INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG1(JPEG) & WG11(MPEG)

ISO/IEC JTC1/SC29/WG1N72033
ISO/IEC JTC1/SC29/WG11N16352
June 2016, Geneva, Switzerland

Source **Joint ad hoc group for digital representations of light/sound fields for immersive media applications**

Status **For publication**

Title **Technical report of the joint ad hoc group for digital representations of light/sound fields for immersive media applications**

TABLE OF CONTENTS

Table of Contents 2

1. Introduction 5

2. Prioritized List of Use Cases 6

2.1 Category 1: Light/Sound field communication 9

2.2 Category 2: Light/Sound field editing 9

2.3 Category 3: Free navigation 10

2.4 Category 4: Interactive all-reality 11

3. Overview of Conceptual workflow 13

4. Data acquisition and content creation 16

4.1 Visual Data Acquisition 16

4.1.1 Introduction Light field 16

4.1.2 Overview sensors and sensed Data for Visual applications 16

4.2 Audio Data Acquisition 23

4.2.1 Introduction 23

4.2.2 Virtual and Augmented Reality 23

4.2.3 Metadata for Virtual and Augmented Reality 24

4.2.4 Sound Fields 24

4.2.5 Sound Field Communication Scenarios 24

4.2.6 Sound Field Sensor Configurations 25

4.2.7 Metadata for sound field capture and recording 29

5. Data Conversion 30

5.1 Visual Data Conversion 30

5.1.1 SLAM 30

5.1.2 Point Clouds to Depth Maps 30

5.1.3 Point Clouds to triangular meshes 31

5.1.4 Point Clouds to Planar primitives 32

5.1.5 Point Clouds vs. Light Fields 34

5.1.6 Light fields to Depth maps 35

5.1.7 Light Field Omni-Directional Stereo 36

6. Technologies supporting Audiovisual Applications 38

6.1 Visual Technologies 38

6.1.1 Single and multi-view texture and depth maps 38
6.1.2 Visual Coding 39
6.2 technologies Supporting Audio Applications 45
6.2.1 Virtual/Augmented Reality Recording and Playback 45
6.2.2 Sound Field Recording 46
6.3 Audio/Visual context processing 46
6.3.1 Visual processing to assist audio processing 46
7. Rendering 47
7.1 Visual Rendering 47
7.1.1 Omni-directional color information 47
7.1.2 Point Clouds 48
7.1.3 3D Meshes 50
7.1.4 Panoramic Textures 53
7.1.5 Omni-Directional Stereo textures 54
7.2 Microlens light fields 55
7.3 Camera array light fields 57
7.3.1 Dense camera array Light Field 58
7.3.2 Sparse camera array Light Field 59
8. Presentation 62
8.1 Projector array light fields and Integral Photography 62
8.1.1 Light Field Displays 62
8.1.2 Super-dense Light Field displays with “holographic” eye accommodation 64
9. User interaction 65
10. Common aspects of data formats 66
10.1 Rendering quality vs. bitrate performance figures 66
10.2 Data representation commonalities 66
10.3 Data compression commonalities and differences 68
11. Application focused workflows 70
11.1 Broadcast SuperMultiView 70
11.2 Immersive bi-directional communication 70
11.3 Personal editing workflow 71
11.4 Post production workflow 72
11.5 Free navigation workflow 73
11.6 Light field based workflow 74

12. Conclusions 77
ANNEX 78
References 78

DRAFT REPORT OF THE JOINT AD HOC GROUP FOR DIGITAL REPRESENTATIONS OF LIGHT/SOUND FIELDS FOR IMMERSIVE MEDIA APPLICATIONS

1. INTRODUCTION

To capture physical environments human beings have sensors like eyes and ears. Whereas ears are acoustic omnidirectional sensors, the eye is a directional visual sensor, which collects light-rays from one direction (head and eyeball position and rotation) and focus to a specific distance (lens). The visual dynamic range in real world is so high, that it has to be limited by the iris to avoid overloading. By processing the sensor signals from either both ears or both eyes additional information can be generated, like the direction of the sound from a sound source or the distance of an object based by stereoscopic visual analysis.

Traditional digital audiovisual representations work closely to the information captured by the human audiovisual system. However this information is only a sparse sampling of the real world at the position of the human being limited by the human sensors in quality and quantity.

With the advent of better sensors, displays and higher computing capabilities the generation and use of virtual environments are becoming more and more interesting (for games, for business applications, for teaching). These virtual worlds can be completely computer generated (CGI) or captured, processed from real world and represented to the human being (virtual reality) or overlaid to the real world (augmented reality). For complete moving in these virtual environments sound and light information at all positions and from all directions are necessary to represent audiovisual samples to the human audiovisual system.

In this report we explain the state of the art of capturing and generation of audiovisual information and their reuse in virtual environments and for improved immersive audiovisual experiences. It shall identify workflows, technologies and potential digital representations of light and sound fields for immersive media applications.

2. PRIORITIZED LIST OF USE CASES

A Generalized Classification of Applications for Framing Use Cases

Use cases applicable to digital representations of light/sound fields for media applications are inherently quite broad. However, it is helpful to separate them into functional categories that capture the scope to the amount of user interaction generally expected and that identify other common characteristics for each category. The following use case categories are based on a combination of input received on the JAhG reflector and from the Plenoptic Imaging: Representation and Processing chapter in [R60]. It is important to recognize that these categories are not necessarily disjoint. Note that in all cases below, the term "captured content" refers both to natural content or synthesized content.

1. Light/Sound field communication: the field is captured and may or may not be transmitted to a receiver over a heterogeneous network. The field also may or may not be compressed, but nevertheless is essentially played back by the receiver in a form that closely matches the form in which it was directly captured, i.e. without any modification of the view or sound points that were captured.
2. Light/Sound field editing: the field is captured and subsequently modified to accent, emphasize, augment, or otherwise improve a particular view or sound point. The captured field may be augmented with computer-generated or natural scene content.
3. Free navigation: the user is free to select from multiple views or sound points or to navigate areas via graphical models. These view or sound points may be synthesized from the content that was originally captured, or may be part of the original captured content.
4. Interactive all-reality: In addition to the above categories, the presence of the user is implicitly or explicitly part of the scene associated with the captured content, and the user interacts directly with the objects in the view or sound content. This category includes virtual, mixed, and augmented reality applications.

Table 1: Use cases and their end-to-end pipeline modules

	use case	sensors	sensed data converter	encoder	decoder	renderer	presentation system	command interactions
1.	**Light/Sound field communication**							
1,1	Broadcast SuperMultiview	MultiView + depth (and/or stereo)	if stereo: extract depth map from multiview	3D-HEVC		DIBR	3D SMV	
1,2	Bi-directional communication	MultiView + depth (and/or stereo), LF camera	if stereo: extract depth map from multiview	most likely 3D-HEVC or extension		DIBR and/or Light ray selection	3D SMV	
1,3	Recording w/o transmission	Light Field camera (MultiView+depth "in-a-box")	extract depth map from multiview, microlens to elemental images	NA if angular data to code		Not applicable	Not applicable	
1,4	Playback w/o transmission	Light Field camera (MultiView+depth "in-a-box")	convert depth to avoid pseudoscopic rendering with inversed depth	NA if angular data to code		Light ray selection	2D and/or 3D SMV	
1,5	Sound Field Recording	Sound Field microphone array		Lossless audio coding with joint channel coding.				
1,6	Sound Field Playback				Lossless audio coding with joint channel coding.		Sound Field loudspeaker array	
2.	**Light/sound field editing**							
2,1	a posteriori editing	Light Field camera (MultiView+depth "in-a-box")	extract depth map from multiview, microlens to elemental images	NA if angular data to code		Light ray selection	2D and/or 3D SMV	
2,2	cinematic, mixed-reality editing	3D modeling software + MultiView + depth (and/or stereo)	extract depth map from multiview	NA if angular data to code		DIBR and/or Light ray selection	2D and/or 3D SMV	
3.	**Free Navigation**							
3,1	omni-directional 360 viewing	MultiView + depth (and/or stereo)	if stereo: extract depth map from multiview	Non-lin extension of 3D-HEVC		ODS, light ray selection, DIBR	2D and/or 3D SMV and/or 3D-VR-HMD	
3,2	free viewpoint live event	MultiView + depth (and/or stereo)	if stereo: extract depth map from multiview	Non-lin extension of 3D-HEVC		DIBR	2D and/or 3D SMV	
3,3	free viewpoint cinema/TV	MultiView + depth (and/or stereo)	if stereo: extract depth map from multiview	Non-lin extension of 3D-HEVC		DIBR	2D and/or 3D SMV	

3,4	auto navigation using 3D maps	point cloud, 3D meshes modeling software	point cloud to 3D meshes, point cloud RGB+depth projection for MultiView+depth	3DMC, PCC in 3DG CfP		splatting, OpenGL shaders	2D and/or 3D SMV	
			SLAM point cloud fusion, Lidar (theta,phi)+depth to (x,y,z)+color					
4.	**Interactive all-reality**							
4,1	surveillance with depth recovery	Light Field MultiView + depth (and/or stereo)	if stereo: extract depth map from multiview, microlens to elemental images	NA if angular data to code		DIBR	2D	
4,2	remote surgery, glasses-free 3D	MultiView + depth (and/or stereo)	if stereo: extract depth map from multiview	NA if angular data to code		DIBR	3D-VR-HMD and/or 3D SMV	
4,3	augmented reality with LF editing	3D modeling software + MultiView + depth (and/or stereo)	if stereo: extract depth map from multiview	Hybrid combination of all above		splatting, OpenGL shaders and DIBR	2D and/or 3D SMV and/or 3D-VR-HMD	
4,4	augmented reality with 3D Audio coding and rendering	Head position tracker	(yaw, pitch, roll) has interface directly to MPEG-H 3D Audio. Need to develop interface for (x, y, z, acceleration)	MPEG-H 3D Audio with metadata extensions to support dynamic acoustic environments	Model-based reverberation that represents acoustic environments	MPEG-H 3D Audio binauralization engine	Headphones	Specify personalized HRTF Continuously acquire head position Receive information about acoustic environment

Legend:
NA = Not Available

2.1 CATEGORY 1: LIGHT/SOUND FIELD COMMUNICATION

Use Case 1.1 Broadcast (one to many) for super multi-view home television/listening

In this use case, the content is captured at an origin, such as a studio or live event with a single or multiple dimension camera array or sound sensor, and is delivered over a network that is connected to multi-view/sound display/listening devices (e.g. super multi-view home television) that are capable of reproducing a high-density array of projected views or sounds. The views/sound captured at the origin is delivered over the network without modification such that the views/sounds that are projected are essentially the same as those that were captured. The views and sound may or may not be compressed at one or more points over the network. At the display/listening device, the user experiences different views/sounds as he/she views/listens to the content from various angles of the projected views and sound. In this use case, the user's interaction with the projected content is limited to the position at which he/she is experiencing the projected views/sounds.

The transmission network may be a managed or an unmanaged network, or a combination of both. The format of the captured content is designed such that the quality of experience for the end user is consistent despite varying network conditions, e.g. increased congestion or latency. This use case is similar to the traditional broadcast scenario including both broadcast over a managed (closed) network with relatively stable network conditions, or broadcast over a network that is more prone to latency or loss (e.g. wireless and mobile networks) for streaming applications.

Use case 1.2: Immersive bi-directional communication (one to one)

In this use case, the content is captured at each point (N) in a single connection or network, and delivered to all other end points (N-1) of the connection. The most typical instantiation of this use case is for bi-directional immersive communication where two users are in different locations. The light/sound fields are captured for each user and are delivered over the network to the user(s) at the other end of the connection; each end is equipped with a display/listening device capable of projecting a dense array of the captured views/sounds. Examples include "holographic facetime" or "immersive tele-presence".

Use case 1.3: Recording without transmission

In this use case, the light/sound fields are recorded and stored without transmission over a network or, potentially transmitted over a network that is not bandwidth-constrained. For this use case, compression may still be needed.

Use case 1.4: Playback without transmission

This use case is the dual to use case 1.3, i.e. the recorded content is played back without transmission over a network, or possibly over a network that is not bandwidth constrained. An example is playback for Blu-Ray players.

2.2 CATEGORY 2: LIGHT/SOUND FIELD EDITING

Use case 2.1: a posteriori editing

In this use case, the user captures a light/sound field, and subsequently interacts with an application that support commands to adjust the resulting view/sound. A generic example of

this use case is the a posteriori adjustment of the focal point in a single light field image that is captured by a light field camera.

A concrete example: Anna loves to take pictures and share them. She just bought a new camera that allows her to do all sorts of nice editing and manipulations after the initial capture. Today, she took a walk through the park during lunchtime and captured some great pictures of ducks in the pond, tree blossoms, and even one of a dog catching a frisbee in mid-air. She immediately uploaded these for her friends to experience. Through a web browser, her friends were able to change the focus and viewpoint of the images that Anna shared. They continued to chat about their favorite versions.

Use case 2.2: Cinematic, mixed-reality editing

A dense array of sensors captures the light/sound field information facilitating the computation of depth information, and the modeling of a scene wherein the objects, focal points, or illumination can subsequently be edited, moved, or deleted, and objects (virtual or natural) inserted into that captured scene. In this case, the user is interacting with an application specifically designed for the editing of light/sounds fields; that application supports a rich set of commands that provide a complex set of features (similar to existing photographic editors such as Photoshop or Da Vinci Resolve). A recently published example of cinematic editing is provided here. This example illustrates the complex interaction between the user and the commands supported by the editing application, and the interaction between the application and the captured light/sound fields. A common example of this use case is the editing of content captured by the dense array of sensors for the production of a movie to be presented in cinematic theaters.

2.3 CATEGORY 3: FREE NAVIGATION

Use case 3.1: Omni directional 360° viewing/listening

A 360° sensor captures scenes/sounds of the existing environment or context. A simple example of this use case is the extraction of a single view selected by the user via a mouse or joystick connected to a display or by motion detected in a motion-sensing head mounted display. In this scenario, the particular view/sound selected by the user is provided from the perspective that the user is at the center of the content with the ability to "look out" or listen to the surrounding environment; the captured light/sound 360° content virtually surrounds the user, and the user selects a particular view of that content by use of the mouse/joystick, or by orientation of the user's head sensed by the head mounted display. The selected view is extracted from captured 360° content and synthesized or rendered by the display/listening device.

Use case 3.2: Free viewpoint live event

This use case is conceptually similar to use case 3.1 with the exception that the cameras/sensors are positioned around an enclosed or partially enclosed scene that is captured. Multiple sensors are positioned to capture multiple views of the live event. The cameras may be fixed (as in the perimeter of a sports stadium) or along cables or in a track that allow the camera to move to capture multiple views of the scene. From the perspective of the user, the user is "looking into" (rather than looking out at) the captured scene. Selection of the desired view is achieved as in use case 3.1. Popular generic examples of this use case include the positioning of multiple cameras along the upper levels of a sports stadium to capture the events on the sports field, e.g. helicopter views.

A concrete example follows: The final match of the World Cup is on tonight. Fred has invited a bunch of his friends to his place to watch the game in his new media room. The picture is stunning and gives a sense of being there through its immersive 3D display and surround sound system. Ten minutes before the half, there is an offside call – some of his friends think the referee made the right call, others are not so convinced. While play continues on the big screen, Fred reviews the footage on his tablet and adjusts the view to the perfect angle to see that the referee was indeed correct.

Use case 3.3: Free viewpoint cinema or television

This use case is also conceptually similar to the earlier use cases with the exception that the rendered view is expected to be of especially high quality to support close examination of objects, or facilitating stop/start, zooming in/out of motion and objects in the scene, e.g. as in the Matrix movie bullet time sequence. To support this use case, a high density array of cameras is placed around or in an arc encompassing a particular scene in a studio. The cameras capture the multiple views of the scene possibly at different frame rates, permitting the production of special effects for the movie.

Use case 3.4: Auto navigation using 3D maps acquired by a network of sensors

In this use case, a potentially dense network of sensors is placed throughout an area (e.g. a city block) to capture real-time views that are used to update a graphical model of the area vicinity in order to facilitate navigation of that area.

Concrete example: Alex does not drive into the city often, but today he needs to visit a store in the center to pick up a gift. He communicates the destination to his self-driving car. The car plans its route and downloads the latest 3D maps, which are stored as 3D point clouds with various attributes. The maps were initially generated by dedicated mapping systems that use laser scanning and camera data, and are updated regularly when changes occur. He notices that there are some lane closures and detours today due to ongoing construction. Using the sensors on his car and the latest updates to the 3D map, his car is able to safely navigate to a nearby underground parking lot. However, he still needs to get to the store and is unsure which direction to go. On top of that, he has no signal in the underground garage. Fortunately, the 3D map of the garage and shopping center has been transferred to his phone to guide him the rest of the way.

2.4 CATEGORY 4: INTERACTIVE ALL-REALITY

Use case 4.1: Surveillance with depth-recovery

This use case extends previous use cases with explicit sensing of depth, rather than implicit computation of depth. In this generic example, a light field camera is mounted on a moving robot that is pre-programmed to survey a particular scene. The captured data is filtered to measure the depth of the objects in the scene.

Use case 4.2: Remote surgery with glasses-free 3D display

This use case is conceptually similar to the immersive bi-directional use case 1.2. In this case, a surgeon interacts directly with a 3D model (possibly a point cloud representation) of the surgical field of a patient in an operating room equipped with robots to respond to commands that are derived directly for the movement of the surgeon's hands. A light field camera or array of

cameras captures the visual field of the operation with explicit measurement of depth. Time of flight cameras capture the movement and positions of the surgeon's hands permitting transmission of the commands that result in the precise execution of the movements required to be performed by the surgical robot. The surgeon views the surgical field either by a super multi-view display or holographic display.

Use case 4.3: Augmented reality surveillance with light field editing

In this use case, surveillance is performed as in the use case 4.1, but with the ability to filter the monitored scene to remove noise (e.g. dust particles on the camera lenses or other unwelcomed objects in the scene). This use case utilizes the editing features from category 2 so that the captured light field information can be processed to remove or focus on specific objects, noise patterns, in the scene.

Use case 4.3: Virtual/Augmented reality with 3D Audio

In a Virtual Reality use case an audio-visual scene is rendered to a user's head-mounted display and headphones. However, the user's head turns or the user's position changes, the scene is rendered consistent with the virtual reality. For audio rendering, this involves changing the orientation of the sound stage as the head position changes and perhaps also altering the sonic rendering, e.g. with varying degrees of reverberation. Since the rendering is of a virtual world, all sound objects and sound processing can be known prior in advance.

An Augmented Reality use case is similar in presentation, but since it is a virtual world (e.g. one or more virtual sound objects) imposed on true reality, aspects of the real world may not be know in advance. For example, the user moves into a physical space with new and different sound properties. Such sound properties (e.g. room impulse response or reverberation) may have to be inferred from other sensors, e.g. position or visual.

3. OVERVIEW OF CONCEPTUAL WORKFLOW

This section intends to present the conceptual processing workflow for the light/sound wavefield from its capture/creation until its display/consumption. It is accepted that the both naturally acquired and synthetic computer generated content will be considered.

The light/sound field conceptual processing flow is presented in Figure 1. It is expected that this conceptual flow may drive the rest of this report as it allows breaking down the overall processing chain into basic fundamental modules which deserve individual study.

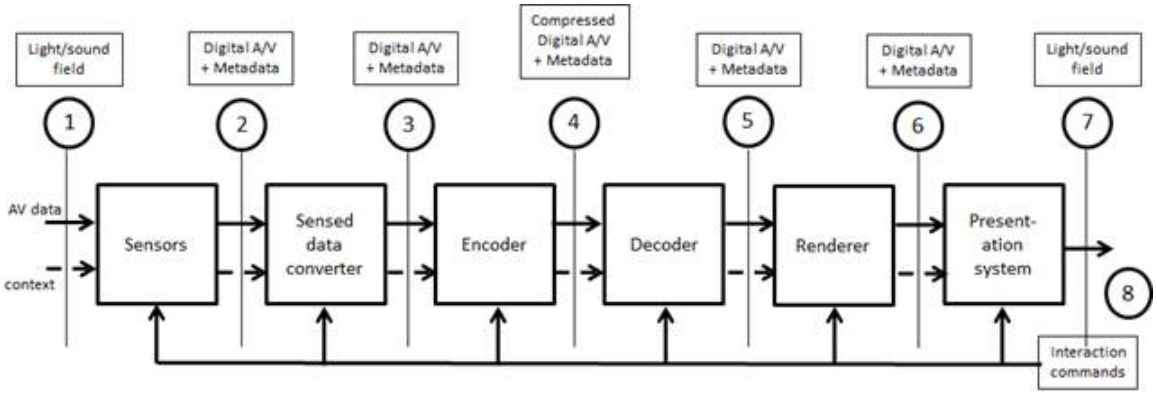


Figure 1 - Light/sound field conceptual workflow

The conceptual light/sound field processing flow considers the following main steps which should be similar both for natural and synthetic content:

1. **Sensors** – The first module deals with the light/sound field data acquisition by using appropriate sensors which are in continuous evolution, especially recently. Naturally, a wide range of sensors may be used from 2D regular camera arrays to time-of-flight depth sensors as well as light field cameras. Each sensor follows a specific data acquisition model which may adopt a regular sampling grid, e.g. light field cameras, or an irregular sampling grid, e.g. non-regular array of cameras. It is considered here also the case where the data is computer created and not acquired from a physical scene. Computer creation may generate data using the same model/format as the natural acquisition. If the synthetic content creation already uses the representation model/format appropriate for the Encoder, the next Converter will not need to make any conversion; this may happen if synthetic content is created as point clouds and point clouds are also selected as one of the representation formats for which an Encoder is available. Moreover, the sensor module may also include the capture/creation of some metadata in some sensor specific format. Examples of sensors are regular 2D cameras, 1D or 2D arrays of cameras and microphones, light field cameras and depth sensors.

2. **Sensed data converter** – The acquisition/creation data model/format may not be necessarily the best model/format for the data representation when considering the overall application requirements, e.g. compression, user interaction, etc. Thus the sensed data converter performs the conversion of the sensed data into a more appropriate representation model/format, e.g. multiple video views and depth data, or point clouds with RGB attributes. Notably for interoperability reasons, the number of representation models/formats should be as small as possible, and even ideally be one universal format. However, considering the variety of

application scenarios and associated requirements, it is more reasonable to target a rather limited number of representation formats, expecting that the remaining formats may be converted to the selected key formats; examples of key models/formats are light fields and point cloud based representations. Naturally, this conversion may happen both for the data and metadata although the data conversion should involve higher complexity and a higher impact on the final visualization experience, depending on the conversion assumptions and constraints. An example of a relevant conversion may be transforming a set of texture and depth views into a set of point clouds with RGB values.

3. **Encoder** – The encoder targets the efficient coding of the data at hand referred to as compression efficiency, while fulfilling an additional set of functionalities such as random access, scalability, error resilience, etc. Compression efficiency means usage of the smallest number of bits for the target quality while fulfilling other identified requirements. It is often considered as the critical functionality. It must be understood that this encoding process may be either lossless or lossy although it is more commonly lossy. If lossy, it is typically important to consider the human perception characteristics, notably visual and auditory characteristics depending on the data modality at hand. While ideally a single Encoder would maximize interoperability, the number of required Encoders will depend on the number of selected representation formats. It may also happen that natural and synthetic adopt different representation models, thus requiring different Encoders (and thus Decoders) to coexist for the same scene. Finally, it is important to highlight the importance of 'quality metrics' already in this context to assess the rate-distortion performance of each encoder and also compare performance; however, contrary to past situations, the decoded data will not be readily ready for visualization, e.g. point clouds shown as video views, and thus 'quality metrics' may have to consider also the impact of the next rendering stage if targeting to express the final visualization quality. Again, the encoding process applies both to data and metadata, eventually considering slightly different requirements.

4. **Decoder** – The decoder is a natural counterpart of the encoder and should recover the data and metadata in the adopted representation model/format after storage or transmission. Depending on the specific lossy encoder, the data itself and the rate used, the decoded data will have a certain amount of distortion with regard to the corresponding original data. For many application domains, the decoding complexity is more critical than the encoding complexity (e.g. broadcasting), but in others the encoder complexity may be more critical (e.g. camera) and in some of equal importance (e.g. two-way teleconferencing).

5. **Renderer** – The renderer has the main function of extracting from the decoded data the appropriate data for the best quality of experience achievable with the presentation system, e.g. display. While in the past decoded content could directly pass to the presentation system without much rendering - although post-processing to improve quality and aesthetics, e.g. filtering, contrast enhancement, etc. could be always applied but were not mandatory - this will not be often the case now as the very rich representation of the scene available will have to be 'mined' to extract the relevant data, e.g. for a specific viewpoint, focus plane, etc., eventually under user control through explicit command interactions or indirect interactions, e.g. with head mounted display motions. Because rendering may become a rather complex and sophisticated piece of technology, it will critically determine the quality of experience, eventually even more than the coding modules which introduce representation data distortion before the rendering. Again the metadata has been processed, eventually in close synchronism

with the data. It should be also possible to locally enrich the overall scene with additional natural and synthetic content, more likely synthetic data like overlays. If locally available already decoded, this content should be added at the Rendered; however, if locally available still coded, it should be inserted first at the Decoder.

6. **Presentation system** – The final presentation system, very likely including visual and audio components, will not only determine the user experience but also the technical solutions (and requirements) for the previous modules. Along with the evolution on the sensors, also the presentation systems, notably displays, have been evolving significantly recently, e.g. autostereoscopic, light field and head mounted displays. Naturally, if a light field display is available, it will only be possible to take full benefit of it if light field data is available. However, especially in the early days, it will be common to have rich data, e.g. light fields or point clouds, which has to be displayed in more conventional displays, e.g. 2D regular displays; this is where the rendering module has to play a critical role by extracting from the available rich data the presentation data to allow the best user experience. Since it is critical to assess the quality of experience, both subjective assessment methodologies and new objective metrics will be required.

7. **Command interactions** – One of the reasons to have richer representations of the scene, is precisely to allow users the types of interactions that naturally happen in the real world but not with technological replicas of the real world, at least until now. By command interacting with the presentation system, the user may not only perform simple actions like sound control, contrast enhancement, etc. which regard to the presentation system itself but he/she may also, e.g. control the point of view, the focal plane, the lighting conditions, etc. This type of interactions mainly regards the rendering module which has to extract from the fully or partially decoded data the appropriate displayable data in reaction to the user request. However, the command interaction may go back to the decoder if only the data relevant for the requested user experience, e.g. specific points of view, is decoded to save decoding resources, or even back to the sending side, e.g. defining the preferred points of view that should be captured or captured with more detail. With time, it is expected that the user interactions with the light/audio data become as natural and powerful as in the real world.

In the next sections, the specific solutions and technologies for some of the modules will be reviewed.

4. DATA ACQUISITION AND CONTENT CREATION

4.1 VISUAL DATA ACQUISITION

4.1.1 INTRODUCTION LIGHT FIELD

A light field is a representation of all light rays in space. It will be described by a vector function *L(x,y,z,Θ,Φ)* that defines the amount of light flowing in every direction through every point.

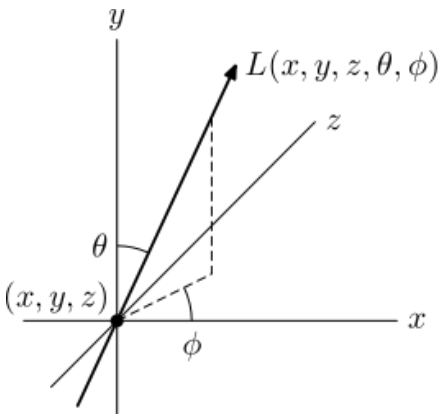


Figure 2 - Light field representation (Wikipedia) [R2]

If the data from a light field is known, then views from all possible positions can be reconstructed, even with the same depth of focus by combining individual light rays. Although the capture of a complete light field is not possible, a light field can be sampled sparsely, enabling the modeling of a virtual environment. This model can be used to reproduce views or to combine the data with computer generated data.

The capturing of light with sensors for movie representations typically follows similar principles of the human eyes. Three different sensor cells capture individual wavelength of the light, allowing representing a scene in color. This increases the dimensions of information additionally. Each camera records the color information, either as RGB or in a luminance/chrominance format, associated with a particular viewpoint. The data captured by a camera may also be referred to as the texture of the scene.

4.1.2 OVERVIEW SENSORS AND SENSED DATA FOR VISUAL APPLICATIONS

SINGLE MOVING CAMERA

The simplest way to capture a light field is to use a single camera and to capture the scene from different viewpoints by moving the camera. However this works only for static scenes and objects, where the scene and the light rays do not change during the capturing process. Such kind of arrangements will be used with cameras mounted on robot systems. This is a standard method in 3D-object scanning.

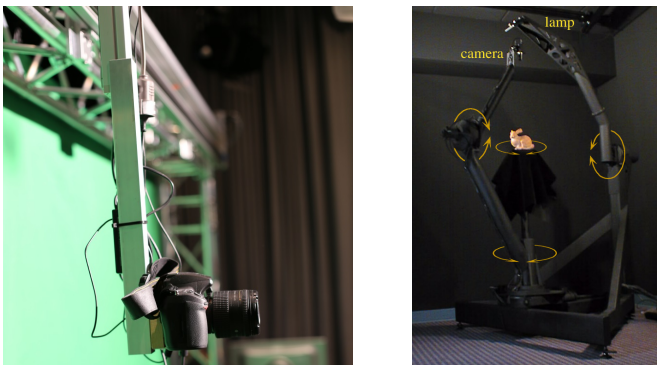


Figure 3 – Camera mounted on robot system (Fraunhofer IIS, Stanford [R3b])

CAMERA ARRAYS

For moving objects and where good reconstruction or processing of scenes is necessary specific light field cameras or camera arrays in regular grid arrangements will be used.

With multiple cameras a sparse light field can be captured in one step. For Free Viewpoint applications irregular arrangements of cameras will be used to generate free viewpoints after processing the camera data. Main application is the creation of virtual viewpoints during sports events.

Figure 4 – Camera arrangement for free view point applications (TU Braunschweig) [R4]

There are many possible arrangements of camera arrays. Figure 6 shows a few example setups including a 1D linear array, a 2D linear array, a 1D array of cameras in a circular arrangement, as well as an unstructured array. 1D linear arrays have typically been used for auto-stereoscopic or multi view display, which support horizontal parallax, while 2D arrays make it possible to support full parallax displays. Arc or circular camera arrangements permit free viewpoint navigation around an area or object of interest, including Matrix-like special effects, see Figure 5.

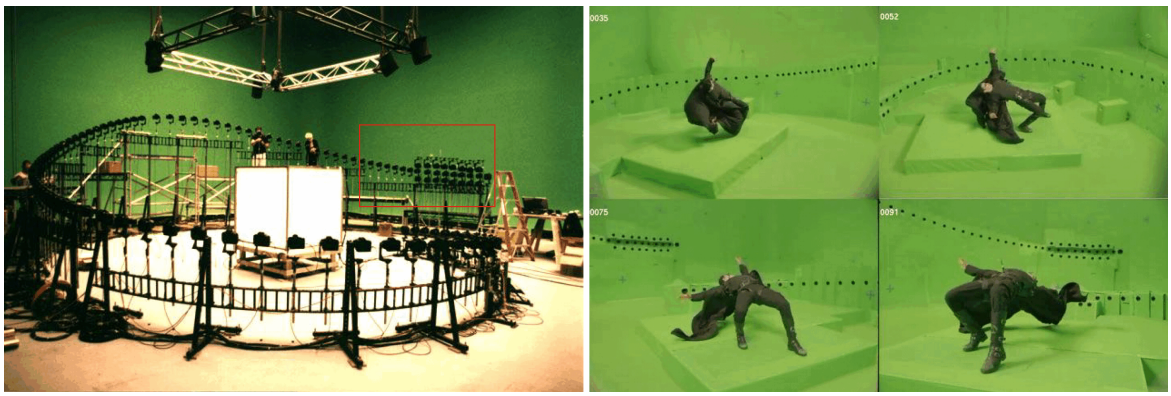


Figure 5 - The Matrix multi-camera setup (mainly 1D non-linear, partially 2D cf. red box) [R5a][R5b]

The capture of video from an arbitrary or unstructured arrangement of cameras occurs in some professional setups such as surveillance camera networks or sports stadiums, but could also occur from simultaneous capture of a scene from mobile handheld devices.

In general, better approximations of the light field will be obtained with denser camera arrangements. Furthermore, a denser camera array will ease the view rendering process and alleviate the need for additional geometric information about the scene, such as depth.

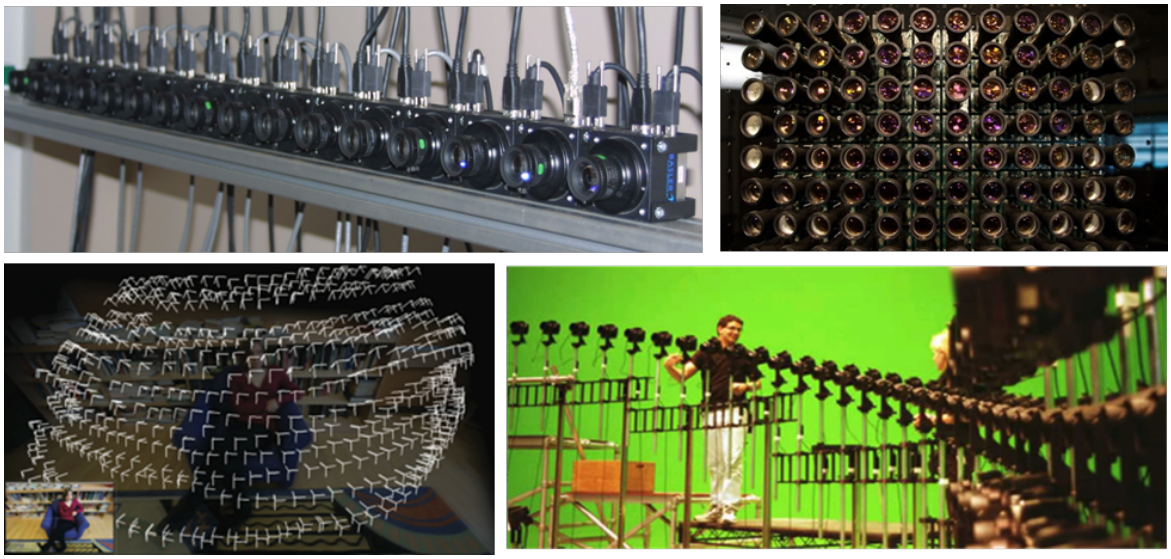


Figure 6 - Example camera arrays (clockwise from top-left): 1D linear, 2D linear, 1D circular and unstructured/arbitrary.

MICROLENS CAMERAS

A multilens camera can be used for specific applications where refocusing, small object scanning or near field distance measurement is necessary. The advantage is the compact design, no need for synchronization of multiple cameras, but the small baseline between the captured images limits the potential applications. There are different types of cameras:

- Cameras with micro-lens array only, so called insect eye cameras

  The main reason to design such cameras is the ultra-compact form factor, which allows camera heights of some few millimeters inclusive optics, which is of specific interest for the smartphone industry. Rough depth estimation in the near field is possible. Light field processing is necessary to reconstruct the typical 2D-image of a camera.

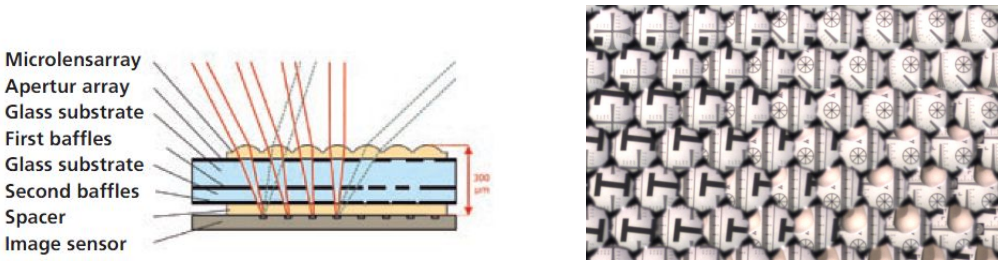


Figure 7 – Insect Eye camera and corresponding image (Fraunhofer Facetvision) [R7]

- Cameras with micro-lens array and main lens

  The multi-directional color information of a point in the scene is captured over different pixels on the CMOS sensor, through a microlens array, see Figure 8. In effect, adjacent viewpoints of the scene are acquired, supporting small perspective changes when rendering the scene from each microlens viewpoint, referred to as the Elemental Images (EI).

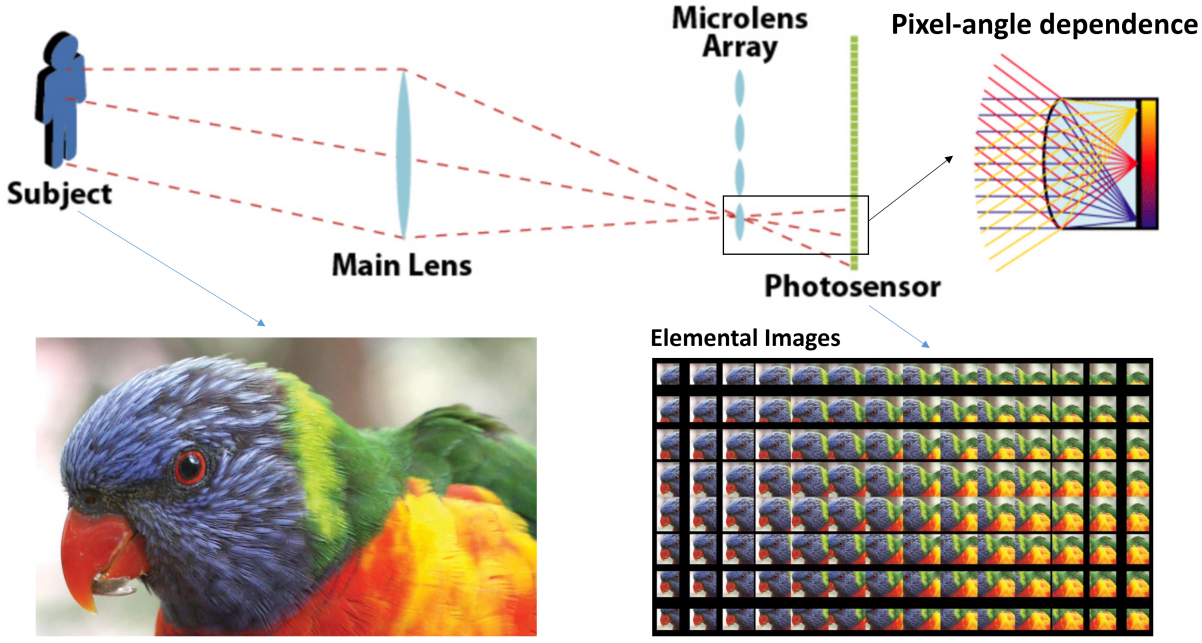


Figure 8 - Microlens based Light Field camera and its Elemental Images

Figure 9 – Plenoptic cameras with microlens and main lens (Lytro) [R9]

360 DEGREE CAMERAS

360 Degree cameras are becoming more and more important for capturing of images in VR scenarios. While stitching is easy when a common nodal point is available for all cameras - which require a mirror rig, it becomes more difficult for standard setups. Light field processing can solve artifact free stitching by calculating depth maps at the overlap of two images.

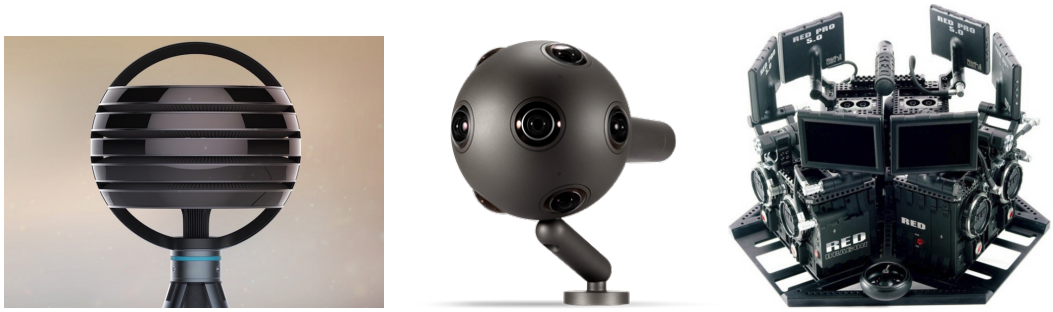


Figure 10 - 360 Degree Camera: Lytro Immerge, Nokia Ozo, NextVR multi-stereo

DEPTH CAMERAS

Information about the depth of objects in a 3D scene provides important geometric information that could be useful for rendering virtual viewpoints. Very accurate, per-pixel depth is often needed for high quality rendering. In computer-generated imagery, which is popular for cinema production, the depth is an inherent part of the data format. However, for natural scenes, the depth must either be estimated from a stereo or multi-camera setup, e.g., using stereo correspondence techniques, or acquired through specially designed depth cameras. Although the resolution is traditionally lower with depth cameras, depth acquisition technology has advanced significantly in recent years with designs based on structured light or time-of-flight (ToF) based imaging. Several example depth cameras are shown in Figure 11.

Infrared cameras: capture the object distance using infrared radiation, typically in wavelengths as long as 14 µm. Infrared cameras may be used as part of structured light projectors and ToF cameras.

Structured light projectors: measure depth by solving a correspondence problem between a reference structured light pattern (grids, horizontal bars or random dots) and the captured light pattern, after projecting the reference pattern onto a scene.

ToF range imaging cameras: measure the distance by means of the time-of-flight of a continuous light signal between the camera and the subject for each point of the image, resulting into depth maps. Some devices work by modulating the outgoing beam with a radio frequency (RF) carrier and measuring the phase shift of that carrier at the receiver side.

LIDAR cameras: measure the distance to a target by illuminating it with laser pulses and analyzing the reflected light, resulting into depth maps. ToF and LIDAR are typically appropriate for shorter (< 5 m) and longer distances, respectively.

Figure 11 - Example depth cameras (from left to right):
Mesa Imaging's Swiss Ranger SR3000 and SR4000, Microsoft Kinect 360.

GRAPHICS AND COMPUTER GENERATED CONTENT

In 3D graphics, content can be synthetically modelled by a 3D modeling software, cf. Figure 12 (3D mesh + 2D texture), or can be captured as point clouds with aforementioned depth cameras, cf. Figure 13.

Figure 12 - 3D synthetic content created from a meshed 3D shape onto which 2D textures are added

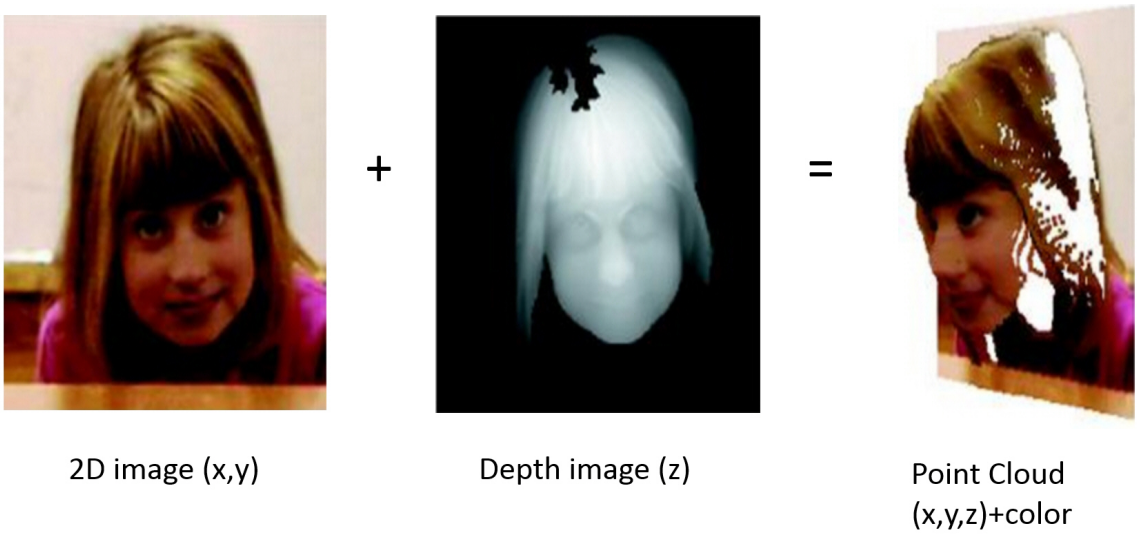


Figure 13 - Point cloud data representation (right) with a projected image (left) and a depth map (middle)

4.2 Audio Data Acquisition

4.2.1 Introduction

MPEG Audio standards have typically addressed the use case of a listener enjoying audio content, where content may be played from storage or streamed from a communications or broadcast channel. Often the audio will be a component of audio/visual content. While the user may be seated at home or walking down the street while consuming content, the user is a largely a passive consumer and the audio content is typically stereo, 5.1 channel or perhaps as immersive as 22.2 channel.

In a first section of this document we propose, a much more interactive use case for audio consumption, as a component of a Virtual Reality or an Augmented Reality session. In a second section, we propose a much richer audio capture and playback use case, that of Sound Field capture and reconstruction.

4.2.2 Virtual and Augmented Reality

Virtual Reality

In virtual reality the user might wear headphones and a head-mounted display that incorporates head tracking sensors. The sensors would provide the audio rendering engine with

- Pitch, roll, yaw

Information so that the audio rendering engine can constantly "orient" the virtual sound stage with respect to the listeners head position. This fits well with the concept of watching a movie with a 360 degree visual and sound stage. In this use case virtual sound objects have only an azimuth, elevation position relative to the user. While the position may be dynamic, sonic aspects of distance change or acceleration are "pre-baked" into the audio object and are not expected to be applied in the decoding and rendering.

For a true virtual world, the sensors would need to deliver

- pitch roll yaw (head orientation)
- x, y, z (user position in virtual world coordinates)
- acceleration (of user in x, y, z coordinate framework of virtual world)

And the audio decoder and render may need to apply aspects of

- doppler shift (due to acceleration)
- reverberation (which may change as x, y, z of user or sound object change)

In a virtual word there would be an "architect" of the world that would specify the reverberant properties at any (x, y, z) position in the world. An audio rendering engine would have to realize desired doper shift and reverberation effects.

Augmented Reality

Augmented reality is very similar to Virtual Reality, but with one very important difference – there is no virtual world "architect" that specifies apriori the reverberant properties of some

arbitrary real-world space that the user may currently occupy. The audio rendering engine can position virtual sound objects in space relative to the user's head orientation, e.g. using anechoic HRTFs, but would be unable to add realistic reverberation as the reverberant properties of the current user environment are unknown.

One solution would be for a head-mounted camera to sense properties of the user location (e.g. I am at an estimated location and orientation in a room of an estimated size with estimated reflectivity properties of any floors, walls and ceilings). This would involve joint audio/visual processing and metadata sharing, which is discussed elsewhere in this document.

4.2.3 Metadata for Virtual and Augmented Reality

As already mentioned, metadata for Virtual and Augmented Reality must supply

- pitch roll yaw (head orientation)
- x, y, z (user position in virtual world coordinates)
- acceleration (of user in x, y, z coordinate framework of virtual world)

of users and virtual sound objects in the world.

In addition, Augmented Reality may require acoustical property information, such as

- Piece-wise planar estimation of bounding enclosures or other sonically reflective objects in the world.
- The acoustic reflectivity coefficients of each planar segment.

Suitable coordinate systems or representation methods for the enclosures or significant acoustic reflection objects in the space will be required.

4.2.4 Sound Fields

The traditional capture, recording and playback of audio signals used discrete microphone to capture and record sound objects and a few loudspeakers (e.g. stereo, 5.1, 7.1, or 11.1 configurations) to play back the audio information. This can be very engaging, but suffers from the deep assumption that a sound object can be adequately represented by capture at a point source and then rendered by one (or a small number) of point sources.

Sound field representations remove this assumption and permit capture and playback of a full audio sound field as the user's ears would experience in the real world.

4.2.5 Sound Field Communication Scenarios

The following is a list of scenarios that imply different workflows involving sound field capture and reconstruction:

1. Sound Field Capture and Recording: To capture and record the sound field with the highest possible fidelity. There is no bandwidth limitation and no transmission channel, but compression might be needed to moderate the volume of data. The number of sensors will dictate the requirements in this scenario.

2. Sound Field Playback and Reconstruction: This is the complement to capture and recording, where the sound field representation is playing from a medium that is not bandwidth limited.
3. Sound Field Unidirectional Broadcasting/Streaming: This is a one-to-many scenario, where the sound field is captured by one entity and location, transmitted through a bandwidth constricted channel, and consumed by many entities at many locations.
4. Sound Field Bi-directional communication: This case envisions a bi-directional communication in which the consumer sends and receives sound field representations.

4.2.6 Sound Field Sensor Configurations

Consider two critical aspects of all of the cases above: sound field capture and sound field reconstruction.

Sensors on a plane

A simple illustration of sound field capture and reconstruction is to imagine an infinite plane between a listener and the sound objects of interest. As sound waves propagate from objects to listener they cross the plane. This is illustrated in Figure 14, below.

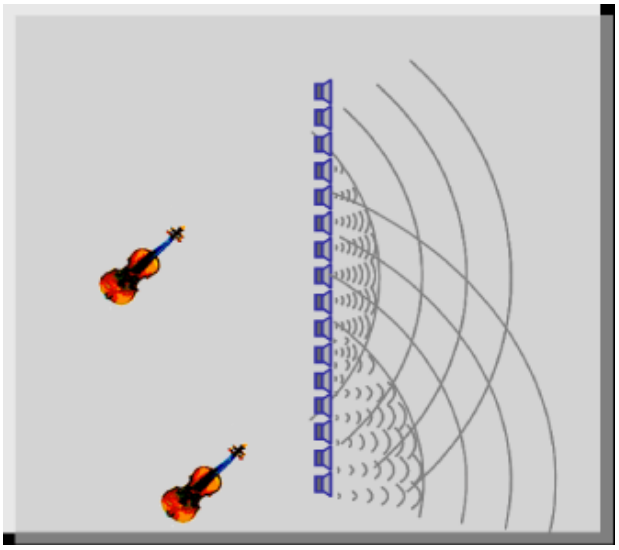


Figure 14 - Wave field crossing a plane

The most efficient placement of sensors in a plane is hexagonal packing, as shown in Figure 15, below, where "X" indicates the sensor position.

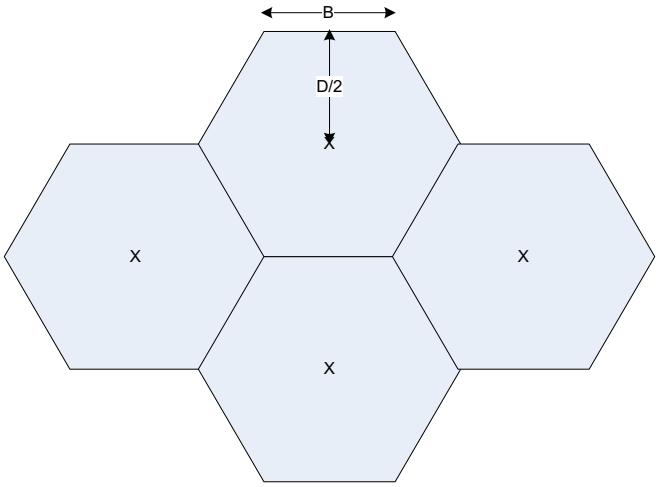


Figure 15 - Sensors with hexagonal tiling

Although the world that humans perceive is analog, we wish to represent aspects of this world as some digital information sequence. For a single audio sensor, we can assume that a (temporal) sampling frequency, fs, of 48 kHz is sufficient to represent audio signals since it is widely assumed that the frequency range of human hearing is 20 Hz to 20 kHz.

The capture of a sound field requires both a minimum temporal sampling frequency and a minimum spatial sampling frequency. The spatial sampling frequency is determined by the spatial location of the sensors. Assume that sensors are uniformly distributed on the surface of a plane using hexagonal tiling, as shown in Figure 15, above. The distance from one sensor to any of its nearest neighbors is D where

$$B^2 = (D/2)^2 + (B/2)^2$$

or

$$D = B\sqrt{3}$$

where B is the length of the hexagon edge and D/2 is the distance from the hexagon center to the mid-point of an edge.

A simpler proposition is to consider sensors on a horizontal line instead of a plane. In this case, sensors would be placed with a distance D apart.

In either plane or line configuration, if sensors are positioned too far apart the reconstructed sound field will suffer from "spatial aliasing" in that it is not able to accurately capture signal components with wavelengths shorter (frequencies higher) than a critical distance fc. This is illustrated in Figure 16, below. The figure shows a plane wave impinging on the sensor plane at an angle Θ. The wave length is λ, as measured from one wave crest to the next.

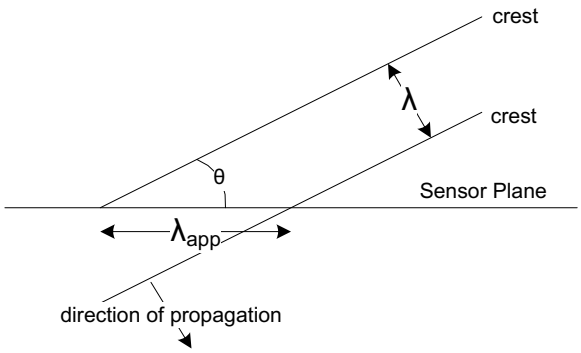


Figure 16 - Plane wave impinging on sensor plane at angle Θ

Because of the angle of incidence, the sensors on the plane sense a longer apparent wavelength λapp which is:

$$\lambda_{app} = \frac{\lambda}{sin\theta}$$

The angle of incidence determines the critical frequency:

$$f_c = \frac{c}{2Dsin\theta}$$

or

$$D = \frac{c}{2f_c sin\theta}$$

where c is the speed of sound and D is the sensor spacing. The critical frequency fc is the maximum frequency that the sensors can detect without aliasing distortion. As the angle of incidence Θ increases, the critical frequency decreases up to the limit when Θ = 90 degrees:

$$f_c = \frac{c}{2D}$$

Although we have assumed an infinite plane (or line) of sensors, this is not possible in practice. A finite-extent sensor line or plane gives rise to a "truncation effect" distortion in the sound field capture and reconstruction.

In signal processing terms, this is spectral leakage in the spatial domain and is caused by application of a rectangular function as a window function on what would otherwise be an infinite array of sensors. The leakage can be reduced if the detection or reproduction level of the outer sensors or transducers is reduced. This corresponds to using a window function which tapers off at the edges. Issues of windowing and spectral leakage are well known and will not be further discussed.

SENSORS ON A SPHERE

A variation of sensors on a plane is sensors on a sphere. As the radius R of the sphere goes to infinity, the sphere becomes a plane, so there is much in common. However, there are compelling aspects to the notion of sensors on a sphere. Here we propose two distinct scenarios.

The first scenario is user-centric, in which the soundfield impinging on a virtual user is captured and then reproduced at a different time and place for an actual user. In this case the sound field is produced by objects farther than some distance R from the virtual user.

The second scenario is object-centric, in which the soundfield emanating from an actual sound object is captured and then reproduced at a different time and place to simulate a virtual object. In this case the soundfield is produced by the actual sound object and perceived by users farther than some distance R from the virtual sound object.

In each case we can consider that the sound field is captured by sensors on the surface of a sphere of radius R. In the user-centric case the sensors (i.e. microphones) face outward, or have their directivity pattern facing outward, to capture sounds outside of the sphere. In the object-centric case the sensors face inward, or have their directivity pattern facing inward, to capture sounds inside of the sphere. If capture sensors have an omni-directional directivity, then the sensor configuration for both cases is the same.

For sound field reconstruction, the user-centric case would have transducers (i.e. loudspeakers) on the surface of the sphere which have their directivity pattern facing inward, while the object-centric case would have transducers with directivity facing outward.

Figure 17 below, shows the sensor/transducer configuration for either user-centric or object-centric cases.

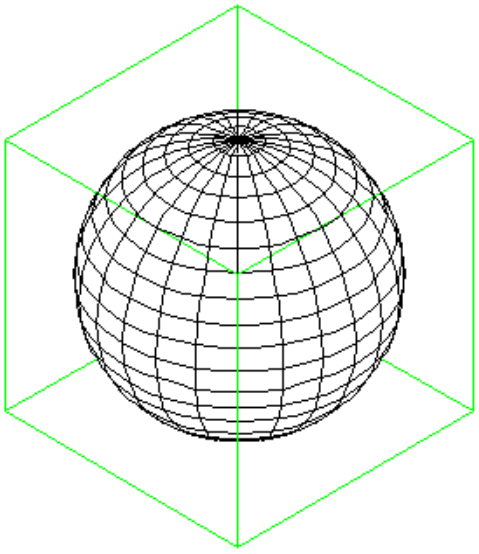


Figure 17 – Sensors on an enclosing sphere. Either sound source object is at center of sphere and listener is outside, or listener is at center of sphere and source objects are outside.

The user-centric scenario can be tailored for a single individual, a small group or even a large group if the value R is sufficiently large. The object-centric case can be tailored for a single object or several objects if R is sufficiently large. Alternatively, for the purpose of capture, there can be a single sphere per sound object and similarly a single sphere per sound object for reconstruction.

If the surface of a sphere of radius R is uniformly tiled with hexagons, then the number of sensors N is the surface are of the sphere divided by the area of the hexagon. Using the dimensions D and B from Figure 15, above, and the equation for the critical frequency fc, above, N is:

$$N = \frac{4\pi r R^2}{(B^2)\left(3\sqrt{3}/2\right)} = \frac{4\pi R^2}{D^2\left(\frac{\sqrt{3}}{2}\right)} = \left(\frac{2}{\sqrt{3}}\right)\frac{4\pi R^2}{\left(\frac{c}{2f_c sin\theta}\right)^2}$$

As can be seen, the assumption concerning the maximum angle of incidence is critical in determining the number of sensors needed. However, if an angle of incidence of up to 60 is desired, then a sphere of radius 4 meters would require N = 600 000 sensors to fully capture a sound field with a 20 to 20 kHz bandwidth.

4.2.7 METADATA FOR SOUND FIELD CAPTURE AND RECORDING

Any capture and recording of an audio sound scene should have associated metadata that sufficiently documents the salient aspects of the event, and such metadata will not be dealt with here. Metadata that is unique to sound field capture and reconstruction might be:

- Sensor geometry (i.e. sphere radius R, number of sensors N)
- Sensor geometry truncation. It might not be meaningful to record the “lower half” of a sphere of sensors for a sound object places on an acoustically solid floor.
- [Others?]

5. Data Conversion

5.1 Visual Data Conversion

All data formats presented in this document show many commonalities; hence bijective data transformations from one representation to another exist and are surveyed in this section.

5.1.1 SLAM

Simultaneous Localization and Mapping (SLAM) is the process of constructing a 3D map (point cloud) from multiple point clouds, each taken from a different localization (scanner position) without actually knowing these localizations precisely. SLAM will fuse these point clouds, gradually updating hypothesis localizations until the distance between the points of the different point clouds is minimized. Figure 18 (left) shows two misaligned point clouds of the same scene, and Figure 18 (right) shows the SLAM alignment, which effectively increases the density of the resulting point cloud.

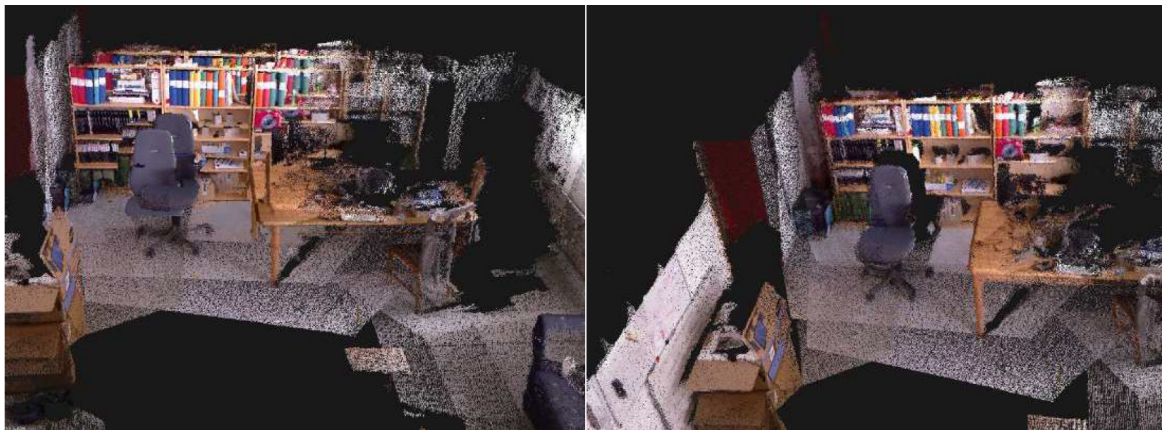


Figure 18 - SLAM Fusion of point clouds before alignment (left) and after alignment (right) [R18]

SLAM is very useful in Lidar scanning where the density of the point cloud decreases with the distance between the scanned objects and the scanner localization. In order to get a more uniform point cloud, the scene is scanned from different positions and the acquired point clouds are fused together using SLAM. The resulting high density point cloud can then be coded as explained in section 6.

5.1.2 Point Clouds to Depth Maps

As explained in section 4 and shown in Figure 19 (top-right), the points captured by a Lidar scanner are actually acquired in spherical coordinates, i.e. angles and depth. Most often, these coordinates are transformed inside the scanner into Cartesian coordinates (x,y,z), cf. Figure 19 (bottom-left).

The authors of Figure 19 (bottom), however, consider the conversion of these Cartesian coordinates (possibly after a SLAM pre-processing) back into projected depth maps, representing the distance of each point to a plane, which is often taken as a face of a cube surrounding the object/scene. Figure 19 (bottom-right) shows such depth map, together with

lowpass and highpass subsampled versions (wavelet decomposition) for further coding. Section 6 will show some competitive coding performances following this data format conversion approach, which is actually equivalent to the Texture + Depth data representation format of Figure 13.

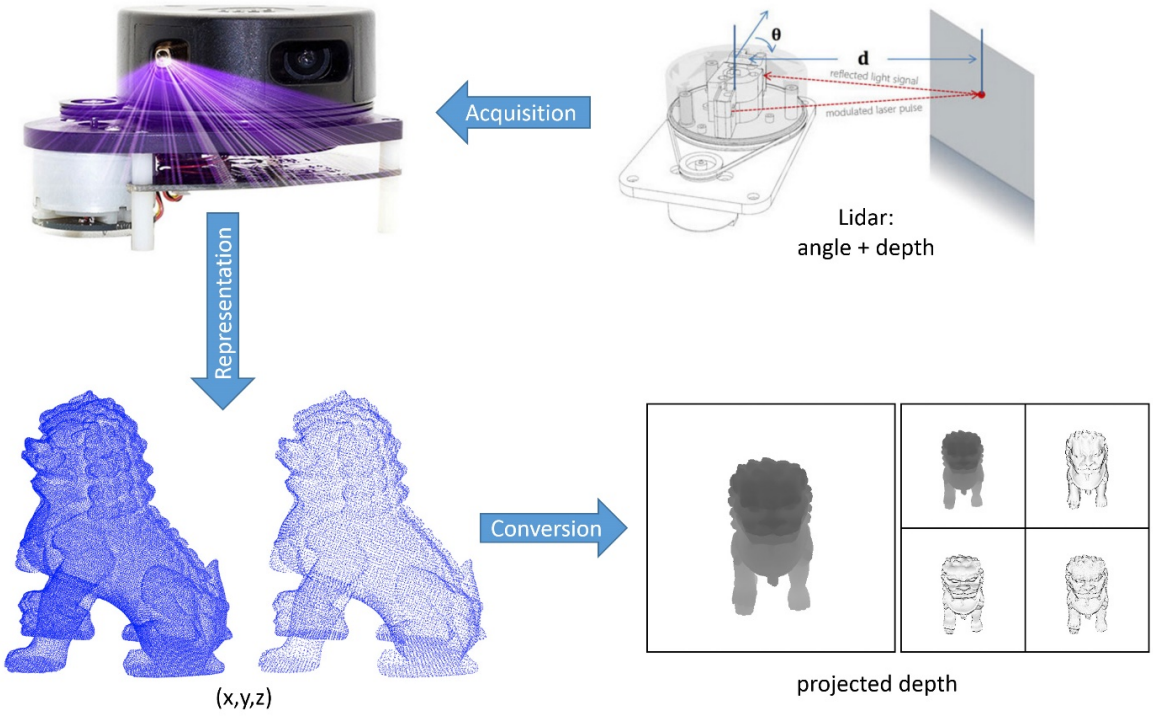


Figure 19 - Lidar acquired (angle+depth) point clouds are most often represented with (x,y,z), but might also be converted to projected depth maps. [R19]

5.1.3 Point Clouds to triangular meshes

Figure 20 shows the conversion of a point cloud into a triangulated mesh for easy rendering in a classical OpenGL pipeline. Note the difficulty in well-reconstructing occluded regions, which can be handled by the aforementioned SLAM techniques.

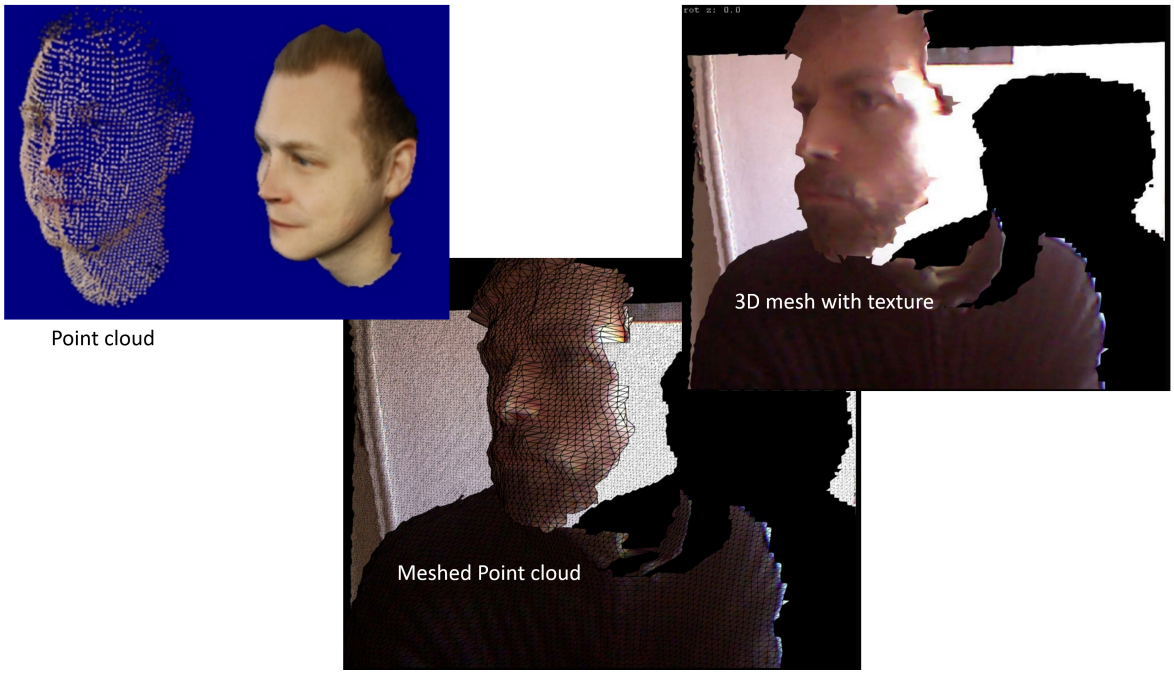


Figure 20 - Conversion of acquired point cloud into 3D meshes [R20]

5.1.4 Point Clouds to Planar primitives

In some applications, e.g. architectural Building Information Modeling (BIM), it is more advantageous to extract planar primitives representing the object than keeping the individual points of the point cloud, cf. Figure 21. Collecting many points into one patch may also yield improved coding performances in mesh-based coding, cf. section 6.

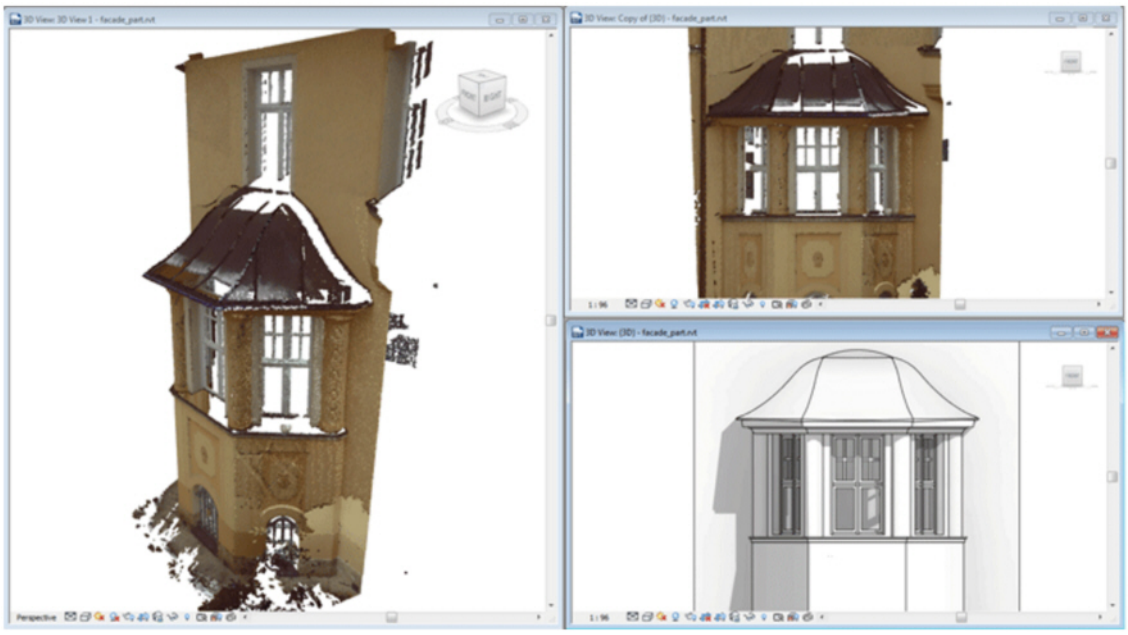


Figure 21 - Planar patch extraction (bottom-right) from point clouds (left) [R21]

For this coding purpose, the authors n Figure 22 have dissociated the background from foreground objects, stitching the background into one texture that can be used as a panoramic image, as often done in virtual reality, where only a window on such large image is visualized in the Head Mounted Device, cf. Figure 23. The remaining points of Figure 22 - or equivalently, following the reasoning of Figure 19 (bottom), the texture + depth maps of Figure 22 (top) - can then be visualized as a regular point cloud, following the rendering approaches of section 7.

Figure 22 - Hybrid point cloud and piecewise planar representation (c) from depth maps (a) and projected views (b) [R22]

Figure 23 - Panoramic texture for Virtual Reality: at any time stamp, one region of the texture is rendered in the Head Mounted Device [R23]

5.1.5 Point Clouds vs. Light Fields

In general, the points in a point cloud do not need to have a uniform color in all directions. Indeed, in nature, many phenomena are specular with a non-uniform color distribution over all directions. Such information can be represented by a so-called BRDF function (more details in section 7) telling for each point in the point cloud its color propagation in each direction. In a sense, the point emits a "field of light" all around itself, and the union of all these fields over all points creates a so-called Light Field. This clearly suggests a bijective equivalence between point clouds and light fields.

As shown in Figure 24, the point cloud/light field represents for each voxel (x, y, z) in space, its color C under a variety of light ray directions (θ, ϕ), i.e. the color C is a function of 5 parameters:

$$C=f(x, y, z, \theta, \phi) \qquad \text{(eq. 1)}$$

In practice, however, this can be reduced to 4 parameters, since the light ray remains unaltered all over its propagation (unless there would be micro-particles, like fog, that would reduce the light intensity along the propagation path). As a result, each light ray can be represented by its intersection (s, t) with a plane parallel to the camera and its propagation angles (θ, ϕ):

$$C=g(s, t, \theta, \phi) \qquad \text{(eq. 2)}$$

Light rays can also be represented by their intersections (s, t) and (u, v) with two parallel planes, see Figure 24:

$$C=h(s, t, u, v) \qquad \text{(eq. 3)}$$

The latter is the most common Light Field data representation used in literature.

In summary, instead of representing the point (x,y,z) and its color in direction (θ, ϕ), one may also represent the light ray emanating from this point by its intersections (s,t) and (u,v) with two parallel planes, one being the camera objective or the display device.

Also observe in Figure 24 the equivalence between the point cloud or light field representation on one hand and the (u,v)-texture with depth data representation from the camera viewpoint on the other hand.

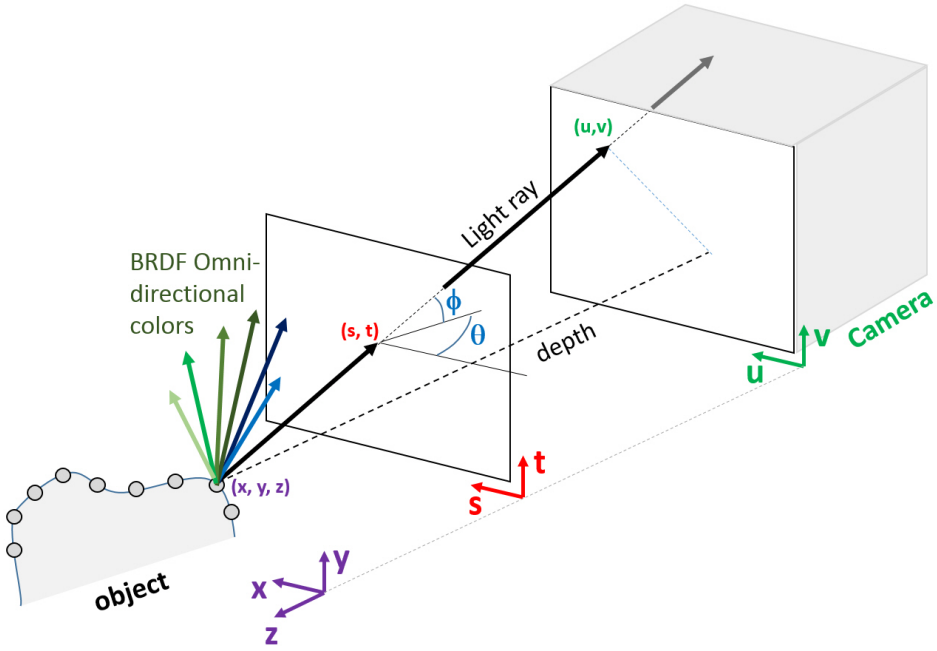


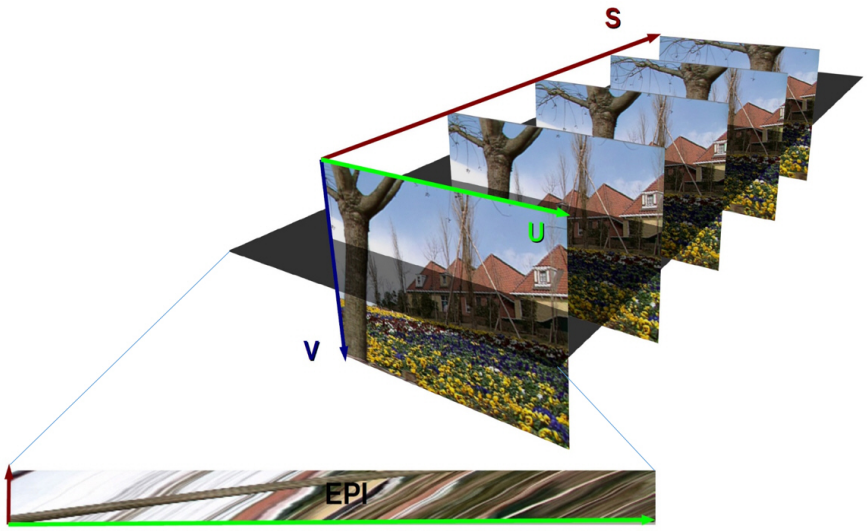
Figure 24 - Description of Light Rays emanating from the points of the Point Cloud

5.1.6 Light fields to Depth maps

As suggested in the previous subsection, depth can be recovered from light fields. Such depth information is often used in Depth Image Based Rendering (DIBR), cf. section 7.

Consider the simple case of a linear camera array in front of a scene. Stacking the images of all the camera positions one behind each other (along the u-axis) creates the image cube shown at the top of Figure 25.

One horizontal slice of such image cube corresponds to a so-called Epipolar Plane Image (EPI) with diagonal lines; see the bottom of Figure 25. Each such line corresponds to how a voxel in space moves when jumping from one camera view to the next. Voxels that are far away (large depth) will almost not move, creating almost vertical lines in the EPI, while voxels in the foreground will exhibit large disparities and hence create diagonal lines with large slopes. The slope of the diagonal lines in the EPI therefore provides information about the depth (inverse proportional to disparity) of the voxels in the scene, and proves to be useful in creating virtual views in sparse camera arrangements, cf. section 7.



Epipolar **P**lane **I**mage

Figure 25 - EPI section of an image stack.

An important challenge is to obtain a dense representation of the EPI, cf. Figure 26 (right), when only having a limited number of camera views, i.e. only a subsampled version of the EPI, cf. Figure 26 (left). Special line detection and interpolation techniques are required for this purpose.

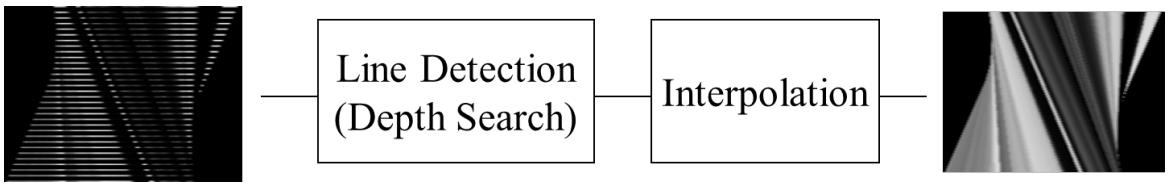


Figure 26 - EPI line slope detection (right) from a subsampled version of the EPI (left)

5.1.7 Light Field Omni-Directional Stereo

Panoramic textures as presented in Figure 23 are sufficient for 360 degree video VR with far-away objects. However, with closeby objects in reach of the user, head motion parallax causing occlusions/disocclusions (see Figure 27) cannot be well represented with panoramic textures. A technique of Omni-Directional Stereo textures (ODS) should then minimally be used (possibly augmented with depth maps), where each vertical line of the texture corresponds to light rays in a specific direction, out of which the complete light field of Figure 24 can be rendered. See section 7 for more details.

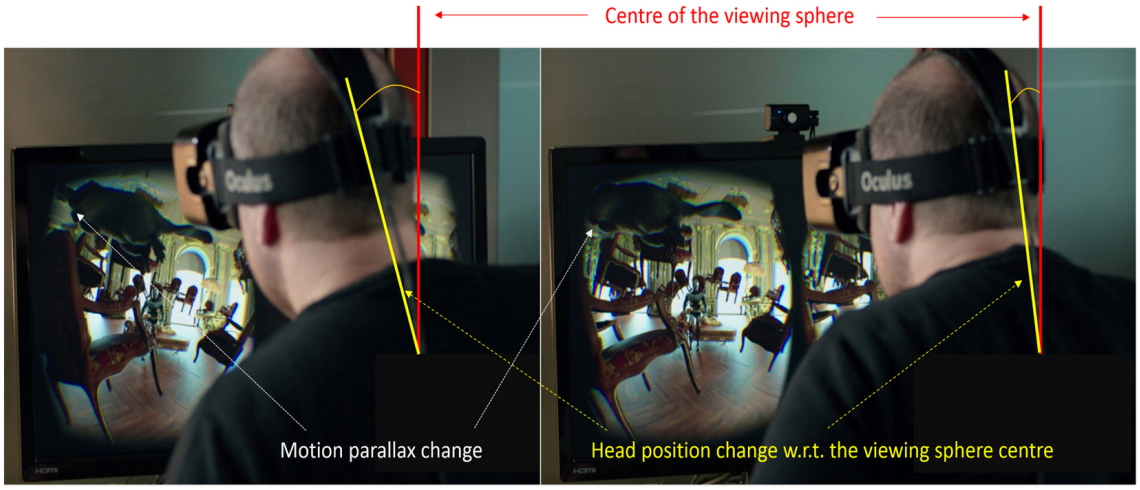


Figure 27 - Head Motion parallax in Virtual Reality calls for a specific Light Field ODS representation [R27]

6. TECHNOLOGIES SUPPORTING AUDIOVISUAL APPLICATIONS

6.1 VISUAL TECHNOLOGIES

6.1.1 SINGLE AND MULTI-VIEW TEXTURE AND DEPTH MAPS

3D scenes can be represented with a combination of camera and supplementary depth information. This format enables the generation of virtual views through depth image based rendering (DIBR) techniques (see section 7 for further details). The virtual views could be used to drive the data requirements for auto-stereoscopic multiview displays and support free viewpoint navigation. For high quality rendering, the videos are often corrected to minimize color mismatches, and for linear camera arrangements, the videos are usually geometrically corrected to reduce misalignments in camera positions and orientations.

A single camera view with depth (often referred to as the 2D plus depth format) enables virtual view rendering, but only within a limited viewing angle and with limited ability to handle occlusions and holes resulting from rendering other views. To overcome these shortcoming, a multiview video plus depth format with a limited number of views and associated per pixel depth is often used instead (see Figure 28).

Video and range data available from multiple viewing angles allows more sophisticated virtual view rendering algorithms and also provides more information for filling occlusion and/or holes when rendering novel views.

Figure 28 - Multiview video plus depth format for 2 views

Two software modules that support depth estimation and view rendering of multiview videos with depth have been developed and made publicly available by MPEG: Depth Estimation Reference Software (DERS) and View Synthesis Reference Software (VSRS).

While multiview video plus depth formats provide include useful geometric data to render virtual viewpoints, it does not include sufficient information to properly handle semi-

transparent scenes or scenes with specular reflection. Additionally, view rendering quality degrades when the input cameras have wider baselines as well as arbitrary (non-linear) positions or orientations.

6.1.2 Visual Coding

Previous section has presented data conversions between different 3D data representations, calling for different coding approaches. This section gives a high-level overview of the associated coding tools.

Multiview + depth coding (3D-HEVC)

Today, the most advanced single camera view coding standard, called HEVC (High Efficiency Video Coding) offers a data rate reduction of two orders of magnitude compared to uncompressed video.

In February 2015, ISO/IEC MPEG and ITU-T VCEG have jointly published a standard for coding multiple views, as well as their depth maps, cf. Figure 29. This codec is knows as 3D-HEVC. It enables the generation of additional views from a small number of transmitted views, supporting glasses-free/auto-stereoscopic 3D display applications with dozens of output views from only a handful of input camera feeds. These additional views are generated with Depth Image Based Rendering (DIBR) techniques, further explained in section 7.

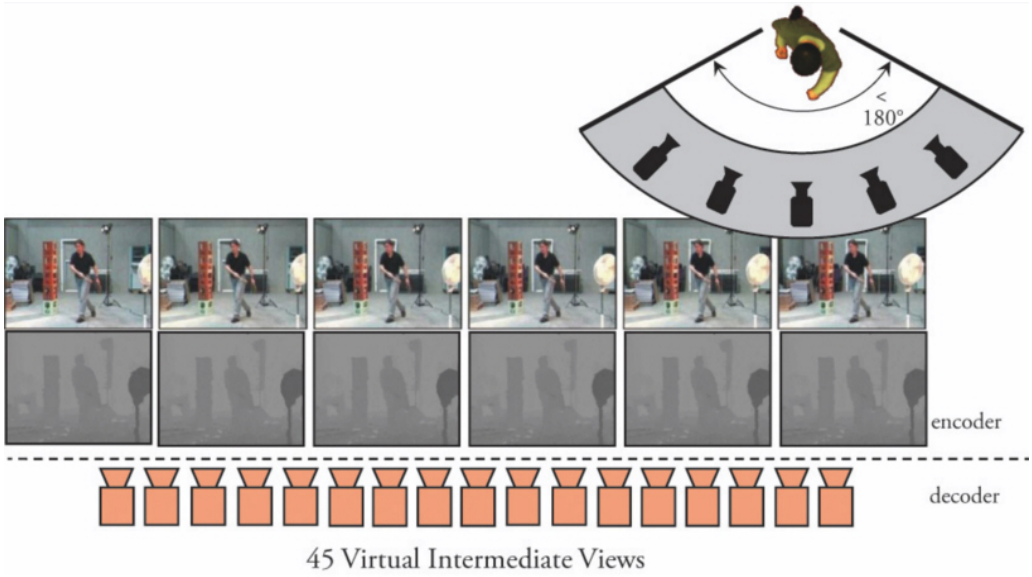


Figure 29 - Multiview + depth coding [R29]

Point Cloud Coding (PCC)

The point cloud data model is a list of points, as shown in Figure 30, and may include some point attributes like the local surface normal (nx, ny, nz). The basic structure of a point sampled surface can often well be captured by organizing the point cloud in an octree structure, see Figure 31. The octree starts from a bounding box enclosing all the points in the 3D space, and then recursively codes empty/non-empty subdivisions of this space into 8 cubes. The octree

division then only continues for non-empty children. The octree data-structure is useful to capture the basic geometric structure of a point sampled surface in an organized and compact way.

Point Cloud			(optional)	(optional)
vertex	(x, y, z)	(r,g,b)	(n_x,n_y,n_z)	
❶	(-1, 0, 0)	(1,2,3)	(1.0,0.0,0.0)	(a1,a2,a3)
❶	(0, -1, 0)	(9,1,6)	(0.0,1.0,0.0)	(a1,a2,a3)
❷	(0, 0, -1)	(9,2,4)	(0.0,1.0,0.0)	(a1,a2,a3)
❸	(0, 0, +1)	(10,1,15)	(0.0,1.0,0.0)	(a1,a2,a3)
❹	(0, +1, 0)	(4,2,19)	(1.0,0.0,0.0)	(a1,a2,a3)
❺	(+1, 0, 0)	(90,10,2)	(1.0,0.0,0.0)	(a1,a2,a3)

Figure 30 - List of points in a point cloud

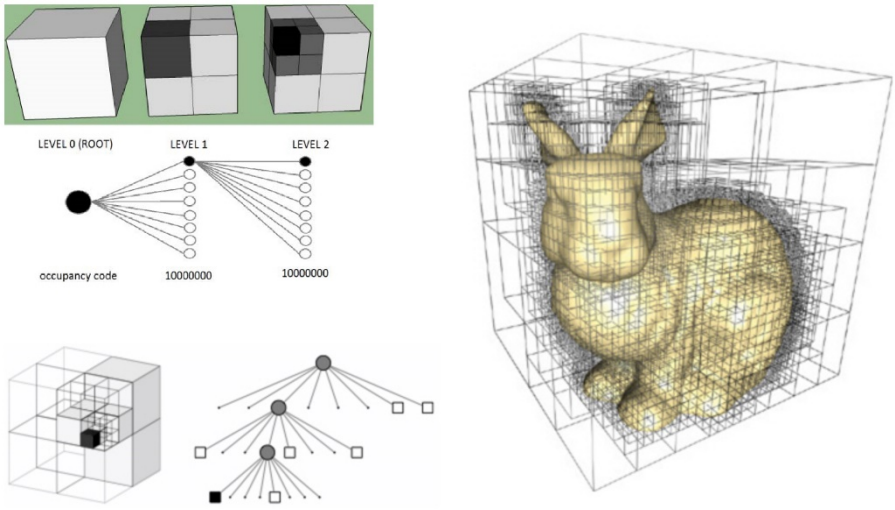
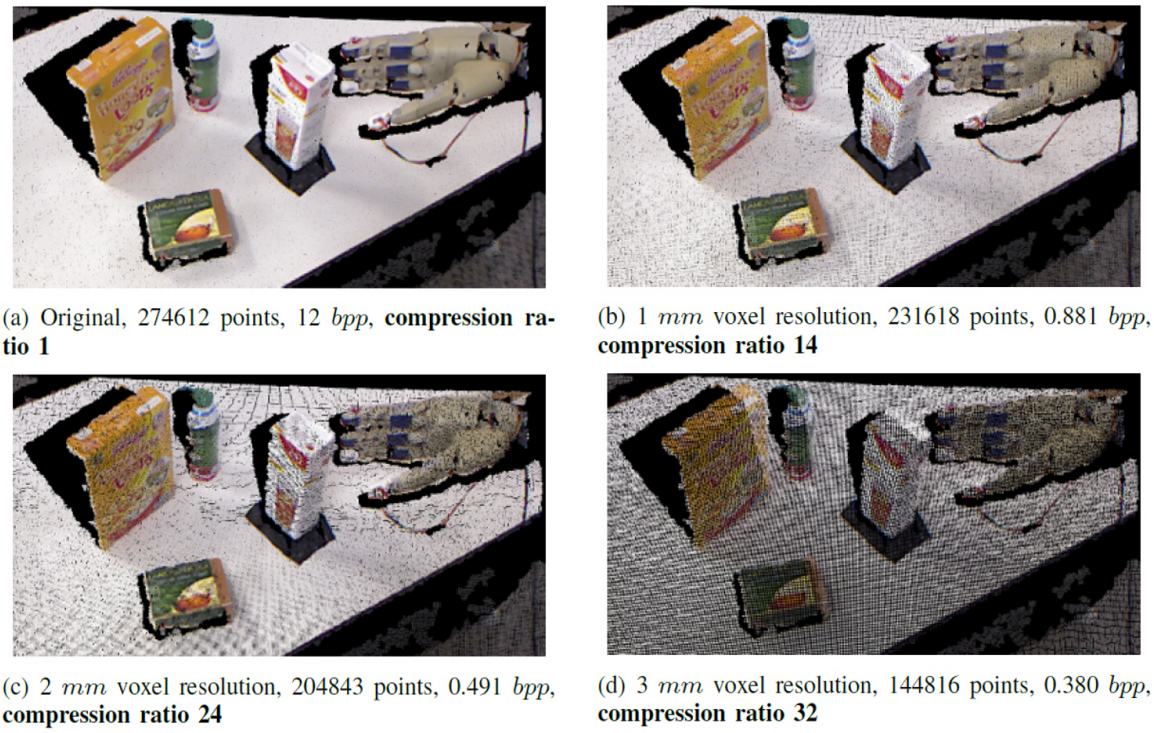


Figure 31 - Octree subdivision of the point cloud [R31]

Figure 32 shows the performances of such point cloud compression tool for scalable coding, i.e. the more information is sent, the higher the details of the decoded and rendered content.



(a) Original, 274612 points, 12 *bpp*, **compression ratio 1**

(b) 1 *mm* voxel resolution, 231618 points, 0.881 *bpp*, **compression ratio 14**

(c) 2 *mm* voxel resolution, 204843 points, 0.491 *bpp*, **compression ratio 24**

(d) 3 *mm* voxel resolution, 144816 points, 0.380 *bpp*, **compression ratio 32**

Figure 32 - Scalable point cloud compression [R32]

Finally, point clouds reconstructed from scanners can also vary in time. This is the case when sequences of point clouds are captured. In this case both the number of points and the position of the points and their attributes vary in time. This is called a time-varying point cloud.

POINT CLOUD WITH DEPTH MAP CODING

Figure 33 follows the (x,y,z) to depth conversion of Figure 19 (bottom) for compressing point clouds with a depth image representation only, achieving competitive performances compared to existing point cloud compression tools. This again shows the equivalence/relationship between point clouds and views+depth coding.

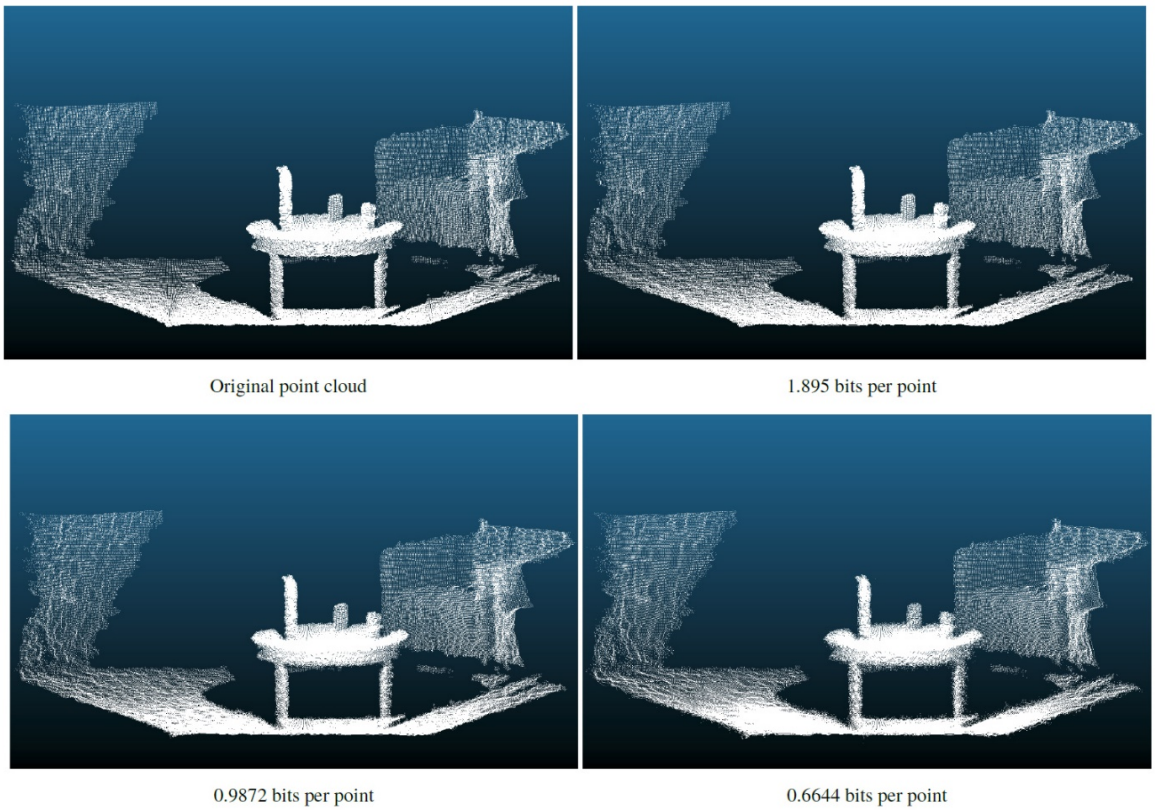


Figure 33 - Scalable Point Cloud compression performances, compressing projected depth maps instead of (x,y,z) coordinates [R19]

3D Mesh Coding (3DMC)

3D meshes add connectivity between the points of the point cloud, see Figure 34, which then become the vertices of triangular patches - the 3D mesh - describing the object's shape.

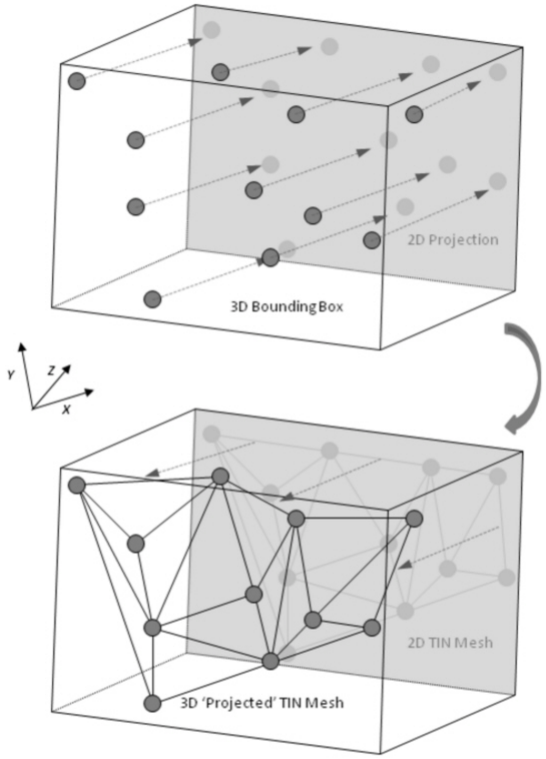
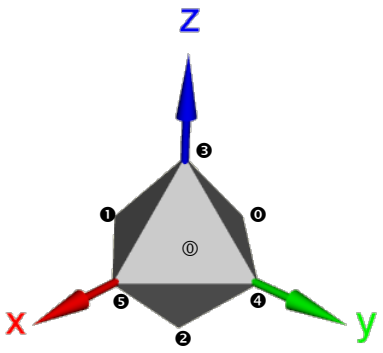


Figure 34 - Adding connectivity between the points of the point cloud to create a 3D mesh.

A 3D triangle mesh is almost invariably represented with two tables, as illustrated by Figure 35: one for geometry, in which the three coordinates (x, y, z) of each vertex are given; and another for topology (or connectivity), listing the indices (i, j, k) in the previous table of the three vertices forming each triangle. This pair of tables is commonly known as an Indexed Face Set (IFS), which is precisely the name (except for the spaces) of a VRML97 node. Given the clear distinction between the geometry information and the topology one, most static 3D triangle mesh compression techniques treat them separately.



Geometry table		**Topology table**	
vertex	(x, y, z)	**triangle**	(i, j, k)
❶	(-1, 0, 0)	⓪	(❸ ❺ ❹)
❶	(0, -1, 0)	①	(❸ ❹ ❶)
❷	(0, 0, -1)	②	(❸ ❶ ❶)
❸	(0, 0, +1)	③	(❸ ❶ ❺)
❹	(0, +1, 0)	④	(❷ ❹ ❺)
❺	(+1, 0, 0)	⑤	(❷ ❶ ❹)
		⑥	(❷ ❶ ❶)
		⑦	(❷ ❺ ❶)

Figure 35 - IFS representation of an octahedron [R35]

As already stated above, the shape is described first with a triangular mesh, and then an appearance is linked to it, by assigning attributes to its elements. Examples of such attributes are a color, hence a 3D vector in the Red, Green, Blue (RGB) color space; or a local normal vector to the surface, hence again a 3D vector, or most typically, a pair of texture coordinates, hence a 2D vector in the (u, v) texture space, pointing to a pixel (hence an RGB color) in some texture/image that is to be wrapped around the shape at rendering time through indirections. Without texture, results of Figure 36 (bottom) are typically obtained.

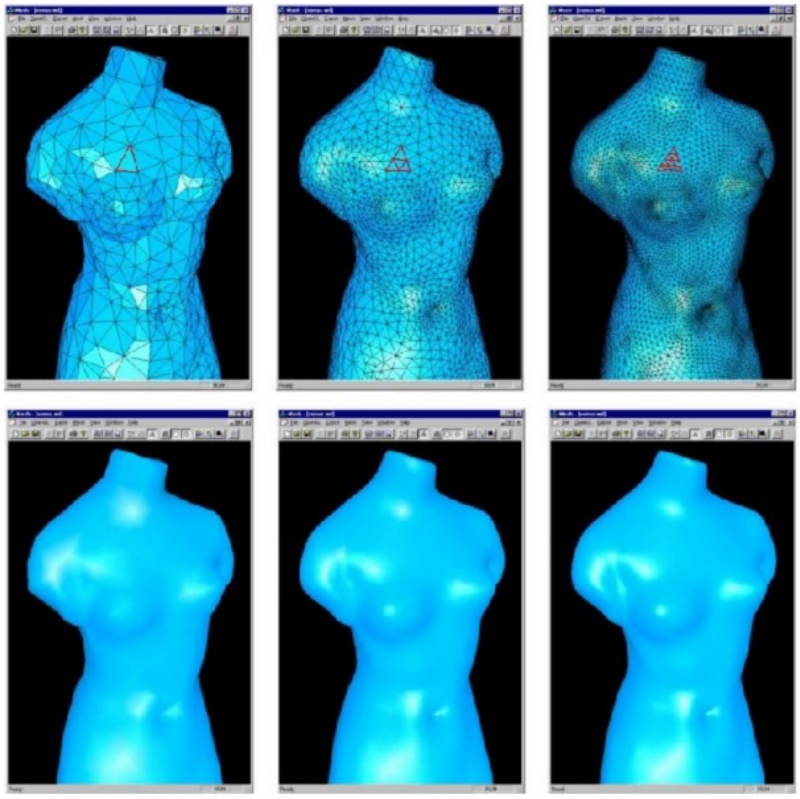


Figure 36 - 3D scalable mesh coding (top) and its OpenGL rendering (bottom) [R36]

Figure 36 shows an example of scalable meshing, where the number and size of the triangles varies with the Level-of-Detail (LoD). Observe the difficulty in keeping track of the connectivity from one LoD to the next. In practice, many meshes are first transformed into regular ones (as the example in Figure 36) before 3D mesh coding is applied, drastically reducing the interconnectivity complexity and bitrate.

OVERVIEW CODING TOOLS

Table 2 gives a tentative overview of the existing data representations and coding tools in visual multimedia applications. Point clouds, 3D meshes, panoramic/omni-directional textures, as well as light fields are considered.

The third column corresponding to the data representation format is highly linked to the rendering approach that will be explained in more details in section 7. Similarly, the differences between the different light field flavors of the last three rows are deferred to section 7.

Category	Functionality	Data representation	Coding tool	Comment
Point cloud	3D rendering, e.g. GIS	(x,y,z) + color	PCC	Under development in MPEG. Uniform color per point in all directions
3D Meshes	3D rendering, e.g. games	(x,y,z) + IFS connectivity + color/texture	MPEG-3DMC MPEG-AFX	Standardized for meshes with non-changing connectivity
Panoramic texture	360^o video VR	Large texture	Discussion in MPEG?	Stitched panorama demos exist. Do they use advanced coding tools?
Omni-Directional texture	360^o video stereo VR	ODS texture	Not available in JPEG/MPEG	Google jump uses this format. Not clear how much coding is involved?
Microlens Light Field	A posteriori refocus	(s,t,u,v) Optional: depth	Not available in JPEG/MPEG	Are there proprietary coding formats available?
Dense Camera array Light Field	Light field display, Horizontal Parallax Only	(u,v) camera + (s,t, θ,ϕ) extrinsics + depth	3D-HEVC: Multiview + depth	DIBR rendering 1D-Linear, dense camera arrangements only
Sparse Camera array Light Field	Light Field display and/or Free Navigation	(u,v) camera + (s,t, θ,ϕ) extrinsics + depth	FTV exploration/CfE with 3D-HEVC extensions	DIBR rendering Targets non-Linear (1D/2D), sparse camera arrangements

Table 2: overview of visual multimedia applications, data representations and coding tools

6.2 TECHNOLOGIES SUPPORTING AUDIO APPLICATIONS

6.2.1 VIRTUAL/AUGMENTED REALITY RECORDING AND PLAYBACK

Virtual and Augmented Reality requires that sound be rendered using a Head-Related Transfer Function (HRTF) or Binaural Room Impulse Response (BRIR) to the listener's headphones. The MPEG-H 3D Audio specification contains such rendering technology and has interfaces for

- HRTF or BRIR information as e.g. FIR filters
- Pitch, roll, yaw information

What is missing is interfaces for

- (x, y, z)
- Acceleration
- Reverberation characteristics

MPEG-H 3D Audio would require additional technology for engines that apply

- Dopler shift
- Reverberation

Such technology could be incorporated into some extensions to 3D Audio, or appropriate interfaces could be specified such that 3D Audio can use new MPEG technology or even external technology to perform e.g. Dopler shift and reverberation.

6.2.2 SOUND FIELD RECORDING

The overriding aspect of sound field recording and playback is that recording and playback must support information streams from and to a very large number of sensors. Furthermore, for many frequency ranges and sound object positions, the sensors could be expected to produce information streams with very high redundancy.

Reduction of redundancy is an opportunity to successfully address the recording problem. For example, use of techniques for purely statistical redundancy reduction may realize significant compression. Exploiting aspects of human perception to achieve reduction of irrelevancy has been shown to be even more successful than reduction of redundancy in achieving signal compression. It is expected that reduction in both redundancy and irrelevancy could be applied to compress sound field information streams.

A list of possible technologies for compressing sound field data steams follows:

- Lossless compression
  - MPEG-4 Audio Lossless coding (ALS)
  - MPEG-4 Scalable Lossless coding (SLS), particularly the “non-core” mode.
  - New lossless compression streams expressedly developed for sound field representation, perhaps with specific sensor geometries.
- Lossy compression
  - MPEG-4 Advanced Audio Coding
  - MPEG-D Unified Speech and Audio Coding

6.3 AUDIO/VISUAL CONTEXT PROCESSING

6.3.1 VISUAL PROCESSING TO ASSIST AUDIO PROCESSING

In rendering audio for Augmented Reality, the audio engine typically has no apriori information about the reverberant properties of some arbitrary real-world space that the user may currently occupy. The audio rendering engine can position virtual sound objects in space relative to the user’s head orientation, e.g. using anechoic HRTFs, but would be unable to add realistic reverberation as the reverberant properties of the current user environment are unknown.

One solution would be for a head-mounted camera to sense properties of the user location (e.g. I am at an estimated location and orientation in a room of an estimated size with estimated reflectivity properties of any floors, walls and ceilings). This would involve joint audio/visual processing and metadata sharing.

7. RENDERING

7.1 VISUAL RENDERING

The rendering engine has a tremendous impact on which essential information to convey for a rich user experience. For instance, though the man in the street takes for granted that 2D photography only projects the acquired pixels "as is" on the display, in reality, many different processing steps occur behind the scenes: color calibration and gamma correction processing produce different color rendering and emotional experiences to the user.

In Immersive Media, the situation is even more critical: 3D spatial information is typically rendered from other perspectives than the ones under which the data was originally acquired, which largely impacts the rendering post-processing and the eventual user experience. In particular, acquiring a scene from a limited number of camera viewpoints and rendering any in-between, virtual viewpoint on a head mounted device, a mono or stereo display and/or a 3D light field display, will need effective perspective re-projection and view interpolation techniques, using multi-directional color information from the points in 3D space.

It is of uttermost importance to make an inventory of the core information that needs to be transmitted as a function of the subsequent image processing and rendering technologies for reaching a high quality viewing experience. This section therefore gives an overview of the rendering technologies used in existing 3D data representation modalities.

7.1.1 OMNI-DIRECTIONAL COLOR INFORMATION

All the 3D visual application technologies of previous sections share in common that for points in 3D space, color information is provided for many different viewing directions. We here repeat the essential characteristics to guide the reader through the following subsections:

- In the light field representation, a point in 3D space emits light rays in different directions, and these light rays are most often described by their intersection coordinates with two parallel planes.
- In point clouds, it's the point position and its color, rather than the emanating light rays, which are stored.
- In 3D graphics, the omni-directional color information is described through the Bidirectional Reflectance Distribution Function (BRDF), i.e. a function that tells in each point of interest how the light behaves in propagating the underlying color in different viewing directions.

Historically, however, many different applications have led to different rendering approaches of these - rather similar - data representations, resulting in varying user experiences, as will be explained in following subsections.

Moreover, this multi-directional color information is often acquired from a sparse subset of all available light directions, which inevitably leads to "holes" when rendering from viewpoints different from the acquisition direction, cf. Figure 37. The process of "hole filling" is an essential part in the rendering pipeline, and heavily impacts the user's Quality of Experience (QoE).

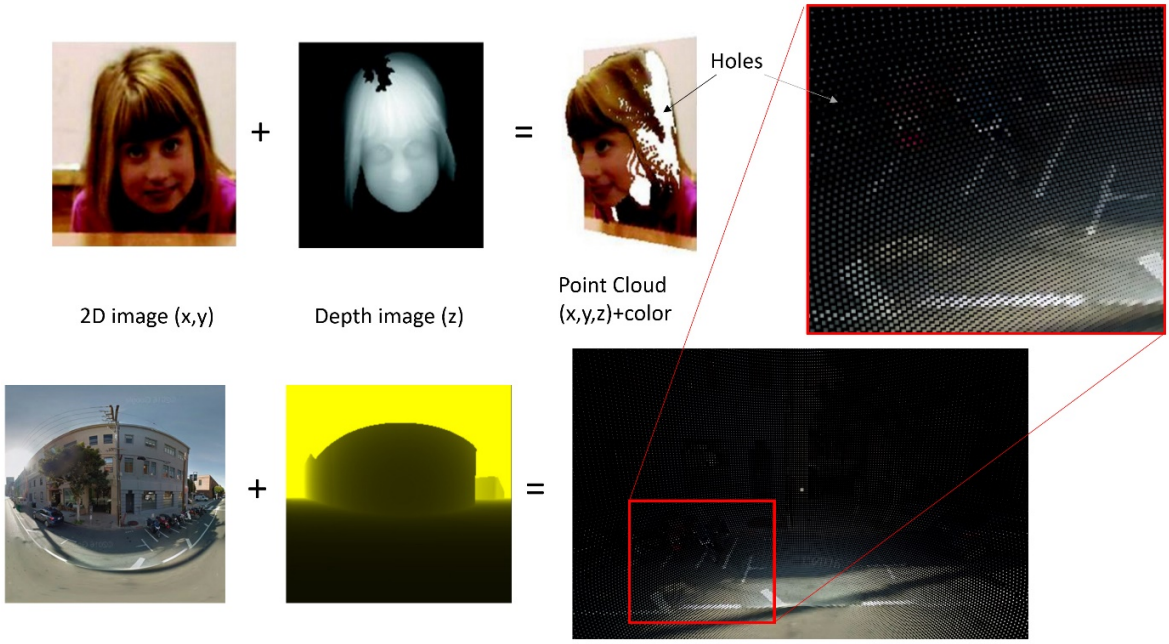


Figure 37 - Colored 3D points from a 2D image (x,y) and Depth image (z)

7.1.2 Point Clouds

A point cloud is a collection of points in 3D space, each holding color information. These points are typically acquired with time-of-flight scanners that collect the depth and color of the points in all directions around the central acquisition position.

The (x,y,z) coordinates of each point can be thought of as a 2D pixel of a photograph that is pushed into the depth z, cf. Figure 37. In remote sensing, the data is arranged slightly differently, where the 2D photograph corresponds to a terrain picture and the z-coordinate then represents the elevation (vertically in height) rather than the depth (horizontally).

To fill the holes in the rendering of point clouds, one typically uses so-called “splatting”, see Figure 38, which extends each point with a rectangular or circular shape, either frontal to the viewing direction or oriented with the surface normal. The overlaps between adjacent splats then hides the holes away, see Figure 39 and Figure 40.

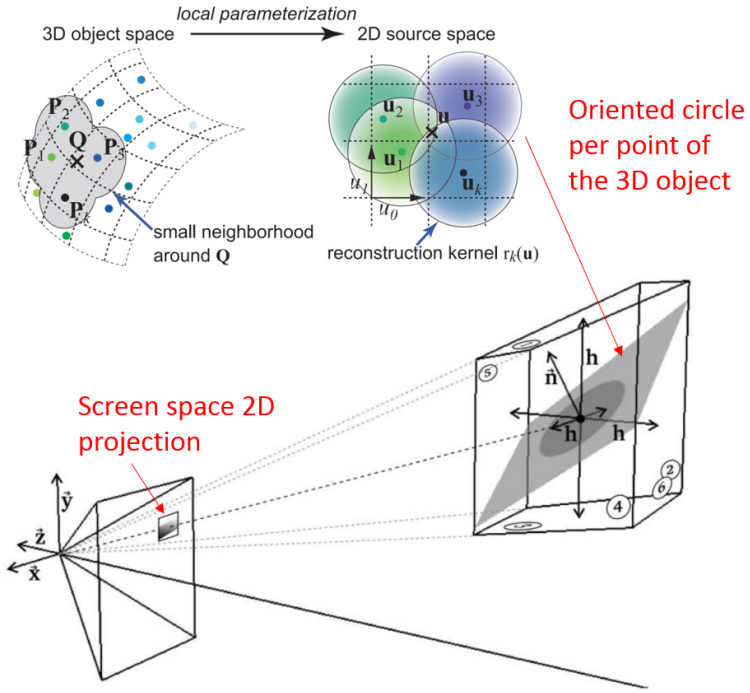


Figure 38 - Splat rendering with oriented circles per point [R38]

Figure 39 - Point Cloud splatting [R39]

Figure 40 - Point Cloud splatting [R39]

The quality of the rendering heavily varies with the parameter settings, but in general - except for very high point cloud densities – an artistic (but not always wanted) "pointillism effect" is obtained, see Figure 40. The advantage, however, of this rendering technique is that it can typically run in real-time in low-cost rendering engines.

Highly dense point clouds often originate from direct fusion of 3D scans from Lidar or RGB + Depth cameras, using Simultaneous Localization and Mapping (SLAM), cf. section 5. View dependent color differences are often aggregated during such reconstruction process to compute a single color per vertex attribute, representing the observed intrinsic object property.

In addition, point clouds can be assigned material properties that determine the view-dependent diffraction/diffusion/reflection characteristics. These properties are used by the 3D renderer for its ambient and global illumination rendering. In such scenario, the direction selective color differences are re-computed by the renderer creating a naturalistic impression. Such representations are, however, more common to represent and capture light fields (see later) than 3D objects.

7.1.3 3D MESHES

3D meshes add connectivity between the points of the point cloud, which then become the vertices of triangular patches - the 3D mesh - describing the object's shape, see Figure 41. The color of a point in a triangular patch is interpolated from its vertices and a patch texture. The latter automatically "fills the holes", playing the role of aforementioned point cloud splats, while providing a pleasant rendering.

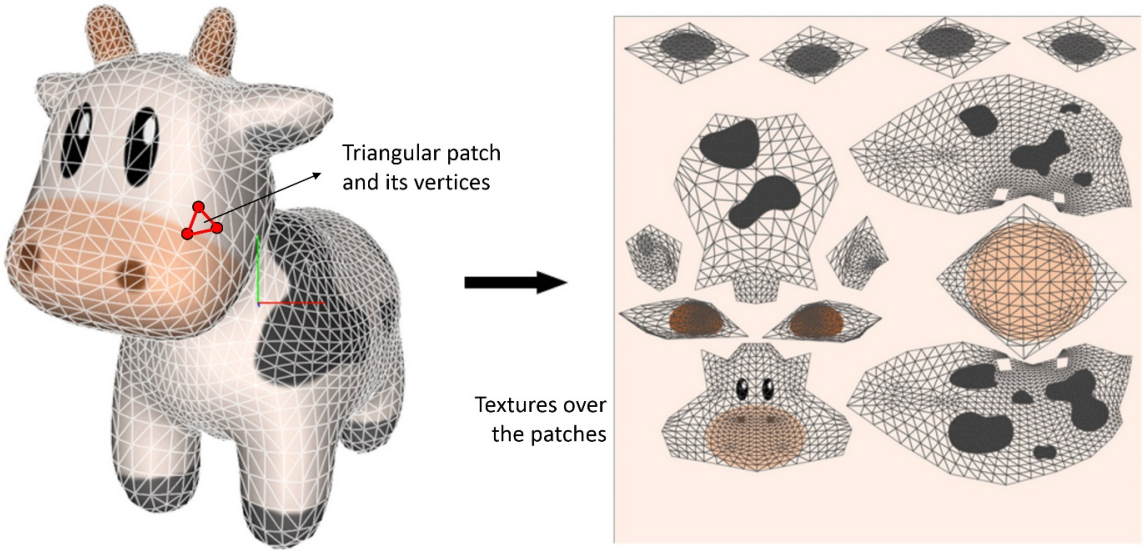


Figure 41 - 3D mesh representation with textures in 3D graphics rendering

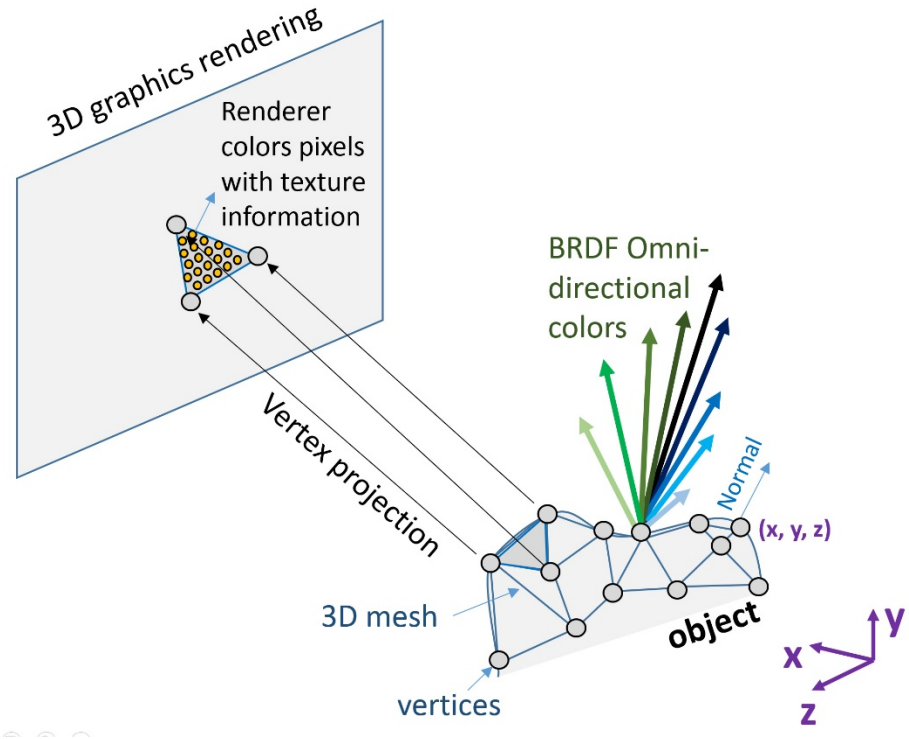


Figure 42 - 3D graphics rendering process: project triangles and fill them with texture pixels

Each vertex of Figure 42 also holds additional attributes, like color, normal, etc. The normal vectors are typically used for lighting calculations that heavily rely on the angle between the surface normal and the viewing direction. A higher level of detail can be obtained with a so-called Bidirectional Reflectance Distribution Function (BRDF) describing the light reflection characteristics in any direction, see Figure 43. The rendering of such content in 3D graphics

pipelines (e.g. Graphical Processing Units, which are massively parallel) can reach, in (almost) real-time, a “quite naturally looking” quality level on desktop computers, though many details (e.g. too sharp object contours) will reveal the “synthetic essence” of the content, cf. Figure 44.

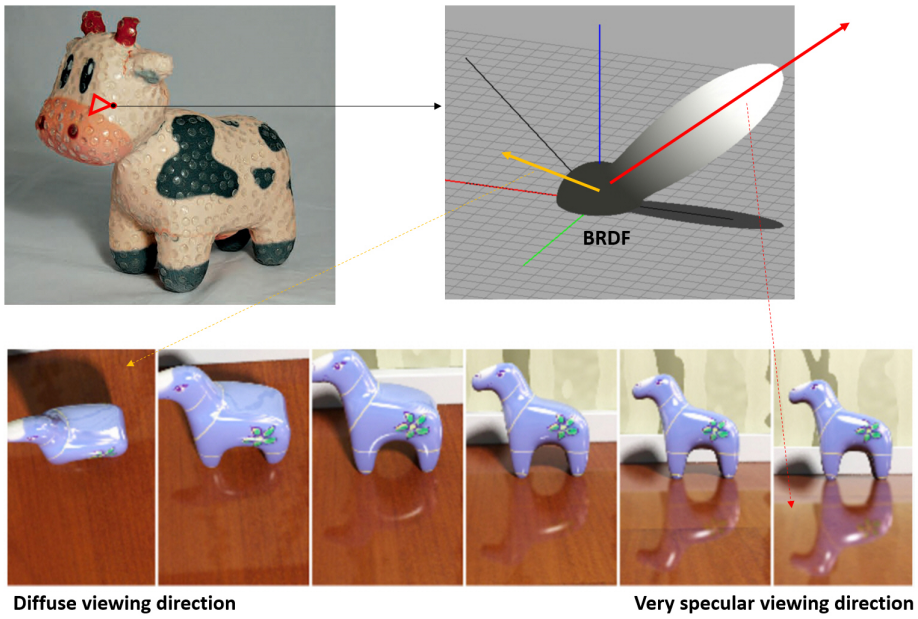


Figure 43 - BRDF directional light characteristics [R43]

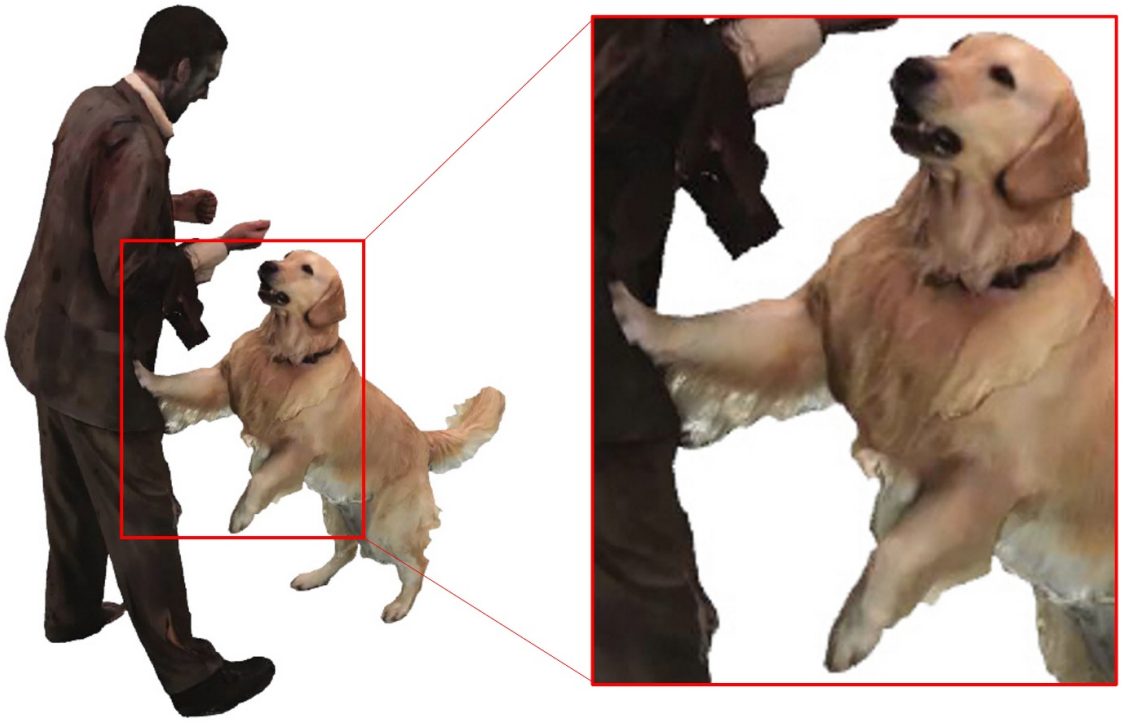


Figure 44 - BDRF assisted 3D mesh rendering, using pre-acquired 2D textures [R44]

It's worth noticing that since a 3D gaming experience relies more on the action in the game than on its intrinsic picture quality, many users accept 3D gaming products based on 3D mesh-based data representations, even though the lighting calculations (cf. BRDF) are often over-simplified to allow real-time rendering.

In contrast, photo-realistic rendering in movies (e.g. the Na'vis in the Avatar movie) relying on all the tiny details of BRDF calculations with raytracing techniques using multi-bouncing light reflections, see Figure 45, are – still today – far from real-time, even on a farm of computers.

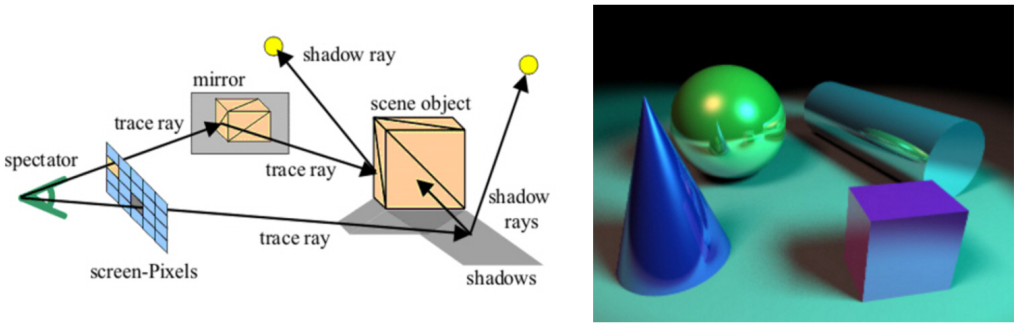


Figure 45 - Multi-bouncing ray tracing correctly render reflective surfaces and shadows

7.1.4 Panoramic Textures

Panoramic Virtual Reality (VR) engines often mimic the projection of points around the user's central position by using a cylindrical/spherical texture, cf. Figure 46, which effectively replaces the projected point cloud of Figure 39 without distracting splatting artefacts. For each viewing direction, a portion of the panoramic texture is extracted for viewing.

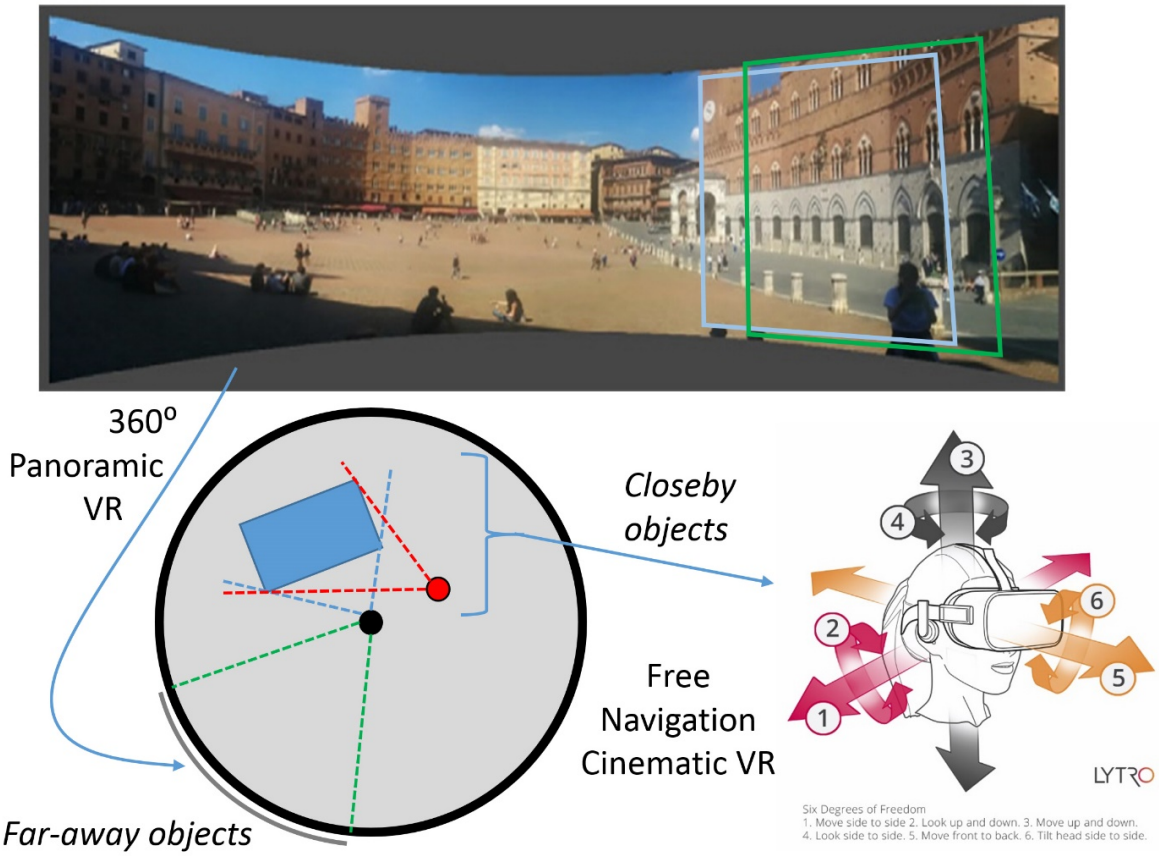


Figure 46 - Panoramic textures for Virtual Reality

7.1.5 Omni-Directional Stereo textures

For stereoscopic viewing, two such cylindrical panoramic textures of Figure 46 should be foreseen, unless one uses Omni-Directional Stereo (ODS) techniques (e.g. google jump) where the eye viewing rays are extracted from the corresponding rays of a single cylindrical texture, corresponding to all possible eye positions/orientations along a circle, see Figure 47. This allows stereoscopic viewing with a single texture, but the user should preferably have his head positioned in the middle of the viewing cylinder for best viewing experience.

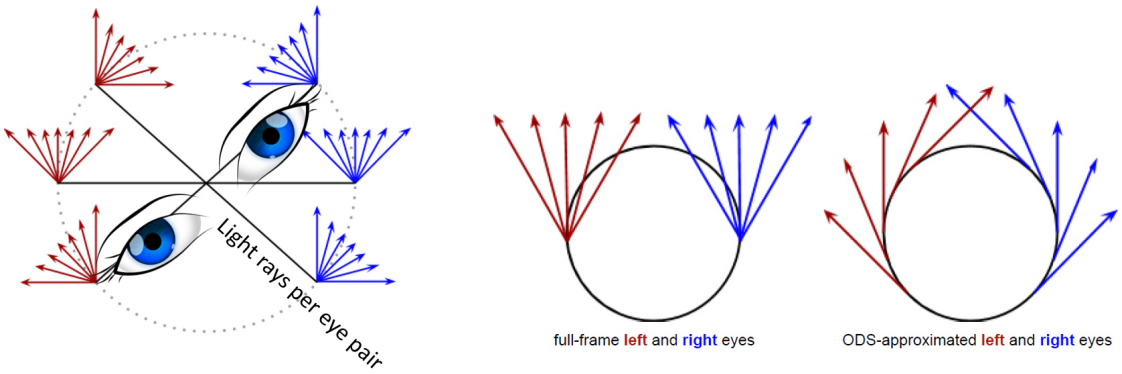


Figure 47 - Omni-Directional Stereo panoramic texture [R47a][R47b]

This technique of extracting the light rays from a collection of already captured light rays will be revisited later. In particular, to support Free viewpoint Navigation (FN) away from the central user position (i.e. motion parallax from the black to the red point in Figure 46), one should interpolate the light rays, additionally extracting a depth map (equivalent to the z-coordinate of the corresponding points) to create any virtual view with DIBR techniques, as will be explained in the Light Field subsections below.

7.2 MICROLENS LIGHT FIELDS

The multi-directional color information of a point in the scene is captured over different pixels on the CMOS sensor, through a microlens array. In effect, adjacent viewpoints of the scene are acquired, supporting small perspective changes when rendering the scene from each microlens viewpoint, referred to as the Elemental Images (EI), cf. Figure 48.

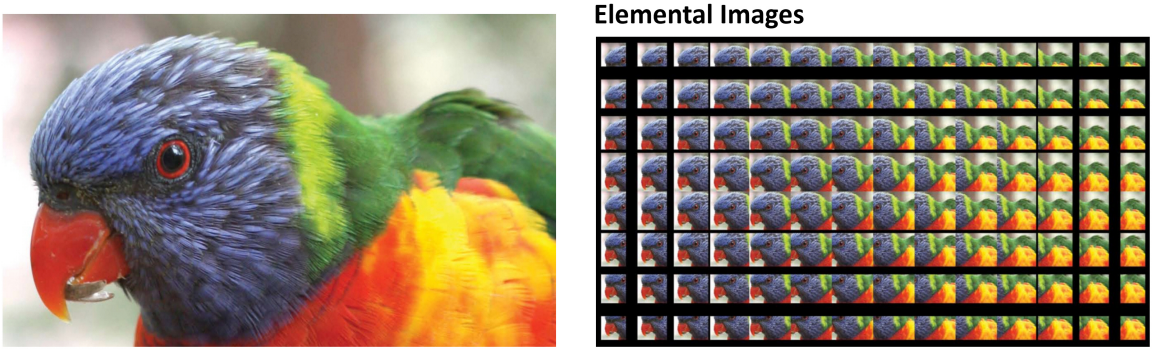


Figure 48 - Light Field Elemental Images

These Elemental Images represent the scene from a discrete set of neighboring perspective viewing directions within the camera's Field of View. All intermediate viewpoints can be synthesized with small baseline synthesis techniques, either using Depth Image Based Rendering (DIBR) methods as described in the next subsection, or by extracting from a "collection of light rays" - i.e. from the light rays that propagate in all directions from the points in 3D space – the ones that correspond to the camera viewpoint, cf. Figure 49 (left). To create a high resolution output image, light rays are then interpolated following the lumigraph approach in Figure 49

(right), i.e. the interpolation is depth assisted. This calls for estimating the depth of the objects from the Elemental Images.

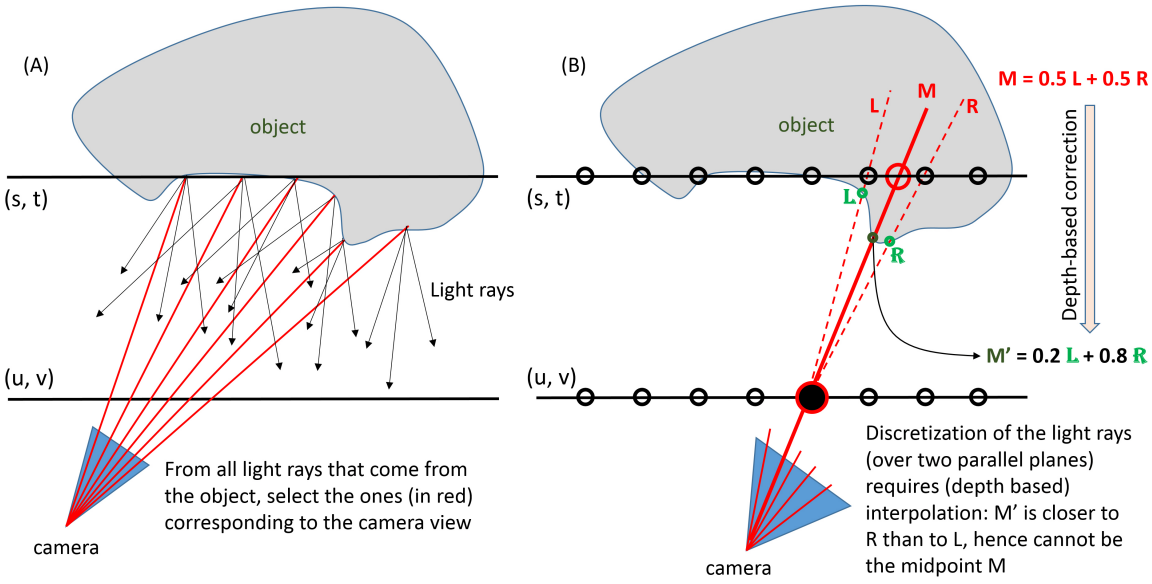


Figure 49 - (A) Extracting a virtual views from the light rays of the Light Field, (B) Depth-assisted interpolation of light rays [R49a][R49b]

Noteworthy is that such light field systems with their depth maps also allow another interesting image post-processing result: the refocus over different depth ranges in the image from a single acquired snapshot, see Figure 50 (left), hence sustaining an "on-axis" free viewpoint generation, besides of the aforementioned off-axis Free viewpoint Navigation (FN) corresponding to the perspective changes.

Figure 50 - Light Field Rendering (refocusing on the left and slightly changing viewpoint on the right) [R50]

The basic principle of such refocus operation is given in Figure 51: focusing on a point Q at focal plane F will integrate all light rays reaching Q, while focusing on a point Q' at focal plane F' will integrate different light rays reaching Q'.

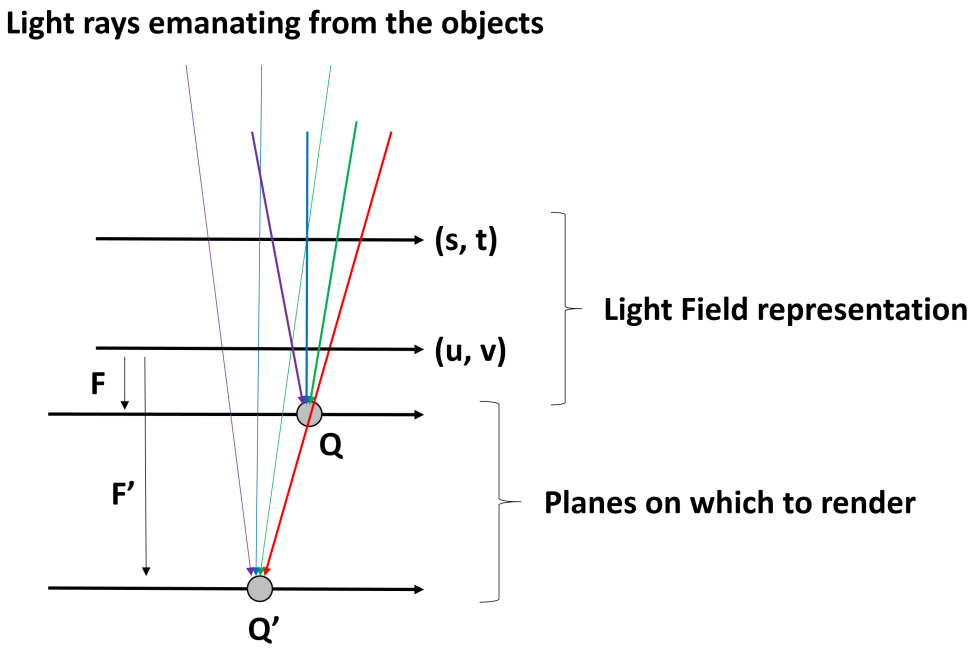


Figure 51 - Light ray intersecting two focal planes (F and F') [R51a][R51b]

In all these light field cases, some processing is performed on the acquired light rays (color and direction). Since the views are all rendered from the captured images (i.e. Image Based Rendering – IBR) without prior 3D modeling like 3D meshes, the final rendering looks very natural. A disadvantage, however, is that the spatial resolution is sacrificed for capturing many different viewpoints (consequently increasing the angular/directional information/resolution), creating output images having typically an order of magnitude lower spatial resolution than the CMOS sensor itself. This prevents its direct use in broadcasting products targeting 4k UHD display resolutions, since this would require sensors with impractically high and/or economically unviable resolutions.

7.3 CAMERA ARRAY LIGHT FIELDS

Camera array light fields differ from the microlens light field cameras in two respects: (i) they cover a larger space of the scene, hence allowing larger perspective changes, and (ii) they have a higher spatial resolution per viewpoint, providing higher output resolution images.

Such setups have been used for the "bullet time effect" in the movie "The Matrix", with hundreds of closely arranged cameras around the scene, mainly in a 1D arrangement, sometimes also in a 2D camera arrangement, cf. Figure 52.

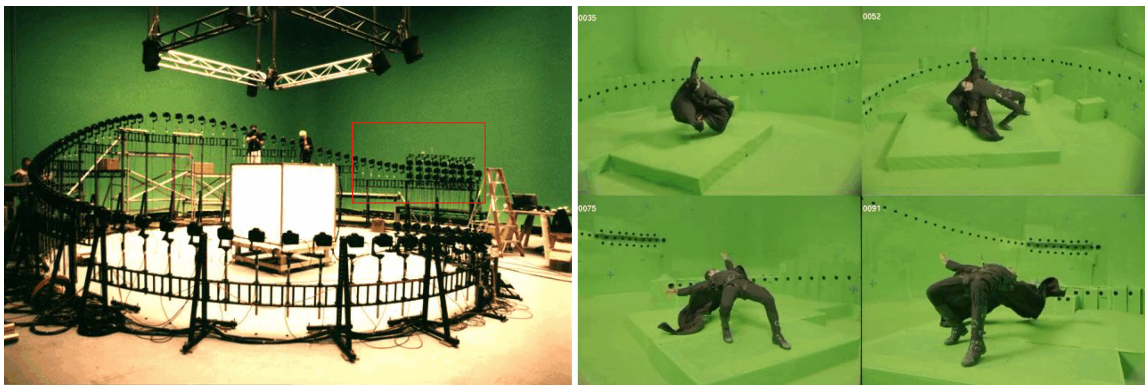


Figure 52 - The Matrix multi-camera setup (mainly 1D non-linear, partially 2D cf. red box) [R5a][R5b]

The relative position of the cameras in such arrangements are given by the so-called extrinsic parameters, describing the relative translation and rotation between cameras. This representation is very similar to eq. (2) in section 5.1.5 with (s, t) being the position of the camera and (θ, ϕ) its rotation in a global coordinate system. For Depth Image Based Rendering (DIBR) purposes (see next subsections), this information is often augmented with the depth d, out of which the (x, y, z) coordinates of eq. (1) can be recovered for DIBR based virtual view synthesis.

7.3.1 Dense camera array Light Field

Consider the simple case of a linear camera array with thousands of camera positions obtained by sliding a camera mm per mm over a 1m rail in front of a static scene. Stacking the images of all these camera positions one behind each other (along the u-axis) creates the image cube of Figure 53. Any section of such cube creates one view to the scene, as presented at the right side of Figure 53.

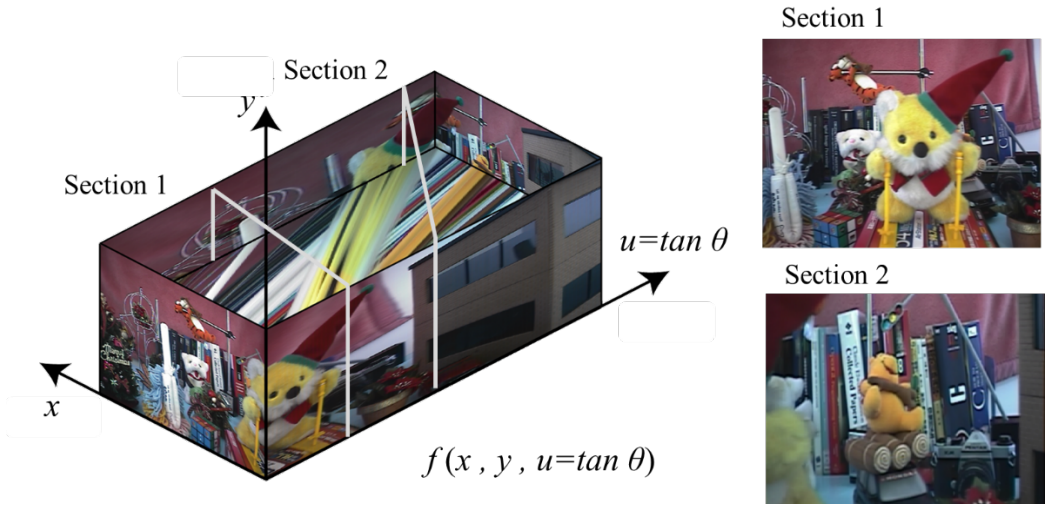


Figure 53 - Image cube for a dense linear camera array

With a spherical, dense camera array, any virtual viewpoint can be extracted from a sinusoidal section of the image cube, see Figure 54. If the camera array is sparse, however, the image stack will be discrete and interpolation will be needed to create any virtual viewpoint, as explained in section 5.1.6.

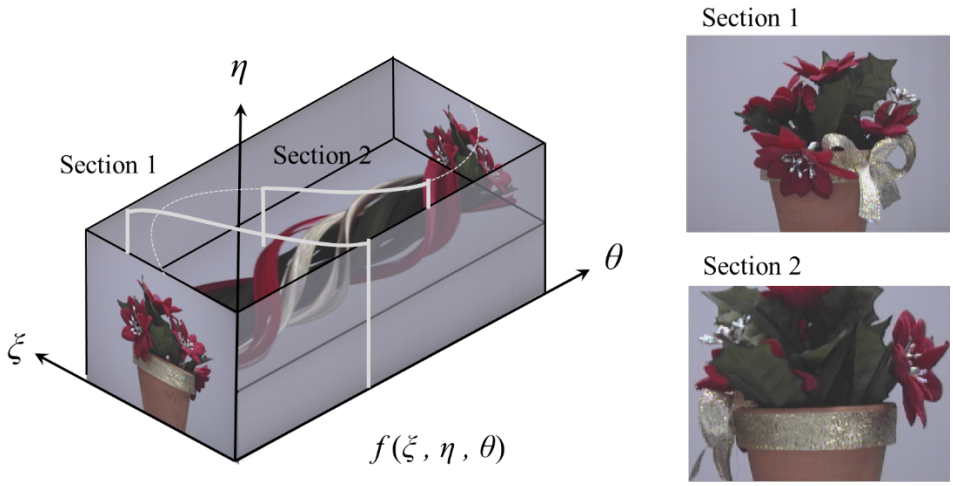


Figure 54 - Image stack (left) and virtual view generation (right) with an arbitrary, dense camera arrangement

7.3.2 Sparse camera array Light Field

The rendering of virtual viewpoints in-between the physical camera positions in a relatively sparse camera array is obtained through Depth Image Based Rendering (DIBR), displacing pixels and objects from a reference view as a function of their disparity: foreground objects move more drastically than background objects when switching from one camera view to the next, and this "disparity effect" has also to be replicated in the synthesis of virtual viewpoints. The disparity itself (which is actually the inverse of depth) is often estimated through pairwise stereo matching techniques, or through EPI techniques as described in previous subsection.

As with point clouds, "holes" appear around disocclusions from some views, which are filled from corresponding pixels re-projected from other views, cf. Figure 55 and Figure 56. The remaining "cracks" can be further filled with pixel interpolation and inpainting techniques.

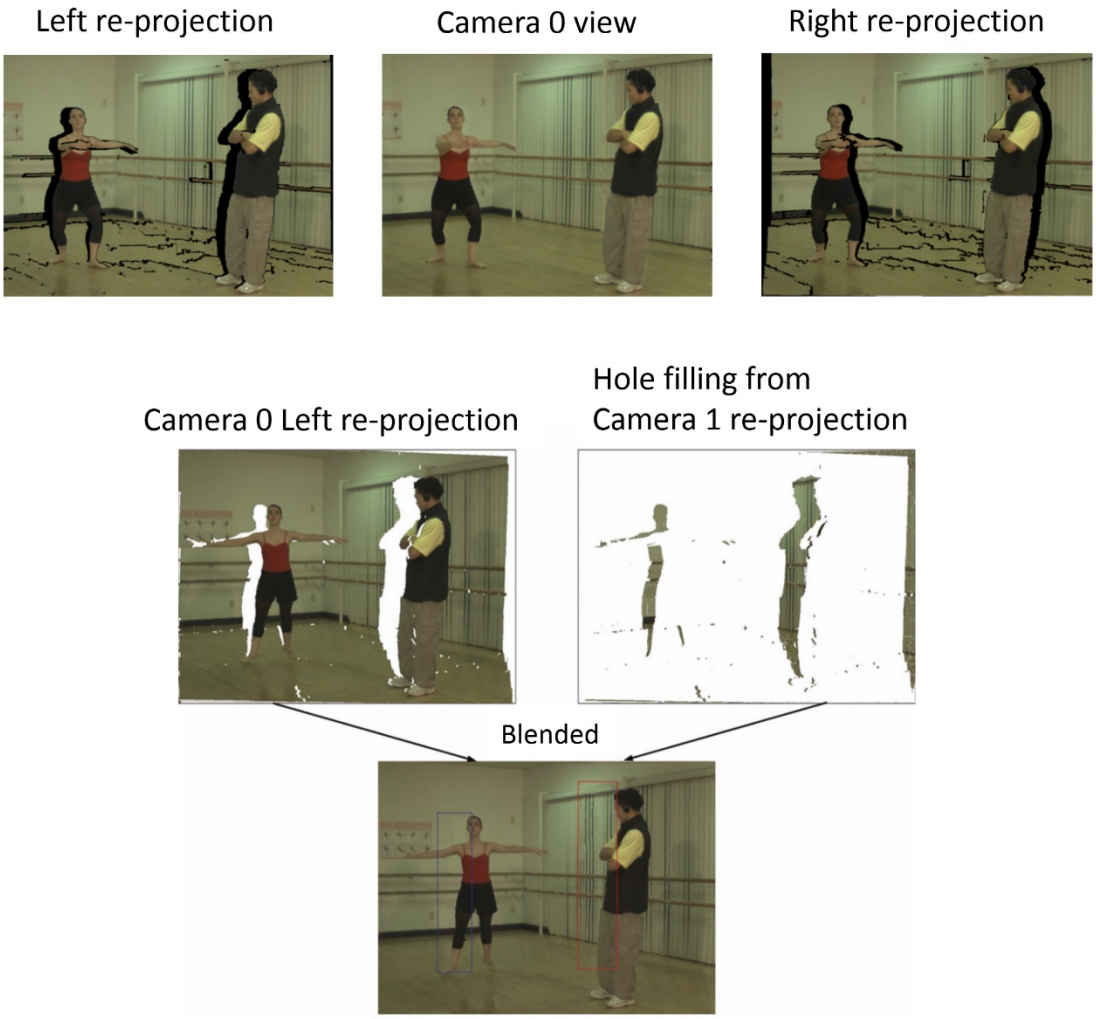


Figure 55 - Disocclusions with re-projections of camera views to virtual views [R55]

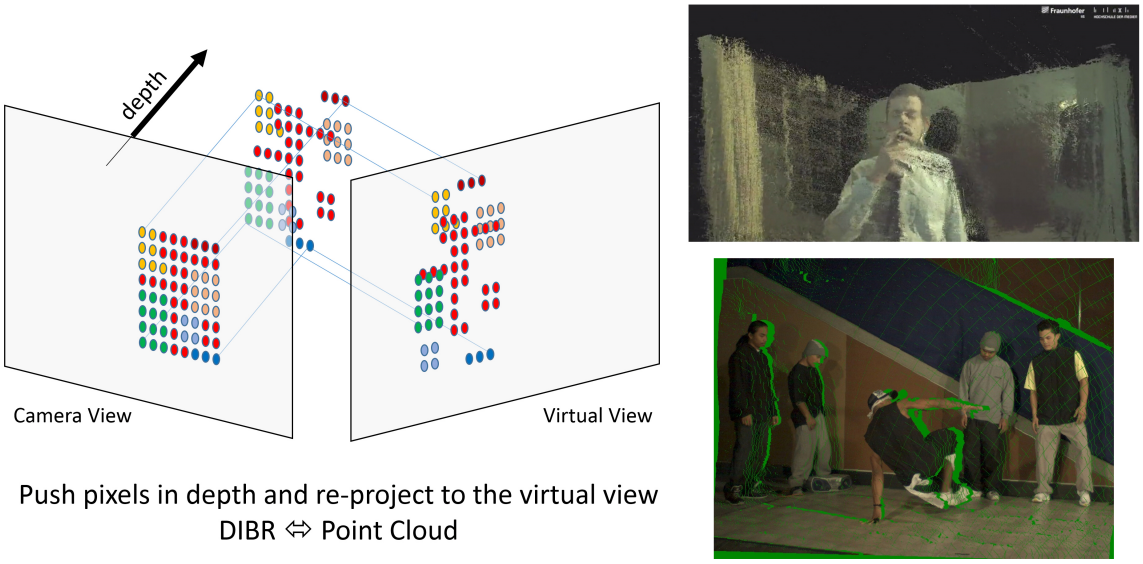


Figure 56 - Close-up of the re-projection of a camera view to a virtual view [R56]

If depth estimation and view synthesis work well in the described DIBR framework (and this heavily depends on the density of the camera setup, and the complexity of the depth estimation and view synthesis), any viewpoint within the camera-enclosed volume can eventually be synthesized. This creates a Free viewpoint Navigation (FN) functionality, much appreciated by Cinematic Virtual Reality (CVR) producers, who provide the VR functionality within the real world, using 3D naturally acquired content rather than 3D synthetic content.

8. PRESENTATION

8.1 PROJECTOR ARRAY LIGHT FIELDS AND INTEGRAL PHOTOGRAPHY

8.1.1 LIGHT FIELD DISPLAYS

Light Fields are not only acquired, but can also be projected in free space, corresponding to the counterpart of the light field cameras. The user gets then embedded into "a field of light", experiencing an immersive feeling, without the need of wearing stereoscopic glasses.

Such light field displays project Elemental Images (EI) in many preferential directions such that the viewer's eyes always capture a stereoscopic pair subset of the projected images, whatever the viewer's position in front of the display (within a working space), cf. Figure 57. The viewer hence experiences correct motion parallax (i.e. a correct change of viewing perspective when moving in front of the rendered scene), as well as a stereoscopic feeling without having to wear 3D glasses.

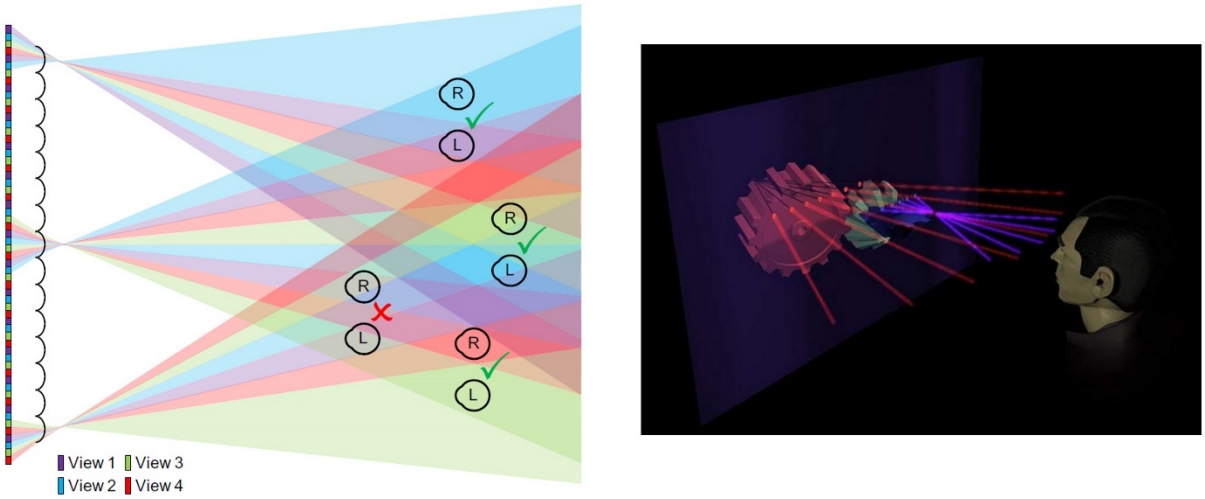


Figure 57 - Light Field display working principle

When both display and camera utilize the same optical arrangement, the light rays captured by a light field camera can be reconstructed by the light field display, but in reversed direction, as indicated in Figure 58. In this case, the displayed 3D object formed by the displayed ray intersections will be identical to the captured 3D object. However, reversing the light ray direction generates object with reversed depth, also known as pseudoscopic objects. In order to present 3D objects with the correct depth, a pseudoscopic-to-orthoscopic conversion is required, which can be achieved either optically or by digital signal processing. In practice, it means that there is always a need to do some signal processing on light fields before they are actually rendered.

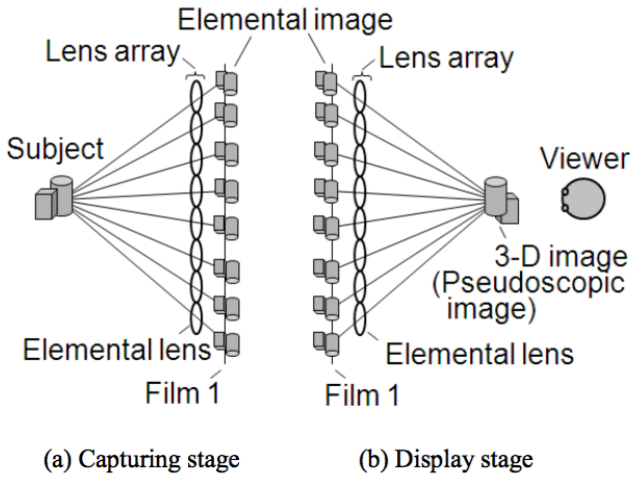


Figure 58 - Perfect symmetry between Light Field capture and rendering [R58]

LIGHT FIELD RESAMPLING

If the capture of Light Fields has not the same characteristics as the display produces, the light field sampled by the acquisition system does not fall exactly at the sampling positions of the light field display, and a resampling operation is required, see Figure 59.

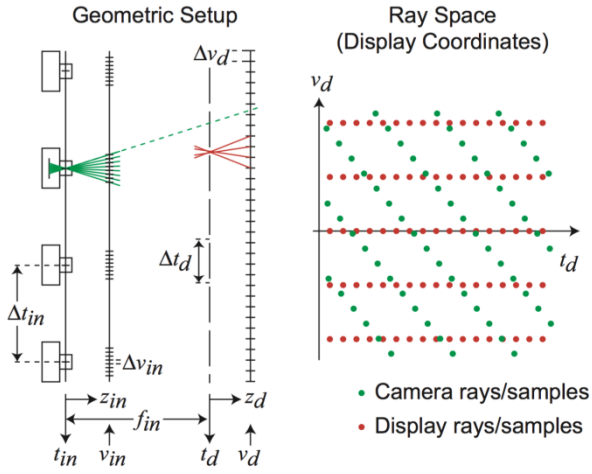


Figure 59 - Light Field resampling [R59]

Moreover, full parallax super-multiview light field displays are capable of reproducing a light field that contains thousands of views aligned horizontally and vertically, which have to be obtained from the camera array that is restricted to dozens and practically no more than hundreds of cameras. In this case, virtual views in-between the existing acquired views have to be rendered using DIBR processing, see Figure 60, similar to what has already been explained in Figure 55 and Figure 56 for creating virtual viewpoints in Free Navigation.

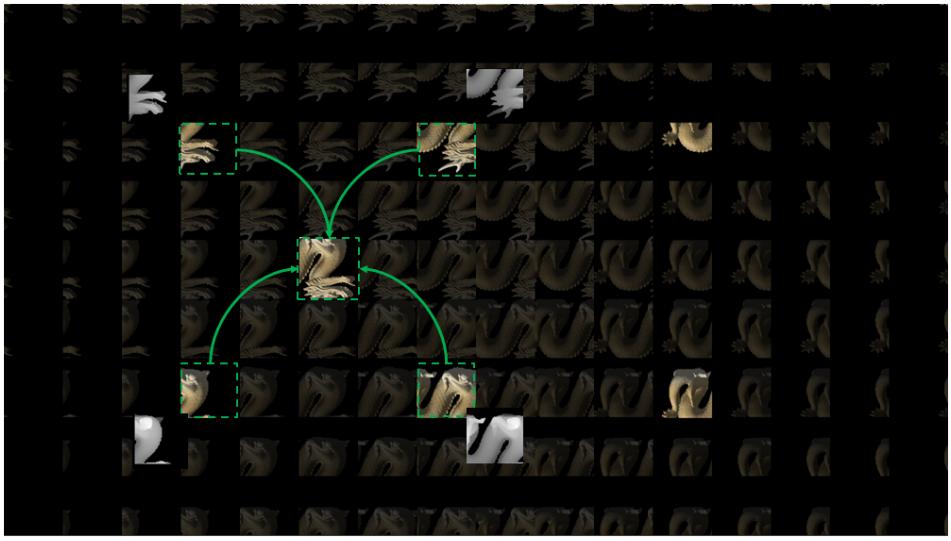


Figure 60 - Light Field being reconstructed with DIBR (reference images are in full color, rendered images are represented with ¼ brightness, the depth maps are provided in gray).

Notice that this also allows for the compression of incoming data to the display system, since fewer Elemental Images are necessary to generate the complete light field.

These Elemental Images can also be obtained by simulating the light transport through the optical elements of the display and requires the complete description of the scene such as light sources, object dimensions, and their positions, as described in Figure 41 and Figure 44 for 3D graphics objects.

8.1.2 SUPER-DENSE LIGHT FIELD DISPLAYS WITH "HOLOGRAPHIC" EYE ACCOMMODATION

There exist Horizontal Parallax Only (HPO) light field displays, as well as Full Parallax (horizontally and vertically) displays, which are often referred to as Integral Photography displays.

If the angular resolution of the Elemental Images (EI) is sufficiently high (i.e. closely adjacent viewing directions correspond to different EIs), or equivalently, many different light rays enter the pupil, all eye accommodation cues are restored. This enables a holographic viewing experience where one can consciously, at his/her own will, re-focus his/her eyes on foreground or background objects, without changing system parameters.

Such "holographic eye accommodation" typically requires the projection of thousands of Elemental Images in many different directions, and bandwidth limitations often prevent the transmission of all these images. Consequently, only a dozen (or maximally hundreds) of directional Elemental Images are transmitted, and all other images of the set of thousands are synthesized following DIBR methods, very close to those already presented in the Light Field subsections (microlens and/or camera arrays) for Free Navigation (FN).

9. USER INTERACTION

User interaction can range from triggering the creation of a new virtual view position in correspondence to a change in the user's viewpoint, to full interactivity with the scene where the user grasps virtual objects in the scene. This level of user interaction might influence the preferred data representation for the use case under consideration.

User position signaling has potentially a large impact on the data flow and processing through the pipeline of Figure 1. Typically, for a particular virtual viewpoint to synthesize, not all segments of the bitstreams are needed, and hence only partial decoding should be enabled for better performance, especially in a VR context where low-latency processing is of paramount importance.

10. COMMON ASPECTS OF DATA FORMATS

10.1 RENDERING QUALITY VS. BITRATE PERFORMANCE FIGURES

To evaluate the relative merits of all aforementioned technologies, it is common practice to create Quality-Bitrate figures that express how quality changes with the compression ratio of the transmitted data.

In the past, single-view video codecs were merely concerned by how well the pixels on a screen could be recovered directly from the bitstream, so that in effect there was little concern about the rendering module in the pipeline. The quality was then mostly expressed by the difference between the original raw pixel value at acquisition and the recovered pixel value after decoding, using the Peak Signal to Noise Ratio (PSNR) metric.

A similar metric has also been used in 3D graphics rendering, but rather expressed as the difference between the original vertex positions and their decoded counterparts, not as the difference in the actual pixel-related quality on the display.

With the advent of more complex rendering processes like Free viewpoint Navigation, where completely new virtual views that were not present in the decoded bitstreams have to be rendered, the quality metric has to be revisited.

For synthetic content (objects modeled in 3D with editing software, rather than acquired with cameras), the generated virtual view (e.g. with DIBR techniques) can always be compared – pixel by pixel, after rendering – with a view directly rendered from a 3D graphics pipeline rendering using the raw 3D data.

However, for natural content acquired through cameras (microlens light field and/or multi-camera light field array), there most often does not exist any reference for the virtual views, therefore disabling the use of objective quality metrics like PSNR. How to exactly handle these cases is still an open question (there are some proposals, but the exact parametric settings are not yet validated), but it is anyway clear that the quality metric should somehow include knowledge of the rendering process.

In summary, the aforementioned Quality-Bitrate performance figure should be thought of as a "Rendering Quality vs. Bitrate" figure, and hence the rendering aspects described in previous sections should be included. Complementary to the rendering, the data attributes related to the bitrate axis should be further analyzed. The next subsections will therefore identify the commonalities and differences between the presented technologies, making recommendations to identify the "greatest common denominator" codec that might encompass all the technologies presented in the previous sections.

10.2 DATA REPRESENTATION COMMONALITIES

All previously discussed visual data representations have in common that color information is conveyed for all light propagation directions in each point in space, either by (A) explicitly

defining the points in space and their color attributes (point clouds and textured 3D meshes), or by (B) defining the color characteristics of the light rays that are emanating from these points.

Category (A) describes the object in space and simulates the light transport during rendering for any requested viewpoint, while category (B) holds information about the physical reality of light transport for the acquisition viewpoints (the rendering comes for free from nature and the results are captured by cameras) and estimates the scene geometry with minimal information (depth) just good enough to be able to do additional "synthetic rendering" (e.g. DIBR) for the missing virtual viewpoints, see Figure 61.

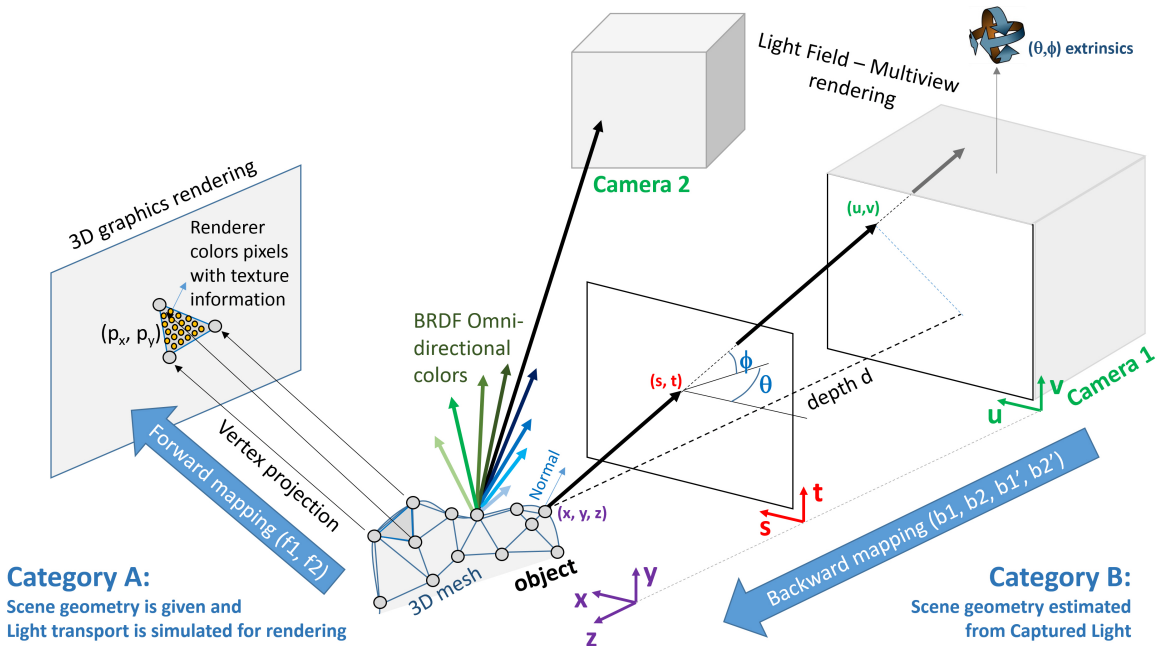


Figure 61 - Object description and Rendering process in Categories A and B

Both categories are conceptually equivalent. A point (x, y, z) of category (A) from which light emanates in direction (θ, ϕ) can indeed be represented in a multitude of data models representing category (B):

I. Category (A) - Point (x, y, z) + light direction (θ, ϕ) :

   The corresponding display pixel coordinates (p_x, p_y) can be recovered as a function of these 5 parameters with $p_x = f1(x, y, z, \theta, \phi)$ and $p_y = f2(x, y, z, \theta, \phi)$, where f1 and f2 represent forward mapping functions.

II. Category (B) - Light direction (θ, ϕ) and point-to-light direction (s, t), with depth d of the point the light ray emanates from:

   The exact position (x, y, z) of the point can be recovered with $x = b1(s, t, \theta, \phi, d)$, $y = b2(s, t, \theta, \phi, d)$ and $z = d$, where b1 and b2 represent backward mapping functions.

III. Category (B) - Light direction (θ, ϕ) and point-light direction (s, t), with depth d of the point the light ray emanates from:

The exact position (x, y, z) of the point can be recovered with x = b1' (s, t, u, v, d), y = b2' (s, t, u, v, d) and z = d, where b1' and b2' represent backward mapping functions.

IV. Category (B) - Light direction {(s, t) & (θ, ϕ)} or {(s, t) & (u, v)} without depth information:

It does not allow a bijective backward mapping, leaving some scaling ambiguity about the recovered point (x, y, z) coordinates.

However, if depth can be recovered (e.g. stereo matching, EPI depth estimation), the bijective relationship between Category (A) on one side, i.e. (I), and Category (B) on the other side, i.e. (II) & (III), is restored.

Categories (A) & (B) are thus equivalent, if (and only if):

- Depth can be exactly recovered or is transmitted in the form of depth maps
- Occlusions can be properly handled in the renderer during the virtual view synthesis stage
- All data can be densely acquired

We can intuitively imagine that practical considerations will have a large impact on these conditions. Table 1 in section 2 and Table 2 in section 6 gives an overview of coding tools that have been developed throughout the long history of multimedia and JPEG/MPEG. The first two rows correspond to category A, while all others correspond to category B.

Be aware that the coding tools in Table 1 and Table 2 merely reflect historical choices; data format conversions as those presented in section 5 have not yet been considered in reaching optimal coding performance.

10.3 DATA COMPRESSION COMMONALITIES AND DIFFERENCES

Knowing that aforementioned Categories (A) and (B) are equivalent from the data representation point of view, can we conclude that all these data representations will have similar performances w.r.t. compression? The answer is intuitively (and most probably definitely): "No".

All compression techniques rely on exploiting redundancies over the data set, and in JPEG/MPEG codecs (see Table 2), streaming conditions even impose some serialization and neighborhood constraints in exploiting these redundancies.

For instance, in Category (A), while 3D meshes have a connectivity between vertices that implicitly defines this neighborhood, point clouds on the contrary have a priori no neighborhood defined in their data representation, unless one imposes a serialization as for example the octree data structure of Figure 31. It is also widely accepted that the scene graph that arranges the logical and spatial representation of a multimedia scene into a hierarchical graph of nodes exhibits a high heterogeneity, hence a low compression ratio (10:1 is common).

In Category (B) also, one might expect some unfavorable conditions. For instance, in multiview video coding, moderate-to-high quality depth map stored into the bitstream might represent already 30% of the total bitrate. In view of the importance of having a depth map for many rendering operations presented in above light field sections, one should be cautious. In fact, the depth map (or disparity map) is the equivalent of motion vectors in single-view codecs, with the

difference that depth/disparity maps can conceptually be recovered at the decoder without prior estimation at the encoder.

The difference between Categories (A) and (B) w.r.t. compression performance will mainly lie in the convenience of a given representation for a given application (cf. interactivity vs. immersivity), as well as its potential in exploiting the data redundancies in this application context. These can only be checked on real-life examples and real coding tools; a philosophical discussion may bring little added value.

An overview of the current status of coding tools is provided in Table 1 of section 2 and Table 2 of section 6. Commonalities between the tools have hardly been considered so far in order to select a single recommended tool; a more in-depth study is required.

11. APPLICATION FOCUSED WORKFLOWS

A list of use cases has been given in section 2 with a table to summarize the different data representation for the different workflows. In this section some specific workflows are presented corresponding to a list of applications.

11.1 BROADCAST SUPERMULTIVIEW

This use case has been described as use case 1.1 in section 2. It corresponds to a broadcast scenario addressing Super Multi-View displays. We consider two different acquisition systems: coplanar camera rig or light-field camera with their respective data representation. The camera rig delivers a multi-view content from which a depth information can be extracted. An extension of 3D-HEVC may be required to address horizontal and vertical dimensions of the acquisition system. Depth Image Based Rendering is applied to render missing views on the display. For light-field cameras (micro lenses in between the sensor and the main lens) data representation may differ to include spatio-angular data that are required to fully exploit the content properties at the rendering side.

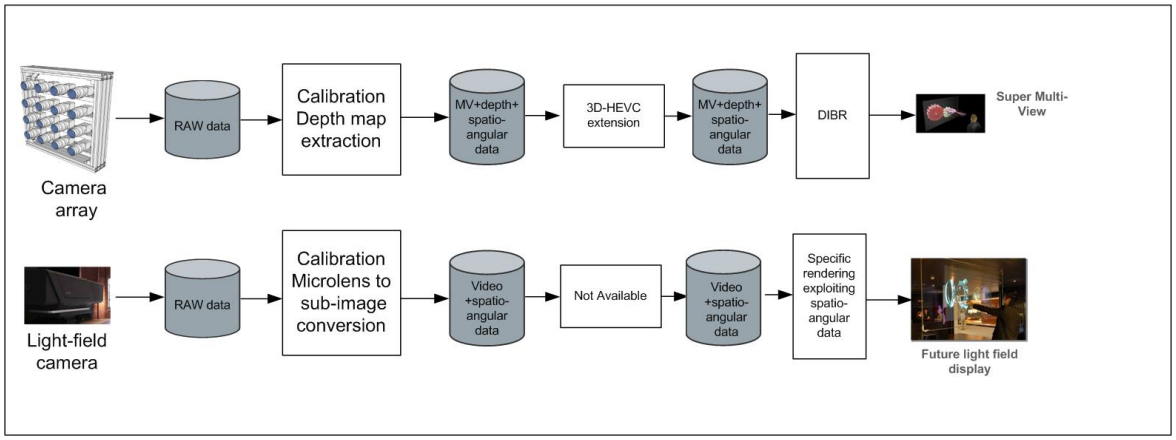


Figure 62 - Broadcast Super Multi-View scenario

11.2 IMMERSIVE BI-DIRECTIONAL COMMUNICATION

This use case has been described as use case 1.2 in section 2. The goal is to deliver between 2 or N users an immersive/realistic content captured by one of the users. The following figure illustrates an augmented reality scenario where the user is able to interact with the renderer. For instance when using see-through glasses the user can move around the object/person. The immersive aspect is clearly depending on the nature of data transmitted.

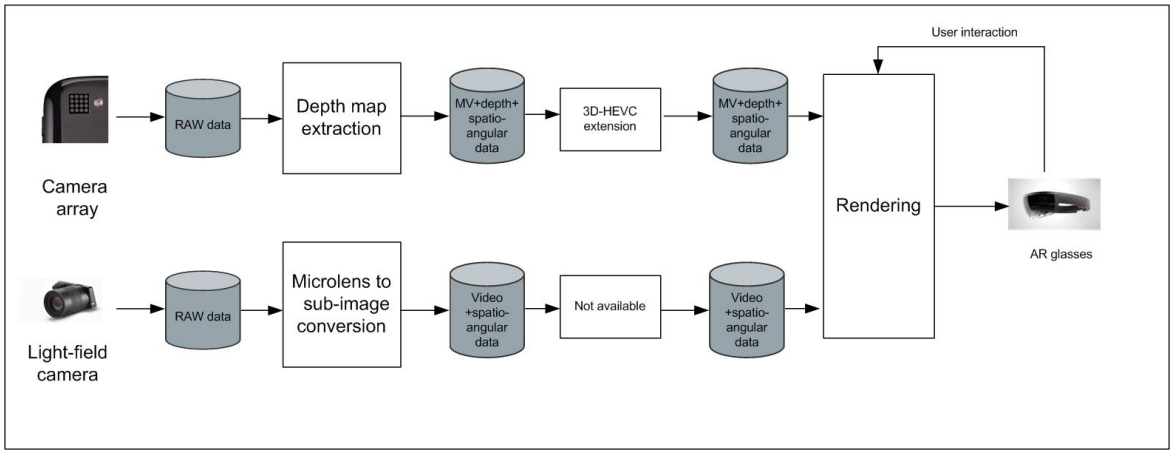


Figure 63 - Immersive bi-directional communication scenario

11.3 PERSONAL EDITING WORKFLOW

This use case has been described as use case 2.1 in section 2. It is corresponding to a personal editing application (e.g. refocusing) with two different scenarios. In the first one (Figure 64) the captured content is a posteriori processed by the person who has made the capture. No need to compress and transmit any data.

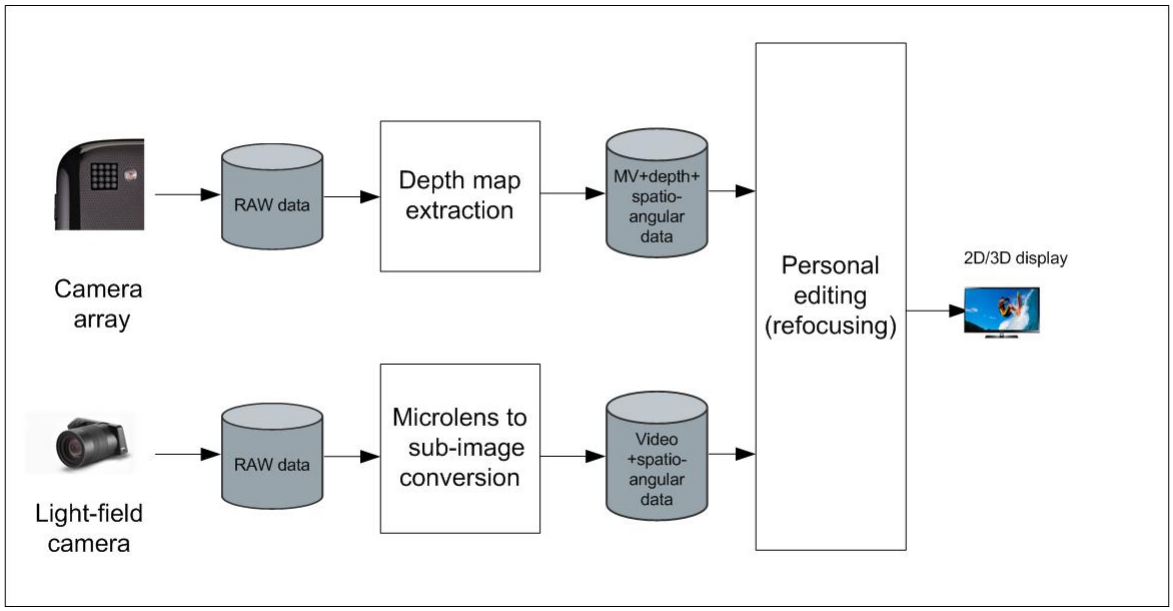


Figure 64 - Personal editing workflow without transmission

In the second scenario (Figure 65), the content is first transmitted to a second person who will make the a posteriori processing. This second case requires to transmit the whole set of data to allow this processing. Two different acquisition system have been considered, one using an array of camera on a cell phone, the other one using a light field camera.

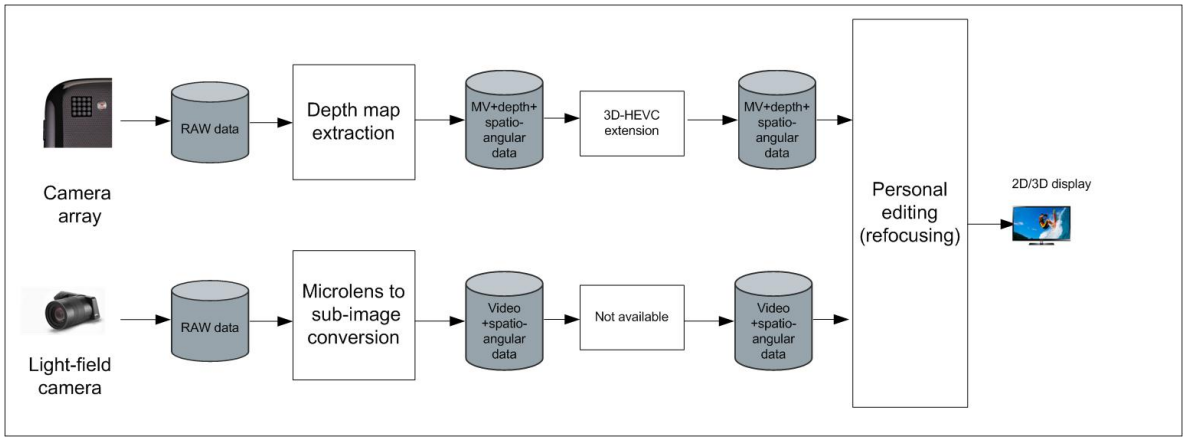


Figure 65 - Personal editing workflow with transmission

11.4 POST PRODUCTION WORKFLOW

This use case has been described as use case 2.2 in section 2. It is corresponding to a cinematic post production application where a given number of sources has to be considered. The following figure (Figure 66) illustrates the different video sources and their corresponding data conversion. The post-production tool shall be able to mix these different contents which means a common understanding of the different data representation. For light field content from camera array or light-field camera, spatio-angular data are required to fully exploit their respective content.

After post-production a standard 2D/3D content is generated and can be transmitted using available standards.

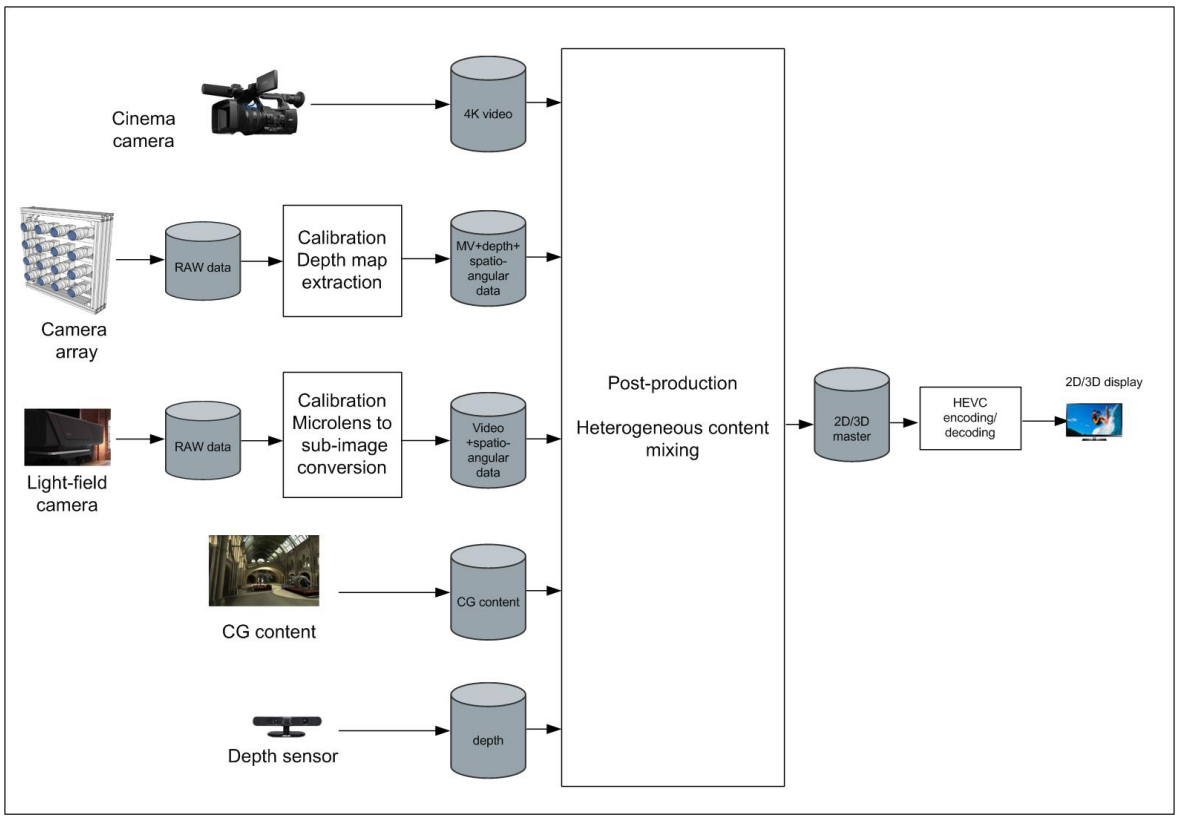


Figure 66 - Post production workflow

11.5 FREE NAVIGATION WORKFLOW

This use case has been described as use case 3.1 and 3.2 in section 2. It is corresponding to a free navigation scenario with different acquisition systems. In the upper part of the following figure (Figure 67), a 360° camera provides a multi-view content to be stitched before transmission. The lower part represents a multi-camera system in a stadium to capture different views of the same scene. Extension of 3D-HEVC is required to handle omni-directional content and multi-camera system. The user interacts with the system to select the right viewing angle.

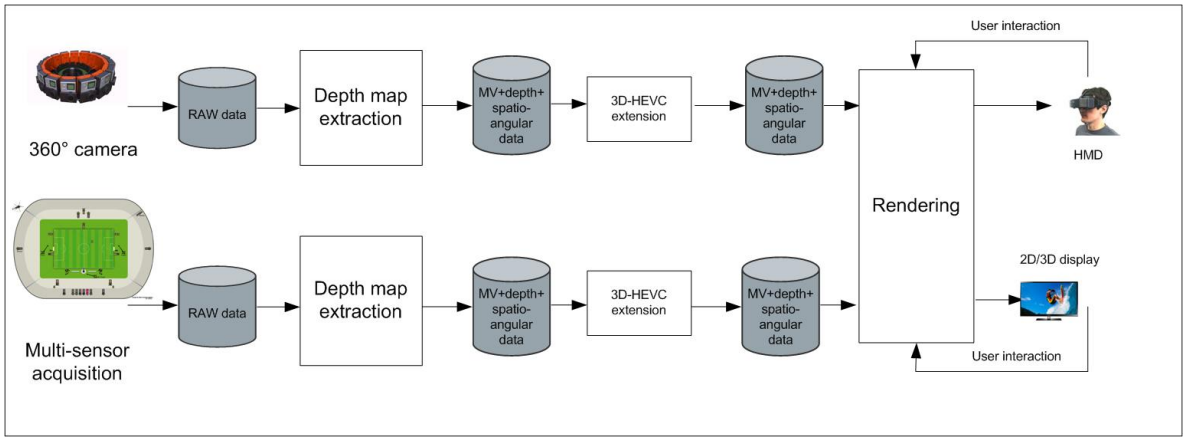


Figure 67 - Free navigation workflow

A more complex workflow is required to be able to address the use case 3.3 or to allow full motion parallax capabilities for the 3.1 use case.

11.6 LIGHT FIELD BASED WORKFLOW

Error! Reference source not found. illustrates a light field display pre-processing use case. In pre-processing case light field data is prepared for use by multiple displays, therefore when the data reaches a specific display a further display specific rendering is required. The scenario illustrated in Figure 68 starts with light field data acquisition then the light field data acquired by multiple sensors are processed to create raw multi view data, depth data and metadata. This raw information goes through a pre-processing stage to create the mastered multi view, depth and metadata. The mastered content is encoded and prepared for distribution and sent to a display for further display specific rendering and display.

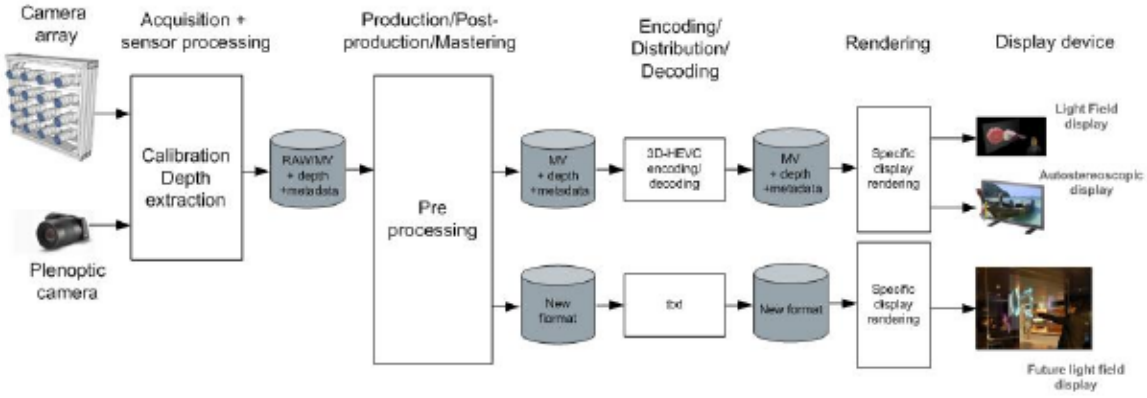


Figure 68 - Light Field Display Pre-Processing Use Case

Figure 69 illustrates a light field display post-processing use case. In post processing case light field data is prepared with a specific display in mind, and it can be displayed with the display directly after its reception. The scenario illustrated in Figure 69 starts with light field data acquisition then the light field data acquired by multiple sensors are processed to create raw

multi view data, depth data and metadata. This raw information goes through a post-processing stage to create the mastered multi view, depth and metadata specifically for the display at the end of the pipeline. The mastered content is encoded and prepared for distribution and sent to a display for direct display without further processing.

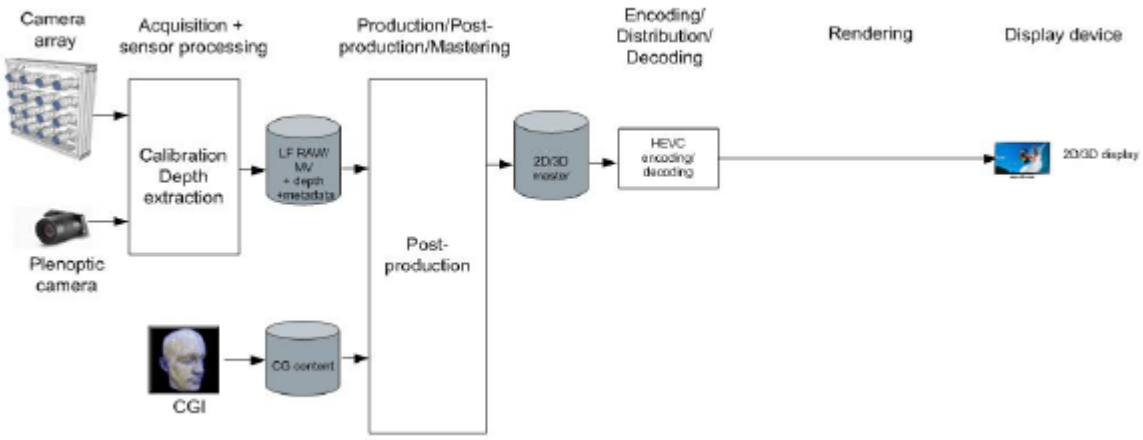


Figure 69 – Light Field Post Processing Use Case

Figure 70 illustrates a light field display interactive real-time processing use case. In interactive real-time processing case light field data is prepared with a specific display in mind, however unlike the post-processing case the data being sent to the display is further processed with the interaction commands. The scenario illustrated in Figure 70 starts with light field data acquisition then the light field data acquired by multiple sensors are processed to create a preferred light field format (i.e., mastered light field data). While the user is working with the display only a specific Region-of-Interest (RoI) is displayed. RoI Extraction stage requests this data from the stored mastered light field data and sends it to the Render/Encode operations. The rendered/encoded data is sent to the display in a compressed format. The compressed light field data can be either partially decoded before the display and finish decoding in the display or sent directly to the display and get fully decoded at the display. User interaction is handled in real-time and it creates a feedback loop to the RoI extraction either to modify the existing data in the storage or to request the next RoI.

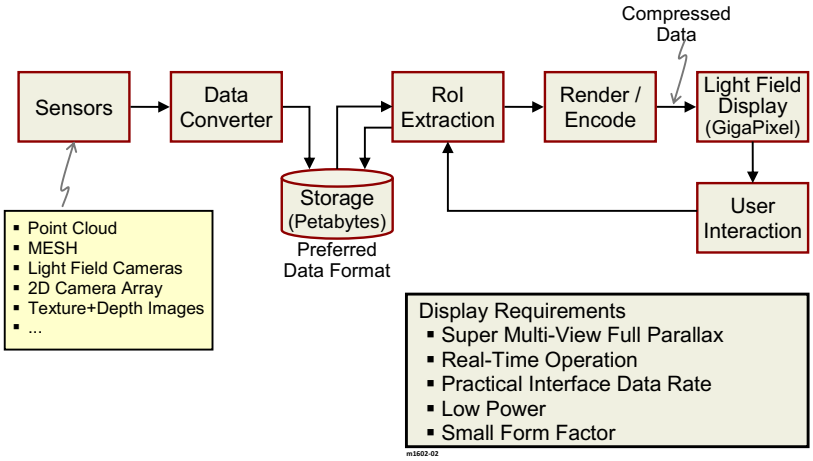


Figure 70 - Light Field Display Real-Time Processing Use Case

12. CONCLUSIONS

We have presented various 3D data representation and coding formats related to immersive use cases. In particular, we have shown the commonalities between light fields captured with lenslet cameras (multi-cameras in-a-box), multiple camera array arrangements and point clouds. An open question remains w.r.t. the trade-off between image-based approaches where the light rays are captured and interpolated on one hand, and the object-based descriptions with material characteristics where the light transport simulation should be performed within the rendering engine on the other hand. Nevertheless, the presented workflows share common depth-based functionalities, as well as the need to represent all points in space, either by their color and precise 3D position in point clouds, or by the light rays emanating from these points described by light fields. Adding sound fields with similar free positioning characteristics will support full cinematic VR use cases.

ANNEX

REFERENCES

[R2] https://en.wikipedia.org/wiki/Light_field

[R3b] http://graphics.stanford.edu/projects/lightfield/

[R4] http://www.cg.cs.tu-bs.de/teaching/seminars/ws1112/CG/webpages/Jannik_Winkel/

[R5a] https://yvonnesmedia.wordpress.com/category/research/

[R5b] https://en.wikipedia.org/wiki/The_Matrix#Visual_effects

[R7] http://www.facetvision.de/fraunhofer.html

[R9] https://www.lytro.com/imaging

[R18] V. Högman. Building a 3D map from RGB-D sensors, MS.c theses at the Computer Vision and Active Perception Laboratory of the Royal Institute of Technology (KTH), Stockholm, Sweden, 2011.

[R19] Arnaud Bletterer, Frédéric Payan, Marc Antonini, Anis Meftah, "Point Cloud Compression using Depth Maps," Electronic Imaging, San Francisco, Feb 14-18, 2016.

[R20] http://therandomlab.blogspot.be/2013/01/visualizing-kinects-3d-mesh-with.html

[R21] http://aecmag.com/59-features/829-from-point-cloud-to-bim

[R22] Sergio García Lobo, Pablo Carballeira López, Francisco Morán Burgos, "Further results on scene reconstruction with hybrid SPLASH 3D models," ISO/IEC JTC1/SC29/WG11/ MPEG2016/M38037, February 2016, San Diego, CA, US.

[R27] http://www.nozon.com/presenz

[R29] http://www.informit.com/articles/article.aspx?p=2418908&seqNum=4

[R31] Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. Autonomous Robots, 34(3), 189-206.

[R32] J. Kammerl, N. Blodow, R. B. Rusu, S. Gedikli, M. Beetz, and E. Steinbach. Real-time compression of point cloud streams. In Robotics and Automation (ICRA), 2012 IEEE International Conference, pages 778–785. IEEE, 2012.

[R35] D. Berjón, F. Morán, S. Manjunatha: "Objective and subjective evaluation of static 3D mesh compression", Elsevier Signal Processing: Image Communication, vol. 28, nr. 2, p. 181-195 (doi: 10.1016/j.image.2012.10.013), February 2013.

[R36] http://www.codeproject.com/Articles/909/Achieving-PostScript-and-Wmf-outputs-for-OpenGL

[R39] https://server.pointcloudviz.com/

[R47a] Google inc, “Rendering Omni-directional Stereo Content,” https://developers.google.com/cardboard/jump/rendering-ods-content.pdf

[R47b] Zhigang Zhu, « Omnidirectional Stereo Vision, » Workshop on Omnidirectional Vision, in the 10th IEEE ICAR, August 22-25, 2001, Budapest, Hungary.

[R49a] S. J. Gortler et al., “The Lumigraph”, in Computer Graphics Proceedings, pp. 43–54, Proc. ACM SIGGRAPH’96, New Orleans, USA, Aug. 1996.

[R49b] A. Kubota, A. Smolic, M. Magnor, M. Tanimoto, T. Chen, C. Zhang, “Multiview Imaging and 3DTV”, IEEE Signal Processing Magazine, vol. 24, no. 6, pp. 10-21, Nov 2007.

[R50] Ng, Ren, et al. "Light field photography with a hand-held plenoptic camera." Computer Science Technical Report CSTR 2.11 (2005): 1-11.

[R51a] R. Ng., Digital Light Field Photography, Ph.D. Thesis, Stanford University, Jul. 2006.

[R51b] D. G. Dansereau, Plenoptic Signal Processing for Robust Vision in Field Robotics, Ph.D. Thesis, The University of Sydney, Australia, Jan. 2014.

[R55] C.M. Palomo, “Interactive image-based rendering for virtual view synthesis from depth images”, PhD Thesis, Pontífícia Universidade Católica do Rio de Janeiro, Brazil, Jul. 2009.

[R56] https://vimeo.com/130983212

[R58] Arai, Junichi. "Three-dimensional television system based on integral photography." Picture Coding Symposium (PCS), 2012. IEEE, 2012.

[R59] Zwicker, Matthias, et al. "Antialiasing for automultiscopic 3D displays." ACM SIGGRAPH 2006 Sketches. ACM, 2006.

[r60] Fernando Pereira, Eduardo A. B. da Silva, Gauthier Lafruit, “Plenoptic Imaging: Representation and Processing,” in R. Chellappa and S. Theodoridis, editors, Academic Press Library in Signal Processing.