Supplemental Material:
Reminiscence Therapy using Image-Based Rendering in VR

Emmanuelle Chapoulie[*] REVES/INRIA Sophia-Antipolis
Rachid Guerchouche[†] REVES/INRIA Sophia-Antipolis
Pierre-David Petit[‡] CMRR Nice
Gaurav Chaurasia[§] REVES/INRIA Sophia-Antipolis
Philippe Robert[¶] CMRR Nice
George Drettakis[‖] REVES/INRIA Sophia-Antipolis

1 Introduction

In this document we detail some technical, algorithmic and experimental aspects related to the work presented in the short paper *Reminiscence Therapy using Image-Based Rendering in VR* presented at the IEEE VR'2014 conference.

We first present the hardware setup with more focus on the devices used, especially the hand-tracking system. In Section 3, we discuss some details of the original Image-Based Rendering (IBR) algorithm [6] and give details of the algorithmic adaptations required for our immersive setup. We also discuss why the algorithm is limited to a single screen.

As explained in the main paper, during the experiments, there is a training session to allow the participants get used to the finger-tracked gesture interface. The technical details concerning this step are given in Section 4. Details of gesture-based object manipulation and navigation are also presented.

In Section 5, we give some additional information on the experimental procedure, in particular the preparation for the experiment.

2 Hardware Setup

The system setup for the experiments is designed in a CAVE immersive space [7]. The front stereo back-projected screen of the CAVE is used as a display; it is a 320×240 *cm* width-height size and 1600×1200 pixels resolution. The CAVE features a tracking system which updates the user view-point according to the position of his head tracked with infrared cameras and targeted markers attached to the stereo glasses. We use the ART tracking system [2].

The projectors use passive Infitec stereo via glasses (Figure 1.a).

In addition, we use the ART wireless finger-tracking of the orientation of the hand and the position of the fingers. We use the three-finger (thumb-index-middle finger) version (Figure 1.b).

The fact that our participants are elderly (adults over 60 years old) implies specific precautions in the hardware setup. To avoid any risk of unsteadiness or falling, participants sat on a chair installed in front of the display screen, at a distance of 1 *m*, during the experiments. A small stool is placed next to the chair for the experimenter (see Figure 2).

3 Image-Based Capture and Rendering

The approach of [6] is designed for free-viewpoint navigation in the scene, making it suitable for interactive VR. Its integration into our system required some adaptations for tracked display.

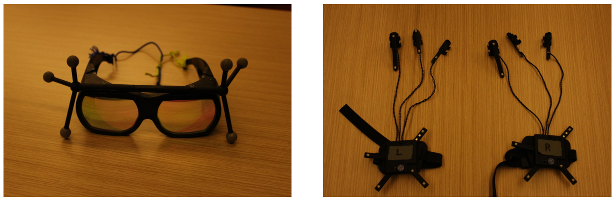

(a) Stereo glasses (b) Hand-tracking device

Figure 1: Tracked devices in the immersive setup. Only a single hand-tracking device is used, depending on the participant: right-handed or left-handed.

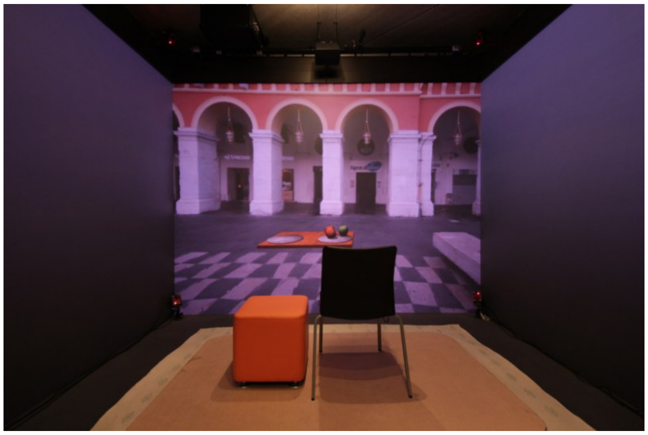


Figure 2: Hardware setup of the experiments.

One important difference with traditional VR is that for rendering we use the calibrated positions and orientations of the cameras of the input photographs. Care must be taken to apply appropriate transformations, especially when combining image-based and synthetic imagery.

In this section, we first briefly recall the basis of the IBR algorithm of [6] in terms of capture and rendering, before presenting how we adapted it to the immersive setup. The limitation of the method to a single screen is discussed in Section 3.3.

3.1 Original Algorithm

3.1.1 Capture and Preprocessing

The method first pre-processes the input images by running standard 3D reconstruction [12, 8] (Figure 3) and oversegments the input images using [1].

The oversegmentation divides the input images into macro regions called *superpixels* (see Figure 4) each containing some 3D depth samples. The approach then synthesizes plausible depth in

[*] e-mail: emmanuelle.chapoulie@inria.fr
[†] e-mail: rachid.guerchouche@inria.fr
[‡] e-mail: pierre-david.petit@unice.fr
[§] e-mail: gaurav.chaurasia@inria.fr
[¶] e-mail: robert.ph@chu-nice.fr
[‖] e-mail: george.drettakis@inria.fr

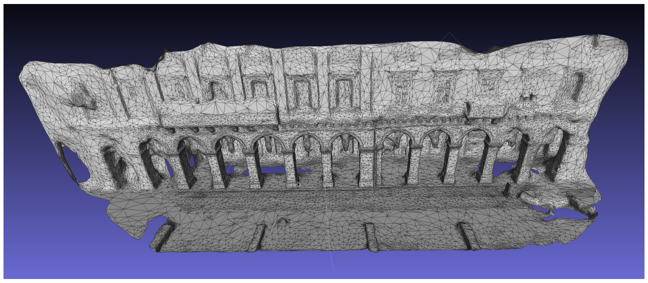
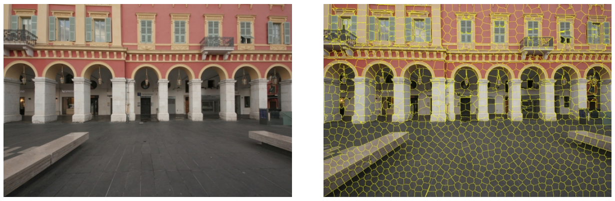


Figure 3: 3D reconstruction (mesh view) of one of the captured scene.

poorly reconstructed regions, compensating for lack of 3D geometry.



(a) Input image

(b) Superpixel oversegmentation

Figure 4: Superpixel oversegmentation.

3.1.2 Rendering

At run time, each novel view is synthesized by pre-selecing 4 input images for the viewpoint. Each superpixel of each selected image is warped to the novel view using a shape-preserving warp. The warped images are then blended to create the final novel view, shown in Figure 5. More details can be found in [6].

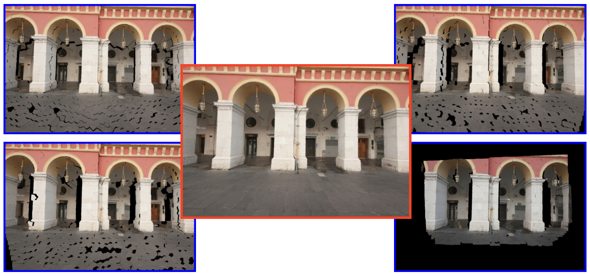


Figure 5: The four side images show the warped superpixels of 4 input images pre-selected to render the current novel view. The central image shows the blended result of the 4 warped images.

3.2 Modifications for Immersive Display

In this section we present the modifications we made to the original method described in [6] in order to display the IBR environments on the front screen of the CAVE. We mainly worked on two components: head-tracked stereo display and rendering synthetic objects with the IBR.

3.2.1 Modifications for Head-Tracked Stereo Display

The approach in [6] is developed for desktop applications where the user navigates using a mouse. In our immersive space setup, the novel camera position V_{novel} is provided by the head tracker in real time. The captured 3D scene is transformed so that the front screen of the immersive space is aligned with the x-axis of the scene. The virtual camera used to display the front screen has orientation $(0,1,0)$ and up vector $(0,0,1)$. These vectors remain the same irrespective of the position and orientation of the head. Thus, in OpenGL terminology, the modelview matrix is given by:

$$M_{\text{f,novel}} = \texttt{gluLookAt}\left(V_{\text{novel}}, V_{\text{novel}} + (0,1,0), (0,0,1)\right) \tag{1}$$

where, the subscript f denotes the front screen. The perspective matrix $P_{\text{f,novel}}$ is constructed using the standard approach of joining the head position with the four corners of the screen [7].

We select input images whose modelview matrices are “most similar” to that of the novel camera. Thus, to render the image, we select the input images whose center of projection is closest to V_{novel} and orientation is similar to $(0,1,0)$ in this coordinate system. Once we pre-select the input images, we warp them using the overall projection $C_{\text{f,novel}}$ given by $P_{\text{f,novel}}.M_{\text{f,novel}}$. The warped images are blended as described in [6].

The above approach seamlessly handles head movement within the immersive space. To allow long range navigation, we use a pointing interface described in Sec. 4, where the user indicates translation. This gives an overall transformation T for the head position. However, recall that the head position is updated asynchronously by the head tracker. Therefore, instead of transforming the head position by T, we transform the scene and input cameras by T^{-1} with equivalent effect. To transform the scene, we apply T^{-1} on each 3D depth sample used for IBR. We transform the input cameras by transforming the center of projection as well as the modelview or extrinsic matrix. We explain this derivation in the next section.

3.2.2 Transforming camera matrices

Assume the entire scene including the input cameras, has to be transformed by the matrix M which comprises a uniform scale s, rotation R_M and translation T_M. Any 3D point x in homogenous coordinates can be transformed by applying M.

$$\bar{x} = M.x = sR_M.x + T_M \tag{2}$$

Clearly, M is invertible because the scale, rotation and translation are all invertible.

Consider an input camera with original perspective matrix (or frustum) F, rotation matrix R and center of projection v. The camera extrinsic or modelview matrix is given by

$$\begin{pmatrix} R & -R.v \\ 0 & 1 \end{pmatrix} \tag{3}$$

The projection of any point x in this camera is given by

$$\begin{aligned} y &= F\begin{pmatrix} R & -R.v \\ 0 & 1 \end{pmatrix} x \\ &= F\left(R.x - R.v\right) \end{aligned} \tag{4}$$

While transforming the entire scene, the camera's rotation matrix and center of projection change but the frustum remains the same because it is an intrinsic property of the camera. Let the new camera position and rotation matrix be $\bar{v}$ and $\bar{R}$ respectively. The projection of a scene point x using the original camera should be the same (up to constant factor) as that of the transformed point $\bar{x}$ using the transformed camera.

$$\begin{aligned} F\left(R.x - R.v\right) &\sim F\left(\bar{R}.\bar{x} - \bar{R}.\bar{v}\right) \\ &\sim F.\left(\bar{R}.M.x - \bar{R}.\bar{v}\right) \end{aligned} \tag{5}$$

This gives the following equations

$$R.x \sim \bar{R}.M.x, \quad R.v \sim \bar{R}.\bar{v} \tag{6}$$

Solving these two, we get the rotation matrix and center of projection of the transformed camera.

$$\bar{R} \sim R.M^{-1}, \quad \bar{v} \sim M.v \tag{7}$$

3.2.3 Rendering Synthetic Objects with IBR

An important part of our system is the ability to manipulate and render synthetic objects (see Figure 6).

To allow this, we modify the IBR rendering pipeline as follows: We first assign a single depth value to each superpixel, i.e., the median of depth samples of the superpixel. When the superpixel is warped to the novel view, we re-project this depth value into the novel view using the following equation:

$$d_{\text{f,novel}} = C_{\text{f,novel}}.C^{-1}_{\text{input}}.d \tag{8}$$

where, d is the depth of the superpixel, C_{input} and C_{novel} are the overall projection matrices of the input and novel camera respectively. While rendering warped superpixels, we write the novel depth into the OpenGL depth buffer. Finally, we render the synthetic objects with the depth test enabled. This places the synthetic objects at the correct depth in the scene giving the correct (dis)occlusion effects.

3.3 Limitation to a Single Screen

The algorithm of [6] is limited to single screen display for two main reasons. First, at every frame the four "closest" input views are chosen, and then their superpixels warped: evidently, this choice is very different for – say – a front and a left screen in an immersive cube, and thus continuity across screens cannot be guaranteed. Second, the shape-preserving warp of superpixels, which compensates for missing depth, assumes that depth over the superpixels has a low gradient, and that superpixels are close to front-facing. The front-facing property and the depth gradients are problematic at screen corners, e.g., when superpixels from the same image are used across two screens.

For our experiments we are thus limited to a single screen. We are actively working on a new approach to combine the quality of [6] and the ability to display in an immersive cube, but this is a fully-fledged research project in its own right.

4 Gestures

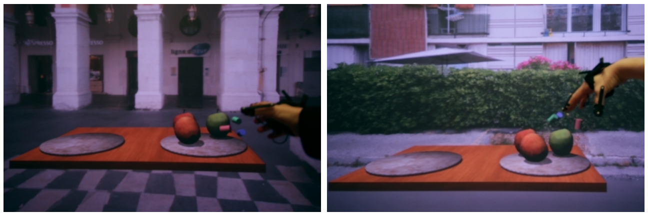
Apart from the ability to quickly and easily capture familiar environments and display them very realistically, a key requirement of our system is support for direct manipulation using finger tracking. We build on the approach presented in [5], which combines ART-based finger tracking with a physics engine, and provides realistic and close-to-natural interaction with objects in the immersive setting.

In the immersive experimental conditions, participants interact with the environment in two ways: direct manipulation of virtual objects and navigation inside the environment. The majority of participants had no previous VR or gaming expertise, and we thus preferred to avoid complicating the task with a wand-like device. The direct manipulation of virtual objects is a training session to so the participant can become accustomed to the use of the hand-tracking system.

4.1 Direct Manipulation of Virtual Objects

The participants can directly grab, release, translate and rotate virtual objects in 3D space using their tracked fingers. Our implementation uses OpenSceneGraph [10] coupled with the Bullet [4, 11], physics library, so that the objects behave as naturally as possible. However the physics library does not control objects used during manipulation. Our gesture heuristic is based on a finite state machine. The participant simply has to place her thumb and index onto an object to grab it, and open her fingers to release it (see [5] for details).

Specifically, a plate with two dishes (one next to the other) is presented to the participant. The right dish contains 3 apples: two red and one green. The apples are dynamic rigid objects following gravity rules, controlled by the physics engine (see Sec. 4). We show these objects in Figure 6. Spatialized sound feedback related to the dynamic virtual objects is provided when the apples are removed or placed on the dishes.

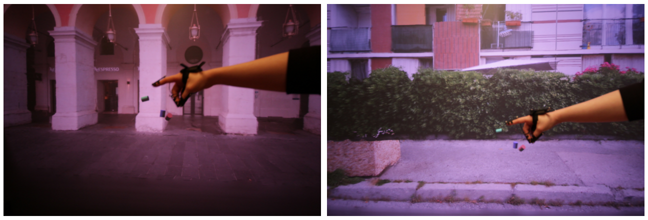


(a) Familiar IBVE (b) Unknown IBVE

Figure 6: Photographs of the screen of the gesture-based interface for direct manipulation of virtual objects. The finger positions are given by the colored cylinders.

4.2 Navigation Inside the IBR Virtual Environments

To navigate inside the IBVEs, the participants have to point in the direction they want to follow with the index finger and a pinch gesture of the thumb and middle finger. Direction is specified by the palm orientation so that the pointing can be approximate, but we instruct participants to point for clarity and simplicity. Given this specification of direction, the gesture is independent of hand location and thus participants can place their hand on their knees to avoid arm muscular fatigue. This is particularly important for our elderly participants. Visual feedback of their thumb, index and middle fingers is presented as red, green and blue cylinders respectively. An example of gesture-based navigation is shown in Figure 7 and in the accompanying video.



(a) Familiar IBVE (b) Unknown IBVE

Figure 7: Photographs of the screen of the gesture-based navigation in the IBVEs.

5 Preparation for the Experiment

Before starting the experiment, two steps are performed: a stereo-blindness test and hand-tracking device calibration.

5.1 Stereo Blindness Test

A fraction of the human population has impaired stereo vision. This inability to see in 3D using stereo vision results in an inability to perceive stereoscopic depth, by combining and comparing images from the two eyes [3, 13].

After clinical inclusion, each participant is invited to sit on the chair in the immersive space. A random-dot stereogram is then displayed on the screen. It subsists on random dots which when viewed with stereo glasses produces a sensation of depth, with objects appearing to be in front of or behind the display level [9].

The experiment continues only if the participant is not stereo blind. All participants had correct stereo vision.

5.2 Hand-tracking Device Calibration

The hand-tracking device has to be calibrated for each user to adapt it to hand and finger size. This step takes less than 2 minutes and it is done directly after the stereo-blindness test.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEE Trans. PAMI*, 34(11):2274–2282, 2012.

[2] ART. `http://www.ar-tracking.com`.

[3] S. R. Barry. *Fixing My Gaze: A Scientist's Journey Into Seeing in Three Dimensions*. Basic Books, 2009.

[4] Bullet. `http://www.bulletphysics.org/`.

[5] E. Chapoulie, M. Marchal, E. Dimara, M. Roussou, J.-C. Lombardo, and G. Drettakis. Evaluation of Direct Manipulation using Finger Tracking for Complex Tasks in an Immersive Cube. Technical Report RT-440, INRIA, Sep 2013. `http://hal.inria.fr/hal-00857534`.

[6] G. Chaurasia, S. Duchene, O. Sorkine-Hornung, and G. Drettakis. Depth synthesis and local warps for plausible image-based navigation. *ACM Trans. on Graphics (TOG)*, 32(3):30:1–30:12, 2013.

[7] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the CAVE. In *SIGGRAPH, ACM Proc.*, pages 135–142, 1993.

[8] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE Trans. PAMI*, 32(8):1362–1376, 2009.

[9] B. Julesz. *Foundations of cyclopean perception*. U. Chicago Press, 1971.

[10] OpenSceneGraph. `http://www.openscenegraph.org/`.

[11] osgBullet. `http://www.osgbullet.vesuite.org/`.

[12] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. *ACM Trans. on Graphics (TOG)*, 25(3):835–846, 2006.

[13] R. Whitman. Stereopsis and stereoblindness. *Experimental Brain Research*, 10(4):380–388, 1970.