D5.2.1 v2
Specification and description of tools for holographic live performance and installation art

Deliverable due date: 31 March 2015
Actual submission date v1: 07 April 2015
Actual submission date v2: 06 July 2015

Project no: FP7 - 61005
Project start date: 01.10.13
Lead contractor: The Foundry
<table>
<thead>
<tr>
<th><strong>Project acronym</strong></th>
<th>DREAMSPACE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project full title</strong></td>
<td>Dreamspace: A Platform and Tools for Collaborative Virtual Production</td>
</tr>
<tr>
<td><strong>Security (distribution level)</strong></td>
<td>PU</td>
</tr>
<tr>
<td><strong>Contractual date of delivery</strong></td>
<td>Month 18, 31.03.2015</td>
</tr>
<tr>
<td><strong>Actual date of delivery</strong></td>
<td>V2. Month 22, 06.07.2015</td>
</tr>
<tr>
<td><strong>Deliverable number</strong></td>
<td>D5.2.1</td>
</tr>
<tr>
<td><strong>Deliverable name</strong></td>
<td>Specification and description of tools for holographic live performance and installation art</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>Report</td>
</tr>
<tr>
<td><strong>Status &amp; version</strong></td>
<td>Outline</td>
</tr>
<tr>
<td><strong>Number of pages</strong></td>
<td>14</td>
</tr>
<tr>
<td><strong>WP / Task responsible</strong></td>
<td>Philippe Bekaert (iMinds)</td>
</tr>
<tr>
<td><strong>Other contributors</strong></td>
<td>CREW</td>
</tr>
<tr>
<td><strong>Author(s)</strong></td>
<td>Koen Goossens (CREW), Vincent Jacobs (iMinds), Philippe Bekaert (iMinds)</td>
</tr>
<tr>
<td><strong>Internal Reviewer</strong></td>
<td>Thomas Knop</td>
</tr>
<tr>
<td><strong>EC Project Officer</strong></td>
<td>Alina Senn</td>
</tr>
</tbody>
</table>

**DOCUMENT HISTORY**

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Reason of change</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>03.02.2015</td>
<td>Initial outline</td>
</tr>
<tr>
<td>0.2</td>
<td>30-03-2015</td>
<td>Revisions in response to feedback on initial outline and evaluation work performed in februari and march 2015.</td>
</tr>
<tr>
<td>0.3</td>
<td>31-03-2015</td>
<td>Additions by Vincent Jacobs and Koen Goossens resulting from further evaluation work.</td>
</tr>
<tr>
<td>0.4</td>
<td>03-04-2015</td>
<td>Revisions by Philippe Bekaert</td>
</tr>
<tr>
<td>0.5</td>
<td>12-05-2015</td>
<td>More additions and revisions resulting from continued evaluation and analysis by Vincent Jacobs</td>
</tr>
<tr>
<td>0.6</td>
<td>04-06-2015</td>
<td>Integration of new input by Eric Joris and final revisions by Ph. Bekaert before internal review</td>
</tr>
<tr>
<td>0.7</td>
<td>03-07-2015</td>
<td>Revisions by Vincent Jacobs in response to internal review</td>
</tr>
<tr>
<td>0.8</td>
<td>06-07-2015</td>
<td>More revisions by Vincent Jacobs in response to internal review. Final edits by Philippe Bekaert.</td>
</tr>
</tbody>
</table>
Table of contents

1. INTRODUCTION ...........................................................................................................4

2. VIDEO PROCESSING SOFTWARE ENVIRONMENTS FOR IMMERSIVE EXPERIENCES...5

3. HOLOGRAPHIC EXPERIENCE SPACE BUILDING BLOCKS.............................................6

4. PROOF OF CONCEPTS....................................................................................................9

5. CONCLUSION ..................................................................................................................14

REFERENCES ......................................................................................................................14
I. Introduction

Task WP5T2 of the Dreamscape project concerns the development and integration of tools for creating immersive environments in which navigable filmed content (depth enhanced omnidirectional video, free viewpoint video) is displayed in an environment allowing the spectator to navigate that content physically. In particular, we aim for view dependent multi-projection environments, creating a “holographic” experience space. The experience space is “holographic” in the sense that the projected imagery is adapting in real time to the viewing position of a spectator, creating an illusion that the projection screen itself disappears, while the reproduction of motion parallax increases the experience of “being inside”.

Not just the image warping and blending, in case of multiple projectors, is adapting to the spectators location, but the free viewpoint video content itself is genuinely 3D in the sense that it can be “navigated” into. Viewpoint dependent display of synthetic 3D CG content is familiar since the beginning of the 1980-ies, and became well known with the virtual reality hype and CAVE environments of the 1990-ies. Dreamscape, in contrast, aims at facilitating displaying “filmed” rather than “synthetic” free viewpoint content in viewpoint tracking display environments for creating navigable experiences with true photorealistic look at affordable cost, both in terms of required hardware and in terms of person-cost for creating and processing the content.

Unlike the “ordinary” 2D resulting from stereoscopic 3D video, the spectator will experience a higher sense of immersion thanks to the reproduction of motion parallax when “moving” in the environment. The reproduction of motion parallax is similar to a hologram, hence the term “holographic”. Unlike true holograms, however, Dreamscape addresses experiences in full color moving imagery, using off-the-shelf IT technology.

Dreamscape does not directly address the contemporary realizations of the “Pepper’s Ghost” effect, that are often called “holographic projection”, e.g. by the London based company Musion. Such displays do not reproduce motion parallax, unlike suggested by their common name, but aim at superimposing live single-viewpoint or stereoscopic projected imagery over live action being performed behind a transparent, yet reflective, Mylar screen in front of a (large) audience. The Dreamscape tools to be developed in WP5T2 will allow the reproduction of free viewpoint content on such displays, if desired, but do not address the Pepper’s Ghost illusion per se.

In a similar spirit, Dreamscape is not directly addressing “integral” displays, such as produced by the Hungarian company Holografika, or other specific high-end multi-scosopic displays. Such displays are also sometimes called holographic displays. The primary goal of WP5T2 is to facilitate the display of immersive video on low-cost and easier to set up multi-projection systems, made from off the shelf common hardware. We believe this will be driving creative exploration more strongly than by addressing high-end environments using high-end hardware. However, Dreamscape tools will not be limited to the aforementioned hardware. We do consider adapting for a Holografika display to be installed soon at iMinds in Brussels.

The work in task WP5T2 builds upon earlier work of Hasselt university, iMinds and CREW, most notably in the 2020 3D Media FP7 integrated project, on multi-projection environment calibration, rendering and display, as well as Dreamscape WP3T2 and WP4T4 work on capture, processing and rendering of depth- and dynamic range enhanced omnidirectional video.

Several software building blocks have been developed, and still are being developed, for the purpose. They have been exploited for creative purposes by CREW and others in the form of monolithic software applications. These applications however are rather closed, and need ad-hoc adaptations each time in response to changing creative needs. In order for creative and artistic people to get the maximum gain out of the developed technology, both existing and new tools must be split up and redesigned in blocks of sufficient fine granularity. Usage should be well documented so that they can be mixed and matched in an easy way.
We believe usage is significantly facilitated by visual graph-based programming approaches, rather than by scripts. A graph based approach has been proven to be successful for real-time (and off-line) audio processing with environments such as cycling74’s Max7, or The Foundry’s Nuke for off-line video processing.

Since the conception and preparation of the project proposal, significant evolution has been witnessed in the industry. For instance, the related technologies of 3D projection mapping have been identified at ISE 2015 as a major trend and opportunity for the coming years in event industry. This evolution required an updated state of the art study concerning hard- and software platforms relevant for our goal.

This document consequently addresses two questions, dealt with in section 2 and 3 correspondingly:

- What audio-video processing software and hardware environments are available that could support live multi-projection display of our immersive video content? Which of these is most suited for our purposes and to build future Dreamspace developments upon?
- What functionality is missing in these environments, and which tools should be developed in the remaining WP5T2 Dreamspace work ourselves?

Section 4 presents proof of concept installations, including a planned demonstration at CVMP 2015.

2. Video processing software environments for immersive experiences

iMinds and CREW have been evaluating the following video processing software environments, that seem suited for creating live immersive holographic experience spaces:

- Cycling 74 MaxMSP
- Derivative Touch Designer
- Ventuz
- The Foundry Nuke (Blink player)
- Pixel Conduit
- Unity 5
- OpenFrameworks and Cinder (C++ creative coding libraries)
- Isadora
- OpenScenegraph
- Resolume Avenue/Arena

The following evaluation criteria were used in the assessment process:

- Freely configurable processing chain, preferably by means of visual graph-based programming
- Live video processing of multiple full HD and preferably 4K streams at 25fps or better
- Flexible in terms of inputs (multiple sources of video streams), and outputs (output to multiple video screens or projectors, preferably also streaming over network). A distinction between hardware inputs (e.g. SDI) vs. software inputs (e.g. different video streams) should be made.
- Support for user interaction devices including gesture based (Microsoft Kinect), optical tracking (Optitrack), IMU (Intersense, PNY Spacepoint, ...), ...)
- Can be integrated in a live performance work flow, e.g. remote controllable by such protocols as OSC, or 0MQ, MIDI, DMX, ...
- Node extensibility: the ability to create new processing nodes able to alter the data-flow. If yes, do these nodes execute on the CPU and/or the GPU. Other optimisations features, if present, should be specified.
- Extensibility in terms of content input sources: 2D/3D graphics, (multichannel) audio, OpenEXR, ...
- Conditions of use in the context of live art performance, during and after the Dreamspace projects
- Platform support
- Community support

The table in appendix summarizes the findings. The table uses the above listed points mirrored as evaluation criteria.
The evaluation takes in account the previous and future work of CREW, since they have extensive experience in building and working with immersion in different forms.

The chosen tools should have a strong real-time component, where parameters can interface on the fly with efficient building blocks or nodes. These nodes (available in the form of plugins, linked libraries or scripts) hide the complexity from the end user, but give programmers the ability to further extend the operation of the software. Most of the evaluated tools provide well-documented extensibility, except Pixel Conduit. Max, Touch Designer, Unity, OpenFrameworks and Cinder provide a plugin system for C++ code with the ability to provide OpenGL code. Max can only work with GLSL 1.2. The Foundry’s Blink provides very efficient kernels and scales itself to the maximum performance of the host system. Resolume only supports FreeframeGL and Quartz composer plugins, focussing its pipeline on video and generative graphic content.

Furthermore should these tools have the ability to work with rich-media content of today, as in being able to play 4k video, to process a variety of 3d formats and to expose its (generated) content to the audience in the form of hardware outputs or software linking mechanisms. Most of the evaluated platforms provide this functionality, with the exception of Ventuz, Pixel Conduit and Isadora. Max still uses Quicktime on Windows.

The software should at least run on Windows, enabling easy hardware extensibility, and preferably also on Mac since this platform is still popular among creative industries. Pixel Conduit only runs on Mac.

We also consider the presence of a wide community an important factor in the evaluation, since emerging technologies can greatly benefit from user experiences but also to have a wide support channel when necessary. Most of the environments have an already established user group and community. Some are quite recent and see their communities still growing, like Blink, Ventuz and Cinder.

Evaluating all the criteria, we decide to further invest research in the Nuke (and its Blink scripting) and Unity environments. Nuke provides a huge toolset for compositing and applying effects and is has the status of being the industries standard. We want to make the calibration and stitching of panoramic and omni-directional content available in its pipeline. By doing so, the pre-production user greatly extends his possibilities being able to use all other present functionality. (S)he can then provide rich content for use and playback in the real-time environment.

We see Unity as the best option for operating real-time and live in a stage environment. This platform agnostic tool has gained much acclaim in the use of live 3D environments, in games but also on set. Its functionality is extensive, and can be easily extended using scripting (Javascript, C#) or creating full native plugins (C++, OpenGL 4.4). Unity enables a creative user to efficiently layout his 3D set and import assets into it, but also provides a good base for programmers to extend this set by code. Therefore, we believe this platform is the best choice, taking in account all evaluation criteria.

### 3. Holographic experience space building blocks

This section presents a high level overview of building blocks that need to be adapted or newly developed for creating holographic experience spaces in the software environment chosen in section 2, being nuke with blink and unity.

The tools proposed here result from an analysis of needs that arose in past installations and productions, and predicted needs of upcoming installations. These installations and productions are being described in Section 4, with reference to the items below.

1. Multi-projection calibration by means of camera feedback (2020 3D Media work and later refinements at iMinds)
   a. Calculation and display (projection) of calibration patterns
   b. Analysis of captured patterns into camera-projector maps
   c. Filtering of such maps, to eliminate noise and reflections in the projection environment
Building immersive experiences can often be a challenging task. It involves combining different technologies, configuring them so that they can operate real-time in a live environment. It’s not unusual for artists and designers to incorporate hardware and software in their work, exploring new possibilities in content generation, creating digital experiences and letting users interact with these worlds.

Suggested workflow implementing the above listed building blocks

Building immersive experiences can often be a challenging task. It involves combining different technologies, configuring them so that they can operate real-time in a live environment. It’s not unusual for artists and designers to incorporate hardware and software in their work, exploring new possibilities in content generation, creating digital experiences and letting users interact with these worlds.

It would be superfluous for this document to give a complete overview of the history of immersive media creation. Instead it will focus on comparing the software tools currently available for creative companies and individuals in dance, performance and installation art for building immersive experiences, followed by some examples. Out of this comparison it will try to deduct a possible workflow, optimized for the creation of immersive experiences.

The tools all have a certain degree of shielding the user from the pure code based internals. However, most of them include state-of-the-art technologies under the hood. As such, many of the described tools are able to work high resolution data and have means to process them in real time. Resolume developed their own video codec to be able to decode 4k+ video at 60fps on the GPU, including support for alpha channels. The communities of OpenFrameworks and Cinder show a heterogeneous mix between VFX developers, trained computer scientists and creative coders. As such, state of the art algorithms for calculating lighting are shared with the community. The most recent version of Unity engine ships with modern rendering techniques for creating high quality real-time renders. All of them support new emerging technologies like Oculus Rift, Microsoft Kinect v2, Leap Motion and provide a framework for supporting future devices with a minimum of work.

The website creativeapplications.net acts as a hub for cross media artworks (including immersive and interactive experiences), creating a portal for artists to display their work and share details about the creation,

---

1 The list of the compared software tools is attached to this document.
2 https://resolume.com/software/codec
4 Techniques include physically based shading, deferred rendering, real-time global illumination, render-to-texture, ...
5 Each tools uses a certain framework for allowing extensions to be created. The inputs and outputs of these extensions (or plugins as you might call them) are abstracted so that the builder can focus on the functionality itself, without having to worry about how the plugs-in to the main system.
technical background and used software- and hardware tools. It’s a good starting point to explore contemporary cross media artworks and the tools they are made with.

Immersive experiences have been explored in the past by artists/researchers such as Jeffrey Shaw⁶ and the Wooster Group⁷, Maurice Benayoun⁸, Paul Bourke⁹ and others.


More specifically the work of Dreamscape partner CREW focuses on creating immersive experiences. Their work is discussed in detail in chapter 5.

We can define three stages in building a workflow for the creation of an immersive experience:

- **Input stage**: eventual capturing and formatting of the input used in the live environment. This can be 360 (stereoscopic) video, a 3d scene with optional live generating elements, planar video content, ... We therefore need a system capable of decoding many formats, being able to process standard projections (3.3), placing this content on certain geometry (3.8) for use in the next stage: the interaction stage.
- **Processing and interaction stage**: the way the content updates and transforms itself as a result of input from one or many users. Examples include: tracking the viewer using time-of-flight camera system and changing his viewpoint on the scene, receiving new scene parameters that change the layout of the scene (lighting, VFX, transformations of certain object, ...), connecting to external triggers like audio/light/sensor data, ... (3.5, 5.6, 3.7)
  The content is updated in real time so that the scene can be rendered for display in the output stage.
- **Output stage**: the scene gets rendered real-time to a certain pre-defined output format. Immersive experiences can be displayed in many ways. An Oculus rift would require a certain distortion on the output, a cave setup with three beamers requires distorting the output to fit on the physical cylindrical screen but also requires to address blending of the three beamers so that they can overlap. A general system being able to project content on arbitrary geometry using an arbitrary amount of projectors or other devices is needed. (3.1, 3.2, 3.3, 3.4, 3.8). I want to mention separately the research towards the use of NVAPI (Windows) and Nvidia-settings (Linux). This seems the only way to create a true application independent blended and warped display. This has the huge advantage of being able to work with the OS desktop in a natural and predictable way. (3.2)

The suggested workflow is not bound to one certain machine, although modern machines coupled with powerful graphics cards housing multiple outputs should have enough processing power to run this pipeline on their own. Eventually, the input stage and interaction stage are possible candidates to run on separate machines, given there is an efficient ‘pipe’ to get data from the input stage into the next. When working with high resolution content, this would likely get less stable and lose real-time behaviour.

---

⁶ http://www.icinema.unsw.edu.au/
⁹ http://paulbourke.net/
¹⁰ http://www.creativeapplications.net/openframeworks/connected-worlds-interactive-ecosystem-for-nysci-by-design.io/
¹² http://www.antivj.com/onionskin/
¹³ http://www.am-cb.net/projets/hakanai/
¹⁴ https://www.behance.net/gallery/18559157/H-OM-E-OMOR-PH-ISM
In our opinion Nuke and Unity, both addressed in depth in chapter 2 fit the above outlined workflow well. Unity is made for real-time rendering of a 3d scene, is able to respond to different kinds of (physical) live inputs, is made cross platform from the bottom-up, is extendable when necessary and can have this output stage be well defined. Combining this setup with the use of hardware blending and warping makes a complete and efficient system for creating immersive experiences, providing enough headroom for future emerging technologies.

The Foundry Nuke has a well-established status as industry standard in the field of VFX, compositing and postproduction. We consider implementing the features documented in point 3.1 inside a Nuke context, mainly using Blink scripts. Looking at the input stage, offloading all the work to an offline process in Nuke would generate better quality input and would free resources for the next stages (interaction and output) in Unity.

An example could be stitching multiple camera images to one equirectangular image, generating a planar representation of a physically captured 3d scene, using scripts in Blink. The images could be color and level corrected using well-known Nuke tools. Extra information for later masking in the interaction stage can be included in the alpha channel. The whole could be exported to a video format that would allow Unity to decode the color information to texture a 3d sphere (recreating the 3d captured environment) and to decode the alpha for other purposes.

4. Proof of concepts

Developed technology will be tested in the field in proof of concept installations, such as the planned demonstrator at CVMP in November 2015 (D6.1.1), as well as in work by CREW. The building blocks described in the previous section have been verified in past proof of concepts, installations and numerous in-lab experiments over 2012-2015. These works provide a solid ground for the selection criteria for new building blocks and the environments they should operate in. Developed technology will also be demonstrated and tested during the research for the upcoming performance named Absense (2015).

The coming section describes two previous works of CREW and connects them with the listed items in section 3 (numbers between parentheses) and more generally with the workflow described in section 2.

- NoHorizon (Hasselt, 2012)

NoHorizon explores new ways of creating immersive experiences without the use of video-goggles. The spectator is presented with a triangular shaped projection environment where a chariot is placed in the middle. This chariot can run on linear rails for about 12m to one of the triangle’s points and back. The spectator can use this chariot to engage on a walk through the environment, surrounding himself with projected content driven by a central computer embedded in the construction. Using a high resolution laser based tracking system the position of the chariot is communicated to this central computer. On the initial state of the chariot (the side opposing the corner), the user sees six grid-like projections, with no blending and warping applied. When he starts pushing the chariot the projection flips to a blended and warped state, effectively joining all separate images to one grid. The equator and prime meridian of this grid are aligned to the (generalized) center-of-view of the user. The plane geometry of the grid appears as standing frontal and perpendicular on the viewer’s position. When the user moves, the position-aware software adapts the blending and warping of the grid so that its perceived position remains unchanged, creating a dissociation between the physical mobility of the user and the virtual persistence of the image.

A lot of the building blocks described in the previous chapter 3 were initiated and used during the development of this installation. Since noHorizon was a multi projector environment, calibration was necessary to enable smooth blending and warping (3.1). A Point Grey Ladybug 3 omnidirectional camera was used during the calibration process for capturing calibration sequences projected onto the environment (3.1). This procedure was repeated at intermittent positions on the rail, effectively ‘storing’ keyframe blend/warps...
on the user’s possible path. The custom software used the actual high-resolution position (in mm) to interpolate between these key frames at runtime.

The computer was equipped with an AMD Eyefinity graphics card providing six separate physical outputs, enough to drive the whole installation (2). The custom application gets the input from the laser and computes the final image, aligned the viewer’s position in space (5).

During rehearsals, experiments were made using video content as opposed to fixed graphical content. The displays software used a VLC backend and was able to play most of today’s video (and audio) file formats (8). It quickly became clear that more time was needed to find the right content to play back in this environment, caused by the added sense for the need of a narrative. The content could be displayed from a viewpoint perspective (texture mapped on inside sphere, viewer in middle of sphere) and equirectangular format (3).

The custom software used in this installation was written in C++ and was created ad-hoc for NoHorizon. As such, it didn’t provide any easy interaction possibilities on site, without having to adjust code on a fundamental level. The VLC backend proved itself as a first step of being able to ‘plug and play’ multimedia content in an immersive environment. Also, the calibration tools were expecting rigid hardware, like the PointGrey Ladybug 3.

Software tools 1,2,3 and 5 of section 3 will allow to further develop such environment with much greater flexibility and freedom than with the current ad-hoc monolytic software.

Figure 1. NoHorizon view dependent immersive projection environment shows 360 video adapted to a spectators location along a track (left). The right image visualisation multi-projection warping meshes in equirectangular format for one particular location along the track. These meshes are adapted in real-time to the spectator location.

- **Double Happiness (2012)**

Double Happiness presents the RGB+Z images captured by the HHI Berlin/Fraunhofer institute, as part of the 2020 3D Media FP7 project. The viewer enters a room where one a projection screen stands on the floor. A Kinect sensor communicates the user’s position to custom software that displays the RGB+Z video frames as a point cloud. This view dependent installation provided the basis for further explorative research by CREW, resulting in the interactive installations Centrifuga and aXes.

Tools 4,5 and 8 of section 3 will help to further develop such installations with increased flexibility and power.

The code for generating viewpoint-dependent content was ad-hoc and opaque. The code could be more generalized as it is a much-explored feature in artistic and creative coding environments. A Microsoft Kinect sensor is frequently used for this purpose. In a generalized form (e.g. in the form of Unity plugin) the feature would be ready-to-use, since it already abstracts some of the necessary key points (a camera in 3d space, meshes, textures ...).
- **Centrifuga** *(Sint-Niklaas, 2013)*

Centrifuga uses the same technical setup used in Double Happiness, but uses omni-directional 360-degree video as content. Compositing tools were used to overlay CG elements to help the user exploring new viewpoints.

The compositing was a difficult process, since there were no ready made tools available on the market to add planar content to a 3d spherical environment, directly applying the correct distortions. A Nuke workflow with the necessary tools available could greatly shorten setup time, and would increase artistic freedom during creation time.

- **aXes** *(Gaasbeek, 2013)*

aXes works on the bridge between state-of-the art depth-acquisition techniques and interactive and immersive presentation techniques. The installation was presented in a neo-Renaissance castle (origin 1240, renovated 19th century) around Brussels. During its renovations, and as neo-Renaissance dictates, the linear perspective theorem was used as a guide in several architectural decisions. For example, the large garden and
lake display the use of an inverted perspective. This inspired CREW to alter the experience of the linear perspective through an interactive installation, where the position of the user is fundamental for the image being constructed on the screen. Through interaction, the viewer learns that our fundamental sense of depth actually stems from our ability to look at objects from different perspectives. We ‘capture’ what we see and our brains construct a 3d ‘scene’ from it.

For aXes, CREW worked together with the Fraunhofer institute, with whom they worked as partners in the 2020 3D media FP7 project. Fraunhofer had developed a depth and color capture system able to capture full HD video using cinema-grade cameras (e.g. Arri). The system could afterwards generate fullHD disparity maps during post-processing, enabling a stereo workflow in post-production. CREW wanted to display this video content in a new way, combining interactivity and the aforementioned idea of perspective perception. Two video walls consisting of 6 42” fullHD monitors mounted in a 3x2 setup were constructed and each linked to a computer with an AMD Eyefinity graphics card. Custom software rendered the captured color frames as anaglyph point clouds in 3D (2). Displacement information for each frame was imported through a parallel stream, reading in depth frames that were generated offline using the disparity information from Fraunhofer’s capture system (4). Frames providing inaccurate disparities were edited in post-production using Eyeon Fusion, which was also used to embed extra depth information (7). The viewer’s position was tracked using the popular Microsoft Kinect and communicated back to the central computer (5). The position influenced the virtual cameras frustum so that it appeared as the user was looking around the object/scene, discovering new perspectives to comprehend its form.

An ambisonic soundfield composition played back through a system consisting of 8 speakers altered the immersive experience and gave the ‘viewer’ clues to explore new things in the environment.

The software was entirely built in MaxMSP. It made heavy use of OpenGL shaders for the displacement of the vertices and for the anaglyph rendering. The node based workflow of MaxMSP provided an easy connection with other systems. As an example, the audio system was running on a separate computer using Ableton Live

The transport (clock) of Ableton Live controlled the playback of the whole system, being transferred over network to all the computers in the network.

The same observations as in the previous discussed installations apply here as well: all the compositing had to be done on a very experimental basis, frame per frame with manual control. The playback process had to implement a timing system (hence the link with Ableton Live) since this was not present in Max/MSP.

If these were to be modules already available on the working platform, one could be experimenting faster with different types of content and seeking better interaction with the sound (the composition was made one day before the premiere). Also Max/MSP was barely performant enough to drive the installation, given its huge resolution (6 X 1920x1200).

- **Explorer and Absence (fall 2015, 2016)**

The former interactive installations prove that one could benefit from a set of modular building blocks. This would enable to use the functionality needed for a specific situation without creating unnecessary overhead but with holding on to a hands-on approach. Parameters inside each building block could give the (artistic) user interaction with the functionality contained inside, without having to worry about implementation details.

CREW has experimented with new tools in their latest production Explorer. Heavy use was made of game engine and live 3d environment Unity together with an optical motion capture system from Optitrack. The same tools will be used in their upcoming performance Absense.
Unity proved to be a versatile tool to work with 3d environments. It also provide to be easy deployable in a performative context, able to communicate with other modules via network or MIDI. It is included in the evaluation list of software environments for immersive experiences.

- **CVMP demonstrator (fall 2015)**

At CVMP 2015 in London will be presented an interactive installation demonstrating view-dependant rendering of a filmed video backlot with parallax (WP4T4) in combination with synthetic CG manipulation (WP5T1). It serves as an intermediate step towards the final showcase to be realized in 2016.

The installation is a corner shaped space, with a large screen or projector that creates the impression of a building window looking out, on one side of the corner. It represents a scene concerning a modern building with a windows looking out over the river or some other open space. The building has construction elements extending outside, visible through the window. At the same time, there are distant elements. In fact, there is a whole continuum between distant and nearby elements visible through the window in the corner of the building. The most nearby elements, such as water plants and building construction elements will be modeled and represented by means of synthetic CG. The (moving) water surface up to distant elements (e.g. a city scape on the opposite side of the river) will be represented by means of a light-field alike model, constructed from capturing a real environment with an array of high resolution cameras placed on a horizontal bar and processed with algorithms being developed in WP4. Different layers, each with their own representation appropriate w.r.t. distance, will need seamless blending. Real-time rendering of the preprocessed parallax video will be performed with tools 4, 5, 6, 7 and 8 mentioned in Section 3.
A motion tracking system will drive view-dependant rendering of the outside view according to a spectators location w.r.t. the screen. The showcase at CVMP will be an interactive video installation. However, replacing the spectator by a motion tracked camera, and the projection of screen now displaying the window view by a green screen, this is equivalent to a virtual studio set-up. It will be realised using tools developed for virtual studio applications for most part.

5. Conclusion

This document reports on the evaluation of existing software environments and the definition of extra building blocks in response to providing the necessary tools for the creation and live control of immersive experiences.

From our extensive evaluation, we conclude that Nuke with blink, and Unity are the most appropriate software environments to further build our developments upon.

A high level description of to be incorporated and developed new functionality in these tools was given in Section 3, motivated by the requirements from past and upcoming installations and performances described in Section 4.

References

1. https://cycling74.com/max7/
2. https://www.derivative.ca/088/Features/
7. http://openframeworks.cc/about/
11. https://resolume.com/software