EG UK Theory and Practice of Computer Graphics (2006)
M. McDerby, L. Lever (Editors)

Augmented Reality with Multilayer Occlusion

Yan Feng[1], Yimin Chen[1], Wen Tang[2]

[1]School of Computer Engineering and Science, Shanghai University, Shanghai, China
[2]School of Computing, University of Teesside, Middlesbrough, Uk

Abstract

An algorithm for realizing multilayer occlusion in augmented reality (AR) is presented in this paper. We have designed a special scene graph tree comprised of some special nodes, namely EMO nodes. According to the location of real moving object, different EMO node will be activated in real-time, consequently realizing the multilayer occlusion. Differing qualitatively from previous work in AR occlusion, our algorithm realizes multilayer occlusion, and its application domain involves indoor-field occluded objects, which are several meters distant from the viewer. Previous related work has focused on monolayer occlusion, and near-field occluded objects, which are within or just beyond arm's reach. In addition, BP neural network is improved to correct the nonlinear error of magnetic sensor, consequently to detect occlusion more effectively. Experimental results are provided to demonstrate the multilayer indoor-field occlusion.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Virtual reality

1. Introduction

The goal of augmented reality technology is to produce environments by seamlessly integrating both real and virtual worlds. Realistic occlusion phenomenon is an important part of making AR environment convincing, as occlusion is well-known to be a strong depth cue. Mutual occlusion between real and virtual objects enhances the user's impression that virtual objects truly exist in the real world. This is an essential feature for certain AR applications [HM99][ARDD*01]. Besides, in terms of cognitive psychology, incorrect occlusion confuses users. However, solving the occlusion problem for AR is so far challenging.

In this paper, we propose an algorithm to realize mutual multilayer occlusion between real and virtual scene. This paper is composed of seven sections. Section 3 introduces our error correction method for tracking system. Section 4 describes the algorithm to represent multilayer occlusion. In Section 5, the experiment prototype system and results are described.

2. Related Work

Before resolving the occlusion problem, we should take account of the technique of visually merging the real and virtual worlds, which will influence our method for occlusion.

At present, there exist two kinds of AR interfaces for showing images in which real and virtual scenes are merged. The one is optical see-through system, and the other one is video see-through system.

Conventional optical see-through system has a crucial disadvantage to resolve occlusion. That is, the synthetic objects always appear as semitransparent ghosts floating in front of the real scene. Consequently, they cannot display mutual occlusion of the real and virtual environments correctly.

Though some researchers have made a few attempts to build new optical see-through displays that attack the occlusion problem [KKO00][KKO01][KKO02], these betterment are very restricted on account of the intrinsic characters of optical see-through system. For example, Occlusion-capable Optical see-through Display [KBCW03] is very bulky and weighty. It must be hung from the ceiling by a rubber band to compensate for its weight. So its flexibility and portability are not very satisfying.

In contrast, video see-through displays allow virtual objects to be shown at arbitrary locations, such as in front of a partner's face [KTY00]. They have potential capability of showing correct occlusion between virtual and real scenes. So, video see-through system has been chosen to resolve mutual occlusion problem.

A key to occlusion is how to best depict occluded object in such a way that the viewer can correctly infer the depth relationship between different real and virtual objects.

The KARMA system [FMS93] built on earlier work in computer-generated illustrations to create an AR system that used ghosting and cutaway views to express depth ordering between real and virtual objects. The apparent conflict created by a virtual object overlapping a real object that should occlude the virtual object is thus resolved by surrounding the virtual object with a "virtual hole" in the real object [SCTB*94].

Furmanski et al [FAD02] utilized a similar approach in their pilot experiment. Using video AR, they showed users a stimulus which was either behind or at the same distance as an obstructing surface, and then asked users to identify the location relationship between the stimulus and the obstruction. Only a single occluded object was present in the test. The parameters were the presence of a cutaway in the obstruction and motion parallax. The presence of the cutaway significantly improved users' perceptions of the correct location when the stimulus was behind the obstruction.

Several authors observed that providing correct occlusion in AR requires a scene model [RF00].

Livingston, M.A. et al[LSGH*03] made user accurately interpret occluded objects using a number of display attributes such as opacity, intensity and so on.

3. Nonlinear Error Correction of Tracking System

In our AR system, magnetic sensor was employed to track real moving object, providing us the basis of detecting occlusion. However there's always complicated and nonlinear error in the tracking data, which will influence occlusion detection. So it's indispensable to correct the error of tracker.

We chose BP neural network (BPNN) and improved it to correct the nonlinear error, which has self-learning and self-adapting ability, and can take account of all kinds of disturbance to tracker.

Conventional method to get weights and threshold of BPNN is to train and adjust them gradually according to a certain aptotic rule until they have a better distribution of values. This method is likely to get an inaccurate distribution of value due to plunging into the state of local extremum.

Correspondingly, genetic algorithm (GA) is a search algorithm based on evolution of population, which is characteristic of randomization and global optimization. It can keep searching always in the whole solution space, and is independent of gradient information, but its capability of local search is deficient.

Therefore, we combine BP neural network with genetic algorithm. Above all, we optimize the initial weights and threshold of neural network using GA fleetly, that is to say, we choose out a better search space from solution space, which serves as the initial weights and threshold for BP neural network. And then, BP neural network searches out the optimum solution in this space with its ability of local search.

3.1 Space Encoding

Above all, to establish the mapping relationship between the space of problem and GA space: Every weight and threshold is expressed as a binary string composed of "0" and "1". We can regard the threshold as a weight whose input value is -1. And then, to concatenate all the binary strings into a chromosome.

In this way, the weights and threshold of BPNN are mapped into a genic string, and this mapping is one-to-one. When to concatenate these binary strings, we should notice that these strings corresponding to the weights connected with the same hidden node should be put together. That's because the hidden nodes perform to extract characters and detect characters in neural network. If we detach these strings, the difficulty of extracting characters will be increased, because the genetic operator is prone to destroy these characters.

3.2 Generating of Population Size

It will generate randomly an initial population comprised of N chromosomes. Optimization of weights and threshold based on GA is only in order to choose out a better search space from the solution space, and it is depend on BPNN to accomplish the latter search. Therefore, the optimization based on GA needn't pay too much attention to complexity of calculation, but should try its best to choose out a smaller space containing the global optimum solution. The results of simulation experiment show that the reproduction generation number needn't to be very large, so we may let the population scale, namely the number "N" be bigger.

3.3 Design of Fitness Function

Suppose that $\overline{Y}_{mk}$ and Y_{mk} are respectively the desirable-output and real-output for the output node no. k in the training sample no. m, (m, $k \in Z+$), then the fitness function of GABP problem is:

$$F(x) = (Etotal)^{-1} = \left[\frac{1}{2} \sum_{m} \sum_{k} (\overline{Y}_{mk} - Y_{mk})^2 \right]^{-1} = \left[\frac{1}{2} \sum_{m} \sum_{k} e_{mk}{}^2 \right]^{-1} \quad (1)$$

In the formula (1), Etotal is the energy function of BPNN, and $e_{mk} = \overline{Y}_{mk} - Y_{mk}$ is the output error of output node no. k in model no. m.

3.4 Genetic Operation

Crossover operation will adopt one-point crossover, and crossover probability is *Pc*. Notices that, the cross training for weights and threshold should be detached to proceed respectively. Mutation operation adopts one-point mutation, and mutation probability is *Pm*.

To repeat the above operation until the Etotal run to a stable state, then to stop the evolution. Here an optimized search space is chose out for the latter neural network search. The optimized results through GA act as initial weights and threshold of BPNN.

3.5 Neural Network Structure

In our project, considering the mapping-theorem and the character of our problem, we chose the BPNN model composed of three layers with 4 input nodes and 3 output nodes. The first input value is changeless -1, and the other three input values are respectively *x*, *y*, and *z*, namely the coordinate-value measured by tracker. The three output values are corrected values.

As for the number of hidden nodes, it's significant in characterizing the performance of neural network. Adding the hidden nodes can improve the capability of processing, but will prolong the training time. In our experiment, we employed the step-up method to choose the number of hidden nodes. In the beginning, we chose 6 hidden nodes to training, then to compare the training error with our predefined error value 10^{-4}. Owing to the training error was greater then the predefined value, we added one hidden node, and the rest may be deduced by analogy. When the hidden nodes were added to 10, the training error begins to be less then the predefined value. Therefore, our neural network model is 4:10:3.

3.6 Learning Rate of Neural Network

For the training of BPNN, the learning rate η is very important. With η becoming bigger, the learning speed will be increased, but too big learning rate will result in sway phenomenon. So, to select an appropriate η is very necessary. In the international conference on neural network in 1998, Kung S.Y. put forward a recommendatory formula about learning rate:

$$\eta = 2 / (n_1+1) \quad (2)$$

The variable n_1 in the formula (2) is the number of hidden nodes.

The learning rate η was set to vary from 0.1 to 2.0 to research the influence of different learning rate to the convergence speed. We found iteration times decrease obviously with the increase of learning rate when $\eta<0.5$, and sway phenomenon of iteration time becomes more severe when $\eta>0.5$.

So, referring to the formula (2), we made η decrease from 0.5 to 0.18 during 1000 times training. η was set to decrease with speed from 0.5 to 0.2 in the fore 500 times training, thereby catching the rough probability-formation of input values. In the next 500 times training, η decreased slowly from 0.2 to 0.18, consequently to adapt the weights subtly and catch the exact probability-formation.

In addition, in order to improve further the convergence speed of neural network, we also used the momentum item in our project:

$$\Delta w_{ji}(n)=\alpha\Delta w_{ji}(n-1)+ \eta\delta_j(n)y_i(n), \quad 0<\alpha<1 \quad (3)$$

Here, owing to the limit of length, the parameter optimization and training, as well as the simulation figures about improved BPNN have to be omitted.

4. Occlusion Representation Algorithm

4.1 Division of the Indoor Space and the Virtual Scene

Most related work has focused on near-field occluded objects, which are within or just beyond arm's reach. And the virtual scene is usually made up of only a simple object. Our efforts differ qualitatively from them, because our application domain involves indoor-field occluded objects, which are several meters distant from the user. In addition, our virtual scene is made up of several parts.

We divided the whole indoor space into several parts from the front to the back. Correspondingly, the 3D virtual scene was also divided into different virtual parts (VP) to modeling and render, namely VP1, VP2, VP3, as shown in Figure 1.

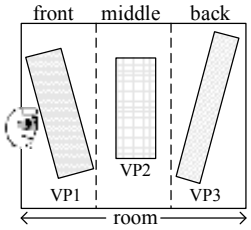


Figure 1. *The layout of indoor space and virtual scene*

When some of the real objects (such as a person) move in the room, we can know the occlusion relationship between the moving real objects and the virtual scene according to in which part of the room the moving objects are, thereby to represent the occlusion effectively. For example, when the moving objects are in the back part of the room, the whole virtual scene isn't occluded by any real object, and then we can directly superimpose the whole virtual scene on the video frames in our process. However, when the moving objects are in the middle or the front part of the room, the real and virtual objects are occluded mutually, and then our rendering process will be more complex.

4.2 Occlusion Representing Technique

At present, in order to realize occlusion effect, z-buffer is usually used in most AR systems. When the real object moves, depth information must be refreshed in z-buffer in real-time, consequently represent mutual occlusion. During this course, however, the band width of display memory is decreased due to using z-buffer.

In our work, the whole virtual scene was divided into three parts according to their depth information, and they were modeling and rendered respectively. We chose TGS Open Inventor to represent the augmented real scene and made full use of the characters of node-tree in Open Inventor to realize mutual occlusion, accordingly save on the band width of display memory without using z-buffer.

The video image captured by CCD is the basis of whole system, but Open Inventor doesn't provide firsthand method of merging real-time video frames, so we have to search a scheme to solve this problem.

Initially, we attempted to make use of SoMovieTexture to manage, because this node can represent the trait of material. We tried to make real-time video as texture and attach it to a big model, consequently to merge the video frames. However, the video frames will become so smaller and smaller that it can not bestrew the whole screen as background when the viewpoint is zoomed out during interaction. So, this scheme was failed.

After repeating research, we found all nodes derive from SoNode, which includes a private member that is GLRender. Further analysis tells us, Open Inventor will always call this member of every node while rendering scene. And yet we couldn't represent the video frames through transforming the rendering mode of model because this member is encapsulated in Open Inventor library in advance. Finally, based on expandability of Open Inventor, we designed a kind of node ourselves and made its GLRender capable of rendering the real video. This node was named Background Node and was put at the uppermost and leftmost location in the scene graph tree, consequently ensured to finish rendering the real video before rendering the virtual scenes. Of course, the virtual object nodes, such as VP1, VP2, VP3, were put in back of the real scene node, as shown in Figure 2.

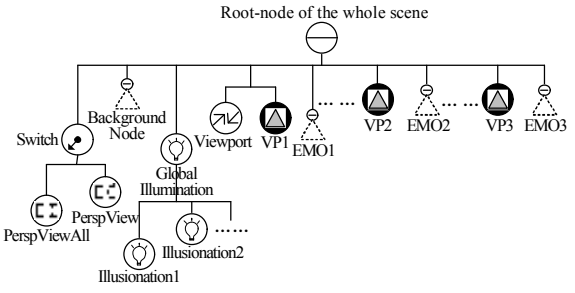


Figure 2. *The structure of whole augmented scene*

In order to realize mutual multilayer occlusion between real and virtual scene, we inserted some special middle nodes in the scene graph tree, such as EMO1, EMO2, EMO3, which denote the extracted moving objects (EMO) in the real scene. According to the rendering character in node tree, we made every EMO node follow a virtual scene node. While some real object moved in the room, we extracted it out from the real-time video frame firstly, and then calculated its new location in the room. According to in which part of the room the moving object was, we could judge which EMO node corresponds to the current moving object. Following commonly adapted terminology, snapshots of a video object at particular time instances are called video object planes (VOP). The shape of VOP can be specified by an alpha map, which may be binary or grayscale. In our system the moving object is just about a VOP. Therefore, setting alpha channel is a necessary step to represent the three-dimensional perspective effect.

Let AR_n represent the alpha plane of the moving object in frame n, defined as follows.

$$AR_n(i,j) = \begin{cases} 1, & if\ pixel(i,j)\ is\ in\ the\ VOP\ in\ frame\ n \\ 0.5, & if\ pixel(i,j)\ is\ at\ the\ VOP\ boundary \\ 0, & otherwise \end{cases} \quad (4)$$

In the above formula (4), we let alpha value be 0.5, which can represents the feathering effect at the VOP boundary for antialiasing, consequently to merge virtual and real scenes more naturally.

After setting alpha channel in the extracted moving object image, we bound it with its corresponding EMO node together. Notice that not all the EMO nodes in the scene graph tree are activated during the rendering course, which lies on the location of the moving object in the room and its occlusion relationship with every virtual scene part.

5. Experiments

5.1 Prototype System

The framework of our AR system is illustrated in Figure 3.

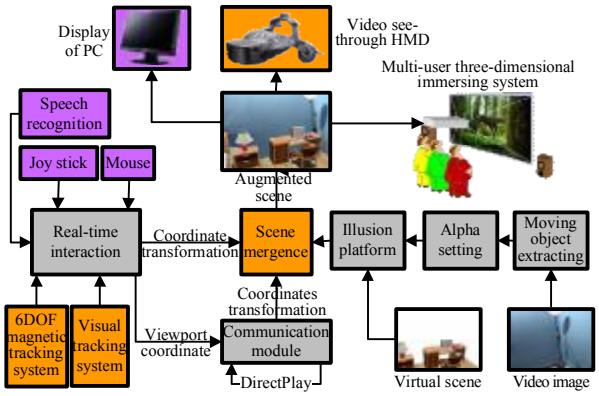


Figure 3. *The framework of prototype system*

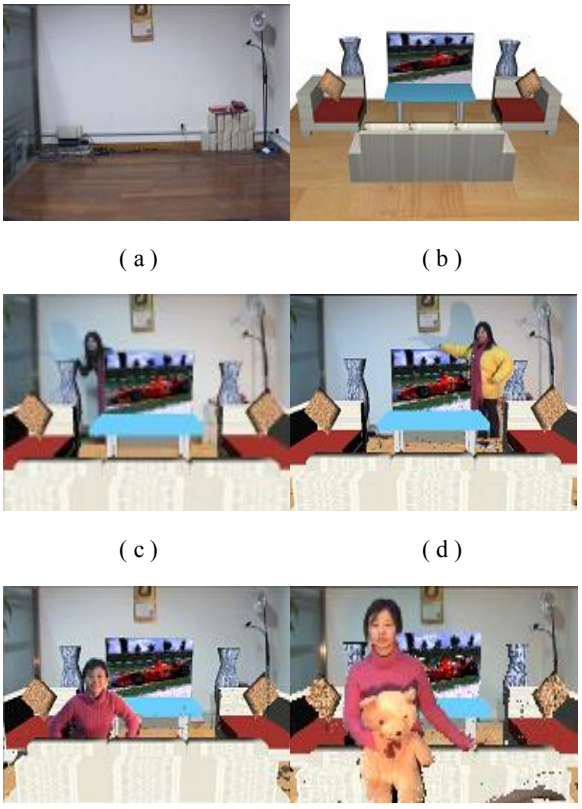
In this system, we chose the V8 model HMD of Co. Virtual Research Systems. This type of HMD was

originally designed for VR system, so we reformed it through building in two CCD cameras (LCH-P49A) and one receiver of magnetic tracker on the HMD. The optical axes of the two cameras are set to be parallel to the viewer's gaze direction. The baseline length between two cameras is set to 65 mm.

The image data, captured by cameras at the rate of 25 frames/second, are sent to PC through a video capture card (10Moons SDK-2000), and then our Illusion Development Platform processes the tracking data and the video image data in real-time, and finally merges real and virtual scenes perfectly.

5.2 Results

Figure 4 shows the result of merged scenes with multilayer occlusion.



(a) (b) (c) (d) (e) (f)

Figure 4. *Result of Multilayer Occlusion*

6. Discussion

In this paper, we have proposed an algorithm of realizing multilayer occlusion in real-time for video see-through augmented reality. As a pilot study, we have developed a prototype system, which is implemented by using existing computer vision techniques on PCs and a HMD with a pair of CCD cameras. The prototype system can produce composite images at video-rate, maintaining correct multilayer occlusions between virtual objects and real objects.

Differing from most previous work, our study emphasizes on multilayer occlusion and indoor-field occlusion, which can represent occlusion more freely. Prominently, owing to our algorithm can resolve occlusion in several meters distant, it can be applied in more domains.

7. Drawbacks of Our System and Future Research

Besides the nonlinear error, magnetic sensor is also limited by its tracking range. So we have considered to superinduce video tracking sensor, and to make the two sets of tracking systems compensate each other for the error and the limited tracking range, consequently to provide us more reliable information to detect occlusion.

Furthermore, in Figure 4 we found there are some unexpected "holes" in the merged scene, which is especially obvious in Figure 4(d) and (f). This is probably because the alpha channel is changeful and prone to be effected by surrounding light. So in the future work, we will farther improve our algorithm, and attempt to introduce threshold idea to eliminate the "holes" phenomenon.

Acknowledgement

Part of this work was funded by the Science Foundation of Shanghai Municipal Commission of Science and Technology (No.025115008).

References

[ARDD*01] Ansar A, Rodriques D, Desai J P, Daniilidis K, Kumar V and Campos M F M. "Visual and Haptic Collaborative Tele-presence", *Computers and Graphics(Pergamon)*, October 2001, 25(5), pp.789-798

[FAD02] Furmanski C, Azuma R, and Daily M. "Augmented-reality visualizations guided by cognition: Perceptual heuristics for combining visible and obscured information". *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2002)*, Sept. 2002, pp. 215–224.

[FMS93] Feiner S, MacIntyre B, and Seligmann D. "Knowledge based augmented reality". *Communications of the ACM*, July 1993, 36(7):52–62.

[HM99] Hirokazu Kato, Mark Billinghurst, "Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System", *Proceedings of 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR '99)*, Oct. 20-21, 1999, pp. 85 - 94

[KBCW03] Kiyokawa K, Billinghurst M, Campbell B, Woods E, "An occlusion capable optical see-through head mount display for supporting co-located collaboration",

Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality, Oct. 7-10, 2003, pp. 133-141.

[KKO00] Kiyokawa K, Kurata Y and Ohno H. "An optical see-through display for mutual occlusion of real and virtual environments", *Proceeding of IEEE and ACM International Symposium on Augmented Reality (ISAR 2000)*, Oct. 5-6, 2000, pp. 60 – 67

[KKO01] Kiyokawa K, Kurata Y, and Ohno H, "An Optical See-through Display for Mutual Occlusion with a Real-time Stereo Vision System", *Elsevier Computer & Graphics, Special Issue on "Mixed Realities - Beyond Conventions"*, 2001, 25(5), pp. 765-779.

[KKO02] Kiyokawa K, Kurata Y, and Ohno H, "Occlusive Optical See-through Displays in a Collaborative Setup", *Proceedings of the ACM SIGGRAPH 2002, Conference Abstracts and Applications (Sketch)* , San Antonio, 2002, p.74.

[KTY00] Kiyokawa K, Takemura H, Yokoya N, "SeamlessDesign for 3D Object Creation", *IEEE MultiMedia*, Jan.-March, 2000, 7(1), pp. 22-33.

[LSGH*03] Livingston M A, Swan J E II, Gabbard J L, Hollerer T H, Hix D, Julier S J, Baillot Y, Brown D. "Resolving Multiple Occluded Layers in Augmented Reality ", *Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality*, Oct. 7-10, 2003, pp. 56-65.

[RF00] Rolland J P, Fuchs H. "Optical versus video see-through head-mounted displays in medical visualization". *Teleoperators and Virtual Environments*, June 2000, 9(3), pp. 287-309.

[SCTB*94] State A, Chen D T, Tector C, Brandt A, Chen H, Ohbuchi R, Bajura M, and Fuchs H. "Case study: Observing a volume-rendered fetus within a pregnant patient". *Proceedings of IEEE Visualization '94*, 1994, pp. 364–368.