking, dynamic recognition system, the definition of dynamic gesture operation, handwriting recognition system, and the definition of handwriting. Section 4 provides a discussion on the experimental results, system design, and theory verification. Section 5 presents the conclusion.

2 Literature review

In previous methods involving gesture control, users have been required to use a handheld device or other electronic input devices to achieve reality-based interaction [5]. Through technological advancement, reality-based interaction for TV and the Internet has become a trend. Users can experience a 3D environment similar to real environments to achieve interaction between the real and virtual worlds, which is frequently incorporated in the design of human–machine interfaces [2], as shown in Fig. 1. Kinect SDK provides a human skeleton model and uses skeleton joints for skeleton tracking [16].

Human–computer interface design has been explored by programmers and digital artists. According to Laurel, interface design has evolved, and its definition has been revised. At the early stages of development, an interface was defined as software and hardware used for communication between humans and computers. Interfaces are based on user cognition and

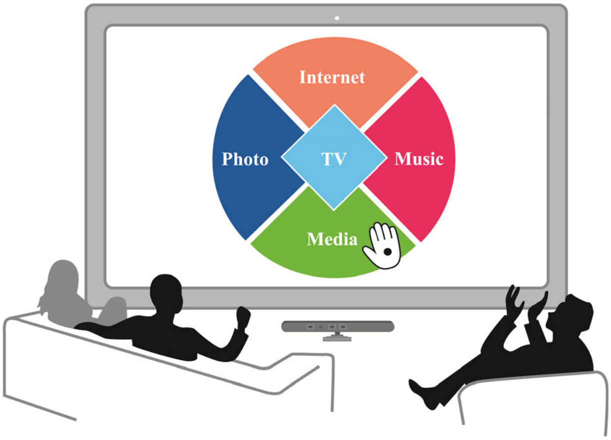


Fig. 1 Gesture interactive control

emotion, and are used to improve user ability. The *Art of Human-Computer Interface Design* by Laurel discusses user friendliness and raises problems related to the actual application of various interfaces [12].

An “interactive” interface means that a product interface plays the active role of auxiliary information provider. Moreover, an interface provides only a one-way action result for users when people operate the product interface. It does not provide any assistance regarding operation information or operation feedback; users are unaware of any program errors or state control until the results of error are evident [14].

Oviatt suggested that interactive user interfaces must demonstrate expressive power, portability, and naturalness [15], which are verified through the principles of interactivity, usability, and human-centric interaction, respectively [13].

1) Interactivity (expressive power): Interactivity is the quality of sensing and reaction, representing the experience after users perform actions.
2) Human-centric interaction (naturalness): From a designer’s perspective, complicated and unnecessary manipulation instructions that can result in redundant operation steps should be avoided.
3) Usability (portability): Usability refers to whether interactive multimedia can be used repeatedly, combined with other design media, and used as a common device in a design environment.

Laurel [11] suggested that people will not be aware of the existence of an interface in future human–computer interaction, and referred to this as the “vanishing interface, by which people can intuitively use computers in daily life for enjoyment. Laurel proposed the design principle of human–computer interaction, which includes direct manipulation and direct engagement. The interacting object is imagined as a character, and interaction can be achieved through enactment, as shown in Fig. 2. Laurel proposed four principles of human–computer interaction design: 1) designing actions; 2) designing characters and thought; 3) designing language and communication; and 4) designing the process of enactment.

Jeng [6] suggested the concepts of perception space, actual space, and virtual space to distinguish the states of interaction events, as shown in Fig. 3. In the perception space, a user performs an action after forming perceptions. In actual space, a user operates a physical representation through control and remotely operates a digital representation, and can feel changes in the space state caused by the digital information in the space. In virtual space,

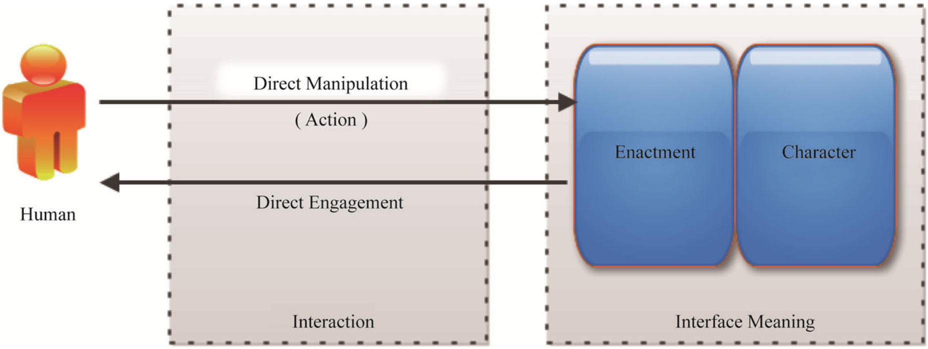


Fig. 2 Laurel’s [11] interaction model

physical representation is connected to interaction events and triggers a virtual interactive mode through interaction events. The digital information in this space is a concrete interactive mode. Users can trigger interaction events through the control of an actual interface, and can directly operate digital information to produce a real and virtual human-centric interaction mode. Hand gesture is one of the natural and intuitive ways to communicate between human and machines, including hand detection and gesture recognition, the 3D positions of body joints, and motion [17–19]. They mimic how people interact with each other.

Neural networks are often used in recognition methods. Among them, back-propagation neural networks (BPNNs) are the easiest to implement. BPNNs are used as a training method for error correction. The training process involves information forward transmission and error back propagation. Given a group of input modes, a BPNN passes the input mode from the input layer to the hidden layer. After being processed by the hidden layer, the input mode is produced and transmitted to the input layer. This process is called feedforward; if the input layer fails to obtain the desired input mode through feedforward, the process is converted to feedback, whereby the error signal is returned along the original connection point path, and minimized by modifying the neuron connection weight of each layer. The feed-forward and feedback processes are repeated until a desired input mode is obtained. In this study, a BPNN was used to convert statistical feature vectors into nonlinear handwriting recognition [1].

3 System architecture and methods

3.1 System architecture

To achieve interactive control over TV and digital video, gesture design was divided into dynamic gesture and handwritten gesture control systems. The dynamic gesture recognition system uses a Kinect sensor to capture the user gesture skeleton, and converts the gesture

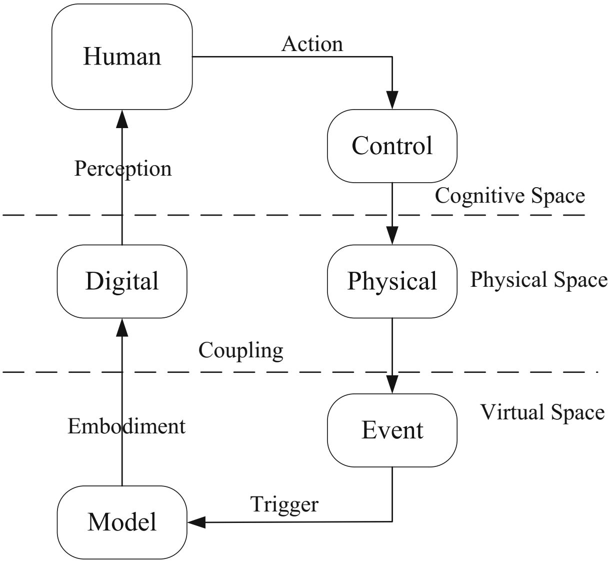


Fig. 3 Jeng's [6] real and virtual human-centric interaction mode

skeleton vector into a mouse operation mode, which is used as the input control in the interactive interface and for interface function selection. The gesture recognition system can be used to control and select TV channels; users can write digits by using gestures to select TV channels [4]. The system architecture is shown in Fig. 4.

3.2 Interactive interface

Based on the human-centric interaction principle proposed by Jeng, a simple interactive TV operation interface was designed. The architecture is shown in Figs. 2 and 3. Users can select TV channels by using Kinect.

The user's hand is regarded as a cursor, and channels can be selected through a handwriting function. The advantages provided by handwriting are listed as follows: 1) If the audience of a TV channel is more, users can directly select channels through handwriting, and the search is not disrupted; and 2) users can simply stand in front of the TV to select channels without using a handheld device.

In this study, a main interface was designed. Figure 5 presents the functions of Areas A and B. Area A is the scroll view. Users can move their right hands to select TV channels in this area. When the cursor stops for 1 s in this area, the scroll can be moved. To watch a TV channel, the cursor must stop for 3 s on the desired channel. Area B is the TV view, in which selected TV channels can be viewed.

3.3 Handwriting interface

Using the main interface, a user can lift their left hand to the height of their shoulder, and then move their hand from the left to the right to launch the handwriting interface. The user can then move their right hand to the handwriting recognition area, and operate a handwriting cursor, as shown in Fig. 6. Initially, the user does not make transient gestures in the handwriting area; after 2 s, the handwriting function is initiated, and the user can move their right hand to write and the handwriting window displays corresponding strokes. The handwriting function is automatically executed if the user holds their hand at the location of the previous stroke for 2 s; the handwritten digit is recognized and the TV switches to the desired channel.

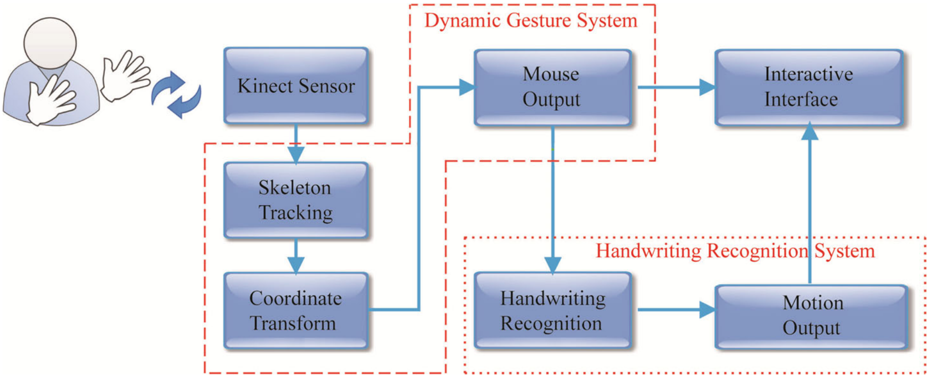


Fig. 4 The dynamic gesture recognition system and architecture

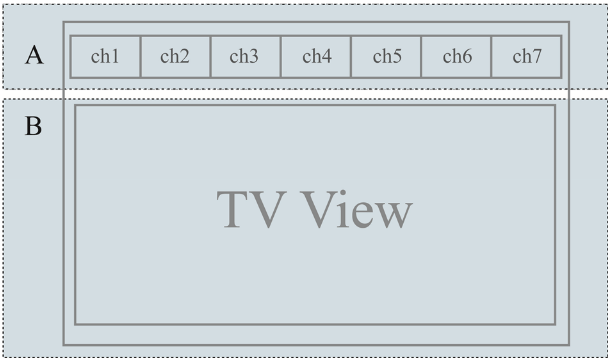


Fig. 5 The main interactive interface of TV

3.4 Definition of interactive gesture operation

The interactive gesture recognition area is shown in Fig. 7. The limited range of the gesture recognition area is demarcated by a yellow square; the maximal height is the height of the user's hands when lifted, and the maximal width is the distance between the user's hands when extended laterally. Gesture detection ceases when the hands are lower than the waist. The center points of the head and waist are shown in Fig. 7, represented by two red points, and the blue dotted line divides the left and right sides. In addition, Table 1 describes the simple interactive gestures used in this study. All of the actions require the user to be in the gesture recognition area for interaction to occur.

3.5 Main interface operation

The main interface operation flow is shown in Fig. 8. The user stands within 2–3 m of the Kinect sensor and waves their right hand to enter the gesture detection mode and start the detection of right or left hand gestures. The program can then detect whether the left hand is waved to the left side to determine whether to initiate interactive TV software. If the software is running, it can be shut down by waving the left hand a second time. If the program starts and

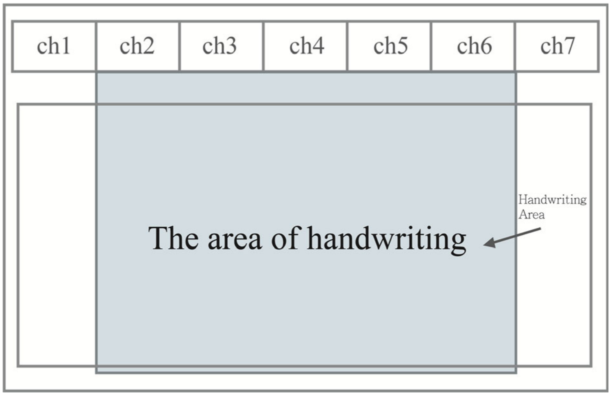


Fig. 6 Handwriting interface

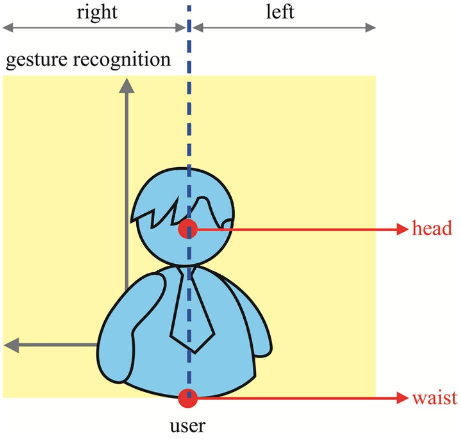
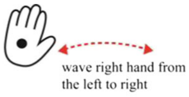
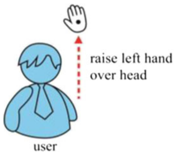
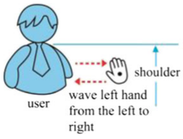


Fig. 7 Dynamic gesture recognition area

is connected to the main interface in parallel, the user raises their left hand to launch the handwriting interface. The user controls the cursor with their right hand to operate the interactive software to select channels or perform handwriting functions.

Table 1 Definition of interactive dynamic gesture rules

Definition of dynamic gesture rules	Description
	1. When the user waves his/her right hand from the left to right in the gesture recognition area, "mouse control mode" is started. 2. When the user puts his/her hand below his/her waist, detection of the mouse control mode stops. If the user wants to restart "mouse control mode", he/she only needs to move his/her right hand to the gesture recognition area, and then wave his/her hand from left to right.
	Instantaneous gesture means interactive feedback function when the user stops action in the detection area. In this study, the instantaneous gesture can be divided into two modes. The first mode is main interface mode, and the second mode is handwriting operation mode. 1. Main interface operation at mode 1 - Stopping 1 second means press-down of the left mouse button for long time, if the function is set, the press-down function of the left mouse button is released after 2 seconds - Stopping 3 seconds means Double Click of the left mouse button. 2. Handwriting operation at mode 2 - Stopping 2 seconds means press-down of the left mouse button for long time. If the user wants to release press-down function, he/she can stop action in the detection area.
	After the user raises his/her left hand over his/her head, the handwriting interface is called out, handwriting recognition mode is started, and handwriting function is performed.
	1. When the user raises his/her left hand to height of his/her shoulder and wave his/her hand from left to right, interactive TV program is started. 2. When the user wants to close down the program, he/she raises his/her left hand to height of his/her shoulder and wave his/her hand from left to right, the program exits.

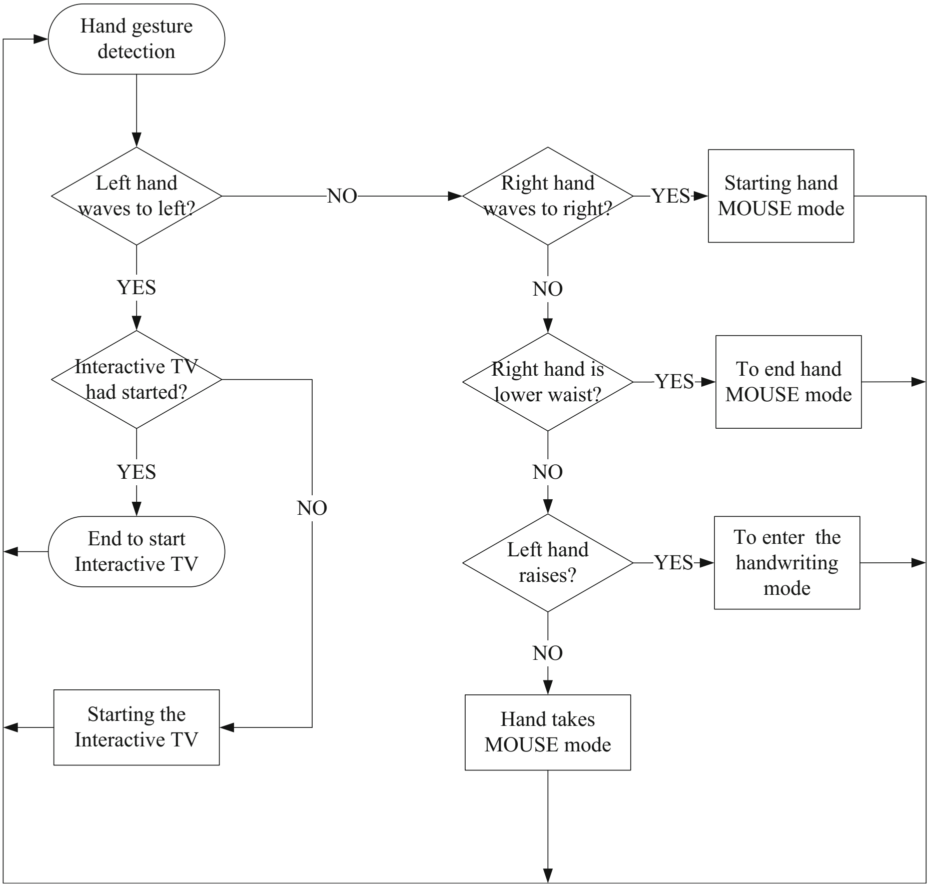

Hand gesture detection
Left hand waves to left?
NO
Right hand waves to right?
YES
Starting hand MOUSE mode
YES
NO
Interactive TV had started?
Right hand is lower waist?
YES
To end hand MOUSE mode
YES
NO
NO
End to start Interactive TV
Left hand raises?
YES
To enter the handwriting mode
NO
Starting the Interactive TV
Hand takes MOUSE mode

Fig. 8 Main interface operation flow

3.6 Definition of interactive handwritten digits and TV system

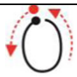
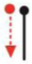
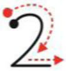
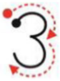
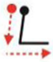
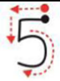
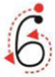
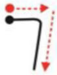
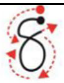
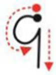
In this study, digits and letters from Palm's Grafitti [10] were used for handwritten digit recognition. Table 2 defines the handwriting rules for digits 0–9; strokes start from the red point and follow the direction of the arrows.

The interactive TV schematic diagram is shown in Fig. 9. The upper area shows the Kinect sensor detecting user movements to achieve the interactive effect. The user must preset the optical zoom distance of the Kinect sensor before operating the interactive TV system. The sensor should be at least 1 m above the ground, and approximately 2–3 m from the user. Regarding the operation method, the user stands in front of the main control terminal and waves their hand from the left to the right to control the interactive selection menu or to select a TV channel.

3.7 Coordinate transformation of the skeleton system

Based on the skeleton information obtained by the Kinect sensor, the coordinates of the gesture vector of the left and right hands are transformed. Kinect is used to obtain depth images. After

Table 2 Digit handwriting direction rule

Digitals	Handwriting Rule
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

detecting ready skeleton streaming, the skeleton coordinates of the user's hands can be obtained. The obtained skeleton information is transformed into relevant cursor coordinates to achieve interactive operation. The Kinect skeleton coordinates differ from the screen coordinates. The Kinect software development kit (SDK) public methods application programming interface (API) *MapSkeletonPointToColor* (*SkeletonPoint*, *ColorImageFormat*) are used. After specific skeleton coordinates are mapped to the color image coordinates, they are placed in a designated area (such as the Windows view area), magnified on an equal scale, and converted to mouse input format to achieve relevant interactive control. The skeleton to mouse coordinates at an equal ratio are shown in (1).

$$\text{Coordinates at an equal ratio}: \begin{cases} ux = kx \times fx = kx \times \dfrac{ux_{\max}}{kx_{\max}} \\ uy = ky \times fy = ky \times \dfrac{uy_{\max}}{ky_{\max}} \end{cases} \quad (1)$$

Here, (ux,uy) is the color image coordinate on the computer screen, (kx,ky) is the skeleton coordinate from Kinect. $(ux_{\max},uy_{\max})$ is the maximum range of color image coordinates on

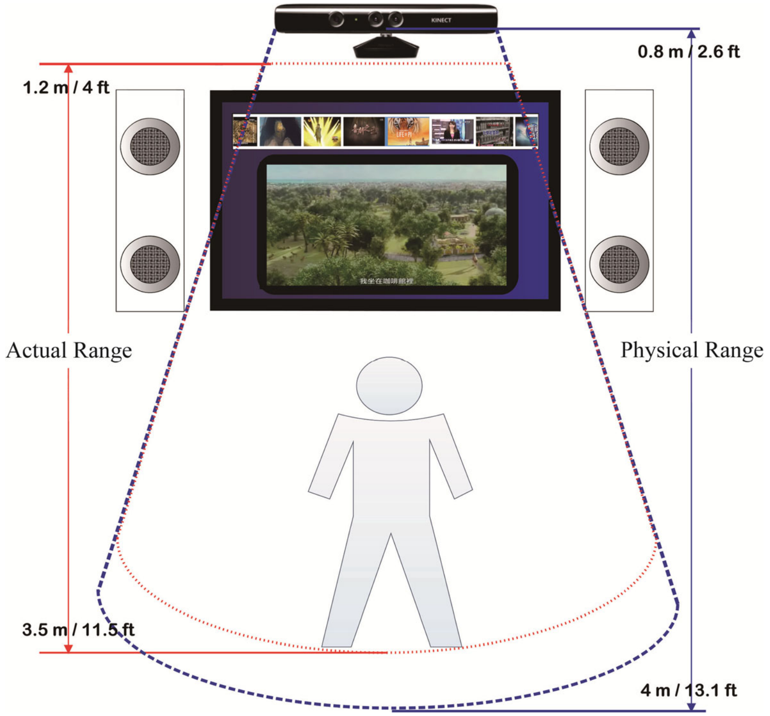


Fig. 9 Interactive TV schematic diagram

the computer screen, and $(kx_{\max}, ky_{\max})$ is the maximum range of skeleton coordinate from Kinect. (fx, fy) is the ratio of coordinate.

3.8 Time transient stability counter

Based on the results of transient time counting, the movement of coordinates X and Y within the threshold time is slight, and the projects in which the current coordinates X and Y are located are extracted. The locus data set is assumed to be $P_i = \{P_0, P_1, P_2, \ldots, P_n\}$ $i = 0 \sim n$ = the number of coordinates of recording the completion time. In the time transient counter (t), the coordinate movement variation $\Delta P_i = \{\Delta P_0, \Delta P_1, \Delta P_2, \ldots \Delta P_{n-1}, \}$, and $\Delta P_i = |P_i - P_{i+1}|$. When $Max(\{\Delta P_i\})$ is smaller than the jitter threshold, stoppage and extraction has occurred.

3.9 Skeleton tracking

The skeleton tracking flow is shown in Fig. 10.

1) The Kinect sensor is used to obtain a depth image frame. Kinect SDK Skeleton Data Joints are used to obtain the 3D coordinates of a user's joints.
2) In the skeleton coordinate system, the skeleton information sent by Kinect contains the 3D coordinates (X, Y, and Z) of each joint in the skeleton, and the unit of measurement is meters. For the right-hand coordinate system, the angle from the Kinect sensor is a zero-

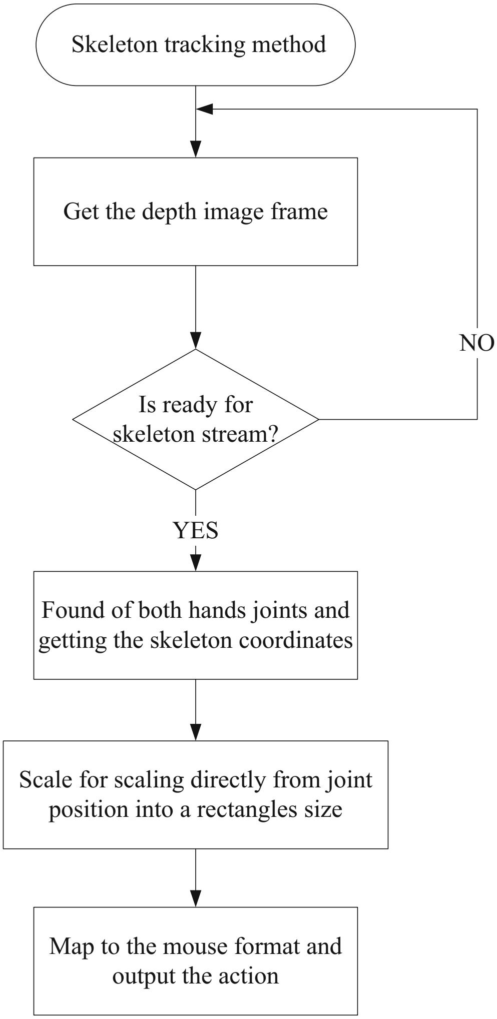

Skeleton tracking method
Get the depth image frame
Is ready for skeleton stream?
NO
YES
Found of both hands joints and getting the skeleton coordinates
Scale for scaling directly from joint position into a rectangles size
Map to the mouse format and output the action

Fig. 10 Skeleton tracking flow

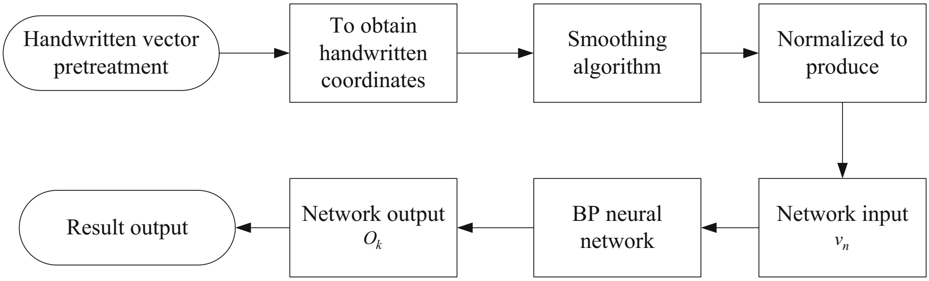


Fig. 11 Handwritten recognition system architecture

axis straight central line. Based on this base point, left is the positive direction on the X axis, and right is the negative direction. For the zero-axis straight central line, the upper level value is in the positive direction on the Y axis, and the lower level value is in the negative direction. The Z axis is small when it approaches the sensor, and is large when it is far from the sensor.

3) The skeleton coordinates are converted to mouse output format; however, Kinect skeleton coordinates differ from screen coordinates. Kinect SDK API is used and after *MapSkeletonPointToColor* has obtained the video coordinates, they are placed in the designated area and magnified on an equal scale, and converted to mouse output format to achieve interactive control.

Second-hand skeleton coordinates obtained from Kinect Skeleton Joint Data are converted into screen coordinates for mouse operation.

3.10 Handwriting recognition system

The handwriting recognition system architecture is shown in Fig. 11. Kegl documented the extraction of a character skeleton by using principal curves [7, 8] and an improved algorithm [9]. Adaptive constraint K -segment principal curves are a type of algorithm [20] that is highly practical in the application of principal curve theory, and is similar to the vector quantization method. The K principal curve algorithm restricts the total curve length, and the vector quantification algorithm constrains the number of samples. The square function of the expected distance between point x of a data set and the curve $f(\lambda)$ is expressed in (2).

$$\varDelta(x,f)=E\left\{\|x-f(\lambda)\|^2\right\} \tag{2}$$

In a closed interval, the length of any random curve in one curve family F is smaller than or equal to length of curve f^*. This means that f^* comprises one curve, and the squared distance from all points of data set x to the curve is minimal. Curve f^* is called the principal curve of the data set. The equation is expressed in (3).

$$f^*(\lambda)=\arg\min_f E\left[\|x-f\|^2\right] \tag{3}$$

Equation (3) indicates that the principal curve must exist when the data set has a finite second moment [8]. According to the definition of a principal curve with a constrained length, Kegl designed a polygonal line algorithm. The curve family F in the closed interval is considered, and each curve consists of $\varDelta(f_k)=\frac{1}{n}\sum\limits_{i=1}^{n}\varDelta(X_i,f_k)$ $f^*(\lambda)$ vertices. Each curve is continuous and smooth when $k\to\infty$. The vertices are connected to form a curve with $k-1$ line segments. Each curve is expressed using $f_k\in F$, and the expected squared distance from the points in the data set to the curve is expressed in (4).

$$\varDelta(f_k)=\frac{1}{n}\sum_{i=1}^{n}\varDelta(X_i,f_k)f^*(\lambda) \tag{4}$$

In (4), n is the number of data points, and $\Delta(f_k)$ is called the mean square error (MSE). A polygonal line algorithm with points can be converged by continually updating the position of the curve vertices. Equation (5) can then be derived.

$$f_k^{\ *} = \arg\min_{f \in F} \Delta(f_k) \tag{5}$$

If the expected distance value fails to reach the global converged MSE threshold, one vertex is added to curve f_k^*. By using the aforementioned method, the principal curve f_{k+1}^* consisting of $k+1$ vertexes can be determined. The computation process is repeated until $\Delta(f_k)$ satisfies the preset threshold value. Determining the principal curve with a constrained length requires two steps: 1) optimizing and updating the vertex position; and 2) adding vertices.

Updating the vertex position includes two phases: 1) projection from the data set to the curve f_k; and 2) the local optimization of the vertex position. Thus, solving the optimization problem requires internal and external cycles and repeated operation. Therefore, a polygonal

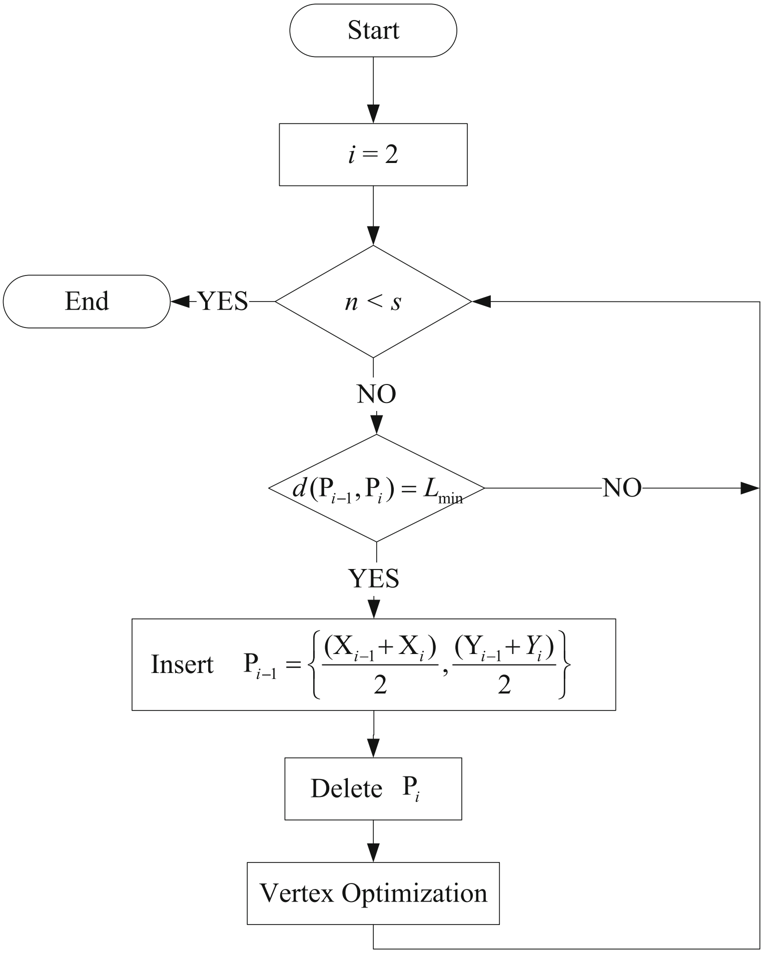

Start
$i = 2$
End
YES
$n < s$
NO
$d(\mathrm{P}_{i-1}, \mathrm{P}_i) = L_{\min}$
NO
YES
Insert $\mathrm{P}_{i-1} = \left\{ \frac{(\mathrm{X}_{i-1} + \mathrm{X}_i)}{2}, \frac{(\mathrm{Y}_{i-1} + Y_i)}{2} \right\}$
Delete P_i
Vertex Optimization

Fig. 12 The flow chart of smoothing algorithm

line algorithm can be used. In this study, Kegl's polygonal line algorithm with K principal curves was used to extract the digital skeleton structure. The algorithm includes the following primary steps:

Step 1, initialization: To obtain the original handwriting skeleton G_{vs}, G_{vs} consists of two sets, V and S, where $V=\{v_1,v_2,\ldots,v_n\}\in R^d$ is a set of vertices and $S=\{(v_{i1},v_{j1}),\ldots,(v_{ik},v_{jk})\}=\{s_{i1j1},\ldots,s_{ikjk}\}$ is a set of edges.

Step 2, smoothing: The purpose is to obtain the optimal graph logical features by adjusting the smoothness of the handwriting skeleton drawing. Based on the given data set $X=\{x_1,x_2,\ldots,x_n\}$, penalty function methods are included in (6).

$$E(G)=\Delta(G)+\Delta P(G) \tag{6}$$

By using the minimal value and optimizing the skeleton diagram, $\Delta(G)=\frac{1}{n}\sum_{i=1}^{n}\Delta(x_i,f)$ can be applied to determine the value of the squared distance from points in the pointer data set to the G curve. Furthermore, when $\Delta P(G)=\frac{1}{k+1}\sum_{i=1}^{k+1}P(v_i)$ is the average penalty function of the G curve, $P(v_i)$ is the penalty curvature of vertex v_i, and

$$P(v_i)=\begin{cases}\frac{1}{k+1}\left(P_v(v_i)+P_v(v_{i+1})\right), & if\ \ i=1\\ \frac{1}{k+1}\left(P_v(v_{i-1})+P_v(v_i)+P_v(v_{i+1})\right), & if\ \ 1<\ i\ <k+1\\ \frac{1}{k+1}\left(P_v\left(v_{i-1}\right)+P_v(v_i)\right), & if\ i=k+1\end{cases}$$

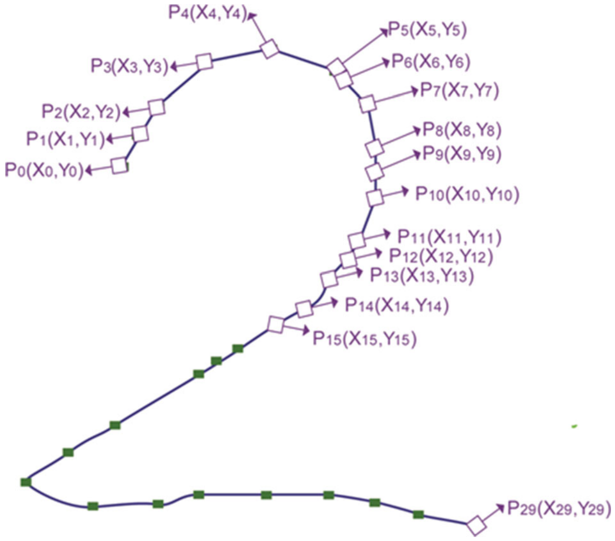


Fig. 13 Handwritten coordinate points of digit 2 obtained by the system

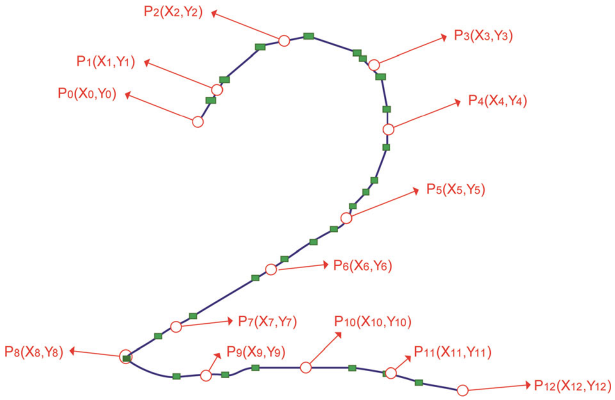


Fig. 14 Sample points after smoothing algorithm

when $\Delta(G)$ is smaller, the characteristic data of the skeleton diagram is improved; when $P(G)$ is smaller, the skeleton diagram is smoother. At this step, the perform projection step, in which the data set X is classified into the area closest to the vertex and edges of the skeleton diagram, vertex optimization is performed by adjusting the G vertex and the edge of the skeleton diagram, causing the penalty function $E(G)$ to produce the local minimal value.

Step 3, reconstruction: At the reconstruction step, the geometric properties of the skeleton diagram are used to delete short branches of the vertex and edge structure, and

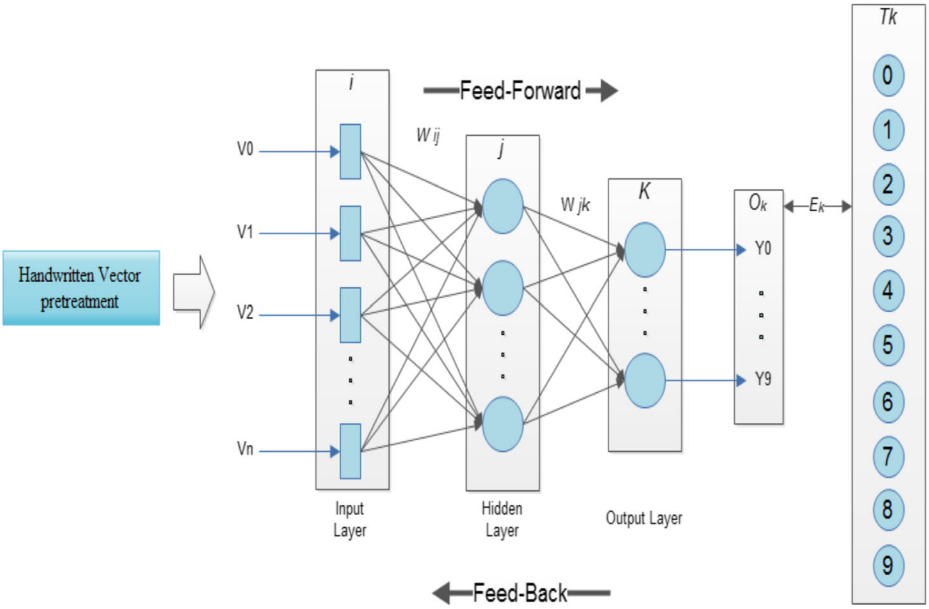


Fig. 15 Neural network training system

integrate the feature extraction, the program flow chart as shown in Fig. 12. In Fig. 12, n denotes the number of handwritten coordinate points obtained by the system; s means the sample number of smooth vectors. The s equals to 13 in this study.

Suppose that the system obtained the data point set of handwritten digit 2 is $P_i=\{P_0P_1,P_2,\ldots,P_n\}$ $i=0\sim n$, where n is the number of handwritten coordinate points, as shown in Fig. 13. The polygon line segment algorithm is proposed by B. Kegl. It is used to determine the distance between the two adjacent points of each vector, delete the coordinates of the shortest distance $L_{\min}$, and reconstruct the combined vector points [7–9]. After using the smoothening algorithm, the vector points are obtained as shown with the red circle in Fig. 14.

After feature extraction, the sample coordinate set $P_i=\{(x_0,y_0),(x_1,y_1),\ldots,(x_n,y_n)\}|i=0\sim n$ is normalized to produce the matched vector set $D=\{v_j,\ldots,v_n\}|j=0\sim n$. This can be used as input vector data in the BPNN, as shown in (7).

$$D=\begin{cases} v_j=\dfrac{x_i-x_{i+1}}{\sqrt{\left(x_i-x_{i+1}\right)^2+\left(y_i-y_{i+1}\right)^2}} & \Big| j=0,2,4,6,\ldots n \\ v_j=\dfrac{y_i-y_{i+1}}{\sqrt{\left(x_i-x_{i+1}\right)^2+\left(y_i-y_{i+1}\right)^2}} & \Big| j=1,3,5,7,\ldots n-1 \end{cases} \tag{7}$$

3.11 Handwriting recognition system

The obtained results are introduced into the BPNN and the input vectors repeat the following training procedure, as shown in Fig. 15.

1) Input the sample feature vector and calculate the network output O_k.
2) Calculate the error between output O_k and target output T_k.
3) Adjust the output layer weight, and repeat Steps 4 and 5 for each hidden layer.
4) Calculate the error of the hidden layers.
5) Adjust the weights of the hidden layers.

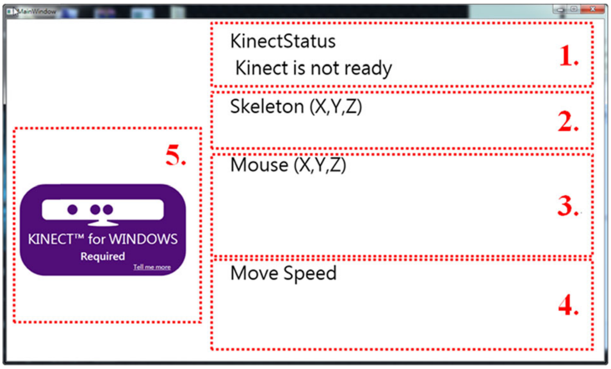
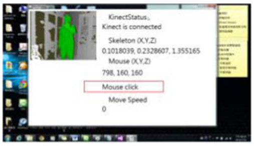
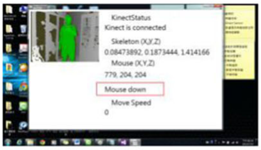
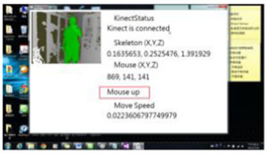
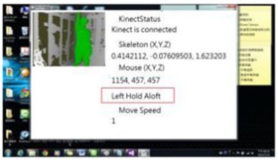
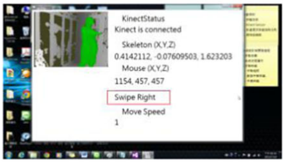
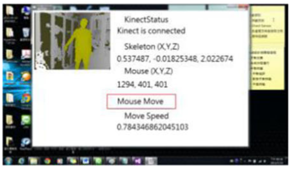
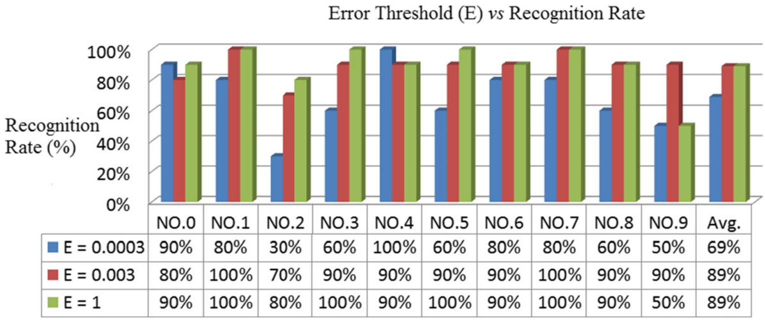


Fig. 16 Dynamic gesture test software interface

Table 3 Experimental results of dynamic gesture operation

Experiment Results	Meaning
 KinectStatus Kinect is connected Skeleton (X,Y,Z) 0.1018039, 0.2328607, 1.355165 Mouse (X,Y,Z) 798, 160, 160 Mouse click Move Speed 0	Mouse click represents the gesture is double click of mouse
 KinectStatus Kinect is connected Skeleton (X,Y,Z) 0.08473892, 0.1873444, 1.414166 Mouse (X,Y,Z) 779, 204, 204 Mouse down Move Speed 0	Mouse down represents that the detected gesture is press-down of the left mouse button for long time
 KinectStatus Kinect is connected Skeleton (X,Y,Z) 0.1635653, 0.2525476, 1.391929 Mouse (X,Y,Z) 869, 141, 141 Mouse up Move Speed 0.0223606797749979	Mouse up represents that the gesture is release of the left mouse button
 KinectStatus Kinect is connected Skeleton (X,Y,Z) 0.4142112, -0.07609503, 1.623203 Mouse (X,Y,Z) 1154, 457, 457 Left Hold Aloft Move Speed 1	Left Hold Aloft represents that gesture is raising left hand
 KinectStatus Kinect is connected Skeleton (X,Y,Z) 0.4142112, -0.07609503, 1.623203 Mouse (X,Y,Z) 1154, 457, 457 Swipe Right Move Speed 1	Swipe Right represents gesture is right hand waving
 KinectStatus Kinect is connected Skeleton (X,Y,Z) 0.537487, -0.01825348, 2.022674 Mouse (X,Y,Z) 1294, 401, 401 Mouse Move Move Speed 0.784346862045103	Mouse Move represents gesture is handwriting recognition verification through mouse movement

The K output neuron is O_k, and the expected target output is T_k. The initial error value E_k of the k output neuron is expressed in (8).



	NO.0	NO.1	NO.2	NO.3	NO.4	NO.5	NO.6	NO.7	NO.8	NO.9	Avg.
■ E = 0.0003	90%	80%	30%	60%	100%	60%	80%	80%	60%	50%	69%
■ E = 0.003	80%	100%	70%	90%	90%	90%	90%	100%	90%	90%	89%
■ E = 1	90%	100%	80%	100%	90%	100%	90%	100%	90%	50%	89%

Fig. 17 Impact of error threshold on recognition rate

$$E_k = (T_k - O_k)O_k(1 - O_k) \tag{8}$$

To change the weight between the *j* neuron of the hidden layer and the *k* output neuron of the output layer, (9) is used, where *L* is the learning rate.

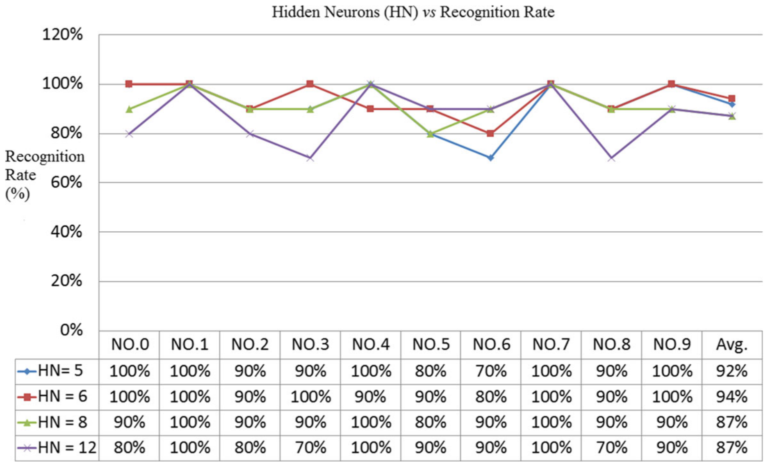
$$W_{jk} = W_{(j-1)(k-1)} + LE_kO_k \tag{9}$$

The *j* neuron weight of the hidden layer is calculated, and adjustments are made by using (10), where *n* is the number of neurons in the output layer.

$$E_j = O_k(1 - O_k)\sum_{k=1}^{n} E_kW_{jk} \tag{10}$$

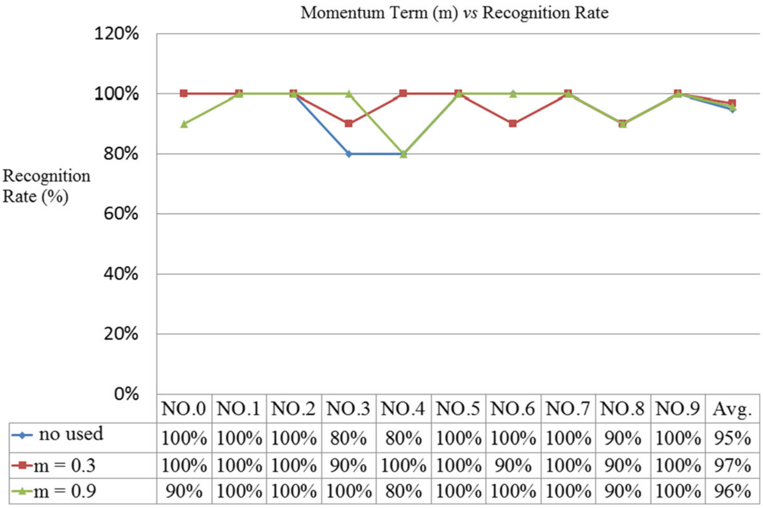
From the *j* neuron of the hidden layer to the input weight *i*, using (11) to update the weighting.

$$W_{jk} = W_{(j-1)(k-1)} + LE_kO_i \tag{11}$$



	NO.0	NO.1	NO.2	NO.3	NO.4	NO.5	NO.6	NO.7	NO.8	NO.9	Avg.
◆ HN= 5	100%	100%	90%	90%	100%	80%	70%	100%	90%	100%	92%
■ HN = 6	100%	100%	90%	100%	90%	90%	80%	100%	90%	100%	94%
▲ HN = 8	90%	100%	90%	90%	100%	80%	90%	100%	90%	90%	87%
× HN = 12	80%	100%	80%	70%	100%	90%	90%	100%	70%	90%	87%

Fig. 18 Impact of number of hidden neurons HN on recognition rate



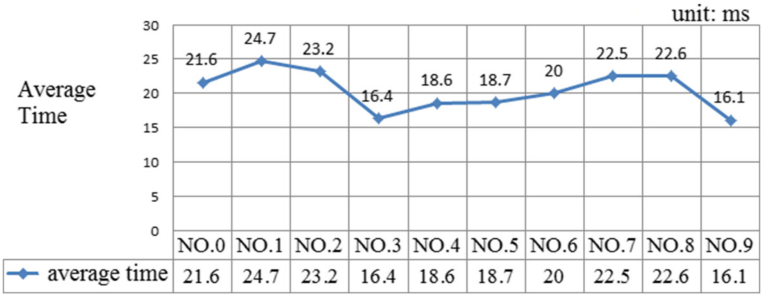
	NO.0	NO.1	NO.2	NO.3	NO.4	NO.5	NO.6	NO.7	NO.8	NO.9	Avg.
no used	100%	100%	100%	80%	80%	100%	100%	100%	90%	100%	95%
m = 0.3	100%	100%	100%	90%	100%	100%	90%	100%	90%	100%	97%
m = 0.9	90%	100%	100%	100%	80%	100%	100%	100%	90%	100%	96%

Fig. 19 Impact of momentum term on recognition rate

The aforementioned steps are repeated until the total error is lower than the specified threshold and, thus, the trained neural network parameters can reach the accepted level. In this study, the target output is T_k, and T_k will be one of the digit samples 0 to 9. And, the match tolerance was greater than 0.96.

Define T_k=[y_0,y_1,y_2,y_3,y_4,y_5,y_6,y_7,y_8,y_9], k will be 0 to 9. It means:

Digit 0: T_0=[1,0,0,0,0,0,0,0,0,0]
Digit 1: T_1=[0,1,0,0,0,0,0,0,0,0]
Digit 2: T_2=[0,0,1,0,0,0,0,0,0,0]
Digit 3: T_3=[0,0,0,1,0,0,0,0,0,0]
Digit 4: T_4=[0,0,0,0,1,0,0,0,0,0]
Digit 5: T_5=[0,0,0,0,0,1,0,0,0,0]
Digit 6: T_6=[0,0,0,0,0,0,1,0,0,0]
Digit 7: T_7=[0,0,0,0,0,0,0,1,0,0]
Digit 8: T_8=[0,0,0,0,0,0,0,0,1,0]
Digit 9: T_9=[0,0,0,0,0,0,0,0,0,1]



	NO.0	NO.1	NO.2	NO.3	NO.4	NO.5	NO.6	NO.7	NO.8	NO.9
average time	21.6	24.7	23.2	16.4	18.6	18.7	20	22.5	22.6	16.1

Fig. 20 Average time of channel selection using handwritten digits

4 Experimental results

In this study, a Kinect sensor and a BPNN were used to integrate dynamic recognition with handwriting recognition to create a simple interactive TV system for gesture control and testing. The test software is based on the Kinect sensor to test user gestures and produce correct mouse feedback. Visual Studio 2012 was also used as a development tool combined with the C# programming language, and Kinect SDK Version 1.7 was used to develop the system verification software. The dynamic gesture test software is shown in Fig. 16, and the interface is described as follows:

Area 1: Kinect Status. Indicates that Kinect is ready.
Area 2: Skeleton (X,Y,Z). Displays the skeleton coordinates of the user.
Area 3: Mouse (X,Y,Z). Displays the skeleton coordinates converted to coordinates in mouse output format and the detected gesture.
Area 4: Move Speed. Displays the movement speed of the user's gesture.
Area 5: User depth image display.

Kinect was connected to a PC through a USB interface. The Kinect sensor was 1 m above the ground. The effective detection distance was 2–3 m.

Table 3 shows the recognized gesture results of the experiment after the dynamic gesture operation was defined. The transient gesture speed time threshold of the movement speed was set to 0.02 ms. The measured movement speed used in the program refers to the movement speed of the right hand. When the user lowered their right hand, the program did not calculate movement speed and set it to 1. If it detected that the user's right hand movement speed was lower than the threshold, the threshold counter was initiated to determine whether the defined mouse stoppage time was reached and send the corresponding preset feedback on mouse operation.

The investigated subjects totaled around 50 people whom are random. Each of them do not know each other. Handwritten digits 0–9 were tested by each person 10 times. Firstly, the better E needs to be selected to consider the recognition rate. In this experiment, the different range of E is consideration. A network layer was trained five times the error threshold E=0.0003, E=0.003, and E=1. Figure 17 shows the impact of the error threshold on the recognition rate. Numbers 0–9 denote the handwritten digits 0–9; E denotes the error threshold; and Avg. denotes the average recognition rate of 0–9. Regarding the experimental result, the impact on the average recognition rate was acceptable when E=0.003 and E=1. But, the average recognition rate is poor when E= 0.0003. Although both of E=1 and E=0.003 affect the same results at average recognition rate, but when E=1 has a bigger difference at each individual digital recognition rate.

In this experiment, when E=0.003, It was used for testing the impact of the number of hidden neurons on the recognition rate. The results are shown in Fig. 18.

The test results for the impact of the momentum term algorithm on the training speed when the E=0.003, the number of neurons *N*=6, and the learning rate L=0.1 are shown in Fig. 19. Regarding the experimental results, the recognition rate increased by 2 % after the momentum term was included in the algorithm. When the momentum term m=0.9, the network convergence accelerated but the recognition rate did not improve.

Regarding handwriting, the implementation time from digit recognition to channel selection was calculated. Figure 20 shows the average time from digit recognition to channel selection when testing the handwritten digits 0–9. The data reveal that TV channel selection was enabled when users implemented the handwriting function. In this study, the handwriting implementation time did not exceed 40 milliseconds, and the average time did not exceed 30 milliseconds after 50 tests.

5 Conclusion

In this paper, an interactive TV user interface is implemented via the Kinect sensor which was used as a gesture skeleton image input tool. It transformed skeleton information into mouse control signals, and used an interactive hand gesture operation mechanism in conjunction with a handwriting gesture and dynamic gesture control system. In this study, a simple interactive TV interface was developed to create an interactive operation mode and thereby fulfill the research purposes. The recognition rate of the tested dynamic gesture and handwriting recognition system was acceptable, it was over 90 % based on the experiment. After testing, the interactive TV interface was integrated. The average time from handwriting recognition to channel selection did not exceed 30 milliseconds and, therefore, users can experience interaction smoothly by using the proposed interface.

Conflict of interests The authors declare that there is no conflict of interests regarding the publication of this article.

References

1. Abbas Q, Ahmad J, Bangyal WH (2010) Analysis of learning rate using BP algorithm for hand written digit recognition application. Proceeding of International Conference on Information and Emerging Technologies, Karachi
2. Freeman WT, Weissman CD (1995) Television control by hand gesture. IEEE Intl. Wkshp. on Automatic Face and Gesture Recognition, Zurich
3. Gustafson S, Bierwirth D, Baudisch P (2010) Imaginary interfaces: spatial interaction with empty hands and without visual feedback. UIST'10 Proceedings of the 23nd annual ACM symposium on User interface software and technology, New York
4. Hichkleym K, Hollan J (2008) Papiercraft: a gesture-based command system for interactive paper. J ACM Trans Comput-Hum Interact 14(4):1–31
5. Jacob RJK, Girouard A, Hirshfield LM, Horn MS, Shaer O, Solovey ET, Zigelbaum J (2008) Reality-based interaction: a framework for post-WIMP interface. Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, Florence
6. Jeng TS, Lee CH, Chen C, Ma YP (2002) Interaction and social issues in human-centered reactive environment. Proceedings of the 7th International Conference on Computer Aided Architectural Design Research in Asia, Cyberjaya,
   ISBN 983-2473-42-X
7. Kegl B (1999) Principal curves: learning, design, and applications. PhD Thesis, Concordia University
8. Kegl B, Krzyzak A (2002) Piecewise linear skeletonization using principal curves. IEEE Trans Pattern Anal Mach Intell 24(1):59–74
9. Kegl B, Krzyzak A, Linder T, Zegger K (1998) A polygonal line algorithm for constructing principal curves. NIPS 501–507
10. Kuhlman LM (2009) Gesture mapping for interaction design: an investigative process for developing interactive gesture libraries. PhD Thesis, The Ohio State University
11. Laurel B (1991) Computers as theatre. Addison-Wesley Publishing Company
12. Laurel B, Mountford J (1990) The art of human-computer interface design. Published by Addison-Wesley Longman, Boston
13. Lee CH (2002) 「Interactive Media」 - A multimodal approach to interface design for human-computer interaction in digital design environments. Master of Thesis, Department of Architecture, National Cheng Kung University
14. Memmel T, Reiterer H (2008) Model-based and prototyping-driven user interface specification to support collaboration and creativity. J Univ Comput Sci 14(19):3217–3235
15. Oviatt S (1999) Ten myths of multimodal interaction. Commun ACM 42(1):74–81
16. Patsadu O, Nukoolkit C, Watanapa B (2012) Human gesture recognition using kinect camera. 2012 Ninth International Joint Conference on Computer Science and Software Engineering (JCSSE), 978-1-4673-1921-8

17. Ren Z, Meng J, Yuan J, Zhang Z (2011) Robust hand gesture recognition with kinect sensor. MM'11 Proceedings of the 19th ACM international conference on Multimedia 759–760
18. Shotton J, Sharp T, Kipman A, Fitzgibbon A, Finocchio M, Blake A, Cook M, Moore R (2013) Real-time human pose recognition in parts from single depth images. Commun ACM 56(1):116–124
19. Song Y, Demirdjian D, Davis R (2012) Continuous body and hand gesture recognition for natural human-computer interaction. J ACM Trans Interact Intell Syst (TiiS) - Spec Issue Affect Interac Nat Environ 2(1): Article No. 5
20. Zhang J, Chen D, Kruger U (2008) Adaptive constraint K-segment principal curves for intelligent transportation systems. IEEE Trans Intell Transp Syst 9(4):666–677

Jia-Shing Sheu received the MS. degree and Ph. D. degree both from the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, respectively in 1995 and 2002. Currently, he works as an associate professor at Department of Computer Science, National Taipei University of Education, Taipei, Taiwan. His research interests include pattern recognition and image processing, especially focus on real time face recognition, and embedded system.

Ya-Ling Huang received the MS. degree from the Department of Computer Science, National Taipei University of Education, Taipei, Taiwan, respectively in 2012. Currently, he works as a R&D engineer and project leader at XAC Automation Corporation, New Taipei, Taiwan. His mainly responsible products are in the field of develop the MSR, contactless/NFC reader and module of product.