http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 11 (1): 110-117, 2012
ISSN 1812-5638 / DOI: 10.3923/itj.2012.110.117

A Research on Hand Position Detection by Camera Array

[1,2]Chai Gongbo, [1,2]Gu Hongbin, [1,2]Wu Dongsu and [1,2]Sun Jin
[1]Institute of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
[2]Research Center of Flight Simulation and Advanced Training Engineering, NUAA, Nanjing 210016, China

Abstract: Hand Position Detection is crucial in real-time interactive virtual reality applications. This study presented a new method to detect a wide range of human hand movement efficiently. Camera array system was used instead of single or binocular vision system to monitor the whole hand existence region. In order to extract stable hand information from video sequence, our method transformed picture's color space from RGB (Red, Green and Blue) to Lab (Lightness, color position between red and green, color position between yellow and blue) and the image histogram was used as the feature parameter of video sequence. Then the differences of single video frequency feature parameter and global feature parameter were putted into neural network. Through neural network training and machine learning the target hand position in three-dimensional space could be calculated. Experimental results showed that this method could detect and locate hand efficiently.

Key words: Hand position detection, camera array, neural network, virtual reality, flight simulator

INTRODUCTION

In flight simulator virtual reality systems, human-machine interaction is one of the key crucial technologies to improve user experience. Instead of explore virtual environment by mouse and keyboard, most people expect to interact with virtual environment by more natural way-by hand (Lai *et al*., 2011; Hsu, 2011).

The conventional approach is using data glove (Kim *et al*., 2009). The user put on a special glove with sensors on hand joint and other locations, for example the back of hand and the fingertips (Ibarguren *et al*., 2010). The more the sensors were putted on glove, the more precise results can be calculated. Although the data glove plan is a sophisticated technology, virtual reality systems applications seldom use it. Not only because the equipment is expensive but also these heave gloves make the operator uncomfortable. Till now the data sensor kind of instrument which is putted on hand or body is mostly used in data acquisition. With the highly accurate information captured by sensors, they can get high quality human movement status and make special effect in movies and computer games (Cao, 2009).

Another method is using camera to detect the object. One major advantage of using camera is that the operator does not need to wear any other accessories. The user can interact with the virtual environment just by naked hand. The other advantage is that the camera is much cheaper than the precision sensors. However, the method using camera costs more central processing unit cycles and requires large amount of storage in computer procession. There are three kinds of detecting and tracking systems based on camera numbers. One approach is using just one camera to achieve detection (Ning *et al*., 2004; Lei and Yongji, 2007). Because of the inevitable intrinsic distort camera lens and the inclined optic axis, the camera need to be calibrated first to get the precise image pixels. Although, there are many auto calibration methods to use, however it needs precision instruments. The normal calibration process is intricate and time consuming (Bullock and Zelek, 2005).

The other approach is using binocular stereo vision (Blake and Wilson, 2011; Peng and Guo-Qiang, 2010; Wan *et al*., 2009) which simulates human being's eye. The advantage is that it can calculate the depth of the object naturally by parallax, just like the way how people get information from outside by eyes. However referring to present artificial intelligence, this method still need complex calibration and distorts correction arithmetic to improve the precision.

The latest approach is using multi-camera or camera array. Multi-camera system is a distributed processing system (Aghajan and Cavallaro, 2009), in which cameras are placed on different corner to supervise the target area.

This system's main advantage is that the cameras in different places can obtain different viewpoint images. So, this system can get more information of the target object, however, it need some sophisticated algorithms to

Corresponding Author: Chai Gongbo, Institute of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China Tel: +86-13776410452

determine the identity of target object acquired in different cameras. The detecting and tracking methods can be divided into calibrated system and machine learning system. The former apply methodologies including multi-view geometry, epipolar geometry, projective transformations (Devarajan *et al*., 2008; Olsen and Hoover, 2001; Hartley, 2008), color correction and distort correction (Li *et al*., 2009, 2010). The latter depends on the sampling, training and learning (Isard and MacCormick, 2001).

On the other side, Camera Array system puts the cameras all together on a flat surface or an arc-surface. Some special characters make camera array system project different with all the methods mentioned above. Some approaches have revealed that the Camera Array system can realize many functions such as synthetic apertures, high performance imaging and high speed video. The synthetic aperture system using the character that the different camera place on the different position has different angle of view. So the occluded surface placed before the camera array can be reconstructed (Vaish *et al*., 2006). The high performance imaging is using combined information collected from cameras to make high resolution pictures (Wilburn *et al*., 2006). The high speed video technology's realization is setting the different sampling time of each camera by computers; through this process one camera with thirty frames per second sampling can become hundred frames per second depending on the number of cameras on the plane (Wilburn *et al*., 2004).

The approach taken in this work is using another character of camera array to detect and track object. The basic principle is that if the camera array is big enough, our target can just be sampled by several local cameras and these cameras' distributions have a certain form. There are certain kind of relationship between the local frame that finds the object and the target object position. However the geometry method needs more precise facilities, so this study use neural network to build this relationship. This paper chooses human hand as the object in order to apply it in virtual environment in future. The experiment shows that this method can locate the position of hand efficiently.

FUNDAMENTAL

The human hand object placed before the camera array will change the color distribution of each camera. In Fig. 1a and b, the camera (named central camera) that just behind the hand will receive the most information of hand target. Cameras which are on the circle around central camera receive less hand information. Cameras on the outer circle can just receive little hand information and far distance cameras can receive nothing target pixel information. In Fig. 1c, if the hand moves to the other position, the central camera will change along with hand. So, this kind of image pixel distribution has a certain relationship with hand position in a plane.

In Fig. 1d, if the hand target keep away from camera array, the image pixel distribution signal will weak and diffuse. So, this image pixel distribution also has a relationship with hand on Z axis position (the axis which is vertical to the camera array).

By the discussion above, the 3D position of target before camera array can be calculated by the image pixel distribution.

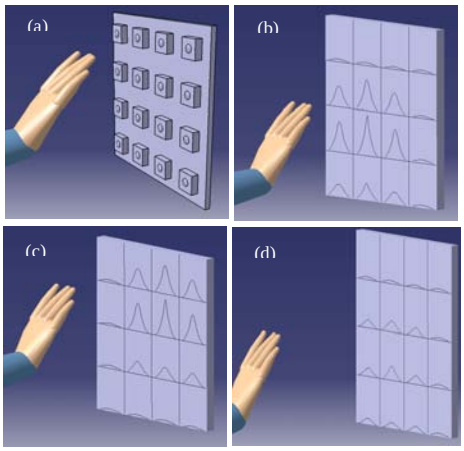

Fig. 1(a-d): Color distribution in lab color space of human hand

COLOR FEATURE EXTRACTION

Color is a powerful character in computer vision. The original signal which is got from cameras is the color information and this information has been divided into different color channel. Most video frequency gathering equipment exports images in RGB format. RGB is a standard color model that can easily transform the color into electronic signals. However, this color space mixes the color with light. The object which is needed to trace should not be affected by different light environment, because the object's diffuse reflection may be different at the different part on one single color object.

In order to reduce the influence of light to the tracking object, color space transformation is used in this study (Chaves-Gonzalez *et al*., 2010), which is from RGB to Lab. Lab is a device-independent model. In Lab color space, L means Luminosity which controls the brightness. So, the brightness influence will be removed just by judging a and b value.

The transform formulas from RGB to Lab are below:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.490 & 0.310 & 0.200 \\ 0.177 & 0.812 & 0.011 \\ 0.000 & 0.010 & 0.990 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

$$\begin{aligned} L &= 116(Y/Y_0)^{1/3} - 16 \\ a &= 500 \cdot [(X/X_0)^{1/3} - (Y/Y_0)^{1/3}] \\ b &= 200 \cdot [(Y/Y_0)^{1/3} - (Z/Z_0)^{1/3}] \end{aligned} \quad (2)$$

where, X_0, Y_0, Z_0 are the coordinates of a reference white point.

In order to make computing easier, the range of the L, a, b value need to be changed in [0, 255]. The next target is finding the distribution of human hand in Lab color space.

In Fig. 2, horizontal axis are equal to a, b. vertical axis is equal to the pixel quantity correspond to a, b values. The distribution is about the hand region which is separated from one image.

The experiment result shows that the distribution of hand color is located at a certain place in the Lab color space. The peak value in Fig. 2 is (a, b) = (133, 134). That means the color of hand is around this point in Lab model. Here the detection area of a, b need to extend:

$$\begin{cases} a \in [129,137] \\ b \in [130,138] \end{cases} \quad (3)$$

If the pixel's a and b value fall in the area above, the pixel presents hand pixel.

COLOR FEATURE FILTER

Because of the noise interference, the color distribution is unstable.

This kind of unstable character makes the sample data imprecise and this error will accumulate and affect the neural network's performance which will be discussed later.

This study use FFT (Fast fourier transform) filter to stable the pixel fluctuation. In Fig. 3, taking the point (a, b) = (133,134) as an example. The red line is pixel quantity fluctuant without any data processing. σ_0 is the standard deviation of this original data. The triangle marker is the pixel quantity after FFT filter. σ' is the standard deviation of this after processing data. The result is shown below:

$$\frac{\sigma'}{\sigma_0} = 0.3470 \quad (4)$$

The relative standard deviation shows that the FFT filter can effectively reduce the fluctuant of image pixel. However this method will bring time delay. So the choice of duration to filter is important. The method in this study select five time frames. The experiment shows that five time frames can reduce fluctuation efficiently without obviously time delay.

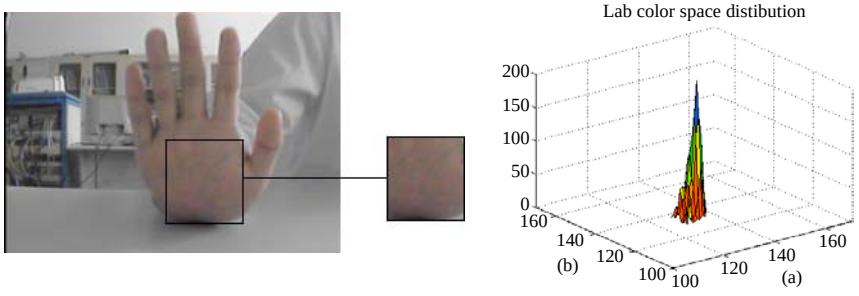

Fig. 2: Color distribution in Lab color space of human hand

Before building single camera feature variable, it needs to make the system remember the background image pixel distribution, because the background image may contain the colors which are the same with hand target.

Figure 4 shows that the image without hand still contains hand color values.

Single camera background feature matrix is defined below:

$$S(b)_{i,j} = \begin{bmatrix} Q(b)_{m_0,n_0} & Q(b)_{m_0,n_1} & \cdots & Q(b)_{m_0,n_k} \\ Q(b)_{m_1,n_0} & Q(b)_{m_1,n_1} & \cdots & Q(b)_{m_1,n_k} \\ \vdots & \vdots & \ddots & \vdots \\ Q(b)_{m_1,n_0} & Q(b)_{m_1,n_1} & \cdots & Q(b)_{m_1,n_k} \end{bmatrix} \quad (5)$$

Single camera feature matrix is defined below:

$$S_{i,j} = \begin{bmatrix} Q_{m_0,n_0} & Q_{m_0,n_1} & \cdots & Q_{m_0,n_k} \\ Q_{m_1,n_0} & Q_{m_1,n_1} & \cdots & Q_{m_1,n_k} \\ \vdots & \vdots & \ddots & \vdots \\ Q_{m_1,n_0} & Q_{m_1,n_1} & \cdots & Q_{m_1,n_k} \end{bmatrix} \quad (6)$$

Formula (6) is the feature variable with hand target.

Matrix $S_{i,j}$ means the ith row jth column camera's feature. $Q_{ml,nk}$ means the quantity of a certain a, b value in Lab color space. This study select $[m_0 \ldots m_l] = [129, 137]$ and $[n_0 \ldots n_k] = [130, 138]$.

Next we define single camera feature variable:

$$D_{i,j} = \sqrt{\sum (S_{i,j} - S(b)_{i,j})^2} \quad (7)$$

In Eq. 7, $D_{i,j}$ is the Euclidean distance. $D_{i,j}$ is the single camera feature variable to reduce the background's influence. However, this presumes that the background is not change during the sample processing. If there is another hand color object come into this system, the parameter $D_{i,j}$ will disable. So, in experiment the background should not be intruded by other hand color things.

If $D_{i,j} > \delta$, it means that the ith row jth column camera found the target and the $D_{i,j}$ presents the target size.

This single camera feature is the foundation to calculate the global feature of camera array.

Global feature difference: After acquired single camera color feature information, the system needs to calculate global feature information of camera array.

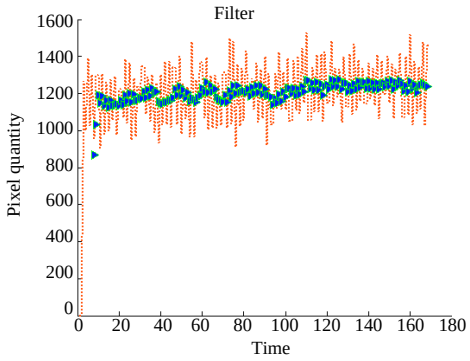

Fig. 3: FFT filter effect

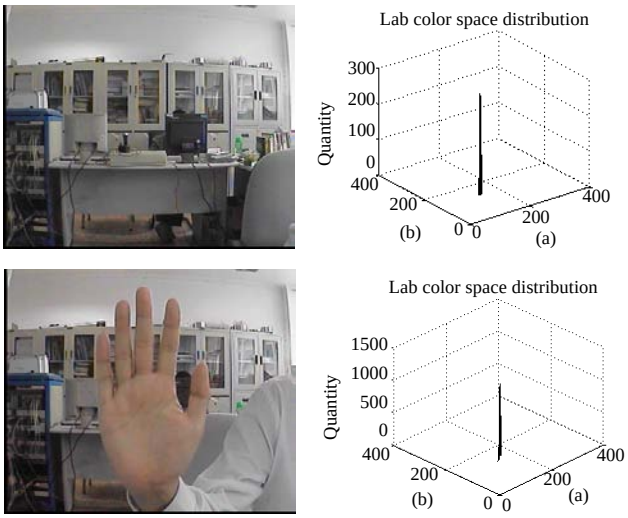

Fig. 4: Camera lab color distribution

In order to describe the global feature, the global feature difference variable is defined below:

$$G = \frac{1}{N}\sum_{i,j}^{N} D_{i,j} \ (D_{i,j} > \delta) \tag{8}$$

where, G is the mean value of the vector $D_{i,j}$. N presents the camera number which found target. In this formula the value of G presents the global color distribution status.

The difference between single camera feature and global feature is defined below:

$$\Delta D_{i,j} = D_{i,j} - G \ (D_{i,j} > \delta) \tag{9}$$

Here we explain why the system still need global feature, because the camera's $D_{i,j}$ value can present color distribution well intuitively.

In 4 by 4 camera array there are 3 by 3 cameras found the target just as shown in Fig. 5. The distribution is on the right side. Parameters is mainly determined by the distance from camera to camera in camera array. From formula (8, 9) we can get the result below:

$$\Delta D_{i,j} = D_{i,j} - G = D - \varepsilon_{ij} - \frac{1}{N}\sum_{i,j}^{N}(D - \varepsilon_{ij}) = -\varepsilon_{ij} + \frac{1}{N}\sum_{i,j}^{N}\varepsilon_{ij} \tag{10}$$

The result in formula (10) shows that $\Delta d_{i,j}$ has no relationship with $D_{i,j}$. From other aspect, $D_{i,j}$ presents the size of our target, because it is the image pixel quantity value. So, calculating $\Delta D_{i,j}$ can reduce the influence of difference hand size or hand incline in a way.

However if someone's hand is small enough or someone has a big hand, the normal hand can be detected by 3 by 3 cameras, their hand will be detected by 2 by 2 cameras or 4 by 4 cameras, this method will disable. For this situation, this tracking system needs a specific video frame sampling and a specific neural network to calculate the result. This study just considers that the hand is normal size.

Matrix P_0 is defined as:

$$P_0 = \begin{bmatrix} \Delta D_{1,1} & \Delta D_{1,2} & \cdots & \Delta D_{1,j} \\ \Delta D_{2,1} & \Delta D_{2,2} & \cdots & \Delta D_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta D_{i,1} & \Delta D_{i,2} & \cdots & \Delta D_{i,j} \end{bmatrix} \tag{11}$$

P is defined below which is the normalization of P_0.

$$P_0 = \frac{P_0}{|D|}, \ |D| = \sqrt{\sum (\Delta D_{i,j})^2} \tag{12}$$

The matrix is P the parameter which will be put into the neural network to train, learn and compute the object's 3D position.

NEURAL NETWORK

Neural network is used to build the relationship between matrix P which is mentioned above and the 3D position of target.

There are many kinds of neural networks. One simple and efficient neural network is net (Rohani and Manry, 1992; Dony and Haykin, 1995).

Figure 6 is BP the network construction. By neural network theory, BP the network with three layers can approximate any nonlinear function. Here the input layer is matrix .

The middle layer is using sigmoid function:

$$f(x) = \frac{1 - e^{-\lambda x}}{1 + e^{-\lambda x}} \tag{13}$$

From formula above we can see that this sigmoid function is differentiable. This character makes the neural network can use method to approximate the weights. The number of neurons in the middle layer is one key parameter, however there is no method to decide this parameter precisely till now. The simplest method is experiment this parameter from a very low value to a certain high value.

$D-\varepsilon_{11}$	$D-\varepsilon_{12}$	$D-\varepsilon_{13}$
$D-\varepsilon_{21}$	D	$D-\varepsilon_{23}$
$D-\varepsilon_{31}$	$D-\varepsilon_{32}$	$D-\varepsilon_{33}$

Fig. 5: Camera array image

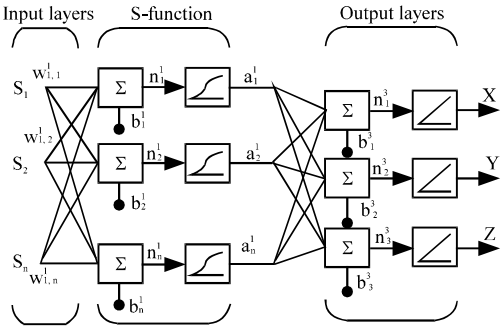

Fig. 6: BP network construction

The output layer is the 3D position X, Y and Z, the 3D position vector is defined below:

$$T = [X, Y, Z] \tag{14}$$

The output layer use linearity function in this paper:

$$F(x) = x \tag{15}$$

After decided the construction of the neural network, the neural network needs a learning method to estimate the weights.

In Fig. 6, the parameter that needed to be estimate is w and b. This study choose momentum BP neural network algorithm. The weights correction is below:

$$w_{t+1} = w_t - \eta_t \frac{\partial E_t}{\partial w_t} + \alpha \Delta w_t \tag{16}$$

$$\Delta w_t = w_t - w_{t-1} \tag{17}$$

$$E_t = \frac{1}{2}\sum (T - T_t')^2 \tag{18}$$

In formula 15, α is the momentum gene. Its value range is from 0.1 to 0.8 generally. Here $\alpha = 0.5$. η is the learning step. The higher of η the faster the neural network will converge, but that will cause the neural network diverge. In our experiment, $\eta = 0.75$ is a good choice.

E is the error function. It's value is the mean square error between network outputs and real position at time t. The neural network learning finish point's judgment standard is whether the neural network has converged the value of E. The construction and the key parameter of neural work have been provided. This paper will not provide lengthy expounding about theories because this method is mature.

EXPERIMENT AND RESULTS

This study use the mini cameras as the image capture equipment. This kind of camera is suitable to construct camera array for its small shape. The camera resolution is 352288. Its frame sample rate is 30 frames per second. The experiment equipment is shown in Fig. 7.

The lights around the camera array are the backup equipment which is used to compensate the environment illumination insufficiency. But it does not mean that the lights condition will be changed during the detection. If this system is placed in a well illuminated environment, these lights should be turned off. After adjusted the environment lights, the lights cannot be changed at will.

Figure 8 is one of video frames for cameras sampling. The number of samples for training is 7×7×3 and for simulating is 10. The samples position presents in Fig. 9. However, there are still tiny fluctuations of matrix P. Here we sample hundred frames at one certain sample position and calculate the mean value of matrix P, then put it into neural network.

In this capture process, the very important thing is the operator's arm must be covered and the things that cover arm cannot be hand color, just as shown in Fig. 9. The face cannot be detected too. The reason has been explained in part 5.

We use high speed image capture card connecting cameras to make sure the image information transmits faster. The capture process is executed by C++ program with the image capture card's SDK.

After acquired data we put these data into neural network.

Here the author use Matlab program to build neural network[26]. Figure 11 is a certain construction neural network's result. The blue line is training sample's output $T_t = [X_t, Y_t, Z_t]$ standard deviation vary with training epochs. The red line is simulation sample's output $T_s = [X_s, Y_s, Z_s]$ standard deviation. Figure 10 we can see the best stop epoch is 19, because after 19 epochs T_s will increase, that means the generalization error increase.

Just as we discussed in part 6, the middle layer's number is an important parameter of neural network. So, here we do some experiment by changing this parameter. The results are shown in Fig. 11. With the increase neuron number in middle layer, the standard deviations have a downward trend. At the point that the middle layer's neuron number is 16, the standard deviations get a local minimum value 0.75. That is the

Fig. 7: Experiment equipment

Fig. 8: Cameras sampling

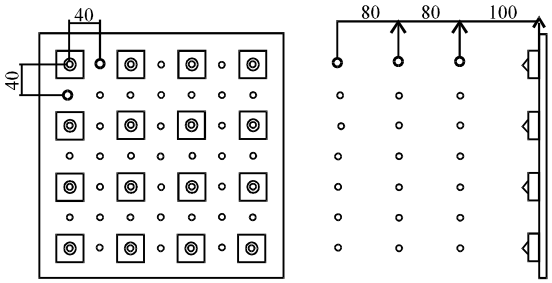

Fig. 9: Sample position (unit mm)

precision of our system's 3D position output. So we choose neural network with 16 neurons in middle layer as our final construction.

After got the 3D position values, we transfer them to the virtual reality systems. Here we use Creator and Vega Prime software to achieve simulation. The hand and aircraft models were built in Creator and the position values were inputted into the Vega Prime SDK program (Multigen-Paradigm Inc., 2005).

Figure 12 shows the virtual hand's movement with naked real hand moving. However there is still tiny

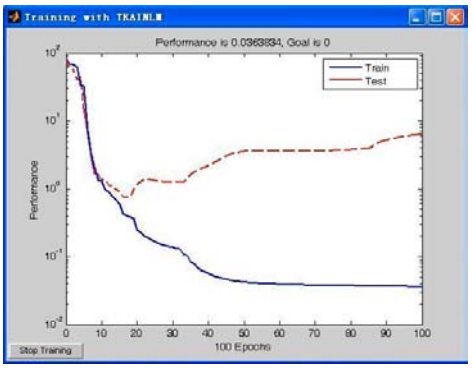

Fig. 10: Neural network training result

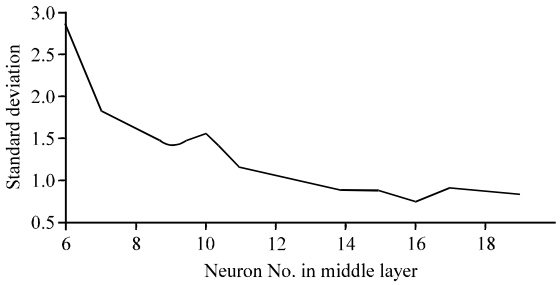

Fig. 11: Results with different neuron number

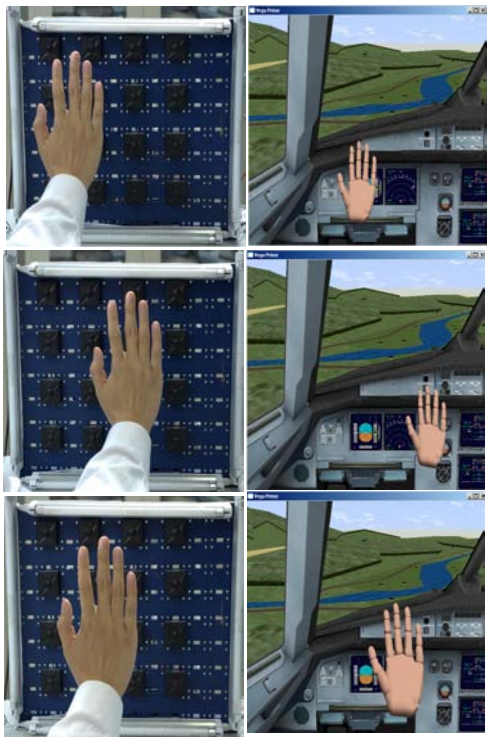

Fig. 12: Virtual reality system simulation

fluctuation. By the experiment results, the final system video frame rate can achieve 15 frames per second.

CONCLUSION

This study presented a new system for flight simulator's virtual reality interacting by using camera array. The advantage of this system lies in that it can detect naked hand in large field without camera calibrations. Because this system only uses the color information, the tracking speed is higher too.

Look to the future, one straightforward improvement to our system would be adding hand region detection and hand gesture recognition. With more information of hand, we can even control the button in virtual reality environment.

REFERENCES

Aghajan, H.K. and A. Cavallaro, 2009. Multi-Camera Networks: Principles and Applications. 1st Edn., Academic Press, USA. Pages: 593.

Blake, R. and H. Wilson, 2011. Binocular vision. Vision Res., 51: 754-770.

Bullock, D. and J. Zelek, 2005. Towards real-time 3-D monocular visual tracking of human limbs in unconstrained. Real-Time Imaging, 11: 323-353.

Cao, Y., 2009. Research on interaction design of virtual reality. Proceedings of the IEEE 10th International Conference on Computer-Aided Industrial Design and Conceptual Design, Nov. 26-29, 2009, Wenzhou, pp: 1340-1344.

Chaves-Gonzalez, J. M., M. A. Vega-Rodriguez, J.A. Gomez-Pulido and J.M. Sanchez-Perez, 2010. Detecting skin in face recognition systems: A colour spaces study. Digital Signal Process., 20: 806-823.

Devarajan, D., Z. Cheng and R.J. Radke, 2008. Calibrating distributed camera networks. Proc. IEEE., 96: 1625-1639.

Dony, R.D. and S. Haykin, 1995. Neural network approaches to image compression. IEEE Proc., 83: 288-303.

Hartley, R., 2008. Multiple View Geometry in Computer Vision. 6th Edn., Cambridge University Press, United Kingdom. Pages: 655.

Hsu, K.S., 2011. Application of a virtual reality entertainment system with human-machine sensor device. J. Applied Sci., 11: 2145-2153.

Ibarguren, A., I. Maurtua and B. Sierra, 2010. Layered architecture for real time sign recognition: Hand gesture and movement. Eng. Appl. Artificial Intell., 23: 1216-1228.

Isard, M. and J. MacCormick, 2001. A bayesian multi-blob tracker. Proc. IEEE Int. Conf. Comput. Vision, 2: 34-41.

Kim, J.H., N.D. Thang and T.S. Kim, 2009. 3-D hand motion tracking and gesture recognition using a data glove. Proceedings of the IEEE International Symposium on Industrial Electronics, July 5-8, 2009, Seoul, pp: 1013-1018.

Lai, Z., G. Hongbin and N. Ben, 2011. Visual hand pose estimation based on hierarchical temporal memory in virtual reality cockpit simulator. Inform. Technol. J., 10: 1809-1816.

Lei, L. and W. Yongji, 2007. Robotic dynamic target recognition and tracking based on the monocular vision. Proceedings of the IEEE Control Conference, July 26-June 31, 2007, Hunan, pp: 193-197.

Li, Z., X. Ji and Q. Dai, 2009. A omni-directional inter-camera color calibration. Proceedings of the 3DTV Conference on the True Vision-Capture, Transmission and Display of 3D Video, May 4-6, 2009, Potsdam, pp: 1-4.

Li, K., Q. Dai and W. Xu, 2010. Collaborative color calibration for multi-camera systems. Signal Process: Image Commun., 26: 48-60.

Multigen-Paradigm Inc., 2005. Vega Prime Programmer's Guide Version 2.0. Multigen-Paradigm Inc., USA..

Ning, H., T. Tan, L. Wang and W. Hu, 2004. Kinematics-based tracking of human walking in monocular video sequences. Image Vision Comput., 22: 429-441.

Olsen, B.D. and A. Hoover, 2001. Calibrating a camera network using a domino grid. Pattern Recognition, 34: 1105-1117.

Peng, Z. and N. Guo-Qiang, 2010. Simultaneous perimeter measurement for 3D object with a binocular stereo vision measurement system. Optics Lasers Eng., 48: 505-511.

Rohani, K. and M.T. Manry, 1992. Nonlinear neural network filters for image processing. Int. Conf. Acoustics Speech Signal Process., 2: 373-376.

Vaish, V., M. Levoy, R. Szeliski, C.L. Zitnick and S.B. Kang, 2006. Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust measures. Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern, 2: 2331-2338.

Wan, L.C., P. Sebastian and Y.V. Voon, 2009. Stereo vision tracking system. Proceedings of the International Conference of Future Computer and Communication, April 3-5, 2009, Kuala Lumpar, pp: 487-491.

Wilburn, B., N. Joshi, V. Vaish, M. Levoy and M. Horowitz, 2004. High speed video using a dense camera array. Proc. Comput. Vision Pattern Recognition, 2: 294-301.

Wilburn, B., N. Joshi, V. Vaish, E.V. Talvala and E. Antunez *et al*., 2006. High performance imaging using large camera arrays. ACM Trans. Graphics-Proc. ACM SIGGRAPH, 24: 765-776.