RGB-D Camera Assists Virtual Studio through Human Computer Interaction

Wei Gao[1, a], and Peng Miao[2, b]

[1] *Shanghai Media & Entertainment Technology(Group) Co., Ltd. Shanghai 200233, China.*
[2] *School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China.*
[a] *gao.wei@opg.cn,* [b] *pengmiao@shu.edu.cn*

Keywords: Virtual studio, human-computer interaction, depth camera

Abstract: Virtual studio has become a common form of television programs production. Traditional virtual studio applies chroma keying technique which requires blue room scene. In addition, for live broadcast program, the lack of on-site human-computer interaction mechanism is not conducive to real-time connection of people and virtual scenes. This paper proposes a virtual studio assistant system based on RGB-D camera. It makes the virtual studio no longer subject to the blue room environment and also provides lots of on-site interactions through gestures between the human and virtual scene. The accuracy and robustness are tested on Kinect device, showing promising performance.

1. Introduction

The virtual studio technique is the combination of computer graphics and chroma keying technique, which has become a popular method for television program production [1]. Figure 1 shows the basic architecture of traditional virtual studio. The traditional technique uses tracking technology to record the motion parameters of the foreground camera in real time, and the virtual camera is controlled to keep the correct perspective relationship between foreground and background by computer graphics method. Then, the real-time tracking signals which are synchronized with the foreground are generated. At the same time, the delayed foreground images are processed by chroma keying and then synthesized with the virtual scene. So that the characters and the virtual background can changed synchronously. Finally, the video signal is output after rendering.

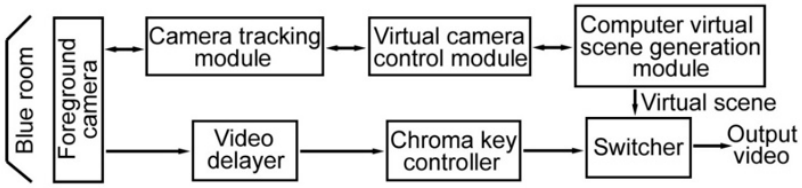

Figure 1. The architecture of traditional virtual studio.

Traditional virtual studio systems are mostly used in single-hosted programs such as weather forecasts and military reports [2]. There are still difficulties in the application of multiplayer

Published by CSP © 2018 the Authors
DOI: 10.23977/msmee.2018.72141

interactive programs such as talk shows. Additionally, it lacks the real-time interaction techniques between hosts and virtual environments in live program [3], usually the virtual content is controlled by the directors and the hosts must learn to perform with no actual material object in the studio.

With the development of machine vision technology in recent years, the RGB-D camera which is able to acquire the depth information simultaneously emerges [4], and some new human recognition and interactive methods based on human skeleton model have been proposed [5]. In this study, we combine RGB-D camera with traditional virtual studio, achieving real-time matting of multiple characters in natural scenes and complex scenes. Moreover, gesture recognition using depth information is developed to provide more natural human computer interaction. By using the proposed system, the virtual studio is equipped with more diverse presentation form and no longer constrained by the blue room environment, providing a new direction for virtual studio development.

2. The Virtual Studio Assisted by RGB-D Camera

The RGB-D camera can simultaneously obtain color images and depth information, where RGB represents color images and D represents depth images. The structured light depth camera is widely-used because of its accuracy of the depth information, high processing speed and high precision with a relatively high resolution, providing a better hardware option for ordinary developers in the fields of 3D information capture, face recognition and gesture recognition etc. The Kinect we use in this study can effectively track and locate 20 skeletal joints at 30 FPS. Furthermore, it can also achieve the simultaneous detection of 6 skeletons.

The system architecture of virtual studio assisted by RGB-D camera is shown in Fig.2. Combining the filming parameters from tracking module and depth information obtained by RGB-D camera, we adjust the parameters such as position information to ensure the real-time and precise segmentation of multiplayer targets and background. The two key jobs for the RGB-D camera are real-time foreground segmentation, namely multiplayer matting and the accurate gesture recognition to generate the corresponding command to virtual studio.

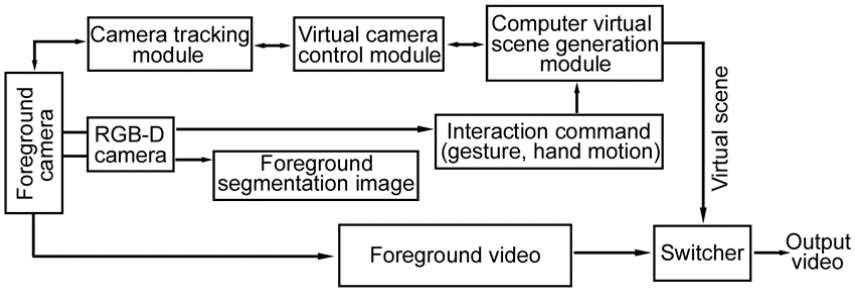

Figure 2. The architecture of virtual studio assisted by RGB-D camera.

In this study, we also developed dynamic gesture recognition, providing the interaction method between human and virtual environment for virtual studio. The gesture recognition can be recognized as a pattern classification procedure. We shall tackle the gesture recognition from 2 aspects, namely feature extraction and mapping method.

3. Gesture Feature Extraction and Mapping

The Kinect we use can capture body silhouettes and generate a corresponded skeleton model in real time, which is composed of 20 joints' 3D coordinates as Fig.3 displays. If we see skeleton as combination of rigid bodies, then the spatial positions of moving limbs is defined by their rotations, therefor we analyze the joints rotational characteristic. For the same gesture, we may get different 3D coordinates due to users' different positions and body build. We use spherical coordinates to

calculate the distance between hip joint and every other joint, along with 2 angles related to this distance.

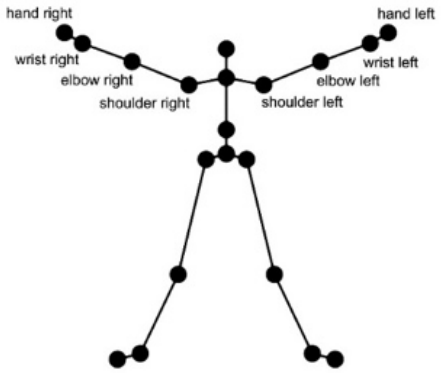

Figure 3. The distribution of skeleton joints

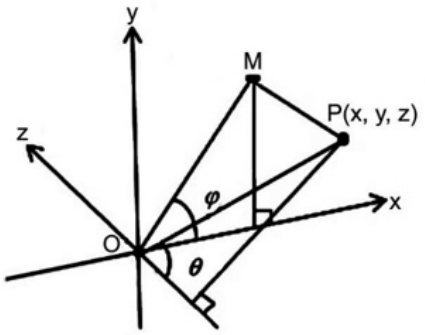

Figure 4. Spherical coordinate

As Fig.4 shows, point P's 3D location (x, y, z) now corresponds the spherical coordinates (γ,θ,φ), where γ represents the distance between point P and point O; θ represents the angle between vector OP and Z axis; φ represents the angle between point M, the point P's projection on plane XOY, and X axis. We chose hip joint as the origin point O and calculate the selected joints' spherical coordinates respectively. By normalizing the joints data obtained by Kinect, we reduce the different body information's influence to experiments. We focus on left and right fingers, wrists, elbows and shoulders, these 8 joints are the main drives to compose gestures. We obtain the positions of these 8 joints and normalize them with the method above, the data set of gesture joints is P = {PLH, PLE, PLS, P_{LK}, PRH, PRE, PRS, P_{RK}}.

DTW (Dynamic Time Warping) algorithm is used in this study. Assuming reference sequence as $R = \{r_1,...,r_i,...,r_{L1}\}$, the test sequence as $T = \{t_1,\ldots,t_j,\ldots,r_{L2}\}$, r_i and t_i represent the rotation angel at time point i, $L1$ and $L2$ are the lengths of the sequences, then the accumulative distance matrix $D(i, j)$ of R and T under nonlinear mapping is :

$$D(i,j) = \min\begin{Bmatrix} D(i,j-1) \\ D(i-1,j) \\ D(i-1,j-1) \end{Bmatrix} + d(r_i,t_j),\ i = 1,2,...,L_1;\ j = 1,2,...,L_2 \tag{1}$$

where $d(r_i,t_j)$ is the distance function of r_i and t_j, usually the Euclidean distance. According to Bellman Equation, then $D(L_1,L_2)$ is the shortest distance, namely the DTW cost between R and T, smaller value come with bigger similarity between R and T.

We choose to use weighted distance [6] and use the formula below to measure a certain joint's contribution to a certain gesture:

$$C_j^g = \sum_{n=2}^{N} Dist^j(f_{n-1}^g, f_n^g) \tag{2}$$

where g represents gesture index, j represents the joint index and n represents the sequence index, $Dist^j(f_{n-1}^g, f_n^g)$ represents the offset of the j^{th} joint, which is calculated by 2 consecutive joint feature vector. Then we normalize the different gestures' contribution by:

$$C_j^g = \begin{cases} C_a & 0 \le C_j^g < T_1 \\ \dfrac{C_j^g - T_1}{T_2 - T_1}(C_b - C_a) + C_a & T_1 \le C_j^g < T_2 \\ C_b & otherwise \end{cases} \tag{3}$$

where C_a, C_b, T_1, T_2 are the thresholds. The weight for a joint to a gesture is finally defined as:

$$W_j^g = \frac{1 - e^{-\beta C_j^g}}{\sum_k (1 - e^{\beta C_k^g})} \tag{4}$$

The distance function $d(r_n, t_m)$ is then modified to: $d(r_n, t_m) = \sum_j Dist^j(r_n, t_m) W_j^g$

The goal is to maximize the distance function of sequences from different gestures and to minimize the distance function of sequences from the same gesture, therefore β is tuned by:

$$\beta = \arg\max_{\beta} \frac{D_B(\beta)}{D_W(\beta)} \tag{5}$$

where $D_B(\beta)$ is the average value of DTW cost of all different gestures pairs, $D_W(\beta)$ is the average value of DTW cost of all same gestures pairs.

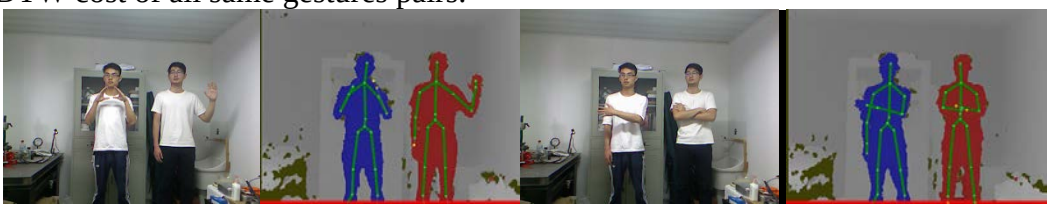

Figure 5 Recognition at runtime

4. Experiments and Results

To assess the accuracy of the gesture recognition method above, we developed a test Kinect program to carry out the experiments. The program allows Kinect to capture human skeleton nodes at 30fps and extract 8 required nodes to process. 7 gestures, Joined Hands, Wave Left, Wave Right, Swipe Left, Swipe Right, Zoom In, and Zoom Out can be defined and identified. Each gesture was performed 12 times by 10 testers with different body builds at different distance to the camera, 4 times each at the distance 1.5~2m, 2~2.5m and 2.5~3m.

We employed leave-one-out cross validation to calculate the accuracy, we take all samples except the samples from one particular tester as the referenced template dataset, and the samples from that tester as test samples. We repeated this procedure for all 10 testers and summed up the results. The recognition at runtime is shown in Figure 5, and the confusion matrix for all testing samples are shown in Table 1. Figure 6 illustrates the accuracy for each gesture at different distance separately. The experimental results show that the method has an average recognition accuracy of 94.88%. The accuracies for SwipeLeft and SwipeRight are relatively higher due to the bigger action reach and speed distinguish them well from other actions. JoinedHands is more likely to be falsely

recognized as ZoomIn and ZoomOut since the samples share more similarity. As long as the Kinect detect the skeleton well enough, gesture can be recognized efficiently at a distance of 1.5~3m.

Table 1. Confusion matrix of the gesture recognition test.

	Joined Hands	Wave Left	Wave Right	Swipe Left	Swipe Right	Zoom In	Zoom Out
Joined Hands	112	0	0	0	0	6	2
Wave Left	0	115	0	5	0	0	0
Wave Right	0	0	110	0	10	0	0
Swipe Left	0	2	0	118	0	0	0
Swipe Right	0	0	0	0	120	0	0
Zoom In	7	0	0	0	0	109	4
Zoom Out	1	0	0	0	1	5	113

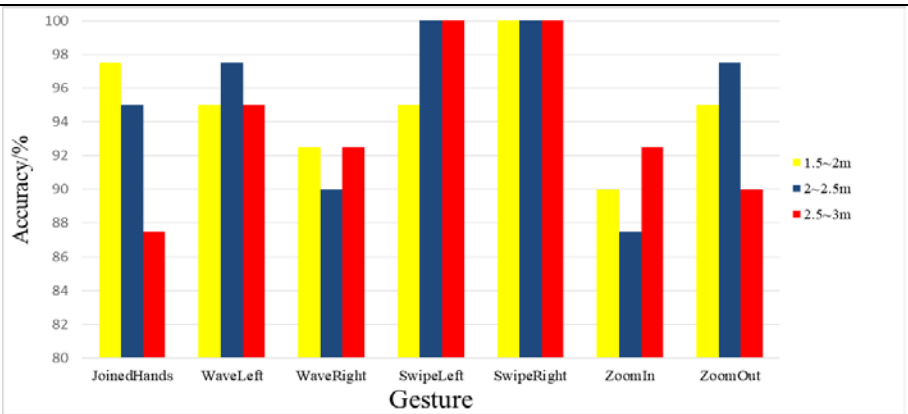

Figure 6 Accuracy for each gesture at different distance

5. Discussion and Conclusion

In this study, the virtual studio assistant system assisted by RGB-D depth camera is presented. The foreground is extracted by threshold segmentation of depth data and transferred to further video fusing, and the gesture is recognized by modified DTW algorithm applied to the depth skeleton model, generating corresponding command to the virtual studio. It achieves multiplayer matting in natural scenes and human-computer interaction by gestures with the help of depth information, bringing new approach for program production and new watching experience for audience.

Acknowledgments

This work is supported by Shanghai Science and Technology Committee (STCSM) (16511105502).

References

[1] L. Fu and X. Liao. Practice and Application of Virtual Studio. Video Engineering, 2011,35(16):95-97
[2] J. He, Z. Han, C. Liang, Y. Jiang, S. Wang and C. Jin. Application of Virtual Studio Technology in Meteorological Service. Radio & TV Broadcast Engineering, 2017, 44(1):63-67.
[3] C. Xia, X. Zhou and H. Wang. Application of Virtual Studio Technology in Live News Program. Advanced Television Engineering, 2008, (10):58-59.

[4] X. Xue, Z. Pan and J. Tong. Depth Camera in Computer Vision and Computer Graphics: An Overview. Journal of Frontiers of Computer Science & Technology, 2011, (06):481-492.
[5] L. Yan and Y. Li. Human Skeleton Extraction Based on Depth Data from Kinect. Electronic Measurement Technology, 2015, (03):39-42.
[6] Z. Yu, W. Cui, and T. Shi. Application and Research on Gesture Recognition by Kinect Sensors. Computer Science, 2016, 43(s2):568-571.