THE DESIGN OF A LOW COST MOTION CHAIR FOR VIDEO GAMES AND MPEG VIDEO PLAYBACK

Chung-Hsi Huang, Jia-Yush Yen,+ Ming Ouhyoung
Communication & Multimedia Lab.
Department of Computer Science and Information Engineering
+Department of Mechanical Engineering
National Taiwan University, Taipei, Taiwan, ROC

Abstract

In this paper, we have developed a low-cost force output device as a user interface and combine it with an MPEG-1 video player to form a virtual reality system for entertainment.

We designed a motor-powered motion chair to create the sensation of being carried in an airplane or a vehicle. To obtain smooth motion, curve fitting and filtering on the motion trajectory are used. The synchronization of video and motion chair is solved with the techniques of motion prediction, while combined with a joystick provides a way for interactive games.

Introduction

In virtual reality applications, there are cases to simulate the driving of a vehicle or to experience a ride on an airplane. Such a system needs a video player or video game plus a motion platform to provide the feedback of motion. But either there are few devices to choose or the prices of existing devices are too expensive. Hydraulic and pneumatic powered devices are commonly used in similar systems but the control system for a small size system is still difficult. Our goal is to reduce the cost for simulation purpose and use digital control for precision and flexibility.

Our design employs a motor-driven 2-degree-of-freedom chair as the force output device and combines the MPEG-1 video decoder to display the scene. We also use a joystick to simulate real-time control of application programs such as video games.

System description

The configuration of the system is shown in Figure 1.

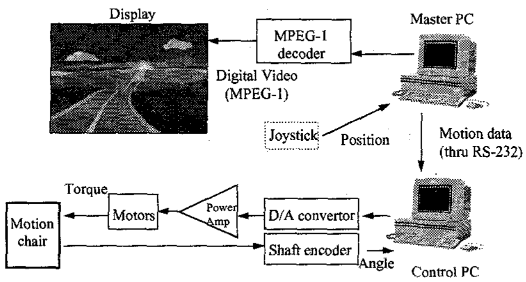


Figure 1. The configuration of the system

As described in the previous section, the motion chair video game system consists of a video game and a motion platform. The video game processor generates two sets of data: a graphic display that emulates the player's view and a two-axis motion trajectory command for the motion platform. As the game is played, the processor continuously generates the motion command for the motion chair to move as if the player is actually riding a flying airplane.

There are several choices of the data format that can serve this purpose. The most common format is for the game processor to generate the absolute attitude for the motion chair. This approach requires fairly fast data link. If the communication is too slow, the chair motion can become discontinuous and can easily hurt the riding comfort. This approach also

Contributed Paper
Manuscript received August 6, 1996

runs the risk of moving across the system singularity. Usually, an interpolator is required in the motion control unit to achieve smooth operation. A second approach in the game process is to generate the attitude difference. In this case the motion can be smoother since the player motion should not be very fast compared with the computing speed. This approach runs the risk of system running into limiting condition and saturating the driver. For the most sophisticate systems, the game process should generate a required acceleration and the motion chair can emulate the required acceleration with the projection of gravity force. In all cases, the system configuration can be represented as in Figure 2.

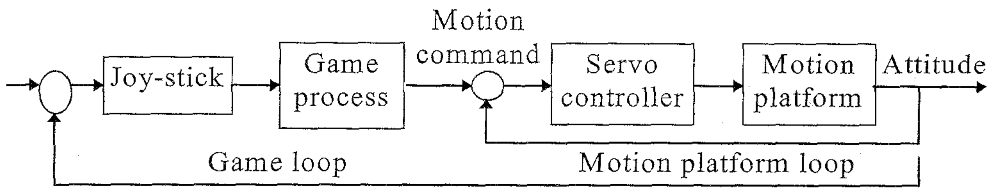


Figure 2. Servo system configuration for the motion chair

In figure 2, the game process generates only the motion command. Feedback path exists only in the motion platform system. The system is therefore a semi-closed loop system. Under this construction, the motion chair loop should have higher servo bandwidth than the game process loop. In this setup, the control system for the motion platform uses a 486 personal computer with A/D board and counters that offers 1 KHz sampling. Since the motor drivers do not saturate, the servo loop can reach around 100 Hz bandwidth. The game process is implemented on a separate 486 PC. The motion command is transferred through the serial communication port that reaches at least 500 Hz sampling rate. The game motion bandwidth can thus go up to around 50 Hz. The motion platform is capable of reflecting the motion command, and the game process is sufficiently fast to achieve a smooth ride.

Structure of the motion chair

The motion chair is designed to maneuver over its base in two axes. We adopt two AC motors with reduction gears as the actuators of the motion chair to create the torque required. Taking a cylindrical shape assumption, the moments of inertia for the human body (assumes to be less than 200 kg) can be represented as

$$I_{xx} = I_{yy} = \frac{m}{4}(R^2 + \frac{H^2}{3})$$

where m is the human body mass $R = 0.2$ m, $H = 2$ m. Therefore, if m = 200 kg, $I_{xx} = I_{yy}$ = 69 kg-m^2. The available angular acceleration can then be calculated from Newtonian 2nd law

$$\vec{\alpha} = \dot{\mathrm{h}} = \frac{d}{dt}(\mathrm{I}_{eff}\vec{\omega})$$

In the above equation, I_{eff} is the effective inertia reflected to the motor through the gear box [4]. If a stepper motor of holding torque 100 nt-m is used, the $\vec{\alpha}$ equals 1.45 rad/s^2. The game system requires a large holding torque and relatively low speed. Therefore, a large gear ratio is used to reduce the effective loading inertia on the servo motor. The basic design of the two-axis motor transmission assembly is shown in figure 3.

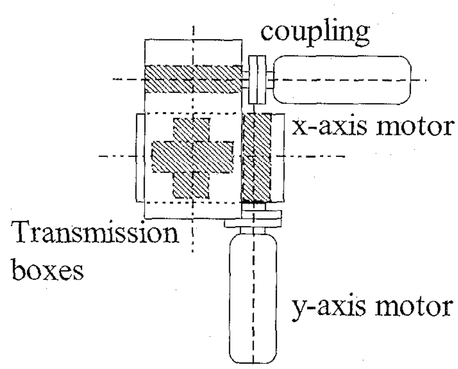
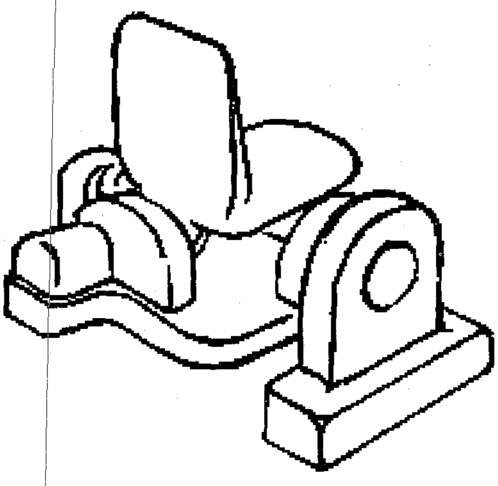


Figure 3. (a) Two-axis motor transmission assembly



(b) *Side view of the Motion Chair*

The motors are coupled in a set of two orthogonal axes. The personal seat is designed to locate the center of mass as close to the coordinate origin as possible to reduce the moment of inertia. Since the rotational axes are orthogonal to each other, the servo information is decoupled. The vector equation of motion for the motion platform reduces to two scalar equations, and each motor axis is controlled by a separate servo loop. The rotation information is measured by two shaft encoders attached at the end of two motors. The mechanical structure of the motion chair is illustrated in Figure 4.

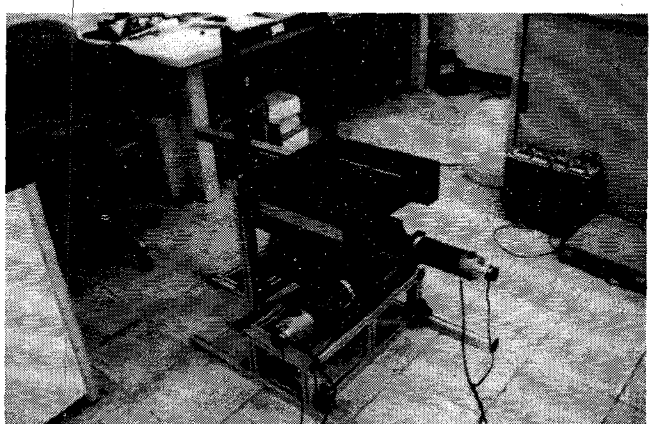


Figure 4. The motion chair

The platform we adopted contains two 486 PC's. A digital-to-analog card, a shaft encoder card and an MPEG-1 decoder card were used in the system for (1) controlling the output torque of motors, (2) collecting rotation information, and (3) playing video files. The communication between two computers is done by an RS-232 cable. In the case of video playing, one PC is used to receive instruction or motion data from RS-232 communication port and controls the motion chair. The other PC uses MPEG-1 decoder card to play video file and send motion data over RS-232 cable to make the video and motion move synchronously.

Preprocessing for motion data

The motion data are the angle values of the motion chair at every sample time. They can be recorded by orientation trackers when the video is taken or from a 2D input device, for example, a joystick.

Although the use of motors can make the cost of the system lower than that of the other actuators, it still has drawbacks. The rotation of gear generates jittering motion from backlash. That is, a user who sits on the motion chair will feel the motion not being smooth. Even a little vibration could be detected when it moves at a small angle. This is caused by two reasons: lower sampling rate and fluctuation of sampled data. The former can be solved by using faster electrical circuit and gears with less backlash. To avoid the latter, we let the motion data pass through a low-pass filter to become a smoother curve. B-spline is adopted for this curve because of its property of local control and second derivative continuity.

B-spline curve

There are many kinds of B-spline curves. We choose the cubic uniform non-rational B-spline to use because the intervals between every continuous pair of motion data are equal and four control vertices are enough to smooth the motion data.

Suppose the ith segment in the B-spline is referred as $\mathbf{Q}_i$. For the four vertices possessed by $\mathbf{Q}_i$ we will refer them as $\mathbf{P}_i$, $\mathbf{P}_{i+1}$, $\mathbf{P}_{i+2}$ and $\mathbf{P}_{i+3}$. Let $\mathbf{Q}_i(t)$ be a cubic spline defined over the interval [0,1] with basis function $\mathbf{B}_k(t)$, (k = 0,..,3) and control vertices $\mathbf{P}_i$, $\mathbf{P}_{i+1}$, $\mathbf{P}_{i+2}$ and $\mathbf{P}_{i+3}$. We can get:

$$Q_i(t) = \sum_{k=0}^{3} P_{i+k} B_k(t)$$

The three later vertices of $\mathbf{Q}_i$ are shared with the three former vertices of $\mathbf{Q}_{i+1}$. By the property of C^0, C^1 and C^2 continuity and convex hull we can solve the basis functions. The matrix representation[3] of $\mathbf{Q}_i(t)$ is as follows:

$$Q_i(t) = T \cdot M_{BS} \cdot P$$

$$= \begin{bmatrix} t^3 & t^2 & t^1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_i \\ P_{i+1} \\ P_{i+2} \\ P_{i+3} \end{bmatrix}$$

The **P** matrix is called B-spline geometry vector. We offer the original motion data into the equation and obtain new data. A pseudo code is given below:

```
for i := 1 to Total_motion_data - 3 do
begin
  take ith to (I+3)th motion data  as values of
            geometry vector P
  set t := 0 and calculate the Qi(t)
  store Qi(t) as new ith motion data
end
```

MPEG-1 video playback

The video applied to the system was a clip of the front view from on top of a flying carpet in a video game called Magic Carpet™. For trial purpose, we captured the video and encoded it in MPEG-1 format. The use of MPEG-1 decoder card can reduce the CPU usage and keep enough frame rate.

The synchronization between motion chair and video

Synchronization is an important consideration in the system. The person who sits on the chair can detect the difference between his view and motion of body if there exists a latency. In the design of a motion chair, the two PC's have their own timers. The PC that plays video uses its timer to get video playing time in second, and subdivide the interval into 20 slices (1/20 second). Then it sends the motion data in every unit to the other PC. The other PC receives the data from its peer and interpolates the data into 5 units. In every time unit (1/100 second) the second PC reads the current angle values from a shaft encoder card, compares the predicted target angles and transmits the appropriate torque values to the DA card to rotate the motors to the correct angles.

Motion prediction

Between the time the motion data sent from the first PC to the torque activated by motors there is a latency. Suppose that the transmission time of RS-232, CPU time and process time of interface cards is constant, the latency can be viewed as constant, too. We compensated the latency by measuring the latency and added an offset to reduce the latency. In other words, the first PC sends the data that should be applied to motors just at the time when it arrives at the motors. So the movement of motion chair should be able to synchronize with the video.

The algorighms used for the motion prediction are either Grey system based [8] or Kalman filter based [9]. We used the Grey system implementation, and given below.

Grey system based prediction method

In the real world, the behaviors of most systems are uncertain. The effects of other systems on the system under monitoring are also unclear. In Grey System theory, the system model is established under a sequence of measured raw data which is generated by a system with unclear system characteristics. The observed tracking data is used to generate a generating sequence on a Grey Generating Space, and a Grey differential model (GM) is applied to fit the generated sequence. By using the established GM, we can predict, analyze, and program the behavior of the original system.

In most cases, the tracking data observed by measuring the system is too random and is insufficient to establish a Grey Model. Some manipulation on the tracking data is needed to get a more regular data sequence, and the obtained sequence is called the generated sequence.

The most commonly used operation of the generated sequence is called the Accumulated Generating Operation (AGO). Let $x^{(0)}$ be the original tracking data sequence and $x^{(i)}$ be the generated sequence for $i > 0$. The AGO is defined as:

$$x^{(i)} = AGO\,x^{(i-1)}, i > 0$$
$$x^{(i)}(k) = \sum_{m=0}^{k} x^{(i-1)}(m), i > 0 \qquad \text{(a)}$$

Since the $x^{(0)}$ are all positive, after applying AGO, the generated sequence $x^{(i)}$ must be a monotonicly increasing sequence and its randomness disappears respectively. Therefore, the prediction model, GM, may be established in the AGO domain.

Let $x^{(0)}$ be the original tracking data sequence with n samples, and $x^{(1)}$ = AGO ($x^{(0)}$), then assume they satisfy the following first order Grey differential model, GM(1,1), with a single variable:

$$x^{(0)}(k) + a\,z^{(1)}(k) = b, k = 1,2,\ldots \qquad \text{(b)}$$
$$z^{(1)}(k) = \frac{x^{(1)}(k) + x^{(1)}(k-1)}{2}, k = 1,2,\ldots$$

which is obtained from the following differential equation:

$$\frac{dx^{(1)}(t)}{dt} + a \bullet x^{(1)}(t) = b$$

Expand Eq.(b) with the n samples in $x^{(1)}$, that is, in the AGO domain, we can obtain:

$$\begin{bmatrix} x^{(0)}(1) \\ x^{(0)}(2) \\ \vdots \\ x^{(0)}(n-1) \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}(x^{(1)}(1)+x^{(1)}(0)) & 1 \\ -\frac{1}{2}(x^{(1)}(2)+x^{(1)}(1)) & 1 \\ \vdots & \vdots \\ -\frac{1}{2}(x^{(1)}(n-1)+x^{(1)}(n-2)) & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \qquad \text{(c)}$$

Let

$$Y = \begin{bmatrix} x^{(0)}(1) \\ x^{(0)}(2) \\ \vdots \\ x^{(0)}(n-1) \end{bmatrix} \text{ and } B = \begin{bmatrix} -\frac{1}{2}(x^{(1)}(1)+x^{(1)}(0)) & 1 \\ -\frac{1}{2}(x^{(1)}(2)+x^{(1)}(1)) & 1 \\ \vdots & \vdots \\ -\frac{1}{2}(x^{(1)}(n-1)+x^{(1)}(n-2)) & 1 \end{bmatrix}$$

Solve Eq. (c) with the minimal square approximation, and we can get a and b from the following equation:

$$\begin{bmatrix} a \\ b \end{bmatrix} = \left[B^T B\right]^{-1} B^T Y \qquad \text{(d)}$$

By solving a, b, and the differential equation, we can get the prediction function $\hat{x}^{(1)}(k)$ for the Grey system in the AGO domain:

$$\hat{x}^{(1)}(k) = \left(x^{(0)}(0) - \frac{b}{a}\right)e^{-ak} + \frac{b}{a}, \text{ for } k \geq 0$$

Applying prediction length, k, in terms of the update rate as k to the prediction function, we can get the predicted data in AGO domain, and then apply it to the operation of the inverse AGO (IAGO) defined in Eq.(e). The output data $\hat{x}^{(0)}(k)$ from IAGO is the predicted data that we need.

$$\begin{cases} \hat{x}^{(0)}(k) = \hat{x}^{(1)}(k) - \hat{x}^{(1)}(k-1), \text{ for } k > 0 \\ \hat{x}^{(0)}(0) = \hat{x}^{(1)}(0) = x^{(0)}(0) \end{cases} \qquad \text{(e)}$$

Here we use an example to show how prediction works by using the 3D tracker data with n=6 previous Qx points, to predict the 7th point. Note that Qx is one of the four parameters in quaternion algebra.

Assume the original sequence $x^{(0)}$ is

$$x^{(0)}(k) = \{x^{(0)}(0), x^{(0)}(1), x^{(0)}(2), x^{(0)}(3), x^{(0)}(4), x^{(0)}(5)\}$$
$$= \{0.0355, 0.0382, 0.0398, 0.0431, 0.0478, 0.0547\}$$

Apply AGO (Eq.(a)) to $x^{(0)}$, which is equivalent to accumulating the sequence

$$x^{(1)}(k)=\{x^{(1)}(0),x^{(1)}(1),x^{(1)}(2),x^{(1)}(3),x^{(1)}(4),x^{(1)}(5)\}$$
$$=\{0.0355,\ 0.0737,\ 0.1135,\ 0.1566,\ 0.2044,\ 0.2591\}$$

Note that the above sequence should be monotonically increasing and solve the differential Eq.(d) and get

$$a\ =\ -0.093907,\ b\ =\ 0.031658$$

Then, the prediction function for the Grey system (in AGO domain) can be formulated as:

$$\hat{x}^{(1)}(k)=(0.0355+0.38953)e^{0.093907k}-0.38953$$

By Eq.(e), we predict the 7th position to be

$$\hat{x}^{(0)}(6)=0.061469$$

Application to video games

Except for the combination with MPEG-1 video playback, the motion chair system can be used in video games. A testing program was written to simulate the real-time control of the system. In the program, we use a joystick to control the chair and make the motion chair move following the movement of the handle of a joystick, see Figure 5.

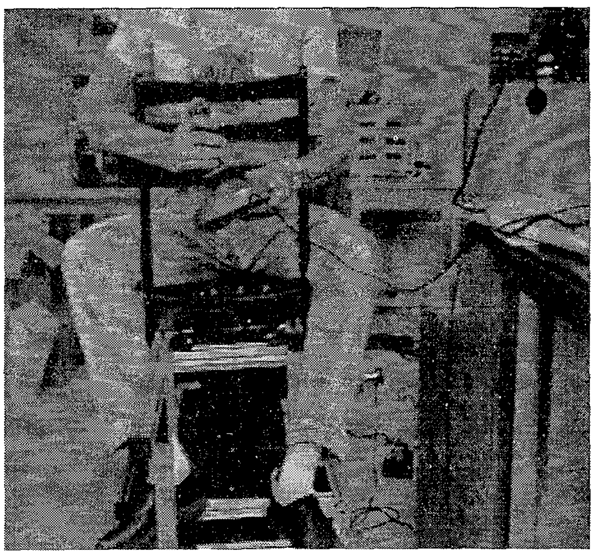


Figure 5. A man sitting on a motion chair holding a joystick for an experiment

The designers of video games can use it as an output device, if they follow the data format sent to RS-232 port.

Data format from motion chair to computer:

31	23	15	7 0
STX	X position	Y position	XXXXXX00

Reserved for future use
Trigger 0 (front) enable=1
Trigger 1 (top) enable=1

Data format from computer to motion chair:

31	23	15	7 0
STX	X force	Y force	Reserved

Figure 6. The communication protocol between a computer and a motion chair through RS-232, where STX means the start delimeter.

We limit the maximum angle that the motion chair can rotate to only 30°, because a user who sits on the motion chair always perceives rotation of a small angle as much wider range. When the user holds the joystick and sits on the motion chair, the system performance is similar to the experience of a flight simulator.

Conclusion

The motion chair is a necessary force output device for some virtual reality systems. With our preprocessing for motion data, the jittering effect of motion can be reduced. Combined with MPEG-1 video or video games, the device will become a virtual reality entertainment system.

With the advantage of easier control using motors and relatively lower cost, the motion chair may be used in a wide range of applications, such as flight simulation, entertainment, education, vehicle driving simulation, remote control, and so on.

Reference

[1] Frederick. P. Brooks, Jr., Ming Ouhyoung, James J,. Batter, P. Jerome Kilpatric, "Project GROPE-Haptic Displays for Scientific Visualization", Vol. 24, No. 4, pp. 236-244, *ACM Computer Graphics*, 1990.

[2] Margaret Minsky, Ming Ouhyoung, Oliver Steele, Frederick P. Brooks, Jr., Max Bhensky, "Feeling and seeing: Issues in force display", Vol. 24, PP. 235-243, *ACM Computer Graphics*, 1990.

[3] Bartels, R., J. Beatty, and B. Barsky, "An Introduction to Splines for Use in Computer Graphics and Geometric Modeling", Morgan Kaufmann, Los Altos, CA, 1987.

[4] NSK Mechanical Design Catalogue, 1993.

[5] M. Ouhyoung, W.-N. Tsai, M.-C. Tsai, J.-R. Wu, C.-H. Huang and T.-J. Yang, "A Low-Cost Force Feedback Joystick and Its Use in PC Video Games", Vol. 41, No. 3, pp. 787-794, *IEEE Trans. On Consumer Electronics*, 1995.

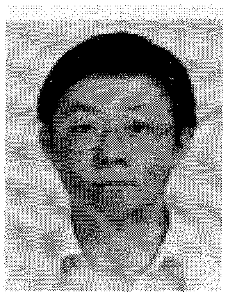
[6] Y.-W. Lei and M. Ouhyoung, "A New Architecture for a TV Graphics Animation Module", Vol. 39, No. 4, pp. 795-800, *IEEE Trans. On Consumer Electronics*, 1993.

[7] C.-W. Shiah, J.-L. Wu, W.-C. Chen and M. Ouhyoung, "A Novel Multimedia Synchronization Model and Its Applications in Multimedia Systems", Vol. 41, No. 1, pp. 12-22, *IEEE Trans. On Consumer Electronics*, 1995.

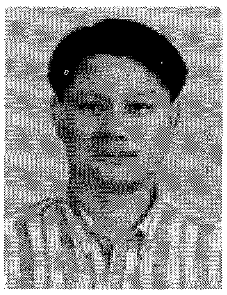
[8] J.-R. Wu and M. Ouhyoung, "Reducing the Latency in Head-Mounted Displays by a Novel Prediction Method Using Grey System Theory", Vol. 13, No. 3, pp. 503-512, *Computer Graphics Forum*, 1994.

[9] R. Azuma and G. Bishop, "A Frequency-Domain Analysis of Head-Motion Prediction", *Proc. of ACM SIGGRAPH* (ACM Computer Graphics), pp. 401-408, LA, USA, 1995.

Chung-Hsi Huang received the B.S. degree in the Department of Computer Science and Information Engineering from the National Chiao-Tung University, Hsin-Chu, in 1991. He received the M.S. degree from the Department of Computer Science and Information Engineering at the National Taiwan University, Taipei, in 1996. His research interests include computer graphics and virtual reality.



Jia-Yush Yen received the B.S. degree from the National Tsing-Hwa University, Hsinchu, Taiwan in 1980, the M.S. degree from the University of Minnesota, Mpls., in 1983, and the Ph.D degree from the University of California, Berkeley in 1989, all in mechanical engineering. During his study at Berkeley, he received the IBM Graduate Fellowship in 1984-1985. Since 1989, he was been with the National Taiwan University, Taipei, Taiwan, where he is currently an Associate Professor of Mechanical Engineering. His research interests are in the areas of modeling and control of computer peripherals, precision measurement systems, and micromechanical systems. He currently serves as the treasurer of the Control System Chapter in IEEE Taipei Section. He is also a member of the ASME.



Ming Ouhyoung received the B.S. and M.S. degree in electrical engineering from the National Taiwan University, Taipei, in 1981 and 1985, respectively. He received the Ph.D. degree in computer science from the University of North Carolina at Chapel Hill in 1990. He was a member of the technical staff at AT&T Bell Laboratories, Middle-town, during 1990 and 1991. Since August 1991, he has been an associate professor in the Computer Science and Information Engineering Department, National Taiwan University and later became a professor in 1995. He has published 67 technical papers on consumer electronics, computer graphics, virtual reality, and multimedia system. He is a member of ACM and IEEE.