Cooperatively Resolving Occlusion Between Real and Virtual in Multiple Video Sequences

Xin Jin, Xiaowu Chen*, Bin Zhou, Hongchang Lin
The State Key Laboratory of Virtual Reality Technology and Systems
School of Computer Science and Engineering, Beihang University
Beijing, China
*Corressponding author: chen@buaa.edu.cn

***Abstract*—The occlusion between real and virtual objects influences not only seamless merging of virtual and real environments but also users' visual perception of orientations & locations and spatial interactions in augmented reality. If there exist a large amount of video sequences for representing the real environment, and each video sequence utilizes computer vision algorithms to deal with all of occlusions between real and virtual, this often drops down the real-time performance of augmented reality system. This article proposed an approach of cooperatively resolving the occlusion between real and virtual based on multiple video sequences in augmented reality scene. Firstly it analyzes the occlusion relations between virtual and real objects in initial video sequences with their intrinsic parameters and poses, and obtains the spatial relations among video sequences through 3D registration information. Secondly, for each video sequence, it divides and codes the perception regions of relative augmented reality scene. Lastly, according to the spatial relations of video sequences, the known occlusion relations in initial video sequences and the code data of perception regions, three types of occlusion relations including real occluding virtual, virtual occluding real and non-occlusion are detected out and represented in augmented reality scene. Some experimental results show that this approach can reduce redundant calculations on the way of resolving the occlusion between real and virtual objects, and improve the performance of generating augmented reality scene, especially which includes plenty of video sequences and occlusion relations of virtual occluding real or non-occlusion.**

Keywords-component; Virtual Reality; Augmented Reality; Video Sequence; Occlusion; Region Code

I. INTRODUCTION

Occlusion between real and virtual objects is a significant visual cue for user to understand the spatial relationship in augmented reality (AR) scene, especially in the case that when a virtual object is occluded by real object. In a shared augmented reality environment, multiple users perceive real objects via video sequences obtained by cameras and users' viewpoint directions are the same as the corresponds cameras'. As shown in Figure 1. and Figure 2. , users are distributed in 3 types of occlusion regions, i.e. non-occlusion (C_1 and C_3), real occluding virtual (C_2) and virtual occluding real (C_4) regions. Usually, for those in non-occlusion or virtual occluding real region, virtual objects could be often drawn directly upon real scene video without any depth estimation. When a new user comes in such shared AR scene, if which type of occlusion region it belongs to could be prejudged, the occlusion handling time may be saved (in C_1, C_3 and C_4). This article aims make computer more intelligent with the help of other computers in occlusion prejudging of the augment reality scene to avoid unnecessary computation cost of depth information.

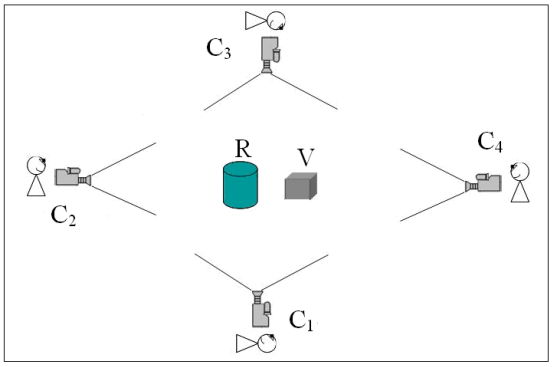

Figure 1. A shared AR scene. The cylinder represents a real object (R) while the cubic represents a virtual object (V). 4 users are in a shared augmented reality scene from 4 typical directions. Screenshots of each view are shown in Figure 2. as an example.

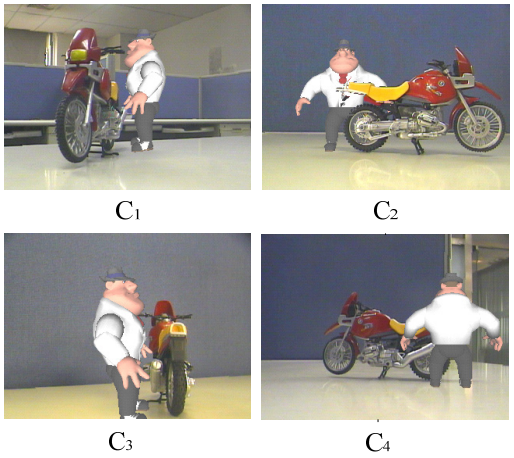

Figure 2. Example of a shared AR scene. The human is a real object while the motocycle is a virtual object. In C_1 and C_3, there are no

occlusions between real and viutual objects. In C_2, the human is occluded by the motocycle while in C_2, the human occludes the motocycle.

In recent literatures, occlusion handling in augmented reality scene has attracted increasing interest. In single view, [1][2] have used 3D (three dimensional) model of real scene while scene depth estimation through stereo matching was employed in [3][4][5][6] and [7] handled occlusions in an interactive way; In multiple views, multiple "clients" shared real scene depth from one depth sensor "server" in [8] while [9] adopted one "tracking camera" from the top to estimate occlusion information of the user ("performance camera"). Obtaining 3D geometrical model or computing depth maps of the real scene are very time-consuming in single view. But to the best of our knowledge, none of current researches have prejudged the necessity of occlusion handling of a particular view. In a shared AR scene, unlike [8] and [9] who took only a single view as the depth information provider, we focus on how to prejudge whether the new views need making such time-consuming depth computation based on the subset view(s) selected from existing view(s).

The occlusion relations between virtual and real objects are analyzed by depth map estimated through stereo matching [10] in initial views. For both initial and new views, the pose parameters are estimated through 3D registrations with spatial relations among them obtained by shared calibration objects (i.e. marker). Then the occlusion region types of initial views are estimated and coded through depth maps. The new views' occlusion region codes (corresponding to occlusion types) are estimated by spatial relations and region codes of a selected subset of existing views. For those in non-occlusion or virtual occluding real region, depth computing will be discarded, while only those in real occluding virtual region need estimating depth map. Some experimental results show that this approach can reduce redundant calculations on the way of resolving the occlusion between real and virtual objects, and improve the performance of generating augmented reality scene, especially which includes plenty of video sequences and occlusion relations of virtual occluding real or non-occlusion.

The contributions of this paper include: (1) in a shared AR scene, a cooperative occlusion handling approach, (2) a method of prejudging occlusion regions based on multiple video sequence.

II. 3D Registration and Spatial Relations of Multiple Video Sequences

3D registration projects a 3D point (X,Y,Z) of real or virtual objects onto a 2D pixel (u,v) video frame through camera intrinsic and extrinsic parameters [11]. Let video sequence C's registration matrix be M_c, then:

$$Z_c\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{M}_1\mathbf{M}_2\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \mathbf{M}_C\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (1)$$

Where M_1 is the intrinsic matrix while M_2 the extrinsic matrix. The spatial relations of multiple video sequences could be estimated through a shared marker as shown in Figure 3.

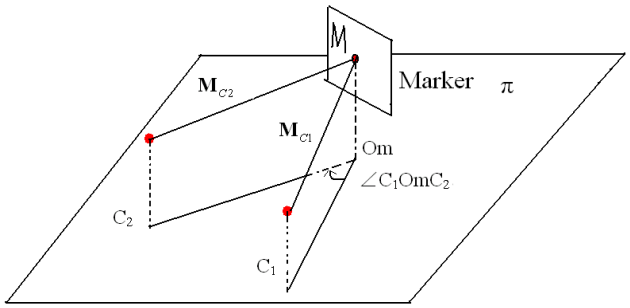

Figure 3. Spatial Relations between 2 video sequences.

Without loss of generality, we assume that users view the AR scene horizontally which is suit for most cases. Occlusions handling in the vertical direction could be similar to the horizontal and more general directions could be synthesized by this two basis directions. In Figure 3. , video sequences C_1 and C_2's registration matrixes are M_{C_1} and M_{C_2}. The two camera which capturing video sequences and a shared marker which could be viewed by both cameras are projected onto horizontal plane π which is perpendicular to the camera's imaging plane as an assumption. Taking the marker as a example reference object to specify the spatial relation between C_2 and C_2. With C_1 as the starting line, clockwise around the O_m point and reaching the line containing C_2 and O_m, there is the angle $\angle C_1O_mC_2$. Then spatial relations "Left" and "Right" could be define as:

$$Sr(C_1,C_2) = \begin{cases} "Left" , \ \angle C_1O_mC_2 \in [0,180) \\ "Right", \angle C_1O_mC_2 \in [180,360) \end{cases} \quad (2)$$

Where $Sr(C_1,C_2) = "Left"$ means C_2 is on the left of C_1.

III. Cooperative Occlusion Handling Approach

The occlusion relations of each view in Figure 1. are shown in Figure 4. If we define the occlusion relations of video sequence C as $\langle V,R,< \rangle_C$, then we have:

$$\begin{cases} \langle V,R,< \rangle_{C_1} = \langle V,R,< \rangle_{C_3} = \phi \\ \langle V,R,< \rangle_{C_2} = R < V \\ \langle V,R,< \rangle_{C4} = V < R \end{cases} \quad (3)$$

When $\langle V,R,< \rangle_C = \phi$ like C_1 and C_3 or $\langle V,R,< \rangle_C = V < R$ like C_2, it is acceptable to draw virtual objects directly upon video sequences, which not only represent correct occlusion relations between virtual and real objects but also save

computations of depth maps. Only when $\langle V,R,<\rangle_C = R<V$, it is necessary to estimate the depth of real scene.

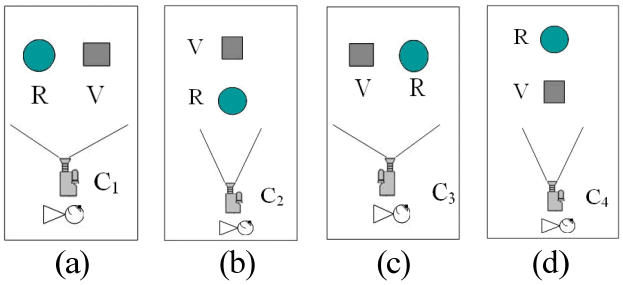

Figure 4. Occlusion Relations between real and virtual objects in 4 views of Figure 1.

So, when new video sequences enter the shared AR scene, if the occlusion relations could be estimated, whose $\langle V,R,<\rangle_C = \phi$ or $\langle V,R,<\rangle_C = \phi$ will save the time-consuming depth calculations.

We propose an approach of cooperatively resolving occlusion between real and virtual based on multiple video sequences in augmented reality scene. In AR scene based on multiple video sequences, when the number of video sequences which describe real scene increases, according to spatial relations among video sequences, prejudge new video sequences' spatial relations cooperatively for reducing the consumptions of real scene depth information.

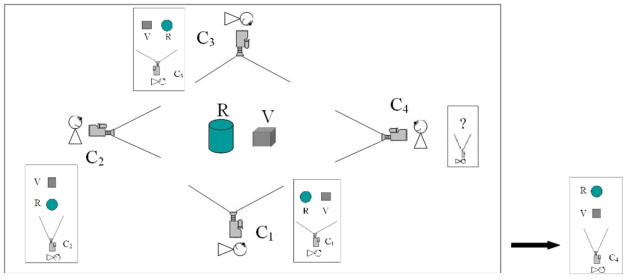

Figure 5. Cooperative occlusion handling example.

For example, as shown in Figure 5. , if these exist video sequence C_1 , C_2 and C_3 with registration matrix M_{C_1} , M_{C_2} and M_{C_3} and the set of occlusion relations $\{\langle V,R,<\rangle_{C_1},\langle V,R,<\rangle_{C_2},\langle V,R,<\rangle_{C_3}\}$, for new video sequence C4 with registration matrix M_{C_4} , using the occlusion judging method which will be described in section IV to estimate the occlusion relations $\langle V,R,<\rangle_{C_4} = V<R$, then virtual object V could be drawn directly upon real scene without depth estimation. And in other situations, if $\langle V,R,<\rangle_{C_4} = \varphi$, C_4 will be treated the same as above. But when $\langle V,R,<\rangle_{C_4} = R<V$, C_4 still needs to calculate the depth to handling occlusions. The whole time of multiple video sequences' occlusion handling will be reduced dramatically especially which includes plenty of video sequences and occlusion relations of virtual occluding real or non-occlusion.

IV. OCLUSSION JUDGING

The key issue of the cooperative occlusion handling approach is how to judge new video sequences' occlusion relations based on spatial and occlusion relations of existing video sequences. This article prejudges the occlusion relations through occlusion region dividing and coding.

A. Oclussion region dividing and coding

As shown in Figure 6. , after projecting the viewer, real and virtual object to the horizontal plan π , two lines L_1 and L_2 divide the whole view region to 4 occlusion regions, including the "real occluding virtual region", the "virtual occluding real region" and the "non-occlusion region" in which user could percept real objects occluding virtual ones, virtual objects occluding real ones and no occlusion between real and virtual objects through video sequences of correspondents region respectively. If the bonding boxes of real and virtual objects are represented by circle, line L_1 and L_2 could be the inner circle tangent.

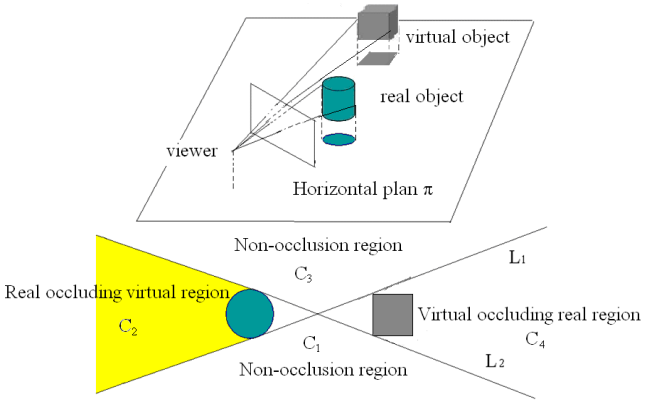

Figure 6. The viewer, real object and virtual object are projected onto horizontal plane π which is perpendicular to the camera's imaging plane. Obviously, video sequence C_2 and C_4 are in the "Real occluding virtual region" and "virtual occluding real region" respectively while video sequence C_1 and C_3 are in the "non-occlusion region".

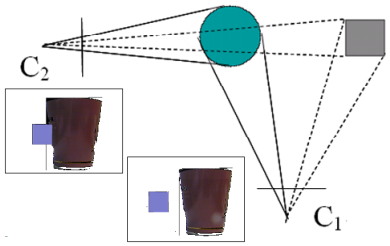

Figure 7. Video sequence snapshots of C_1 and C_2 after background subscription. Before occlusion handling, the virtual object ("the cube") is directly drawn upon video sequences. On the horizontal plane, the "Circle" and the "Rectangle" are the real and virtual object's projections respectively. The contours of virtual objects and real ones ("the cup") are projected to 1D imaging plane of C_1 and C_2.

In practice, without full 3D model of real scene, it is hard to obtain the projection of real objects on the horizontal plane π and the region dividing lines. The silhouette contours of real objects are obtained by background subscription. Then the occlusion relation of each video sequence is coded through the relative positions of real and virtual objects' silhouette contours.

As shown in Figure 7. , the silhouette contours real and virtual objects have no intersection on the imaging plane of C_1, so there is no occlusion relation between real and virtual objects. While, real and virtual objects' silhouette contours have intersection on C_2's imaging plan, so there are occlusion relations in this view, however, it could not be sure that where the virtual object is before or after the real one without depth information. So we need to estimate the occlusion relations via spatial and occlusion relations of existing video sequences.

As shown in Figure 8. , projection of C_1's imaging plane is line S_1, projecting the circle and the rectangle onto S_1, facing the imaging plane S_1 into the direction of the AR scene, scanning from left to right (clockwise on the plane π), when meets the virtual object's contour (the dotted line's projection point on S_1), marking 0, while when meets the real object's contour (the solid line's projection point on S_1), making 1. Then the occlusion code of C_1 is 1100, while occlusion codes of C_2, C_3 and C_4 are 1001，0011，and 0101 respectively. We define such codes as both video sequence and regions' occlusion codes.

B. Oclussion judging method

The non-occlusion region could be specified by the code 1100 or 0011, where real and virtual objects contours have no intersections on the imaging planes. Since occlusion codes of video sequences in virtual occluding real and real occluding virtual regions may be 0101, 0110, 1001 and 1010. It is impossible to judge whether the virtual object is before or after the real one only through occlusion codes. This will be done via existing video sequences.

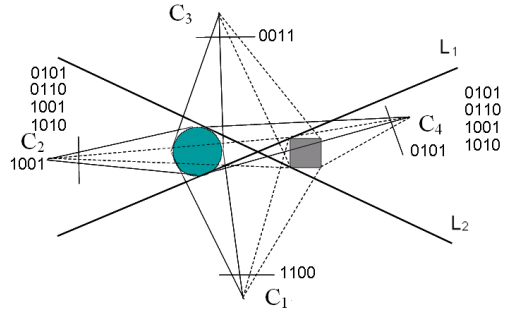

Figure 8. Occlusion region coding. Video sequences in non-occlusion region are coded as 1100 or 0011 while occlusion codes of video sequences in virtual occluding real and real occluding virtual regions may be 0101, 0110, 1001 and 1010.

It could be observed that, selecting the 2 dividing line L_1 and L_2's intersection point as the reference, similar to the definition of "Left" and "Right" in section II, clock wisely:

(1) Video sequences in the region with occlusion code 0011 are on the "Left" of video sequences in the real occluding virtual region.
(2) Video sequences in the region with occlusion code 0011 are on the "Right" of video sequences in the virtual occluding real region.
(3) Video sequences in the region with occlusion code 1100 are on the "Left" of video sequences in the virtual occluding real region.
(4) Video sequences in the region with occlusion code 1100 are on the "Right" of video sequences in the real occluding virtual region.

Based on the above observation, the occlusion judging algorithm could be summarized as in Figure 9. If the output is $\langle V,R,<\rangle_{C_{j+1}} = R < V$ which means the real object occluding the virtual one or "failure judging case" which mean existing video sequences is not sufficient for occlusion judging, depth map will be calculated for occlusion handling; If the output is $\langle V,R,<\rangle_{C_{j+1}} = V < R$ which means the virtual object occluding the real one or $\langle V,R,<\rangle_{C_{j+1}} = \phi$ which means no occlusion between the real and virtual object exists, the virtual object will be draw directly upon the video sequence.

Algorithm. Occlusion judging via occlusion codes

Input: Existing video sequence $\{C_1, C_2, \ldots, C_j\}$ with their registration matrixes $\{M_1, M_2, \ldots, M_j\}$ and occlusion codes $\{OC_1, OC_2, \ldots, OC_j\}$. New video sequence C_{j+1} with its registration matrix M_{j+1}.

Output: Occlusion relations of the new video sequence $\langle V,R,<\rangle_{C_{j+1}}$ or failure judging case.

Calculate the occlusion code of C_{j+1}: OC_{j+1}

(1) If OC_{j+1} is 0011 or 1100, then $\langle V,R,<\rangle_{C_{j+1}} = \phi$, return; Else go to (2).

(2) If 0011 $\in \{OC_1, OC_2, \ldots, OC_j\}$, go to (3); If 1100 $\in \{OC_1, OC_2, \ldots, OC_j\}$, go to (4); Else, return with failure judging case.

(3) Suppose OC_x = 0011, calculate the spatial relations between C_x and C_{j+1} according to M_x and M_{j+1}, if C_{j+1} is "Left" to C_x, then $\langle V,R,<\rangle_{C_{j+1}} = V < R$, return; Else if C_{j+1} is "Right" to C_x, then $\langle V,R,<\rangle_{C_{j+1}} = R < V$, return.

(4) Suppose OC_x = 1100, calculate the spatial relations between C_x and C_{j+1} according to M_x and M_{j+1}, if C_{j+1} is "Left" to C_x, then $\langle V,R,<\rangle_{C_{j+1}} = R < V$, return; Else if C_{j+1} is "Right" to C_x, then $\langle V,R,<\rangle_{C_{j+1}} = V < R$, return.

Figure 9. The occlusion judging algorithm.

V. COMPLEXITY ANALYSIS

Suppose the video resolution is $i \times j$ pixels, the computation complexity of each step in occlusion handling approach based on depth information are rectifying original stereo video frames $O(ij)$, stereo matching $O(iD\lg D)$ (D is the maximum disparity on the scanning line of left and right video frame corresponds pixel)[10],depth testing between real and virtual objects $O(ij)$ (depth testing of every pixel). So, for a single video sequence, the total computation complexity of occlusion handling approach based on depth information is $F = O(iD\lg D + ij)$. For n video sequences in the shared AR scene, the total cost is

$$T_{depth} = nF \tag{4}$$

For the proposed approach, suppose there are n_0 initial video sequences from different views, which need depth estimation, depth testing for occlusion handling and occlusion region coding. The cost of occlusion region coding is $H = O(ij) + O(i)$ (including background subtraction), so the total time cost of each initial video sequence's occlusion handling is $n_0(F+H)$. Suppose the number of new video sequences is n_{new} ($n = n_0 + n_{new}$), the ratio of video sequences in non-occlusion and virtual occluding real region against n_{new} is λ ($0 \le \lambda \le 1$). The occlusion judgments cost of these video sequences are $\lambda n_{new} G = O(ij + c)$ (c is the time of obtain other occlusion region codes, supposed as a constant), for those judged as in real occluding virtual region, which need not only occlusion judgment but also depth estimation and occlusion handling, the cost is $(1-\lambda)n_{new}(F+G)$. So the total cost of n video sequences (with n_0 initial and n_{new} new video sequences) based on cooperatively occlusion handling is

$$T_{coo} = n_0(F+H) + \lambda n_{new} G + (1-\lambda)n_{new}(F+G)) \tag{5}$$

And the difference between the two costs is

$$T_{depth} - T_{coo} = \lambda n_{new} F - (n_{new} G + n_0 H) \tag{6}$$

When most of the video sequences are in real occluding virtual region or get the failure results of occlusion judgments, $\lambda \approx 0$, $T_{depth} - T_{coo} \approx -(n_{new} G + n_0 H) < 0$, since the occlusion judgment increases costs, the cooperative approach cost more time than the depth based approach.

When most of the video sequences are not in real occluding virtual region, which means $\lambda \approx 1$, the two approaches' cost difference is $T_{depth} - T_{coo} \approx n_{new}(F-G) + n_0 H$, since usually G is relative smaller than F, so $T_{depth} - T_{coo} > 0$, the cooperative approach cost less time than the depth based approach.

In more general cases, suppose uniform distribution of video sequences on plane π, $0 < \lambda < 1$, when n increases, since usually G and H is smaller than F, the total cost of cooperative approach may be less than the depth based approach, as shown in Figure 10.

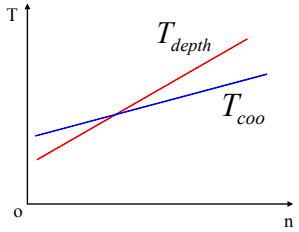

Figure 10. The predict performance of the cooperative approach and depth based approach when the number of video sequences increases.

VI. EXPERIMENTS

In this section, some experiments demonstrate the occlusion handling progresses, results (Figure 11.) and the time consuming of the cooperative occlusion handling approach compared with the depth based approach (Figure 12. ~ Figure 16.).

The equipments used in these experiments including video cameras (JVC® R TK-C1481BE C 1/2") with lens (Computa® R H6Z0812M 1/2"), video capturers with resolution 320*240, and graphics workstation (HP® R xw4200). For implementation, the program is developed using OpenGL1.2, GLUT3.7.6, DirctShow, Intel OpenCV library and Intel Image Processing Library (IPL) [12], and ARTookit 2.7.1 [13].

With experiment setup as in Figure 1. , Figure 11. illustrate 2 typical results after occlusion handling by our approach. The top picture of Figure 11. illustrate that occlusion relations of video sequence C_1 is estimated as non-occlusion, so the virtual object is directly drawn upon the video sequence and the cost of depth map estimation is saved. The bottom picture of Figure 11. illustrate that video sequence C_2 is judged as in real occluding virtual region, so the depth map is estimation through stereo matching for occlusion handling.

In a shared AR scene with multiple video sequences, the time consuming of proposed cooperative occlusion handling approach and depth based occlusion handling approach (which calculates depth map in each video sequence) are compared to show which approach is more efficient in some specific situations.

Figure 12. ,Figure 13. and Figure 14. illustrate the 10 times comparisons in single video sequence's time consuming of the two approaches in the 3 types occlusion regions respectively.

In Figure 12. ,those in the real occluding virtual region have to calculate depth maps after being judged as real occluding virtual. With the proposed approach, they usually cost around 230 ms in a single video sequence which is a little more than depth based approach, with which each video

sequence only calculate depth map without occlusion judging cost (about 160 ms);

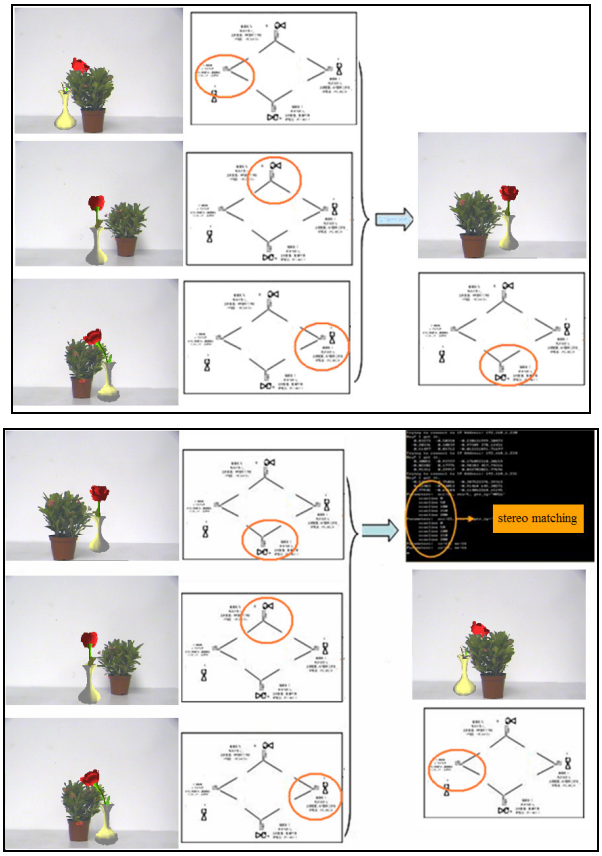

Figure 11. Non-occlusion and real occluding vritual detected. The red flower is a virtual object and the green tree is a real object. Top: Video sequence C_1 enter the AR scene after C_2, C_3 and C_4 have finished occlusion handling and the occlusion codes of them are estimated. The judgement of C_1's occlusion relations return when C_1's occlusion code is 1100, which means there is no occlusion between real and virtual in C_1. Bowttom: Video sequence C_2 is the new video sequence this time. The occlusion code of C_2 is 0101. Since C_2 is to the "Left" of C_1 whose occlusion code 1100, according to step (4) of occlusion judging algorithm shown in Figure 9. , C_2's occlusion relation is real occluding virtual. So after stereo matching and depth map estimation , the virtual flower is registrated partly behind the real tree.

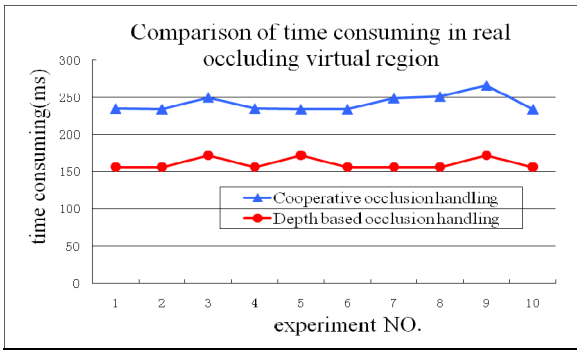

Figure 12. Comparison of time consuming in real occluding virtual region.

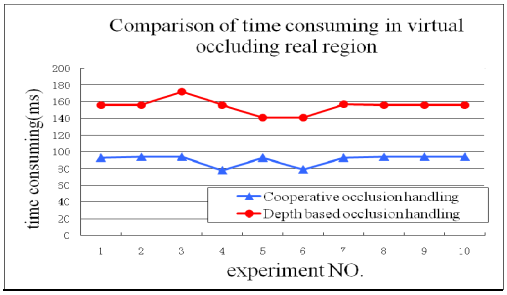

Figure 13. Comparison of time consuming in virtual occluding real region.

In Figure 13. , video sequences in virtual occluding real region need not depth map estimation. They stop at step (3) occlusion judging algorithm shown in Figure 9. and usually cost around 90 ms for occlusion handling while in Figure 14. those in non-occlusion region stop at step (1) which cost about 50 ms. As described in section V, complexity of occlusion prejudging is often much less than depth estimation, so video sequences in this 2 region saved time of depth estimation and often cost less than those directly use depth based approach without prejudging whether they really need to do.

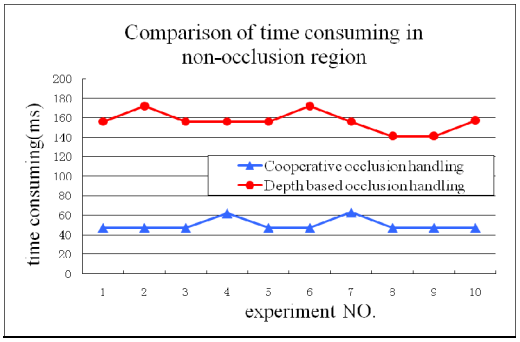

Figure 14. Comparison of time consuming in non-occlusion region

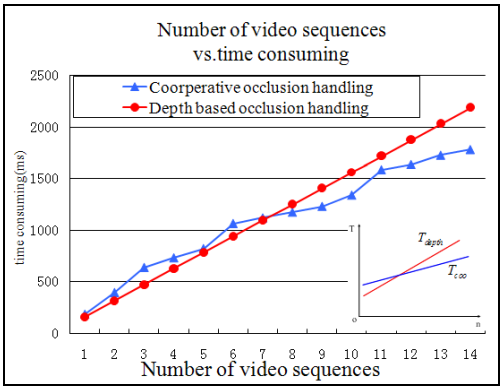

Figure 15. Number of video sequences vs.time consuming.

In Figure 15. we show the performance of the two approaches when the number of video sequences increases in shared AR scene. With uniform distribution of video sequence, similar to what we predict in section V (see the small right down picture), when the number of video

sequences increases, those in virtual occluding real and non-occlusion regions are often more than those in real occluding virtual region. As illustrated in Figure 15. , at the beginning, the cooperative approach cost more than depth based approach with time consumed in occlusion judgments; after 8 video sequences enter the shared AR scene, some video sequences saved time of depth estimation after occlusion prejudging and the total cost become less than what depth based approach does.

If these exist plenty of video sequences in virtual occluding real or non-occlusion regions, the total cost will decrease quickly. As in Figure 16. , we fix the total number of video sequences to 14 with 1 initial video sequence in real occluding virtual region. Through changing cameras' positions, we get 14 types of user distributions which contains 1~13 video sequence(s) outside real occluding virtual region respectively. The time consuming of occlusion handling in all the 14 video sequences in each type of user distribution are compared between cooperative and depth based approaches. When the number of video sequence outside real occluding virtual region is more than 7, the total cost of cooperative approach is less than depth based approach.

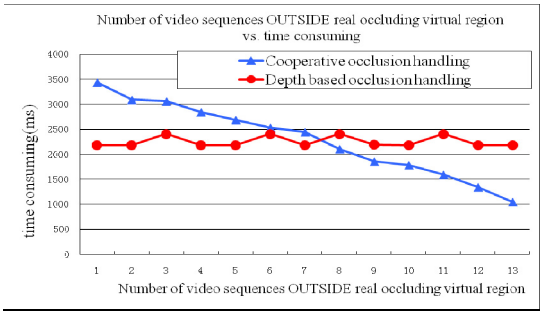

Figure 16. Number of video sequences OUTSIDE real occluding virtual region.

The above experimental results show that this approach can reduce redundant calculations on the way of resolving the occlusion between real and virtual objects, and improve the performance of generating augmented reality scene, especially which includes plenty of video sequences and occlusion relations of virtual occluding real or non-occlusion.

VII. CONCLUSIONS

Occlusion between virtual and real objects not only influences seamless merging in augmented reality but also affects users' visual perception and spatial interaction in augmented reality scene. This article proposes a approach of cooperative occlusion handling in shared augmented reality scene with multiple video sequences, which use relative spatial and occlusion relations of multiple video sequences to resolve occlusion between virtual and real objects cooperatively. When a user enters the shared AR scene, the occlusion in his view is estimated according to the spatial and occlusion relations of existing video sequences with occlusion region codes. If the estimation results show that real objects do not occlude virtual ones, the time and effort on computing depths of the real scene and occlusion handing can be saved. Some experimental results show that this approach can reduce redundant calculations on the way of resolving the occlusion between real and virtual objects, and improve the performance of generating augmented reality scene, especially which includes plenty of video sequences and occlusion relations of virtual occluding real or non-occlusion. This approach could be used more efficient with 3D registration based on nature features, multiple video sequence placement and more types of relations between multiple video sequences.

VIII. ACKNOWLEDGEMENTS

This work was partially supported by Key Technology R&D Program of China (2008BAH29B02), National Natural Science Foundation of China (90818003&60933006), and Specialized Research Fund for the Doctoral Program of Higher Education (20091102110019).

REFERENCES

[1] K. Ahlers, D. Breen, C. Crampton, E. Rose, M. Tuceryan, R. Whitaker, and D. Greer, "An augmented vision system for industrial applications," in Telemanipulators and Telepresence Technologies, vol. 2351, pp. 345-359, SPIE Proceedings, October 1994.

[2] A. Fuhrmann, G. Hesina, F. Faure, and M. Gervautz, "Occlusion in collaborative augmented environments," In EGVE '99 Conference Proceedings, 179-190, 1999.

[3] J. Schmidt, H. Niemann, S. Vogt, "Dense disparity maps in real-time with an application to augmented reality," In: WACV '02: Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision. Washington, DC, USA: IEEE Computer Society, 2002. 225-230.

[4] X. Chen, P. Milgram. "Integration of pointed-based interposition with binocular disparity alignment in stereoscopic augmented reality environments," The Conference on IRIS, 2002.

[5] M. Berger. "Resolving occlusion in augmented reality: a contour based approach without 3D reconstruction," In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Washington, DC, USA: IEEE Computer Society, 1997, pp. 91-96.

[6] K. Hayashi, H. Kato, S. Nishida. "Occlusion detection of real objects using contour based stereo matching," In: ICAT '05: Proceedings of the 2005 international conference on augmented tele-existence. New York, NY, USA: ACM, 2005, pp. 180-186.

[7] V. Lepetit and M. O. Berger. "Handling occlusions in augmented reality systems: A semi-automatic method," Proc. Int'l Symp. Augmented Reality 2000 (ISAR 00), IEEE CS Press, Los Alamitos, Catif., 2000, pp. 137-146.

[8] O. Yuichi, S. Yasuyuki, I. Hiroki, O. Toshikazu, T. Kaito. "Share-Z: Client/Server depth sensing for See-Through Head-Mounted Displays," Proc. 2nd Int'l Symp. Mixed Reality (ISMR 2001), MR Systems Lab, Yokohama, Japan, 2001, pp. 64-72.

[9] P. Fortin, P. Hebert, "Handling occlusions in real-time augmented reality: dealing with movable real and virtual objects,CRV06(54-54),2006.

[10] S. Birchfield, C. Tomasi, "Depth Discontinuities by Pixel-to-Pixel Stereo," Proceedings of the 1998 IEEE Int'l Conference on Computer Vision, Bombay, India

[11] Z. Zhang. "A flexible new technique for camera calibration," IEEE Trans. Pattern Anal. Mach. Intell. 2000, 22(11): 1330-1334

[12] http://www.intel.com/research/mrl/research/opencv

[13] http://www.hitl.washington.edu/artoolkit