

ARFMS: An AR-based WYSIWYG Filmmaking System

Chen Ling*, Wenjun Zhang

School of Film and TV Arts & Technology, Shanghai University

149 Yanchang, Shanghai 200072, China, Ph: 56332030

*Corresponding author, e-mail: lcex@shu.edu.cn

Abstract

This paper proposes a novel "What You See Is What You Get" filmmaking system, called as ARFMS. It provides users with a low cost and easy to use system to make movies. Users cannot only real-time visualize the performance with computer-generated objects, but also visual effects films will be finished without post-production. After finished the screenplay, the performance will be shot on-site using an ordinary camera. CG objects will be simply and effectively controlled by natural interaction. Then users can apply the DR-marker just like an actual scene element, because it takes the advantages of marker-based and markerless-based registration approaches. This paper implements ARFMS to achieve pre-visualization and real-time finish the film using augmented reality technology. The user can perform with the CG character and stage property just like real ones.

Keywords: filmmaking, visual effects, pre-visualization, augmented reality, diminishing reality

Copyright © 2014 Institute of Advanced Engineering and Science. All rights reserved.

1. Introduction

Being the highest peak of entertainment, the film industry has a huge market and potential. Technologies which increase efficiency in filmmaking are required. In the mid 90's, creation of film and TV was adopted the Virtual Reality (VR) technology. 3D animation is aided to shoot in the pre-production stage. This is pre-visualization (PreViz). PreViz is a widespread pre-production technique used in various film productions to plan camera work before the actual shoot. It allows a preparation of often very complicated shots that have to be clear not only to the director but also to the director of cinematography, the set designer, the lighting crew, and other members of the film team including the special effects and visual effects units. Since 2000, PreViz has become an increasingly important technology in the Hollywood. Most the academic community focuses the PreViz using the VR or 3D game technology [1]. However, filmmaking can't be the only indoor shot. Unlike game visualizations that most often concentrate on the representation of the virtual 3D world alone. A novel PreViz, MR-PreViz, using augmented reality (AR) [2][3] has been proposed [2]. In contrast to VR aiming for completely replacing the natural reception of our environment by the provision of artificial input cues, AR has always allowed the user to operate from a common and well know ground, only sparsely modifying the environment by additional artificial virtual content [5]. Adding artificial content or even superimposing the real content completely is something which has been done in the area of visual perception for decades. Therefore, actors can perform with computer-generated objects in the real environment and the outdoor. Sony, Studio Output and Marshmallow Laser Feast tried to produce three short movies "Great films fill rooms" (<http://www.greatfilmsfillrooms.com>). These movies are all in real-time and no post-production.

In this paper, we take the advantages of the MR-PreViz and "Great films the fill rooms" to present a new type film and TV shoot system. We call it as AR Filmmaking System (ARFMS). It is combined with computer vision, human-computer interaction, VR and AR technology to achieve WYSIWYG (What You See Is What You Get) visual effects filmmaking.

An overview of the ARFMS is introduced in Section 2. In Section 3, we describe our ARFMS and present several results. In the end, we discuss possible enhancements.

2. ARFMS

On-Site PreViz uses the digital data of the camera-works synchronized recorded in the actual shooting to achieve registration. And the green screen is replaced with the final VFX content. Then the director can see the final synthetic image to determine whether to retake. However, tracking sensors are complex and expensive. Now, filmmaking also takes place in the actual environment and is commonly for independent filmmaking. Since the introduction of DV technology, the means of production has become more democratized. Filmmakers can conceivably shoot and edit a film, create and edit the sound and music, and mix the final cut on a home computer. With Internet movie distribution, independent filmmakers can exchange with each other. More and more amateurs produce their own movies. However, it is hard for the amateur to make a VFX movie without high-end devices and professional skills. Thus, the goal of ARFMS is to offer a method to make a VFX film, which is easy to use for users without professional skills. Workflow of filmmaking using ARFMS is as follows (Figure 1): (1) preparing the continuity, (2) preparing the virtual objects, (3) actual shooting and (4) modification and compositing.

We proposed a new filmmaking system ARFMS. However, it is a huge project to implement. In the experiment, we implement a simple ARFMS prototype. We employ a laptop computer with two 2.13GHz Intel Core i3 processors and 2.5GB RAM, an ordinary inexpensive camera, a Kinect and a special marker. The framework of ARFMS is shown in Figure 2. After finished the screenplay and continuity, the performance will be shot on-site. In fact, we use an ordinary camera instead of the professional equipment, which is low cost (just 50 RMB in the simulation) and easy for amateur to get. One of ARFMS goals is to satisfy amateurs. Then virtual objects can be added by DR-marker. CG objects can be built in the pre-production stage. And they can be simply and effectively controlled by Kinect. Thenceforward the VFX movie will be finished real time. It works throughout the typical filmmaking three stages process, which are pre-production, production and post-production stages. After the director finished previous work, the film can be shot by ARFMS. In the remaining section, we will introduce two important components of ARFMS below.

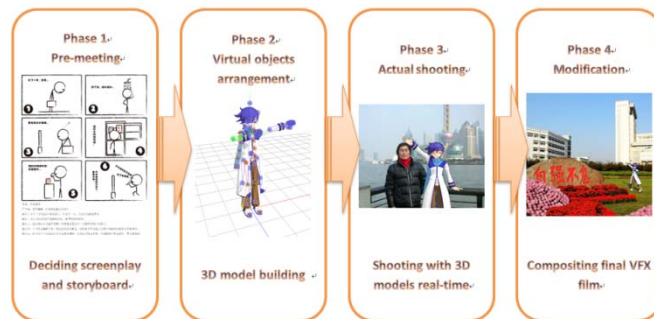


Figure 1. General ARFMS Workflow

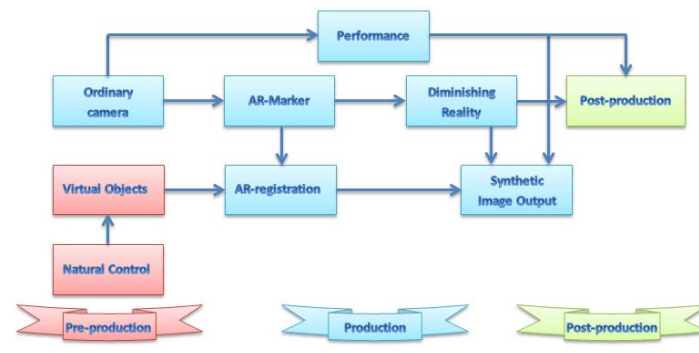


Figure 2. The Framework of ARFMS

2.1. CG Character Control

A range of tools originally designed for high-end graphics are used for PreViz: from Maya and Motionbuilder, to 3D Studio Max, Softimage XSI, and Poser. Each of these programs has its own strengths and weaknesses but all of them allow for some kind of staging of events on virtual sets as well as a definition of virtual camera angles toward the resulting scenes. They are widely used in traditional PreViz nowadays. However, traditional technology to handle motion data uses motion captured data which are attached to the CG character. Then the user can naturally control the virtual character. However, the motion captured equipment is expensive, inconvenient and uncomfortable.

The state of the art is used several cameras in the studio [6] to get the model and action. But it needs a professional studio. It does not suit for amateur users. We take Kinect as a cheap and easy motion captured device. Kinect is attached to the MikuMikuDance (MMD) to control the CG character as shown in Figure 3. The top right corner figure is the user's real action got by a Kinect, and it controls CG boy to do the same action. In this way, we can easily gain the rough action. It will be approximately acceptable for an animation. Then the user can control the character manually to get the better action. Then the action will be saved as a motion file "vmd". The file will be connected with the different character model in the actual shooting on-site.

After finished the virtual character and motion, the user can use them in the ARFMS. Firstly, the CG character is added on the actual on-site image, its size position and direction will be manually adjusted. Then, the actor can play with the CG objects in realtime.



Figure 3. Kinect is Attached to the MMD to Control

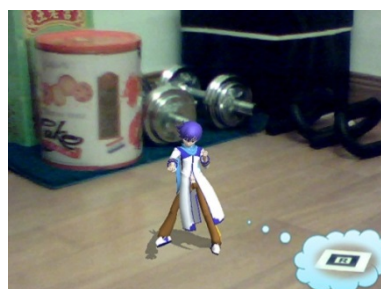


Figure 4. The Disappeared Marker is the Action of the CG Character under the Virtual Character

2.2. DR-marker

There are two types of AR registration nowadays: AR registration based on marker method [7] and AR registration based on markerless method [8]. Due to highly visual feelings, a special marker will be unacceptable. Hence the markerless way will be popular. However, it always occurs to CG objects' flutter using the markerless approach. Flutter means the computer-generated models will shift their position, due to calculation error. Thence we use marker method. Then we use diminishing reality (DR) technology for removing the special marker from a live video stream of the user's real environment. We call this approach as DR-marker [9]. Our approach is based on a simple setup and neither requires any pre-processing nor any information on the structure and location of the objects to be removed.

While AR has always been restricted to adding artificial content to the real environment, DR allows for removing real world content. Existing approaches require complex setups and are not applicable in real-time to seamlessly delete visual content from the observer's view in unconstrained environments. Our DR-marker approach does not have any distance or 3D environment structure information and uses a single camera only. The illustration is shown in Figure 4 that a CG character is superimposed into a real scene. The marker is under the CG model. However, it is not been seen.

2.2.1. Marker Detection and Projecting Model

DR-marker begins with the marker detection. It can be fast and stably detected in real time using the approach ARToolkit (<http://www.hitl.washington.edu/artoolkit/>). This is the most

famous methods in AR. After the quadrate marker detected, we can obtain the transformation matrix. Meanwhile, the all-round marker coordinate can be built. The coordinates of the artificial objects will translate into the marker coordinate by a transformation matrix.

In fact, the marker has a lot of prior knowledge. One of the important information is the size and the location of the square marker. According to this knowledge, the region of interest (ROI) can be picked up. Firstly, we can get the actual scene image without marker. Secondly, we put the marker into the image. The marker will be detected and the vertexes V_i ($i=1, 2, 3, 4$) are gained, clearly shown in Figure 5. Here, we decide the ROI by the points around the vertexes. The points P_i can be defined using the following formula:

$$P_i = \text{Size}(P_i) / \text{Size}(V_i) \cdot (V_i - V_0) + V_0 \quad (1)$$

$$V_0 = 1/4 \cdot \sum_{i=1}^4 V_i \quad (2)$$

Where $\text{Size}(V_i)$ denotes that the side length of the marker. $\text{Size}(P_i)$ denotes that the side length of the ROI, which is the outer rectangle in the Figure 5. V_0 presents the centroid of the vertexes. The ROI is bigger than the marker. The reason is that the area around the marker has a priori knowledge which will be used in the following section. Because the marker is square, the projecting model is the rectangular model. Once the ROI is determinate, the image of the ROI will be saved as the texture of the projecting model. As illustrated in Figure 6, a quadrate marker is on a picture in Figure 6(c). Thenceforward the ROI is computed and the texture in Figure 6(b) will superimpose the marker. In this way, the special marker has been removed from the scene, as shown in Figure 6(a). When the AR marker is detected first time, the ROI is picked up. The affine transformation matrix can be determined. Then the quadrate ROI will be transformed, which takes place distortion. The image of ROI will be a quadrate model. This is the projecting model. When the live video stream comes, the transformation matrix will be calculated fast after marker detection. And the projecting model will change it, just like a computer-generated model in the AR application.

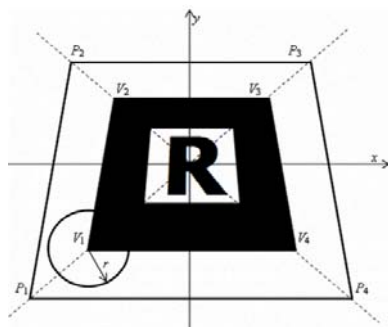


Figure 5. The special marker of Figure 6

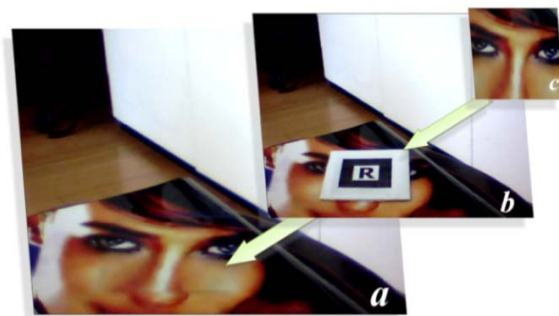


Figure 6. Projecting model of DR-marker (a) marker DR-marker has been removed from the scene; (b) original image; (c) the texture of projecting model

2.2.2. Projecting Model Adjustment

Due to lighting change, the texture of the projecting model will be not consistent. So we need to adjust the projecting model in time. Here we adjust the HSV (Hue Saturation Intensity) of the texture to match the environment. The ROI is around the marker and it is bigger. When the camera captures the images, part of ROI can be seen. That means the area is overlapped between the real scene and the projecting model. So we adjust the texture HSV via comparing with the same points' HSV in the texture and the real image. It is determined as follows:

$$\lambda_{HSV} = \sum_{i=1}^4 HSV(P_i) / HSV(V_i) \quad (3)$$

Where $HSV(V_i)$ is the HSV of the vertexes of the marker. $HSV(P_i)$ is HSV of the ROI vertexes. λ_{HSV} is the coefficient that the projecting model will change. λ_{HSV} ranges from 0 to λ_{max} . λ_{max} is the superior limit. In our experiment, it takes two. Once λ_{HSV} is determined, the new texture $HSV_{new}(tex_j)$ will be obtained by:

$$HSV_{new}(tex_j) = [1 + \alpha_{HSV} \cdot (\lambda_{HSV} - 1)] \cdot HSV(tex_j) \quad (4)$$

Here $HSV(tex_j)$ denotes that HSV of the pixel in the projecting model and $HSV_{new}(tex_j)$ is the new one. α_{HSV} is the weight for different functions of hue, saturation and intensity. In the actual situation, the intensity changes the most, so the α_V will be bigger than α_H and α_S . In this paper, α_H is 0.15, α_S is 0.15, and α_V is 1. In fact, light will not change at all times, so the projecting model is not adjusted every frame. In this paper, we make the texture adjusted automatically every 30 frames. Note that the automatic exposure of the camera should be turned off. Otherwise the images will change at any time. The brightness will be out of control.

A DR-marker simulation result is exhibited in Figure 7, which is a real-time continuous sequence. According to the top left corner of every image, the frames per second are average 30fps. There is a special marker on the ground in Figure 7(a). It is a raw captured frame. And the other three figures are other three post-processing frames, using DR-marker. In Figure 7(b)-(d), an arrow indicates the area which the special marker exists in the original frame. If you watch not carefully, you will not find the difference. It's worth noting that the projecting model adjustment can fit the illumination environment which is relatively stable. However, it is hard to be suitable in the actual complicated outdoor environment.

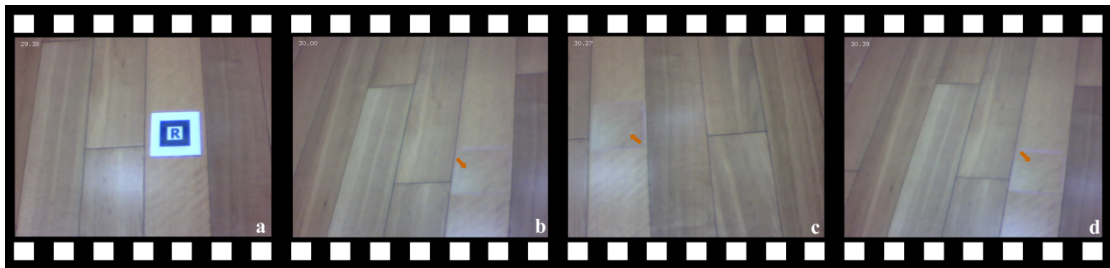


Figure 7. Projecting Model Adjustment Simulation Results. (a) original frame, (b) the 31st frame (the arrow is pointing to the region of DR-marker), (c) the 241st frame, and (d) the 421st frame

2.2.3. Tracking

When the live video stream comes, the AR marker will be detected. It processes fast. However, for the influence of the light, the transformation matrix will happen to change a little. The visual feeling is flutter. So we try to wipe out it by Euclidean distance formula as follows:

$$\begin{cases} \text{not change } T & \|V_i^{last} - V_i^{current}\|_2 < r \quad (i = 1, 2, 3, 4) \\ \text{change } T & \text{others} \end{cases} \quad (5)$$

Here T is the transformation matrix. V_i^{last} presents the vertexes of the marker in the last frame. $V_i^{current}$ denotes the vertexes of the marker in the current frame. r is the threshold of the allowed moving range, which is known in Figure 5. Equation (5) tells that if all four vertexes move a little, the transformation matrix will hold on. The experimental result shows that the computer-generated models do not flutter after this method processes.

In the film and TV shoot, images will be shot steady. That means the pictures will not rapidly change. And 80% shots are static. Therefore, we utilize a hold-on mode to achieve this

goal. Figure 8 shows the result of the hold-on mode. When the camera is ready and will not move. The transformation matrix is saved. Then the marker coordinates will not change. In this way, the marker can be removed from the scene and the CG objects can also be registered in the scene. Of course, the marker can be occluded clearly shown in Figure 8. Although the marker is occluded by the hand, the CG object is still able to register.



Figure 8. Hold-on Mode

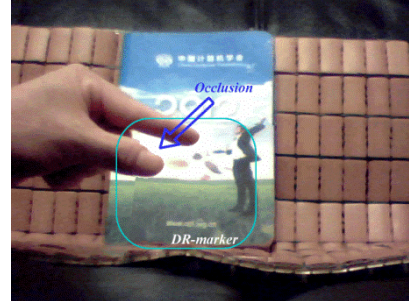


Figure 9. Illustration of Occlusion

2.2.4. Virtual-to-real Occlusion

DR-marker has been implemented as stated above. In an ordinary AR registration, graphical objects are added on the marker. On the other hand, it means real objects will be occluded by computer-generated ones. Real objects always appear occluded by virtual ones, regardless of their actual spatial relationship. These simple approaches cannot handle occlusion between different types of objects in the scene. However, DR-marker is always the background of the scene in the practical situation. If the projecting model is in front of the real object, it does not meet the actual experience. It seems not real. So DR-marker considers the occlusion between projecting model and the real scene.

Here, we consider that detection only in front of previously defined textured planes in the real scene. The method is inspired by F. Jan et al [10]. First, a reference background is selected and saved. Then, once a frame comes, the reference background will compare with the current camera image. We adopt "adaptive HSV" criterion to achieve pixel comparison:

$$o(x, y) = \alpha(x, y) \cdot \Delta H(x, y) + (1 - \alpha(x, y)) \cdot \Delta V(x, y) \quad (6)$$

Where $\Delta H(x, y)$ is the hue difference of pixel (x, y) in the reference background and the current camera image. $\Delta V(x, y)$ is the intensity difference of two images, $\alpha(x, y)$ is a weight. Formula (6) is an adaptive HSV method. Ordinary background subtraction only thinks the intensity difference. That means they simply utilize the influence factor of brightness. If the color of the foreground is close to background, it can't be recognized. Therefore, adaptive HSV criterion is employed. It considers the hue effect. The weight functions is determined as follows:

$$\alpha(x, y) = \Delta H(x, y) / (\Delta H(x, y) + \Delta V(x, y)) \cdot \beta \cdot \min(V^{ref}(x, y), V^{current}(x, y)) \quad (7)$$

Here, $V^{ref}(x, y)$ is the intensity of pixel (x, y) in the reference image. $V^{current}(x, y)$ is the intensity of the corresponding pixel in the current image. $\min(\bullet)$ is minimized value. β is a weight. In this function, we know that if $\min(V^{ref}(x, y), V^{current}(x, y))$ is bigger, then $\alpha(x, y)$ will be bigger. It means when the image is too bright, the image will look pale. The color is hard to be recognized. So the intensity is the principal factor. When the $\Delta H(x, y)$ is bigger, the $\alpha(x, y)$ depends on the color. In a similar reason, when $\Delta V(x, y)$ is large, $\alpha(x, y)$ will be small. So we can utilize lightness to compare. For holes in the comparison image, it should be smoothed. Then we consider whether the pixel should be occluded or not:

$$occlusion(x, y) = \begin{cases} 1 & o(x, y) > \alpha(x, y) \cdot t_h + (1 - \alpha(x, y)) \cdot t_v \\ 0 & \text{others} \end{cases} \quad (8)$$

Where t_H and t_V are the respective threshold of hue and intensity. After the formula (8), we decide which pixel needs not to be drawn. Ultimately the projecting model is modified and the image will be rendered. The result is shown in Figure 9. The DR-marker is on the book and hard to be recognized by the naked eye. The fingers occlude the DR-marker. After occlusion algorithm, it looks like valid due to the actual spatial relationship.

3. Results and Discussion

In this paper, we employ Microsoft Visual Studio 2008 to implement the software. ARToolkit, OpenCV and OpenGL are combined to implement the proposed method. A square marker, like in Figure 6(b), is detected first. Then a projecting model is built, which is the texture in Figure 6(c). Finally the output image will not see the AR marker in Figure 6(a). Next the artificial CG objects can be superimposed on the projecting model. The result can be seen in Figure 4 and Figure 7. The computer-generated character is just the real scene without seeing special markers. The virtual-to-real occlusion is shown in Figure 9. Fingers are in front of the AR marker. After our occlusion algorithm, the real objects will not appear occluded by the virtual objects as other AR applications. In this case, the projecting model has the actual spatial relationship. Here, due to a priori knowledge of the marker, we can just consider the area of marker, omitting the extra pixels to simplify calculation.

In the experiment, we make some short VFX films, and each one is about 2 minutes. In Table 1, marker detection and tracking process is in real time, which costs about 32.7ms per frame. After the projecting model first time, the total process time will be 39.7ms. That means the frame per second of DR-marker is 25fps. However, if we consider the occlusion algorithm, due to the data read time, it will have 211ms computing time. So the frame per second will drop to 4fps. In the future work, we will operate GPU to reduce the computing time.

DR-marker takes the advantages of two AR registration methods. Because it uses the marker method, DR-marker is stable and low-computation. It will rarely occur to flutter. Furthermore, DR-marker removes the marker from the image. It achieves the result of the markerless method. Sometimes it is hard to diminish reality. Therefore, we can put the marker in a simple environment, such as the simple texture desktop. For instance, the sofa is only one color, black, in Figure 10. In Figure 10(a), there is an AR marker above the sofa. It is insufferable for users. Then we remove the marker using the proposed DR-marker. The result is shown in Figure 10(b). Next, we place a CG character to sit in the sofa in Figure 10(c). So this method can be used in filmmaking system as we proposed ARFMS in the other paper.

Table 1. DR-marker Processing Power

	Computing time	Frames per second
Marker detection and tracking	32.7ms	30.6fps
DR-marker without occlusion total process time	39.7ms	25.2fps
DR-marker with occlusion total process time	211ms	4.7fps

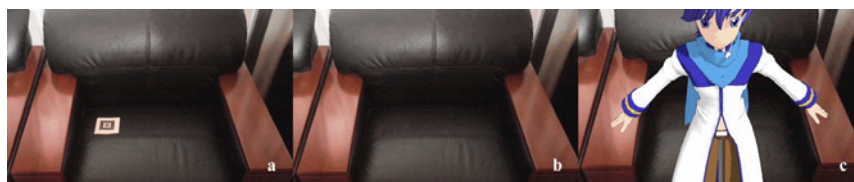


Figure 10. Conceptual Illustration of DR-marker (a) An AR marker above the sofa; (b) The result of taking out the marker; (c) A CG character added on the image.

In fact, the DR-marker method is simplicity of the operator for the amateur users. They only need to put a marker on the ground. Then he can shoot their VFX film. First, as an amateur, a user wants to make a short VFX film. The story is that a virtual CG boy dances in the study. Then a Kinect is attached to the MMD, and he dances by himself to control the CG character as shown in Figure 3. A rough action of CG boy is obtained quickly. Afterward the user controls the character manually to get the better action. The motion file is finished. Next,

the amateur user puts a special marker on the bookcase, as shown in the top left corner image in Figure 11. He uses a laptop to remove the marker and add the CG boy on the real on-site image, whose size position and direction is manually adjusted. Then the dancing motion file is connected to the boy. Finally, the user can shoot the film real time that the CG boy dances in the study in Figure 11. If the VFX film is wonderful, post-production modification phase will skip and the VFX film has been finished. If some details want to be changed, he can utilize the CG boy and raw sequence to synthesize a better film. In sum, it is very easy for an amateur to use.



Figure 11. DR-marker Simulation Result. The top left corner image is the raw frame, and the other five images are real-time processing output frames

4. Conclusions and Future Work

This paper presents a new type film and TV shoot system, ARFMS. Its goal is to achieve WYSIWYG VFX filmmaking. As the beginning of the ARFMS project, we implement an ARFMS prototype. ARFMS demonstrates it is low cost and easy to use. The amateur user will make a VFX film with just an ordinary camera, a special marker and software.

However, ARFMS still has some problems. It is just a prototype. The first problem is the processing power of occlusion algorithm. In the prototype, due to the data read time, the occlusion algorithm of the DR-marker is not real time, which is 4fps. In the future work, we will operate GPU to reduce the computing time. Another issue of DR-marker is illumination consistency. Due to lighting change, the texture of the projecting model will be not consistent. Although projecting model adjustment wants to solve this problem, the method can be only used in the illumination environment is relatively stable. It is hard to be suitable in the actual complicated outdoor environment. The next phase, we will consider the outdoor application.

This paper is just the beginning of the project. We wish more researchers would pay close attention to the WYSIWYG filmmaking system.

References

- [1] Nitsche M. *Experiments in the use of game technology for pre-visualization*. Proceedings of the 2008 Conference on Future Play: Research, Play, Share. 2008: 160-165.
- [2] Weng ENG, Khan RU, Aduce SAZ, et al. Mobile Camera as a Human Vision in Augmented Reality. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 11(7): 3568-3575.
- [3] Ismail AW, Sunar MS. Intuitiveness 3D objects Interaction in Augmented Reality Using S-PI Algorithm. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(7): 3561-3567.
- [4] Ichikari R, Tenmoku R, Shibata F, et al. Mixed reality pre-visualization for film making: on-set camera-work authoring and action rehearsal. *International Journal of Virtual Reality*. 2008; 7(4): 25-32.
- [5] Sziebig G. *Achieving total immersion: Technology trends behind augmented reality - A survey*. WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering. 2009: 458-463.
- [6] Starck J, Maki A, Nobuhara S, Hilton A, Matsuyama T. The Multiple-Camera 3-D Production Studio. *IEEE Transactions on Circuits and Systems for Video Technology*. 2009; 19 (6): 856-869.
- [7] Kato H, Billinghurst M. *Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System*. 2nd IEEE and ACM International Workshop on Augmented Reality. San Francisco. 1999: 85-94.
- [8] Klein G, Murray D. *Parallel tracking and mapping for small AR workspaces*. 6th IEEE and ACM International Symposium on Mixed and Augmented Reality. 2007: 225-234.
- [9] Ling C, Zhao J, Zhang W. *DR-Marker: A Novel Diminishing-Reality-Based AR Registration*. 2013 Seventh International Conference on Image and Graphics. Qingdao. 2013: 716-720.
- [10] F Jan, R Holger, B Gregory. *Detecting dynamic occlusion in front of static backgrounds for AR scenes*. Proceedings of the workshop on Virtual environments. Zurich. 2003: 153-161.