Interim Specification of Generic Data Representation and Coding Scheme

Deliverable D2.1.2

FascinatE identifier:	Fascinate-D212-HHI-InterimSpecGenericDataRepresentationCoding-v06.docx
Deliverable number:	D2.1.2
Author(s) and company:	O. Schreer, P. Kauff (HHI), J.-F. Macq, P. Rondão Alface (ALU), J. Spille (DTO), R. Oldfield (UOS), G. Thomas(BBC)
Internal reviewer:	J.F. Macq (ALU), G. Kienast (JRS)
Work package / task:	WP2
Document status:	Final
Confidentiality:	Public

Version	Date	Reason of change
1	2011-11-18	Document created
2	2011-12-02	Document updated by BBC, UOS, ALU and DTO
3	2012-01-04	Further input received by partners
4	2012-01-09	Final version for internal review
5	2012-01-24	Revision after receipt of internal review
6	2012-01-26	Final version

The work presented in this document was partially supported by the European Community under the 7th framework programme for R&D.

Table of Contents

List of figures:

List of tables:

1 Executive Summary

One of the key challenges of FascinatE is to capture a natural scene with many different audio-visual sensors and to offer this information in a commonly accessible framework in order to allow users to freely navigate in a scene by avoiding current limitations in a conventional production workflow.

Due to this, the large variety of sensors (microphones and cameras) needs to be related in a common framework, which we call Layered Scene Representation (LSR). This requires on the one hand a joint geometrical framework, which relates all sensors with respect to their spatial position and orientation. On the other hand, the mapping process of visual input from different cameras of different types is described in order to provide a schema for coherent registration of all the visual information in a common visual representation format.

As this layered scene representation defines the relationship between all the sensors in order to allow managing and rendering all the audio-visual information, the deliverable also discusses in a separate chapter coding aspects for audio and video.

This is an interim deliverable after the second project year (M24, January 2012), revisiting the generic data representation for format-agnostic production, the type and structure of calibration data needed as operative metadata for the further processing chain as well as a coding scheme suited to the generic data representation. It will provide updated information required for system specification in WP1 and investigations on coding in WP4.

The major changes in this second interim version of the Specification of Generic Data Representation and Coding Scheme are as follows:

- The mapping of any camera view into the panoramic view has been described in an idealistic way. Due to first experiments and results achieved in the first project year, it has been recognized that in practical situations the disparity between cameras cannot be neglected. Hence, this requires other approaches of merging content from different cameras. This fact is elaborated in detail in the section 3.2.6 Mapping Rules.
- The section 3.3 Audio Scene has been revised completely in order to consider the research results during the first year. Now, a detailed description of Audio Objects (AO), Sound Fields (SF) and Sound Descriptions can be found.
- The tables that contain all the different parameters of the layered scene representation have been revised.
- Some references to other meta data (production scripts, …) and meta data formats have been removed as these topics are elaborated in much more detail in *D3.1.2 Metadata and Knowledge Models and Tools* of WP3.

This deliverable will be updated during the project and a final version will be issued as D2.1.3 in Month 42.

This interim version of the Specification of Generic Data Representation and Coding Scheme has been delivered due to a request by the reviewers in the first project review. A set of comments have been made, which have been considered to our best in this revised version of the document. The comments have been as follows:

"Several open questions for the generic data representation and coding scheme still exist."

The consortium fully agrees and the provided interims version tries to give a few more answers.

"Now, it seems to be that the assumption at page 15 is not true i.e. perspective differences from different cameras are negligibly small. Further investigations are needed and should be documented."

An extra section has been included to consider this fact. However, the solutions and approaches to this problem are not yet available. Research and development is still under progress and results will be presented in upcoming WP2 deliverables.

"Depth of Field is not discussed"

The depth of field is not part of this deliverable to our understanding. The usage of a 3D laser scanner to create 3D model data has been discussed in "*D2.2.1 Specification and first test implementations of capture and hardware components*". The exploitation of this data for calibration and possibly encoding is currently under investigation. Results will be presented in upcoming deliverables of WP1, WP2 and WP4.

"HDR integration is not adequately addressed."

The HDR integration is performed in the FascinatE system by using the new Alexa M in the OMNICAM. As here in this deliverable data formats are specified, the HDR capabilities are reflected in the section 2.2.3 where the colour space is defined. A more detailed presentation of HDR acquisition is presented in *"D2.2.1 Specification and first test implementations of capture and hardware components"*.

2 Introduction

2.1 Purpose of this Document

This deliverable is an updated version of *D2.1.1 Draft Specification of Generic Data Representation and Coding Scheme*, which revisits the generic data representation for format-agnostic production, the type and structure of calibration data needed as operative metadata for the further processing chain as well as a coding scheme suited to the generic data representation. It will provide updated information required for system specification in WP1 and investigations on coding in WP4. It will be updated during the project and a final version will be issued as D2.1.3 in Month 42.

2.2 Scope of this Document

This deliverable presents an overall framework, showing how an audio-visual scene can be described in a hierarchical order by partitioning a scene into an audio and video scene. The audio and video scene is further broken down to sensor level and the geometrical relations as well as the parameters describing each layer are listed. One section describes the mapping from different video sources into a common panoramic image, which is relevant for the final presentation of the format-agnostic production workflow.

The deliverable represents the status of the definition of the layered scene representation at month 24. As explained in the Executive Summary, some changes have been performed reflecting the progress and research results during the last two years. However, we will keep some level of flexibility and will adopt the layered scene representation according to upcoming needs and requirements during the next period. The necessary but expectant minor changes will be reported in D2.1.3 by end of the project.

2.3 Status of this Document

This is the final version of D2.1.2.

2.4 Related Documents

Before reading this document it is recommended that the reader is familiar with the following documents:

- *D1.1.1 End user, production and hardware and networking requirements* defines the FascinatE scenarios, use cases and requirements, from which requirements on the metadata representation can be derived.
- *D1.5.1 First System Integration* describes the status of the development of key modules in the system as of Month 21 (as shown at the demonstration at IBC 2011)
- *D2.2.1 Specification and First Test Implementations of Capture and Hardware Components* describes the components of the acquisition part of the system, and includes some results from the first test shoot.
- *D3.1.2 Metadata and Knowledge Models and Tools* describes the specification of the metadata and knowledge models used in the FascinatE system

3 Generic Data Representation Scheme

3.1 General Structure

The generic data representation scheme follows a hierarchical tree structure describing the spatial and temporal relation of all captured audio-visual data in one common framework. The highest level is called layered scene representation and may consist of different scenes. Each scene describes a closed area (e.g. room, stadium) that can be watched by several cameras and contains a set of corresponding audio sources. The different scenes might occur simultaneously (different locations of same sport event like different sections of a race or different arenas of the same Olympic Games) or subsequently following a time line or a story board like different locations of a fictive story (or a combination of both).

Each scene contains one audio scene and one video scene as shown in Figure 1. The audio scene consists of individual audio objects and one or more ambient sound representations. Audio objects are recorded with close up microphones or estimated using an array of microphones including gunshot microphones. Sound fields, transmitted as Ambisonics signals, are recorded using microphone arrays and/or composed out of individual signals. The video scene usually consists of several spatially distributed camera clusters. Each camera cluster represents a spatial arrangement of different close-distant cameras, showing almost the same view of a scene from one specific viewpoint, but with different properties. The cameras can be as well of different type such as broadcast cameras or an omni-directional camera. A video scene should contain at least one camera cluster. The next sections describe the structure of video and audio scenes in more detail.

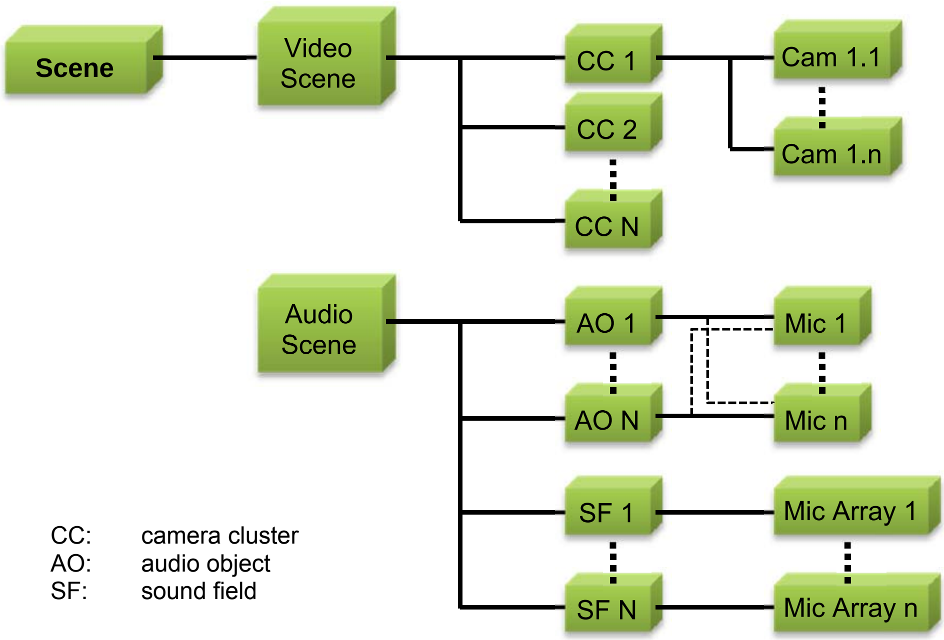

Figure 1: Hierarchical structure of the layered scene description

3.1.1 Geometry of the audio-visual scene

In Figure 2 the different coordinate systems of the audio and video scene are depicted in order to show their global relationship and their meaning. The individual coordinate systems are then defined in detail in the next sections 3.2 Video Scene and 3.3 Audio Scene.

The world coordinate system of the audio-visual scene is described in Cartesian coordinates whereas the origin $\mathbf{O}_w(X,Y,Z)$ is located at a predefined position, e.g. the kick-off of the football field. The video scene as well as the audio scene is referring to this origin in order to have the same reference in 3D space for audio and video.

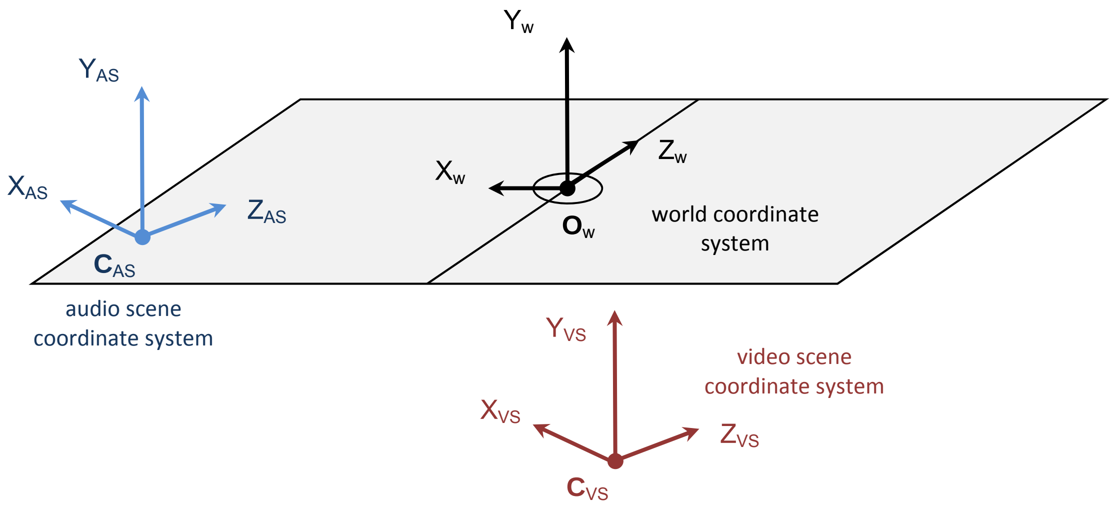

Figure 2: Definition of audio and video scene coordinate systems related to the world coordinate system

The transformation between different coordinate systems is performed by a standard 3D coordinate transformation (see Figure 3). The point in world coordinates refers to the origin in the scene. The world coordinate system and any arbitrary located and oriented coordinate system are linked via rotation R and a translation $\mathbf{t}$ such as

$$\mathbf{M}_O = I \cdot \mathbf{M}_w \quad \text{and} \quad \mathbf{M}_c = R\mathbf{M}_w + \mathbf{t} \tag{1}$$

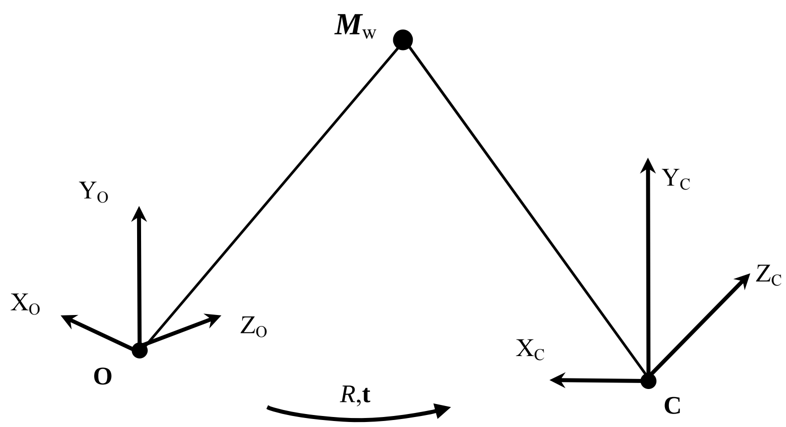

Figure 3: Relationship between two Cartesian coordinate systems

In order to express a point, defined in an arbitrary coordinate system, in the world coordinate system, the following coordinate transformation has to be applied:

$$\mathbf{M}_w = R^T(\mathbf{M}_C - \mathbf{t}) \Rightarrow \mathbf{M}_O = R^T(\mathbf{M}_c - \mathbf{t}) \tag{2}$$

The scene object as part of the layered scene representation only contains header data. Some are mandatory, others are optional.

Mandatory data are

- time code data specifying the temporal relations between the different scenes along a common timeline
- the metric of the spatial dimensions within one scene
- a reference world coordinate system following the above metric

Optional data are

- a text description of the scene content
- the meaning of the world coordinate system (e.g. kick-off of a football field)

Below in Table 1, the proposed parameters for the scene header are listed.

Scene Header				
Parameter	Symbol	Description	Units	Required = R, optional = O
TimeCode	t_c	Time code specifying the temporal relations between different scenes	ms	R
Metric		Metric of the spatial dimension	m, mm	R
RWCS	$\mathbf{O}(0, 0, 0)$	Reference world coordinate system	m	R
SceneDesc		Text description of the scene content		O
MeaningWCS		Meaning of the world coordinate system		O
Rotation AS	$R_{AS\to W}$	Rotation of the audio scene against world coordinate system		R
Translation AS	$\mathbf{t}_{AS\to W}$	Translation of the audio scene against world coordinate system		R
Rotation VS	$R_{VS\to W}$	Rotation of the video scene against world coordinate system		R
Translation VS	$\mathbf{t}_{VS\to W}$	Translation of the video scene against world coordinate system		R

Table 1: Parameters for the scene header

The transformation from a point in world coordinates to the audio scene coordinate system and the video scene coordinate system is defined as follows:

$$\mathbf{M}_{AS} = R_{AS\to W}\mathbf{M}_W + \mathbf{t}_{AS\to W} \quad (3)$$

$$\mathbf{M}_{VS} = R_{VS\to W}\mathbf{M}_W + \mathbf{t}_{VS\to W} \quad (4)$$

3.2 Video Scene

3.2.1 The geometry of the video scene

The video scene consists of a layer, which is called camera cluster. A video scene may contain different camera clusters, each of them containing a set of individual cameras of different type. The individual coordinate system of each camera cluster is defined in relation to the coordinate system of the video scene as depicted in **Figure 4**.

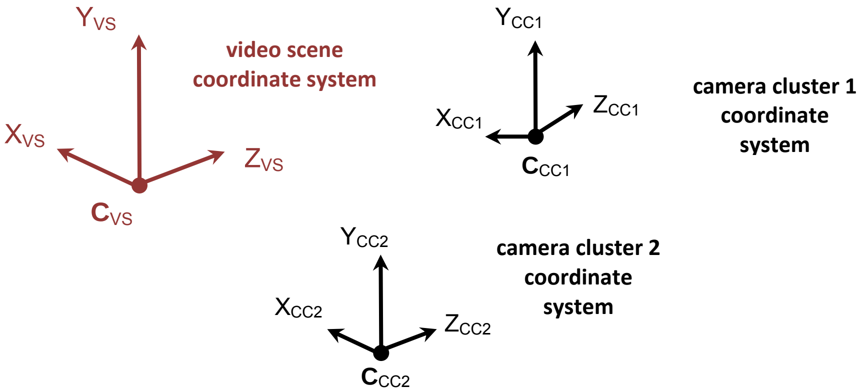

Figure 4: Definition of camera cluster coordinate systems related to the video scene coordinate system

The header of the video scene contains the number of camera clusters in this particular scene and at least the geometrical relationship between one camera cluster and the video scene. It might also contain a text description of the used camera clusters and their particular view of the scene.

In the following Table 2, the required parameters of the video scene header are shown. The parameters of the audio scene header are presented in section 3.3 Audio Scene.

Video Scene Header				
Parameter	Symbol	Description	Units	Required = R, optional = O
NumCC	n_{cc}	Number of camera clusters		R
Rotation CC1	$R_{CC1\rightarrow VS}$	Rotation of the camera cluster 1 against the video scene system		R
Translation CC1	$\mathbf{t}_{CC1\rightarrow VS}$	Translation of the camera cluster 1 against the video scene system		R
Descr1		Description of the viewing area of the camera cluster		O
Rotation CC2	$R_{CC2\rightarrow VS}$	Rotation of the camera cluster 2 against the video scene system		O
Translation CC2	$\mathbf{t}_{CC2\rightarrow VS}$	Translation of the camera cluster 2 against the video scene system		O
Descr2		Description of the viewing area of the camera cluster		O

Table 2: Parameters for the video scene header

The transformation from a point in video scene coordinates to the camera cluster coordinate system is defined as follows:

$$\mathbf{M}_{CC\,i} = R_{CC\,i\rightarrow VS}\mathbf{M}_{VS} + \mathbf{t}_{CC\,i\rightarrow VS}, \quad i = 1...\mathrm{n_{cc}} \tag{5}$$

3.2.2 Camera cluster

Each camera cluster consists of several physical cameras and a related panoramic coordinate system. The reason for the panoramic coordinate system is that in the final rendering, the visual input of all physical cameras of a camera cluster is rendered in a common panorama. For the sake of generality, we distinguish here between the camera cluster coordinate system and the panorama coordinate system, while in practice both coordinate systems will be the same. The relationship between the different coordinate systems is depicted in Figure 5.

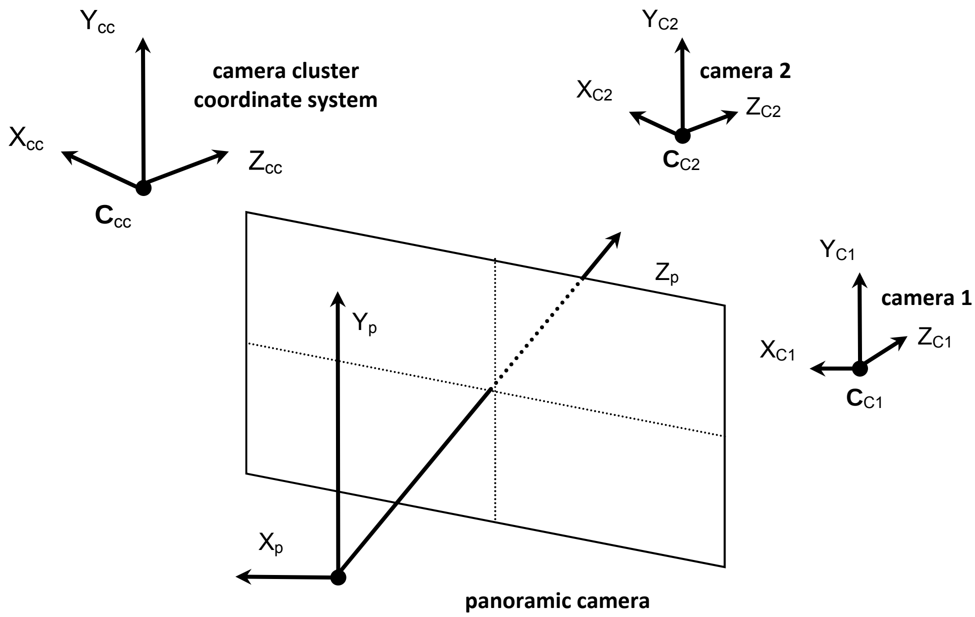

Figure 5: Definition of the camera cluster coordinate systems, the panoramic camera and the individual camera coordinate systems

The panoramic camera can be a planar, a cylindrical or a spherical camera, while the other cameras within the camera cluster are planar cameras. At the current stage, planar and cylindrical cameras will be used definitely in the FascinatE system. Hence, these two types of camera will be defined in the next sections. Spherical cameras are not considered as relevant candidate sensors for panoramic video acquisition. This is mainly due to the limited resolution and the significant amount of geometric distortion. However, for initial test and some specific investigations, spherical cameras have been used (see deliverable *D4.4.1 Definition and First Evaluation of Delivery Mechanisms*).

If there are additional camera clusters available, their perspective will be significantly different from each other. Hence, these additional camera clusters will have again their own panorama coordinate system in order to allow rendering of all the different video sources in a common panoramic scene.

Each camera cluster has a header with information about the number and types of cameras used in this particular cluster. Furthermore each camera in the cluster has a local reference coordinate system that is specified by a translation vector and a rotation matrix relating each camera to the common camera cluster coordinate system. In order to be flexible, the parameters of each camera coordinate system can be either static or dynamic. The term “dynamic” means here that a specific camera is moving with respect to the camera cluster coordinate system. The parameters of the camera cluster can be summarized as follows in Table 3:

Camera Cluster Header				
Parameter	Symbol	Description	Units	Required = R, optional = O
NumCams	n_{cams}	Number of cameras in this clusters		R
Rotation C1	$R_{C1\rightarrow CC}$	Rotation of camera 1 against the camera cluster coordinate system		R
Translation C1	$\mathbf{t}_{C1\rightarrow CC}$	Translation of camera 1 against the camera cluster coordinate system		R
Rotation C2	$R_{C2\rightarrow CC}$	Rotation of camera 2 against the camera cluster coordinate system		O
Translation C2	$\mathbf{t}_{C2\rightarrow CC}$	Translation of camera 2 against the camera cluster coordinate system		O
Rotation CN	$R_{CN\rightarrow CC}$	Rotation of camera N against the camera cluster coordinate system		O
Translation CN	$\mathbf{t}_{CN\rightarrow CC}$	Translation of camera N against the camera cluster coordinate system		O

Table 3: Parameters for the camera cluster header

The transformation from a point in camera cluster coordinates to each individual camera coordinate system is defined as follows:

$$\mathbf{M}_{C\,i} = R_{C\,i\rightarrow CC}\mathbf{M}_{CC} + \mathbf{t}_{C\,i\rightarrow CC}, \quad i = 1 \ldots \mathrm{n}_{\mathrm{cams}} \tag{6}$$

All the different cameras forming a single camera cluster can be of two different types, either a planar camera or a cylindrical camera. Both camera models are revisited in the next two sections.

3.2.3 A single camera

On top of the parameter list describing each camera, some general parameters are defined in Table 4 in order to specify each camera.

General Camera Parameters				
Parameter	Symbol	Description	Units	Required = R, optional = O
CamID	ID	Camera identification number		R
CamType		The type of the camera (planar, cylind., spherical).		R
ColourSys		Colour system of the camera (RGB444, YUV444, YUV422, logC, Rec.709, …)		R
TempRes	Fps	Temporal resolution	Frames/s	R

Table 4: General camera parameters

3.2.4 The model of a planar camera

Figure 6 illustrates the central projection model of a camera with the centre point in the origin of the camera coordinate system. The axes describe the camera coordinate system (X, Y, Z) and the image coordinate system (x, y). In general, a camera is described by its extrinsic parameters that give the position and orientation in the world coordinate system and by a set of intrinsic parameters.

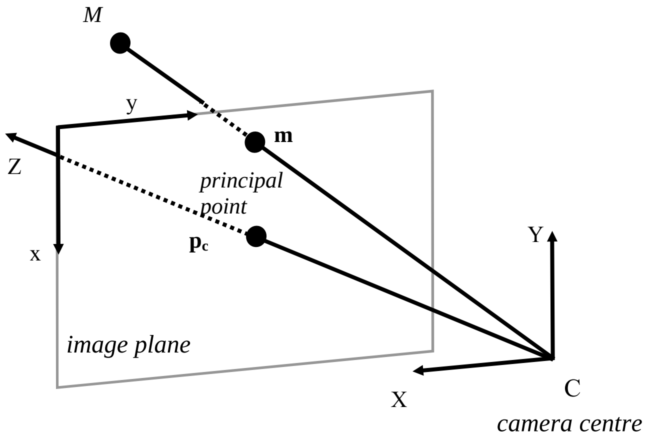

Figure 6: Central projection camera model

The origin $(0,\,0)^T$ of the image coordinate system is addressing the centre of the top-left pixel.

The projection defines a transformation of a point $\mathbf{M}$ defined in the world coordinate system into point $\mathbf{m}$ on the image target. The projection can be expressed in homogeneous coordinates with

$\mathbf{M}=(X,Y,Z)^T \rightarrow \tilde{\mathbf{M}}=(X,Y,Z,1)^T$ and $\mathbf{m}=(x,y)^T \rightarrow \tilde{\mathbf{m}}=(x,y,1)^T$.

The projection is then defined as:

$$\tilde{\mathbf{m}} = KR[I_3\ \mathbf{C}]\tilde{\mathbf{M}} \tag{7}$$

The vector $\mathbf{C}$ specifies the location of the camera centre point. The rotation matrix R describes the rotation of the camera coordinate system against the world coordinate system and will be described in section 3.1.1. I_3 is the 3 x 3 identity matrix.

To get in-homogeneous coordinates the following mapping applies:

$$\tilde{\mathbf{x}}=(x,y,w)^T \rightarrow \tilde{\mathbf{x}}=\left(\frac{x}{w},\frac{y}{w},1\right)^T.$$

The calibration matrix K is defined as:

$$K=\begin{bmatrix} f/s_x & 0 & p_{Cx} \\ 0 & f/s_y & p_{Cy} \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

With f in $[\mathrm{m}]$, the pixel size (s_x, s_y) in $\left[\frac{m}{pel}\right]$ and the principal point

$$\mathbf{p}_C=(p_{Cx},p_{Cy})^T=\left(\frac{N_x-1}{2}+\frac{h_x}{s_x},\frac{N_y-1}{2}+\frac{h_y}{s_y}\right)^T \tag{9}$$

with the image size N_x, N_y.

In an ideal camera, $\mathbf{p}_C$ would be in the centre of the image target $\left(\frac{N_x-1}{2},\frac{N_y-1}{2}\right)^T$. The vector $\mathbf{h}=(h_x, h_y)^T$ therefore describes a centre point shift (in $[m]$).

Note that the centre point shift and radial distortion terms are kept in physical dimensions of the camera target. With this notation the parameters stay unchanged when for example the image resolution is changed.

The inverse projection of a point $\mathbf{x}=(x,\,y)^T$ on the image target gives a line rather than a 3D point unless the depth of that particular point is known. This line is called line of sight of point $\mathbf{x}$ and can be expressed as:

$$\mathbf{r}(\lambda) = \mathbf{C} + \lambda R^{-1} K^{-1} \mathbf{x}^h \quad (10)$$

Camera rotation

Camera position and orientation are defined by the camera centre point $\mathbf{C}([m] \in \Re^3)$ and the camera coordinate system defined by three base vectors $(\mathbf{i}, \mathbf{j}, \mathbf{k})$. These vectors are unit vectors forming a right-handed orthogonal system. In this specification, we follow a common definition of the camera orientation by using two vectors (viewing direction and up-vector).

Hence, the camera orientation is defined as follows:

$\mathbf{a} = (a_x, a_y, a_z)^T$, unit vector, specifies the direction of the optical camera axis and is equivalent with the local camera coordinate axis $\mathbf{i}$ in Figure 7.

$\mathbf{up} = (up_x, up_y, up_z)^T$, unit vector, specifies the upwards orientation of the camera. This is equivalent of $(-1)\mathbf{k}$ in Figure 7. Both vectors have to be perpendicular to each other!

This orientation describes a camera looking into the negative z-axis with an up-vector parallel to the y-axis as depicted in Figure 7.

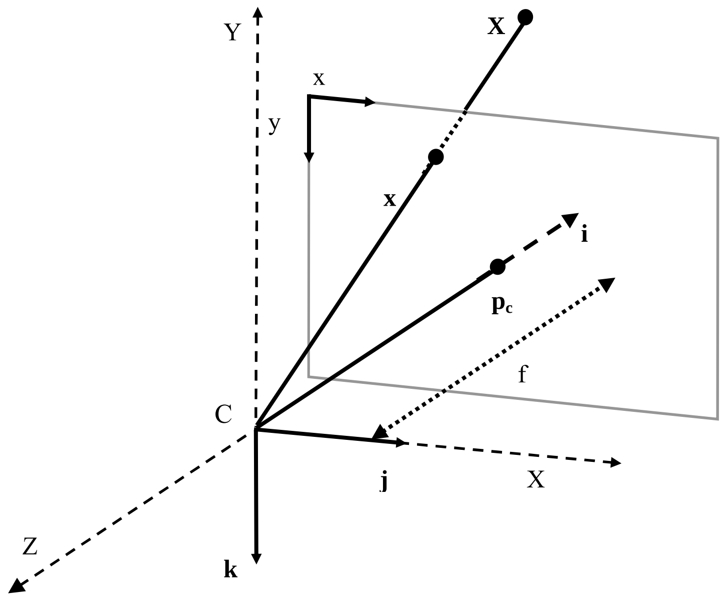

Figure 7: Default camera orientation

The resulting rotation matrix based on the direction vector $\mathbf{a}$ and the up-vector $\mathbf{up}$ is defined as follows:

$$R = \begin{bmatrix} (\mathbf{a} \times \mathbf{up})^T \\ \mathbf{up}^T \\ \mathbf{a}^T \end{bmatrix} \quad (11)$$

The camera position $\mathbf{C}$ refers to the translational shift of the camera with respect to the camera cluster coordinate system.

Lens distortion

In literature a rich variety of distortion models can be found, which apply to different types of lenses, e.g. [Brown, 1966], [Devernay, 2001] and [Ma, 2003]. Since modern broadcast cameras uses lenses, which cause only moderate pincushion and barrel distortion a more general fourth order polynomial lens-distortion model can be applied [Mallon, 2004]. Therefore the lens-distortion model is given by

$$\mathbf{x} - \mathbf{x}_c = (\mathbf{x}' - \mathbf{x}_c) / L(\|\mathbf{x}' - \mathbf{x}_c\|_2) \quad (12)$$

with $L(r) = \kappa_1 r + \kappa_2 r^2 + \kappa_3 r^3 + \kappa_4 r^4$ (13)

$\mathbf{x}_c$ is the centre of distortion and $\mathbf{x}, \mathbf{x}'$ denote the distorted and un-distorted image coordinates. Eq.(12) is used on one hand to estimate the distortion coefficients κ_1 - κ_4 during calibration. On other hand the same equation is used in order to perform the distortion correction. As the distortion correction has to be performed by back-ward mapping, i.e. the calculation of pixel values in the target image (the undistorted) from the original image (the distorted image), the distortion of undistorted pixel positions needs to be calculated.

Since the centre of distortion is almost identical with the principle point, we use $\mathbf{x}_c = (u,v)^T$ for simplification. However, in order to achieve high precision, an individual treatment for every colour channel is needed. Thus the four kappa values are estimated for each colour channel separately.

Chromatic aberration

For each wave-length, light takes a different way through a lens system. The distortion caused by this effect is known as chromatic aberration. Therefore the lens cannot focus all colours to the same convergence point. An example for the visual impact of chromatic aberration is shown in Figure 8.

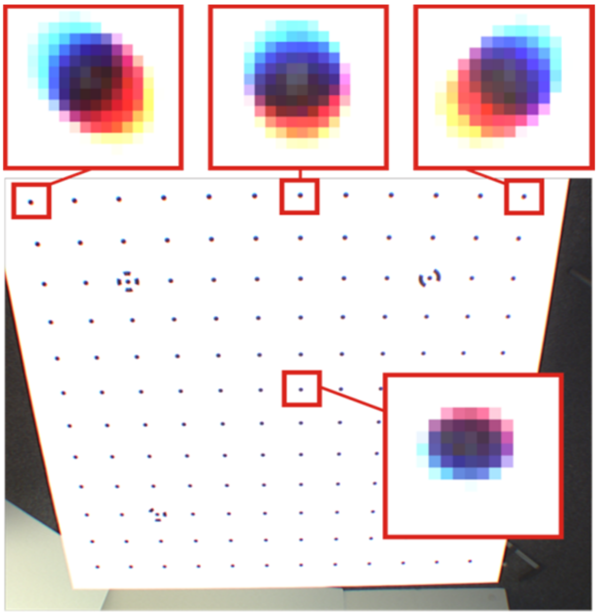

Figure 8: Examples for chromatic aberration distortion. Apparently, beside the radial lens-distortion, an offset and a different scaling for each colour channel constitute the main elements of distortion

In order to reduce distortion, diffractive optical elements or achromatic doublets can be applied. However, image post-processing leads to a good reduction as well. As the example in Figure 8 shows, the distortion mainly consists of a different scaling for each colour channel and a translational offset. Since the green channel is usually the most illuminated one, it is used in our framework for estimating the reference position for red and blue channel pixels. Thus the chromatic aberration correction emerges as a registration process that contains a colour dependent scaling factor s_C and a translation $\mathbf{t}_C$, with $C \in \{R,G,B\}$ respectively. Please note since the registration is done with respect to the green channel, $s_G = 1$ and $\mathbf{t}_G = (0, 0)^T$ by definition.

To conclude with the section about the camera model, we summarize the imaging process for a space point $\mathbf{X} = (X, Y, Z, 1)^T$ to a distorted image point $\mathbf{x}_C = (x_C, y_C)^T$, $C \in \{R,G,B\}$ of a certain colour channel. After the combination of the camera model components, the imaging equation is given by

$$\mathbf{x}_C = (s_C \mathbf{x}' + \mathbf{t}_C - \mathbf{x}_c) / L(\|s_C \mathbf{x}' + \mathbf{t}_C - \mathbf{x}_c\|_2) + \mathbf{x}_c \qquad (14)$$

with $(\mathbf{x}'^T, 1)^T = (x, y, 1)^T \sim P\mathbf{X}$. Since $s_G = 1$ and $\mathbf{t}_G = (0, 0)^T$ by definition, we finally end up with 3 + 6 = 9 linear parameters, 3 × 4 = 12 parameter for the lens-distortion model and 2 × 3 = 6 parameter for the chromatic aberration model, which is in total 27 parameters for each camera. In the following we refer to the 6 linear extrinsics as *extrinsics* and to the remaining 21 parameters as *intrinsics*.

Global parameters

In Table 5, global parameters are specified for each planar camera, which remains unchanged during the whole capturing process.

Planar Camera Parameters – Global				
Parameter	Symbol	Description	Units	Required = R, optional = O
Intrinsic parameters				
Width	N_x	Horizontal spatial resolution		R
Height	N_y	Vertical spatial resolution		R
PixelSizeX	s_x	Horizontal size of the pixel target		R
PixelSizeY	s_y	Vertical size of the pixel target		R
Primaries	(p_R,p_G,p_B)	Primaries for the three colour components		O
WhitePoint	WP	White point of the sensor		O
BlackLevel	BL	Black level of the sensor		O
Gamma	γ	Gamma of the sensor		O
BitDepth	bpp	Number of bits per pixel		R

Table 5: Planar camera parameters - global

Frame-based camera parameters

In the Table 6 below, the frame-based camera parameters are listed. These parameters may change with time depending on the specific camera.

Planar Camera Parameters – Frame-based				
Parameter	Symbol	Description	Units	Required = R, optional = O
Extrinsic parameters				
CVec	$\mathbf{c} = (c_x, c_y, c_z)^T$	Camera centre point		R
AVec	$\mathbf{a} = (a_x, a_y, a_z)^T$	Camera viewing direction		R
UpVec	$\mathbf{up} = (up_x, up_y, up_z)^T$	Camera up-vector		R
Intrinsic parameters				
FocalLength	f	Focal length	mm	R
CenterPointShift X	x_c	(metric) horizontal centre point shift on the image target	mm	R
CenterPointShift Y	y_c	(metric) vertical centre point shift on the image target	mm	R
DistK1(A,B,C)	$\kappa_1(A,B,C)$	1st order distortion coefficient vector for colour components (A,B,C)		O
DistK2(A,B,C)	$\kappa_2(A,B,C)$	2nd order distortion coefficient vector for colour components (A,B,C)		O
DistK3(A,B,C)	$\kappa_3(A,B,C)$	3rd order distortion coefficient vector for colour components (A,B,C)		O
DistK4(A,B,C)	$\kappa_4(A,B,C)$	4th order distortion coefficient vector for colour components (A,B,C)		O
ScaleChroma A	s_A	Scale of A component for chromatic aberration		O
ScaleChroma B	s_B	Scale of B component for chromatic aberration		O
ShiftChroma A	t_A	Shift of A component for chromatic aberration		O
ShiftChroma B	t_B	Shift of B component for chromatic aberration		O

Table 6: Planar camera parameters – frame-based

3.2.5 The model of a cylindrical camera

The panorama camera is defined as the reference coordinate system to which all the other views are mapped. This panorama camera can be either a planar camera or a cylindrical camera. In the case of a planar camera, the model of the planar camera in section 3.2.4 can be used. In the case of a cylindrical camera the following parameters are defining the imaging process.

For the cylindrical panorama, the origin coincides with the centre of the cylinder at position h/2 in vertical direction. The height of the cylinder is defined as h= N_y * s_y . The z-axis points along the radius perpendicular to the cylindrical plane. If the viewing angle in horizontal direction is θ, then the center of the cylindrical plane in horizontal direction is at $\theta/2$. The focal length of the cylindrical panorama is equal to the radius of the cylinder.

The orientation is again defined by using the direction vector $\mathbf{a}$ and the up-vector $\mathbf{up}$. The direction vector $\mathbf{a}$ coincides with the Z-axis in Figure 9, while the up-vector $\mathbf{up}$ relates to the Y-axis.

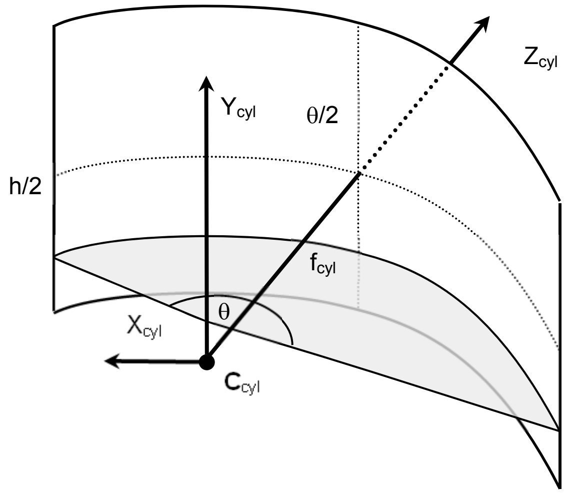

Figure 9: Cylindrical camera

Global parameters

In the following Table 7, global parameters are specified for each cylindrical camera, which remains unchanged during the whole capturing process.

Cylindrical Camera Parameters - Global				
Parameter	Symbol	Description	Units	Required = R, optional = O
Intrinsic parameters				
Width	N_x	Horizontal spatial resolution		R
Height	N_y	Vertical spatial resolution		R
PixelSizeX	s_x	Horizontal size of the pixel target	*mm*	R
PixelSizeY	s_y	Vertical size of the pixel target	*mm*	R
ViewAngle	θ	Viewing angle		R
FocalLength	f_{cy}	Focal length (the centre of the cylinder)	*mm*	R
CenterPointShiftX	x_c	(metric) horizontal centre point shift on the image target	*mm*	R
CenterPointShiftY	y_c	(metric) vertical centre point shift on the image target	*mm*	R

Table 7: Cylindrical camera parameters – global

Frame-based camera parameters

In the Table 8 below, the frame-based camera parameters are listed. These parameters may change depending on the specific camera.

Cylindrical Camera Parameters – Frame-based				
Parameter	Symbol	Description	Units	Required = R, optional = O
Extrinsic parameters				
CVec	$\mathbf{C} = (c_x, c_y, c_z)^T$	Camera centre point		R
AVec	$\mathbf{a} = (a_x, a_y, a_z)^T$	Camera viewing direction		R
UpVec	$\mathbf{up} = (up_x, up_y, up_z)^T$	Camera up-vector		R

Table 8: Cylindrical camera parameters – frame-based

3.2.6 Mapping rules

For any two cameras that are sufficiently close together that parallax effects can be ignored (closer than the smallest length easily discernable on the nearest object in the image, assuming that the most distant object is significantly further away), it is possible to determine a unique pixel-based mapping between the images based on the calibration information outlined above. In this situation, the mapping of any camera view into the panorama reference system can be performed using the following steps:

- Geometrical distortion correction for each colour channel of each camera view
- Geometrical mapping:
  - Mapping of satellite camera plane to tangential plane of the panorama reference system
  - Mapping from tangential plane to cylindrical plane, if the panorama reference system is defined as a cylindrical panorama (non-linear mapping)

In many practical situations, however, the camera systems will be located too far apart to allow a depth-independent mapping to be carried out. This situation arose at the project’s first test shoot, and was discussed in Section 5.3.1 of D2.2.1. In this situation, it is impractical to attempt to produce a pixel-wise mapping between the images, as even with the use of sophisticated depth-aware processing, there would be regions of one camera image that were not visible in the overlapping part of the other image due to occlusions. The best that can be achieved in this situation is to identify the most subjectively-important object in the satellite camera view, and extract a window from the panorama so that this object appears in the same place as it does in the image from the satellite camera, and then perform a blend or similar kind of transition from the panorama to the satellite camera to present the viewer with a higher-resolution view. This ensures that the object on which the viewer’s attention is focused will not appear to move during the transition. This requires knowledge of the depth of the object of interest: either explicitly using approaches such as multi-camera object tracking and triangulation, or implicitly through identifying where the same object of interest appears in both the satellite camera and the panoramic camera. This can make use of the metadata derived by the automated metadata extraction tools, described in D3.3.1. The investigation of approaches is on-going and results will be presented in *D2.2.2 Implementation of preliminary capture test bed* due in M27 and other WP2 deliverables.

In the ideal situation, where the camera systems are essentially coincident, the distortion correction and mapping processes described below may be used.

Geometrical and photometric distortion correction

In this step, the geometrical distortion of each colour channel is corrected, which is defined by Equ. (14). This correction contains the chromatic aberration as well as the pincushion and barrel distortion resulting from the lens and the CCD chip.

Linear mapping onto a tangential plane

This step performs a mapping from any satellite camera view onto the tangential plane of the panorama reference system. As stated above, the key assumption in this step is that the perspective difference between any satellite camera view and the panorama reference system is negligibly small. In this situation, the registration and mapping can be performed by a simple linear transformation, called homography.

Here, two different cases need to be distinguished. If the panorama reference system is a planar panorama then this mapping represents the last mapping step.

If the panorama reference system is a cylindrical panorama then another non-linear transformation needs to be applied in order to map the tangential plane onto the cylinder. The mapping from a satellite camera view via a tangential plane onto the panorama reference system is displayed in Figure 10.

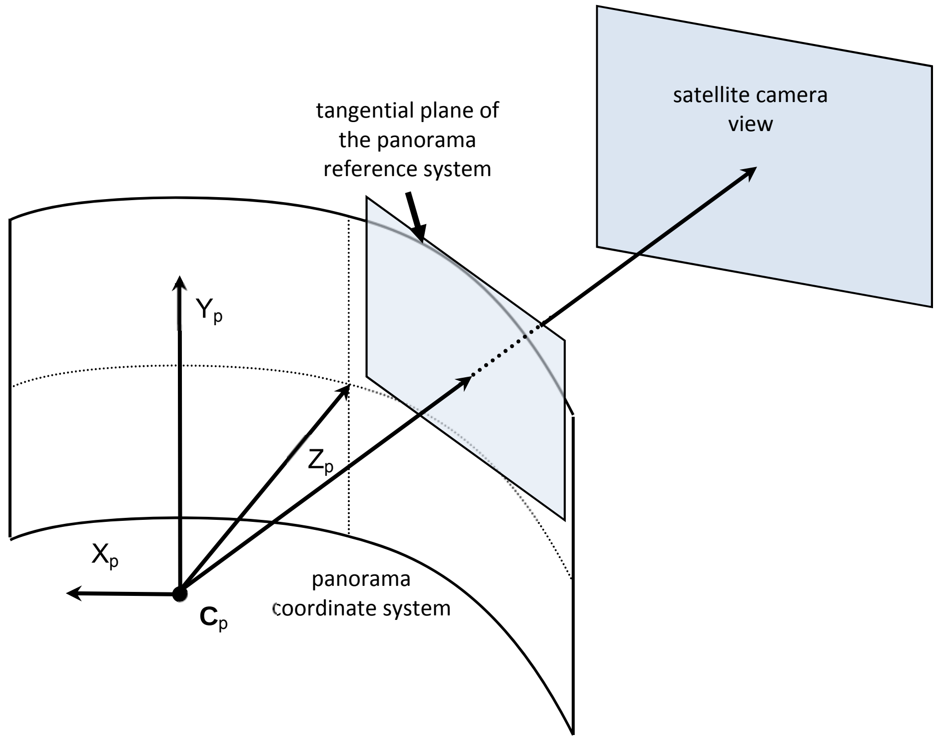

Figure 10: Mapping of a satellite camera view to the tangential plane of the panorama reference system

The mapping from one plane to another in the projective space $\mathcal{P}^2$ is defined by the following linear transform:

$$\tilde{\mathbf{m}}_2 = H\tilde{\mathbf{m}}_1 \tag{15}$$

The matrix H represents an affine transformation, which allows scaling, shearing and translational shift. The coefficients of the matrix need to be defined during the registration process between the satellite view and the final panorama reference system. In Figure 11, the warping process is visualized.

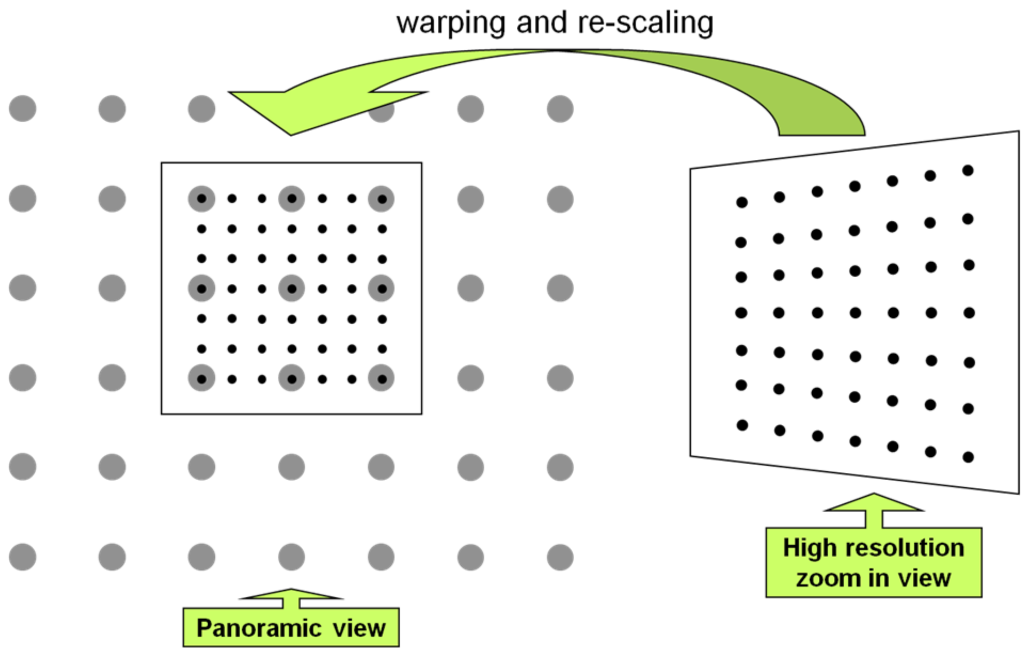

Figure 11: Linear warping from a satellite camera view onto the panoramic reference view

Non-linear transformation onto the cylinder

In the case of a cylindrical panorama, the mapping result of the previous step needs to be transformed onto the curved screen of the panorama reference system. Therefore, the image on the tangential plane needs to be distorted in a way that straight lines are preserved in the projected image.

The concept of FascinatE foresees a panoramic view, which is enriched by visual information from satellite cameras. As we want allow enrichment of any part of the panoramic scene, no assumptions on the orientation of the tangential plan with respect to the panoramic circle can be made.

The mapping along the x-coordinate can be derived due the following geometrical relationship shown in Figure 12.

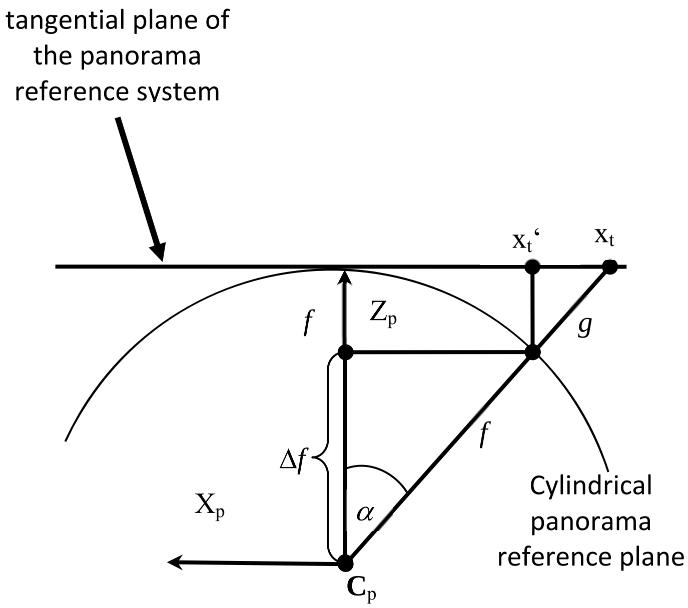

Figure 12: Mapping the horizontal coordinates of the tangential plane to the curved screen of the panorama reference system – top view onto the cylinder

The horizontal angle α can be calculated based on the x-coordinate x_t and the focal length i.e. the radius of the cylindrical panorama as follows:

$$\alpha = \operatorname{atan}\left(\frac{x_t}{f}\right) \tag{16}$$

In order to calculate the new position x_t', the length Δf needs be known. By exploiting the angle α, we end up with

$$\Delta f = \cos\alpha \cdot f \tag{17}$$

The new position finally results in

$$x_t' = \sqrt{f^2 - \Delta f^2} = \sqrt{f^2 - (\cos\alpha \cdot f)^2} = \sqrt{f^2 (1-\cos\alpha)^2} = f\sqrt{(1-\cos\alpha)^2} = \\ = f \cdot \sin(\alpha) \tag{18}$$

The missing distance g on the projection ray from the centre of the cylinder to the point x_t on the tangential plane is required for the calculation of the vertical projection and can be derived as follows:

$$g = \frac{f - \Delta f}{\cos\alpha} = \frac{f - (\cos\alpha \cdot f)}{\cos\alpha} = f\,\frac{1-\cos\alpha}{\cos\alpha} \tag{19}$$

The mapping in y-direction also depends on the angle α. The new y-coordinate y_t' can be derived due the following geometrical relationship shown in Figure 13.

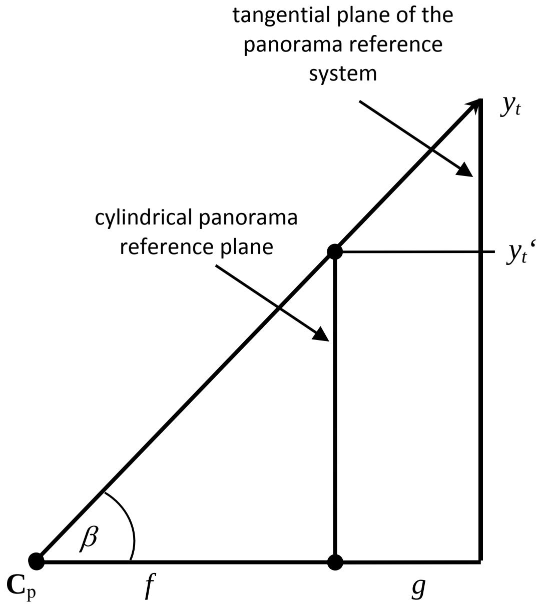

Figure 13: Mapping the vertical coordinates of the tangential plane to the curved screen of the panorama reference system – side view

$$y_t' = f\,\frac{y_t}{f+g} = f\,\frac{y_t}{f + f\,\frac{1-\cos\alpha}{\cos\alpha}} = \frac{y_t}{1 + \frac{1-\cos\alpha}{\cos\alpha}} = y_t \cdot \cos\alpha \tag{20}$$

In the Figure 14, a regular point grid is shown (left), which needs to be distorted as presented on the right in order to preserve the correct display on a cylindrical plane.

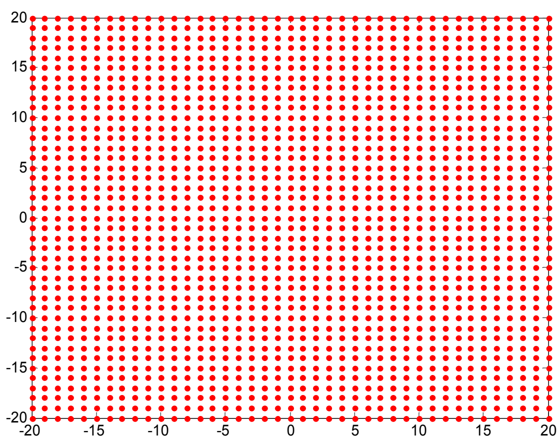

Figure 14: Regular point grid (left) and distorted point grid required for correct cylindrical projection (right)

3.3 Audio Scene

An audio scene in the context of FascinatE can be considered as a combination of background/ambient sound components also known as a sound field and a collection of spatially discrete sound sources also known as audio objects. An audio object describes a sound source of which the content, position, onset time and duration are known. Broadcasting a stream of audio objects with an accompanying sound field allows a spatial mix at the render end rather than at the production end so any audio system can be accommodated from a simple stereo system right up to a 3D sparse loudspeaker set using higher order ambisonics, a line of loudspeakers using wave field synthesis and even reproduction systems not yet in use. For the more complex reproduction systems, broadcasting audio objects and sound fields rather than loudspeaker signals is of particular importance as the number of loudspeakers used for pure loudspeaker systems, like 22.2 and more, could potentially resulting in a huge demand on bandwidth. The use of audio objects and sound fields also mean a truly format agnostic broadcast as the loudspeaker signals are derived at the user end and the user can more or less freely choose the loudspeaker layout and position. This is not true for pure loudspeaker systems.

Audio Objects

Audio objects can be loosely grouped into two main categories: explicit and implicit. The grouping of audio objects depends on the method and accuracy of capture. A brief description of each type can be found below.

Explicit audio objects (EAO) are objects that directly represent a sound source and have a clearly defined position within the coordinate system. This could include a sound source that is recorded at close proximity either by a microphone or by a line audio signal, and is either tracked or is stationary with defined coordinates. An example of an explicit audio object could be instruments close miked in a performance which have little or no crosstalk from other sources and whose position with respect to time is known. Close miking and position tracking however are not always possible in practice – as in the case of football where individual players cannot be close miked. Thus, not all audio objects are explicit.

Implicit audio objects (IAO) therefore, represent sound sources in a more indirect manner; these could include signals that are picked up by more distant microphone techniques or by microphone arrays where the source of sound is distinct from the receiving device or where the audio object may be derived from several recording sources. Example: in a football match the sound of the ball being kicked is picked up by one or more shotgun microphones around the pitch, the content and position of this audio object therefore has to be derived from these microphone feeds. In this instance the ball cannot be tracked so areas of the pitch are defined as implicit audio objects that are either active or inactive depending on whether there is relevant sound activity in that region at any given time. More information on the techniques used for the recording of football matches is outlined in the FascinatE deliverable D5.1.2 and in [Oldfield, 2011].

Sound Fields

Ambient sound fields can be captured using microphone arrays, such as the SoundField® or Eigenmike® microphones or can be generated out of individual sound signals [Batke, 2010]. The recorded sound field represents the incident sound energy from all principal directions at one analysis point. A sound field can conveniently be represented using spherical harmonics in a technique called ambisonics. The number of capture devices and the number of microphone capsules used for the recording determine the resolution of the recorded sound field and hence the ambisonics order. For the computation of valid ambisonics coefficients the order N has to fulfil (N + 1) ^2 ≤ O, where O is the number of capsules, which results in a maximum order of N = 1 for the SoundField® and a maximum order of N = 4 for the Eigenmike®.

So the recorded audio scene can be considered to be a combination of audio objects and sound field components (this could additionally include one or more commentary, and vision as well as hearing impaired feeds). Unlike the video scene where several camera clusters make up the complete scene, there is no necessity to additionally split the audio scene into different audio clusters. The reason for this is that camera clusters are distinct in their view point and viewing direction onto the scene and therefore cover different regions. The nature of the audio issues within the FascinatE project however are such that the audio scene is not directly linked to any particular camera cluster and the audio is captured from a location independently. The audio scene will be captured in a holistic manner for each complete scene and links with the camera cluster currently selected by the user and will be carried out at the rendering stage.

3.3.1 The geometry of the audio scene

The coordinate system should support a 3D metric system to be able to take the sound propagation delay into account. The OpenGL system has been discussed, however the mapping to the metric system and the mapping between a 2D Video and 3D Audio System is not obvious. As an alternative, VRML was proposed for FascinatE, specifically MPEG-4 Advance Audio BIFS V.3 (BInary Format for Scenes). It is a right-handed Cartesian coordinate system with the origin in the centre of the visual screen. The x-axis is oriented to the right side. The BIFS ‘Transform3DAudio’ node has 2D input vectors and 3D output vectors and fields for the completion of the 2D input vectors to 3D output vectors for the location, rotation, scale, orientation and translation fields. It can therefore perfectly be used in 2D visual scenes together with associated 3D audio [Schmidt 2004].

As defined in the previous section, a set of microphone signals are used to represent the audio scene in terms of sound fields and audio objects. The complete audio scene has its own origin, which is assigned by the audio scene coordinate system. This audio scene will consist of multiple audio objects with different positions in space but all of them related to the origin of the audio scene coordinate system.

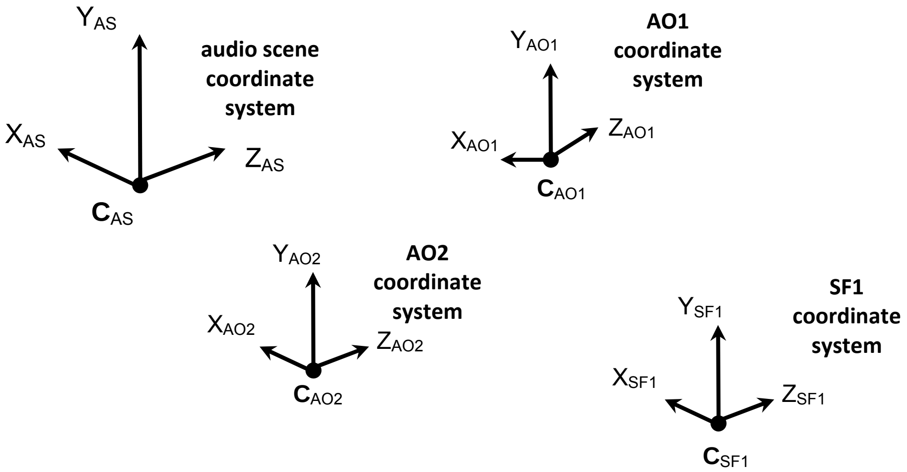

Figure 15: Definition of microphone coordinate systems related to the audio scene coordinate system

The header of the audio scene describes the number and type of audio objects in this particular scene. The related geometry parameters are listed in the Table 9 below.

Audio Scene Header				
Parameter	Symbol	Description	Units	Required = R, optional = O
NumEAO	n_{EAO}	Number of explicit audio objects in the scene		R
Translation EAO	$\mathbf{t}_{EAO \to AS}$	Translation of explicit audio objects (if static) against the audio scene coordinate system	*mm*	O
EAODirection	$R_{EAO \to AS}$	Rotation of explicit audio objects (if static) against the audio scene coordinate system	*Radians*	O
NumMicsIAO	n_{IAO}	Number of microphones from which to derive the implicit audio objects (the number of implicit objects varies with time)		R
NumSF	n_{SF}	Number of sound fields in the scene		R
Translation SF	$\mathbf{t}_{SF \to AS}$	Translation of Sound Field against the audio scene coordinate system	*mm*	R
Rotation SF	$R_{SF \to AS}$	Rotation of Sound Field against the audio scene coordinate system	*Radians*	R

Table 9: Parameters of the audio scene header

3.3.2 Sound descriptions

In FascinatE, two principal approaches are used to describe an audio scene as described in [Batke, 2011], these descriptions are based on the higher order ambisonics (HOA) and wave field Synthesis (WFS) and techniques.

Higher order ambisonics (HOA): This first approach is to describe the sound field of the audio scene using an ambisonics description. The necessary ambisonics coefficients are taken from microphone arrays or are calculated for a synthetic sound scene. If a natural sound field is acquired no modifications of the audio scene are possible although the sound field can be retrospectively rotated to match the video scene. Using this technique, the elements of the audio scene are first encoded into an ambisonics B-format signal, this encoding can contain both the audio objects and the sound field component. At the user end this B-format signal can then be decoded according to the user's loudspeaker setup. The recorded sound field can be rotated and potentially zoomed into and out of as the user navigates the video scene.

Wave field synthesis (WFS): The second approach is to describe all elements (audio objects) of an audio scene, i.e. the source signals, the positioning of the sources and if required also the acoustical environment of the audio scene. In this case the sound field component of the audio scene should be decoded into plane wave components and rendered as plane waves in the WFS renderer. Audio objects can be rendered as point sources in the WFS renderer. The relative positions of the point sources and the angles of the plane waves can be adjusted as the audio scene rotates to match the navigation of the video scene

The elements of an audio scene may be estimated from the audio information if the parameters of the microphones (positioning, directivity, etc) are known. However, the quality of this information is highly dependent on the microphone setup. This parametric approach allows modifications of the audio scene. A drawback is that dry sound sources are required, i.e., the sound source signals do not contain any spatial sound information. This spatial sound information (reverberation, echoes, etc) can be added as an extra sound effect when the audio scene is rendered. In practice, typical microphone signals are often wet. i.e. they contain the dry and acoustical signal components. However they are treated like the dry signals as described above.

(Listener) Sound field descriptions

An Ambisonics sound field representation describes the sound field to be reproduced at the position of the listener.

An Ambisonics sound field description is characterised by a set of following metadata parameters:

- Acquisition Position(s)
- Orientation (of Microphone array / Coordinate system)
- Spatial dimension 2D or 3D
- Ambisonics order, i.e., number of channels

Sound source (audio object) descriptions

Sound sources can be recorded with close-up microphones or using direct instrument outputs.

A sound source is characterised by a set of following metadata parameters:

- Position (fixed or time varying parameter)
- Onset/offset times
- Direction (fixed or time varying parameter)
- Directivity Pattern
- Optional: temperature, air absorption, velocity of sound

Audio sources without position or orientation

Audio sources do not always have a specific spatial origin in, e.g. commentary, background music, etc. In this case the metadata parameters could be:

- Language
- Details of commentator or composer etc
- Onset/offset times

Microphone descriptions

Microphone positions and directions can be used to estimate real sound source positions. Producers should try to capture/estimate sound source descriptions instead of microphone descriptions. If sound source descriptions cannot be provided, microphone descriptions give at least some hints for the audio object extractor and renderer about the real sound source behaviour.

It is anticipated that a range of microphone types will be utilised in capturing the audio scene representation. The type and number of devices used to capture the sound of an event is anticipated to vary considerably depending on the nature of the event, however it may be possible to define template setups that can be applied to more than one scenario. Each microphone position will be identified as a single microphone or audio sound source object although in some cases the capturing device may be a multichannel device utilising more than one microphone capsule (e.g. Coincident stereo pair or sound field microphone).

Audio capture devices that are to be assessed as part of WP2 and considered likely to feature in implementation are as follows:

- Static multi-capsule sound field microphone (SoundField® or Eigenmike®)
- Static and mobile shotgun (hypercardioid) microphones
- Static coincident stereo pairs of cardioid microphones
- Static omni microphones
- Mobile omni microphones (e.g. referee microphone in rugby coverage)
- Possible other microphone arrays to be confirmed based on user requirements

An audio capture device is characterised by the following set of metadata parameters:

- Position (fixed or time varying parameter)
    - For mobile microphones (e.g. shotgun microphones running touchline at a rugby match) this may consist of a vector descriptor setting the boundaries of that microphone's movement within the audio scene.
- Direction (fixed or time varying parameter)

- Number of capsules
- Capsule arrangement (omni, pairs like X-Y, A-B or ORTF, circular, spherical, line/flat array)
    - Positioning
    - Directivity / Polar pattern (e.g. filter for various directions and frequencies)
- Transport format (mono, stereo, surround, B-Format, etc.)

3.3.3 Listening Point

The audio scene should be constructed to match the desired framing of the shot, but will not necessarily adapt continuously to changes of field-of-view or choice of camera cluster.

A listening point is characterised by a set of following metadata parameters:

- Position
- Direction
- Binding with visual orientation and zoom

If there is no Listening Point given in a scene, the apparent listener position will become the active View Point.

4 Coding Scheme

The objective of this section is to describe the coding schemes used for the audio and video data that come out of the production system.

In order to motivate this discussion on the coding schemes required, we first need to describe some assumptions on the roles of the various system components in the end-to-end chain.

- A first assumption is that the production system outputs A/V signals in their *"raw"* form, meaning that all the A/V processing blocks (for instance the functions of warping multiple video signals in a common reference frame and possibly blend them) are left for a rendering functional block located further in the chain. As a result, each audio/video layer coming out of the production system is made available in its native format. The rationale of this assumption is that the quality of the rendered scene at the terminal device will be maximized if the rendering function is fed with the A/V signals in their most original form.
- A second assumption is that the first functional block, which follows the production system for delivering the produced A/V data, is an "*A/V Ingest"* node. Its main roles are the following:
    - o Perform some basic A/V processing: for instance to tile large video frames, to change video resolution, etc.
    - o For each A/V layer to transport, perform some compression, possibly at multiple bitrates
    - o Initiate A/V transport: A/V data encapsulation and packetization

The processing done in the "A/V Ingest" essentially aims at facilitating the delivery mechanisms and can be steered by the production scripts (e.g. describing ROI, how they evolve over time, semantic links between shots, …). But, in principle, it should not interfere with the processing required at production and rendering sides.

Also note that this "A/V ingest" function is not seen as part of the production system, but as the first block of the delivery system. In practice, its actual implementation will typically depend on the nature of the service provider (e.g. as an IPTV head-end of a telecom operator, or as video servers of an over-the-top service)

The scope of the present discussion is on the A/V data that flow from the production system to this A/V ingest node. A nice overview of the current state of the system specification of the FascinatE system can be found in *D1.4.2 Interim System Specification*. The coding schemes used for the delivery itself are mainly studied in WP4 and discussed in D4.4.1 and will be updated in D4.4.2. As explained above, we assume that the production systems output A/V data in their most original form, and thus ideally without (lossy) compression. However, in order for the system to scale with the order of magnitude of the data rate required by the FascinatE usage scenarios, it is clear that some compression must be performed. As a simple example, consider a FascinatE scene made of an OMNICAM (6k x 2k) input and the equivalent of 10 High-Definition camera feeds. Given that a raw 1080p video stream (24bit RGB at 30fps) requires around 1.24Gb/s, the transmission of the whole captured video content (equivalent to 16 1080p streams) would require around 20Gb/s!

Looking again at the whole system, A/V data can potentially go through a two-stage compression scheme: the first between production system and A/V ingest, and the second from the A/V ingest on for the actual delivery. There is thus a natural analogy with the traditional contribution/distribution compression cascade, which happens in traditional broadcasting systems. The role of the contribution network is to transport A/V content at almost lossless quality, while the distribution network requires much more severe compression rate, thus leading to two very different sets of coding requirements. In the following subsections, we therefore discuss the coding schemes that can compress the production A/V output at a "contribution-like" quality.

4.1 Video Coding

Many criteria can be considered to choose the most suited video compression framework. Among them, one can cite

- Compression efficiency, which is influenced by the quality/bit rate operating point required by the application

- Support for specific video format, such as the colour space (e.g. RGB, YUV), the type of chroma sampling (e.g. 4:2:0, 4:2:2, 4:4:4), bit depth per sample (8, 10 or 12 bit)
- Scalability; that is the ability to extract from the video bit stream a representation of the video at a lower temporal or spatial resolution, at a lower fidelity, or within a given Region-of-Interest (ROI).
- Random access that is the ability to access and start the decoding at any frame. When scalability is present, one can also desire random access at any lower layer representation.

We present in the following paragraph a brief discussion on the video compression technologies under consideration for the A/V output of the production system, as well as some first insights on the compression performance one can expect for the FascinatE production video streams.

4.1.1 Very short overview of state-of-the-art video compression technologies

In order to compress the video stream, one can basically use the two following approaches

- Either chooses a still image compression technology to encode each video frame independently.
- Or choose a inter-frame compression technology, that can exploit the redundancy between consecutive frames

The former has the advantage to naturally possess random access capabilities, as there are no inter-frame coding dependencies, while the latter has the advantage to generally yield better compression rates, although several sources report that the advantage of inter-frame compression vs. intra-only tends to decrease with higher resolution and higher bit rate, as will be discussed below.

The latest-generation technical solutions for these two approaches are respectively the JPEG 2000 and H.264 standards.

- JPEG 2000 is a wavelet-based compression standard for still images. Motion-JPEG 2000 is specified as Part 3 of the JPEG 2000 standard and provides the necessary support to independently encode each frame of a video stream using the core image compression technology of JPEG 2000. A key benefit of JPEG 2000 is its inherent support for scalability (spatial, quality, ROI). Although JPEG 2000 has known little success so far in consumer-grade applications requiring still image compression, JPEG 2000 has been chosen by the Digital Cinema Initiatives (DCI) as the main video coding standard for Digital Cinema applications.

  Note that other wavelet-based compression schemes were developed at the same time as JPEG 2000. One of them, called Progressive Graphics File (PGF), was developed with a prime focus on compression speed. While its compression performance approaches JPEG 2000, PGF encoding and decoding times are reported to be up to 10 times faster than JPEG 2000 [Stam, 2002]. Little is known about commercial utilization of Motion-JPEG 2000.
- AVC/H.264 is the latest video compression standard by the ITU-T Video Coding Experts Group (VCEG) together with the ISO/IEC Moving Picture Experts Group (MPEG). It can be seen as the successor of MPEG-2 as the de facto standard for the video broadcast and video entertainment (e.g. Blu-Ray) industry. Although the basic profiles of H.264 have no support for scalability, the standard has a recent extension, known as Scalable Video Coding (SVC), with support for temporal, spatial and quality scalability. SVC comes with a limited cost in encoding complexity and compression overhead.

Many studies in the literature provide comparison between H.264, H.264 restricted to intra-coding and (Motion-) JPEG 2000. Their results are typically difficult to aggregate into general conclusions, as the encoding software used, the choice of parameters and other testing conditions can vary greatly between papers. We cover three references hereafter.

In [Marpe, 2004], a comparison between Motion-JPEG 2000 (M-JPEG 2000) and H.264 (Main Profile) in pure intra-mode is performed. The authors conclude that H.264 Intra usually outperforms M-JPEG 2000 at lower resolutions (up to SD) and bitrates and in particular for interlaced content, whereas M-JPEG 2000 offers better compression for higher resolutions (from 720p or 1080p) and higher bitrates.

More recently, [De Simone, 2007] gives an overview of previous studies that globally claim a superiority of H.264 for monochromatic images and of JPEG 2000 for colour image. The authors conduct their own comparison of JPEG 2000 against H.264 Intra 4:4:4 Profile for high-resolution (2k x 2k up to 4k x 2k) images. Although this is not the case for all test images, JPEG 2000 tends on average to outperform

H.264 Intra, again especially at higher bit rate. The results hold both for PSNR and MSSIM quality metrics.

The above papers restrict the use of H.264 in intra-mode and thus do not “allow” H.264 to benefit from its inter-frame compression tools that are based on the traditional motion estimation/compensation loop. In [Smith, 2004] Intra-frame JPEG 2000 is compared against inter-frame H.264. Although a very limited set of sequences is tested, their study tends to show that the gain of inter-frame compression over intra-only becomes negligible for very high resolution (e.g. 4K) and very high bit rate (e.g. above 1 bit per pixel).

As already discussed, we need here to ensure a sufficiently high quality of the video after this first stage decompression, in order to avoid a propagation of errors in the second-stage compression and video processing/rendering. Thus image-based compression schemes, such as JPEG 2000 or PGF, without motion estimation/compensation loop, could be good candidates next to the traditionally used video codecs, such as H.264.

4.1.2 Preliminary comparison of compression performance for beyond-HD video content

To complement the studies cited above, this section gives a first overview of the compression performance one can expected when dealing with very high-quality (lossless or nearly lossless) and very high-resolution compressed video. We focus on these working conditions because, when coding production video, one primarily aims at maintaining high fidelity, rather than requiring maximal compression performances.

As discussed above, several technologies are available to perform video compression: frames can essentially be compressed independently, using any image-compression technology or any video-compression technology (in intra mode), or they can be compressed using inter-frame compression.

In the compression tests performed so far, our primary objective is to have a first insight on whether these compression technologies are all valid candidates for future research work in this project or, on the contrary, have significant disparities in performance. In the comparisons we focused on specific compression techniques: a (wavelet-based) image compression technology and a (DCT-based) video coding technology (both in intra and inter):

- PGF was chosen as the image compression format to be evaluated. We used the libPGF library to perform the encoding and decoding tasks. It was used instead of JPEG 2000 for its claimed faster speed of execution, while maintaining compression in the same range of performance.
- H.264 was chosen as the plain video compression format. In particular, we used the x264 encoder and the libavformat library for the decoding task.

In order to already capture some specificities of FascinatE at this stage of the project, the tests described below were performed on content acquired by the OMNICAM set-up developed at the Fraunhofer Heinrich-Hertz Institute, see the press release at [TiME-Lab]. As illustrated in Figure 16, the content used is the output of the processing of 6 HD cameras, resulting in a 6000x2050 panoramic video. Content-wise, this panoramic video fragment can be characterized by a fixed background covering a large portion of the frame with a limited number of moving foreground objects and people.

Figure 16: Sample frame of OMNICAM 6000x2050 video

Whereas image compression formats are usually designed to operate at such large resolutions, 6000x2050 is actually above the maximum resolution allowed by the highest level (level 5.1) of the H.264 specification. We therefore tiled the video along its width so as to have two set of tiles of resolution 3000x2050 before the H.264 encoding, while PGF compression was directly performed over the entire frames.

Another major difference between the PGF and H.264 encoders used is that the former operates directly on RGB4:4:4 images, while the latter assumes a pre-conversion in YUV4:2:0. This has an impact on our method to compute PSNR. All PSNR values shown below have been computed in the YUV space. For the PGF tests, this requires to first convert the original frames and the reconstructed compressed frames into the YUV4:2:0 representations before computing their PSNR values.

Figure 17 depicts portions of the rate-distortion curve for PGF, H.264 intra and H.264 inter at high fidelity points, i.e. targeting a PSNR in the 40dB-50dB range. It shows that H.264 (both intra and inter) seriously outperforms PGF. Figure 18 depicts the same results with a focus on H.264 performance and shows that, even at such high resolution and fidelity, motion estimation still plays a significant role and outperforms intra-only compression schemes.

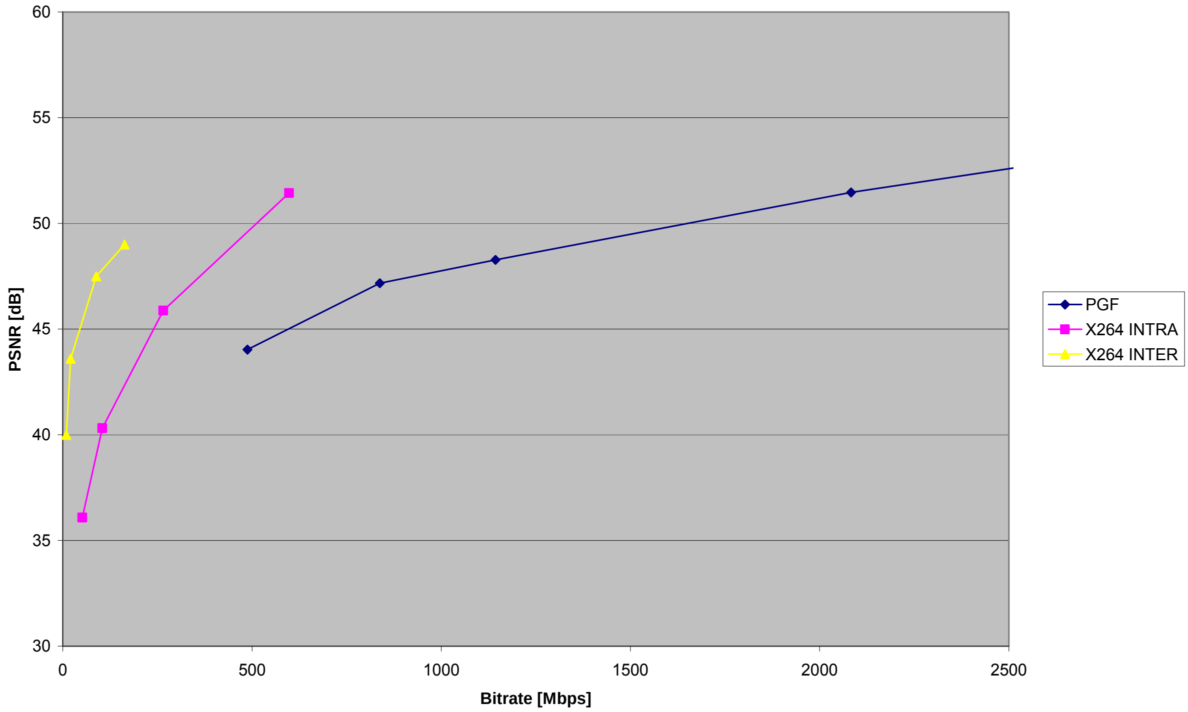

Figure 17: Comparisons of PGF, H.264 intra and H.264 inter compression

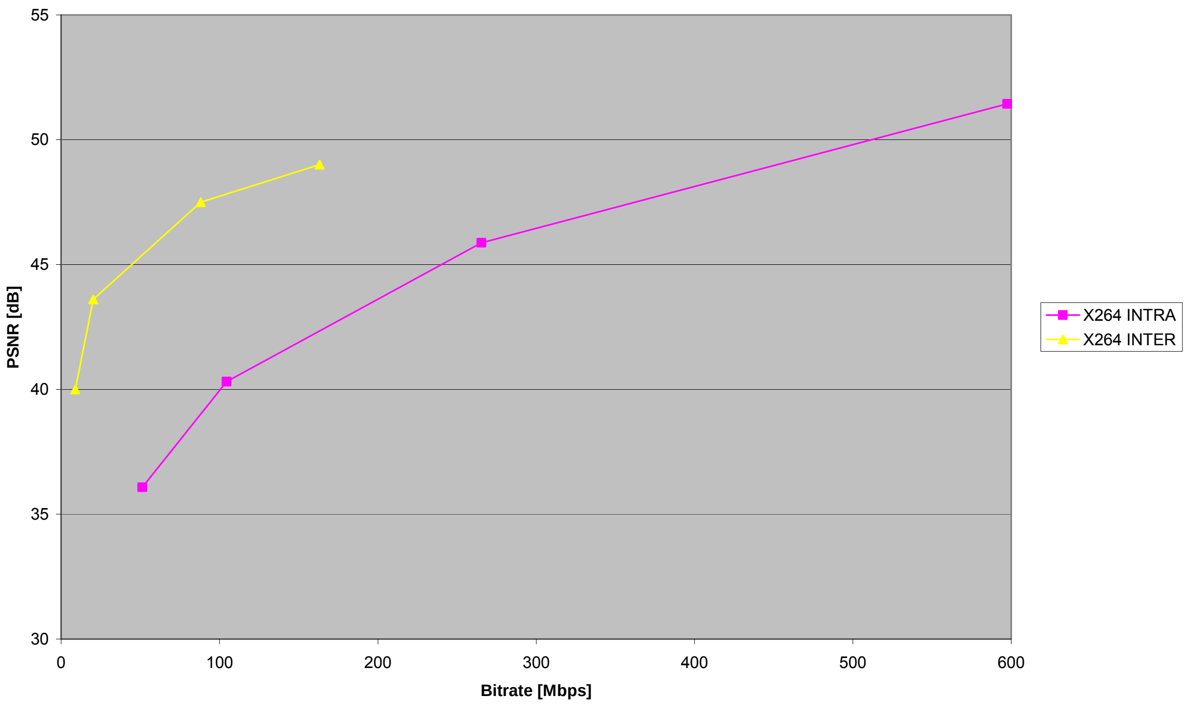

Figure 18: Zoom of Figure 17 on H.264 results

Finally, by pushing the fidelity requirement to the extreme, one ends up with lossless compression. Again, remark here that the H.264 video encoder used operates on YUV4:2:0, while the image encoder operates on RGB4:4:4, containing twice as much raw data as YUV4:2:0. Therefore, to have a relevant view on the lossless compression performance, the results are separated depending on the colour format used by the compression scheme. In Figure 19 and Figure 20, one can compare the bitrates required for the uncompressed video vs. the bit rate after lossless compression for encoder operating in RGB4:4:4 and YUV4:2:0 respectively. In Figure 19, next to the PGF results, lossless compression with JPEG 2000, using the OpenJPEG library, is reported and shows that JPEG 2000 can here compress by a ratio of 3, i.e. around twice as much as PGF (compression ratio of 1.46). In Figure 20, H.264 Intra has similar compression performance with respect to the uncompressed YUV4:2:0 (compression ratio of 3.2), while H.264 inter again provides some additional compression (compression ratio of 4.05)

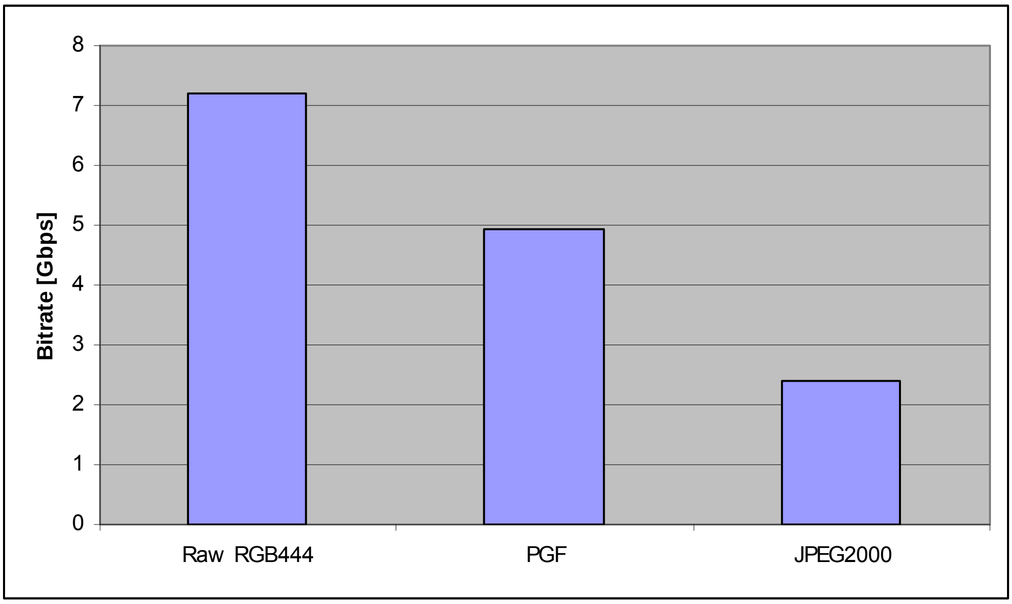

Figure 19: Bit rate of Raw RGB4:4:4, lossless PGF and lossless JPEG 2000

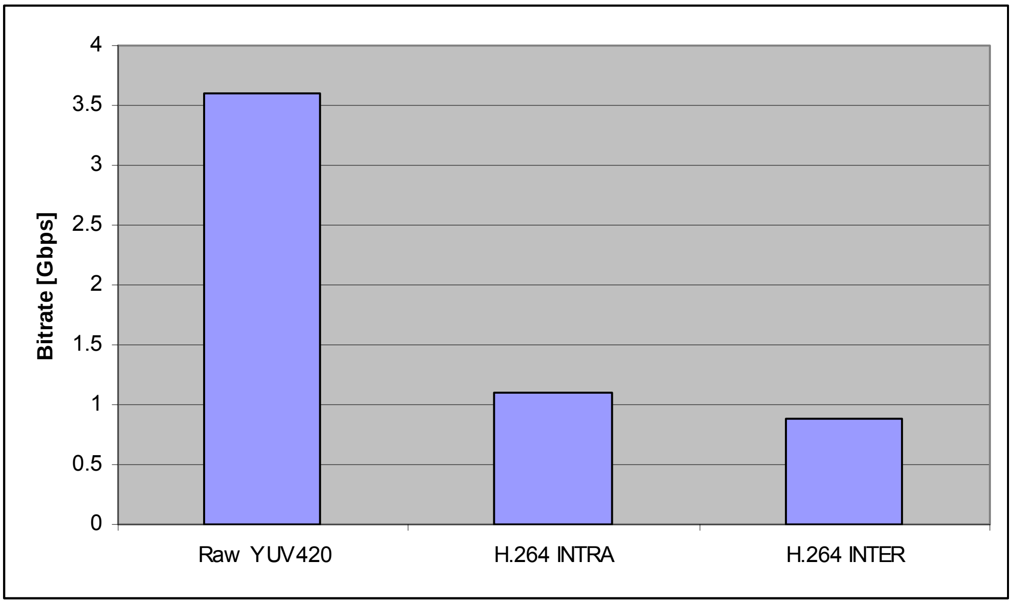

Figure 20: Bit rate of Raw YUV4:2:0, lossless H.264 intra and lossless H.264 inter

From the results reported in this section, one cannot yet draw definitive conclusions on the coding format best suited for the production video flows. However we can make a few observations and proposals for the upcoming investigations:

- Scalable image compression, in particular PGF, seems to have compression performance significantly below H.264 (including in intra-only mode) even at beyond HD resolutions and high-fidelity.
- From the first compression results performed with JPEG 2000 (so far only at lossless mode), it seems that better compression ratios can be obtained, approaching the performance of H.264 intra. But this needs further evaluation addressing in particular a broader range of fidelity points, and operating in a common colour format.
- At lossless compression, the compression gain of H.264 inter vs. intra-only seems lower than for lossy compression. But note that the rate-distortion performance are so far only measured in terms of PSNR, which may not be the most appropriate metric at these high-fidelity operating points. Other metrics, such as SSIM, could be evaluated as well.
- For other functions of the FascinatE end-to-end chain, such as delivery and rendering , some specific requirements (to be defined in D1.1.1) in terms of decoding complexity, scalability, random access, … will also influence the choice of the coding format.
- An extension of these results including the new emerging High Efficiency Video Coding (HEVC) standard can be found in D2.2.1.

4.1.3 Video coding parameters

Table 10 provides a preliminary list of coding parameters which should be part of the video coding specifications. Many of them are optional because

- Either they can, by default, be deduced from the camera parameters (resolution, colour space…). But still, they might be modified prior to the compression.
- Or, they can be deduced from the video coding syntax itself.

Also note that the tiling scheme proposed here implicitly assumes tiles of constant size to be processed in a predefined order (e.g. raster-scan). But more complex tiling could be specified as well.

Video Coding Parameters				
Parameter	Symbol	Description	Units	Required = R, optional = O
Format	F	Video coding format : H.264, JPEG 2000, Raw		R
PictureSizeX	P_x	Horizontal size of each frame, if modified by coding scheme		O
PictureSizeY	P_y	Vertical size of each frame, if modified by coding scheme		O
NumTileX	N_x	Number of columns of the tiling		R
NumTileY	N_y	Number of rows of the tiling		R
TileSizeX	T_x	Horizontal size of each tile (required if cannot be derived from the syntax of the coding format)	pixel	O
TileSizeY	T_y	Vertical size of each tile (required only if $T=1$)	pixel	O
CodingColourSys		Colour system, if modified by coding scheme (RGB444, YUV444,YUV422, YUV420,…)		O
CodingBitDepth	Bpp_C	Number of bits per pixel, if modified by coding scheme		O
CodingOptions		Optional list of parameters describing encoding options and conditions: e.g. encoder description, compression ratio, fidelity (PSNR, SSIM, …), …		O

Table 10: Video coding parameters

4.2 Audio Coding

Unlike for video data, it is not yet clear at the moment whether compression will be required for the audio data flowing out of the production. This decision as well as the choice among existing standard compression schemes will be made later in the project, and will be reflected in later versions of this document, in particular in the final Specification document D2.1.3.

4.3 Channel Coding, Encryption and DRM

We consider here being out of scope of this document the channel coding between the production and the network. Channel coding consists in adding forward error correction codes (FEC) which enable to cope with transmission errors on the network. This is usually done at a relatively small cost in terms of bandwidth. Error correction codes add some redundancy which permits to detect and correct erroneous or missing bits. The use of such channel coding techniques has already been done for example on satellite broadcasting (DVB) with classical Reed Solomon and Viterbi codes. We believe this channel coding is of paramount importance prior to transmission after video (re-)coding. An alternative to channel coding to improve error resilience are retransmission-based mechanisms, which require a feedback channel, and usually lead to higher latency.

The same can be said of encryption of the payload data to be transmitted (metadata and audio-visual content). If encryption is used between production and the network, any network rendering operation will have to first decrypt the content and re-encrypt it after processing by making use of another secret key than the one used at the production side. The terminal should then be able to decrypt material which would have been received directly from the production or from the network. Among popular encryption schemes, one can cite AES (advanced encryption standard) which is used in AACS (Advanced Access Content System) for Blu-ray Disc or HD DVD.

Finally in a commercial application of the project, coupled to encryption, DRM (Digital Right Management) systems are needed in order to deal with the access rights of the end-users. This allows for differentiation of services in terms of quality, applications, interactivity modes etc.

5 Conclusions

This deliverable defines the current version of the layered scene description as of Month 24. It contains the different layers and the related parameters. The FascinatE scene is split in two parts, the video scene and the audio scene. In both descriptions the geometrical relationship and the technical parameters have been elaborated that cover the complete set of audio-visual sensors currently foreseen in the overall system. The definitions so far are converging to the final state due to the development in the past two project years. However, some parameters will be changed and some further ones will be added as the project is progressing.

Compared to the initial version of this deliverable, namely D2.1.1, a detailed discussion has been included regarding the merging and blending of video content from broadcast cameras into the panoramic view. It has been identified that a straightforward mapping does not lead to satisfying results due to remarkable parallax between the view from OMNICAM and the broadcast cameras mounted next to it. This was one of the major outcomes of the first test shoot and the experiments and research afterwards.

Some aspects raised in the first review such as usage of depth of field have not been included in this document as it will be discussed in other WP2 deliverables. Regarding the HDR sensors, the definition in the layered scene description takes such sensors already into account as parameters for camera type, colour system, frame rate and bit depth are part of the camera parameters. More detailed discussion of the new Alexa M HDR camera is presented in *"D2.2.1 Specification and first test implementations of capture and hardware components"*.

As mentioned in this document, there exist a lot of relationship and overlap to work in other work packages. Hence, the section 4.2 on Audio Coding highlight a few key issues, and make references to deliverables where further details may be found.

Nevertheless, this document provides all the necessary information in order to describe the FascinatE Layered Scene Representation from the top level down to the audio-visual sensors, i.e. the different type of cameras as well as the microphones.

6 References

[Batke, 2010] JM. Batke, F Keiler, S Kordon "Using 3rd Order Ambisonics Signals for 3D Concert Recording and Playback", 26. Tonmeistertagung – VDT International Convention, 2011

[Batke, 2011] JM. Batke, Jens Spille *et al*, "Recording Spatial Audio Signals for Interactive Broadcast Systems", *Forum Acusticum*, Aarlborg, Denmark, June 2011.

[Brown, 1966] D.C. Brown: "Decentering distortion of lenses," PhEng, vol. 32, no. 3, pp. 444–462, 1966.

[De Simone, 2007] F. De Simone, M. Ouaret, F. Dufaux, A.G. Tescher and T. Ebrahimi: " A comparative study of JPEG 2000, AVC/H.264, and HD Photo", Proc. SPIE Optics and Photonics, Applications of Digital Image Processing XXX, Vol. 6696, 2007.

[Devernay, 2001] Frederic Devernay and Olivier D. Faugeras, "Straight lines have to be straight," Machine Vision and Applications, vol. 13, no. 1, pp. 14–24, 2001

[Ma, 2003] Y. Ma, S. Soatto, J. Kosecka, and S. Shankar Sastry: "An Invitation to 3-D Vision: From Images to Geometric Models", Springer Verlag, 2003.

[Mallon, 2004] J. Mallon and P.F. Whelan: "Precise radial un-distortion of images", ICPR 2004, Proc. of 17th Int. Conf. on Pattern Recognition, Aug. 2004, vol. 1, pp. 18–21 Vol.1.

[Marpe, 2004] D. Marpe, V. George, H. L. Cycon, and K. U. Barthel: "Performance Evaluation of Motion-JPEG 2000 in Comparison with H.264 / AVC Operated in Intra Coding Mode", Proc. SPIE , Vol. 5266, pp. 129-137, Feb. 2004.

[Oldfield, 2011] R. G. Oldfield and B.G. Shirley, "Format agnostic recording and spatial audio reproduction of football broadcasts for the television", *Institute of Acoustics Reproduced Sound Conference*, Brighton, UK, November 16th, 2011.

[Schmidt 2004] J. Schmidt, E. F. Schröder, "New and Advanced Features for Audio Presentation in the MPEG-4 Standard", AES Convention Paper 6058, 2004 May 8-11 Berlin, Germany.

[Smith, 2004] M. Smith and J. Villasenor: "Intra-frame JPEG-2000 vs. Inter-frame Compression Comparison: The benefits and trade-offs for very high quality, high resolution sequences", In Proceedings of SMPTE Technical Conference and Exhibition, pp 1-9, October 2004, Pasadena, CA.

[Stam 2002] C. Stamm, "PGF: A New Progressive File Format for Lossy and Lossless Image Compression", Proceedings of WSCG02, pp 421-428, 2002.

[TiME-Lab] http://www.hhi.fraunhofer.de/de/veranstaltungen/veranstaltungs-und-messenarchiv/official-opening-of-the-hhi-time-lab/time_ov/opening-time-lab/

7 Glossary

Terms used within the FascinatE project, sorted alphabetically

AACS	Advanced Access Content System
AES	Advanced Encryption Standard
AVC	Advanced Video Coding
BIFS	BInary Format for Scenes
DCT	Discrete Cosine Transform
DRM	Digital Right Management
DVB	Digital Video Broadcasting
FEC	Forward Error Correction Codes
HDR	High Dynamic Range
ITU	International Telecommunication Union
JPEG	Joint Photographic Experts Group
LSR	Layered Scene Representation
MPEG	Motion Picture Expert Group
OMNICAM	Omni-directional camera for ultra high resolution panoramic video capture
PGF	Progressive Graphics File
PSNR	Peak Signal-to-Noise Ratio
VCEG	Video Coding Experts Group

Partner Acronyms

ALU	Alcatel-Lucent Bell NV, BE
ARI	Arnold & Richter Cine Technik GMBH & Co Betriebs KG, DE
BBC	British Broadcasting Corporation
DTO	Deutsche Thomson OHG, DE
HHI	Heinrich Hertz Institut, Fraunhofer Gesellschaft zur Förderung der Angewandten Forschung e.V., DE
JRS	JOANNEUM RESEARCH Forschungsgesellschaft mbH, AT
SES	Softeco Sismat S.P.A., IT
TII	The Interactive Institute, SE
TNO	Nederlandse Organisatie voor Toegapast Natuurwetenschappelijk Onderzoek – TNO, NL
UOS	The University of Salford, UK
UPC	Universitat Politecnica de Catalunya, ES

Notation

$\Re^n$	Euclidean space of dimension n
$\mathfrak{P}^n$	Projective space of dimension n
$\mathbf{v}$	Vector
$\mathbf{X}=(X,Y,Z)^T$	(3D-) point $\in \Re^3$

$\tilde{\mathbf{X}} = (X, Y, Z, 1)^T$	Homogenous (3D-) point ∈ $\mathcal{P}^3$
$\mathbf{x} = (x, y)^T$	(2D-) point ∈ $\Re^2$
$\tilde{\mathbf{x}} = (x, y, 1)^T$	Homogenous (2D-) point ∈ $\mathcal{P}^2$
$\mathbf{A} \cdot \mathbf{B}$	Scalar product of two vectors
$\mathbf{A} \times \mathbf{B}$	Cross product of two vectors
R	3 x 3 rotation matrix
I	3 x 3 identity matrix
M	Matrix

