D4.1.1 Description of the SRA

Project ref. no.	FP7-ICT-2011-7 FP7- 287639
Project acronym	SCENE
Start date of project (dur.)	1 November, 2011 (36 months)
Document due Date :	30 April, 2012 (6 month)
Actual date of delivery	14 June, 2012
Leader of this deliverable	Christopher Haccius
Reply to	haccius@intel-vci.uni-saarland.de
Document status	Final

Deliverable Identification Sheet

Project ref. no.	FP7-ICT-2011-7 FP7- 287639
Project acronym	SCENE
Project full title	Novel Scene representations for richer networked media
Document name	SCENE_D4.1.1_DDMMYYY
Security (distribution level)	PU
Contractual date of delivery	Month 6, 30.04.2012
Actual date of delivery	Month 8, 14.06.2012
Deliverable number	D4.1.1
Deliverable name	Description of the SRA
Type	Document (.doc)
Status & version	Final
Number of pages	26
WP / Task responsible	WP4 T1
Author(s)	Christopher Haccius, Thorsten Herfet, Ingo Feldmann
Other contributors	Tom Fogal, Jörn Jachalsky, Jean-Yves Guillemaut
Project Officer	Manuel.CARVALHOSA@ec.europa.eu
Abstract	Conceptual description of the new Scene Representation Archtitecture
Keywords	Representation Architecture
Sent to peer reviewer	BM - 21.05.2012
Peer review completed	*14.06.2012*
Circulated to partners	Yes
Read by partners	Yes
Mgt. Board approval	Pending

Table of contents

1 Public Executive Summary

The Scene Representation Architecture (SRA) developed in SCENE is intended to go beyond the ability of either sample based (video) or model-based (CGI) methods to deliver richer media experiences. The new SRA is a layer based architecture, consisting of a Base Layer, a Scene Layer and a Directors Layer. This layer based architecture is movie production oriented with the intention to merge real and generated content on the lowest possible level for facilitated post processing and an enhanced consumer experience.

The Base Layer represents the lowest level of the SRA. This layer stores the information as provided by different sources, which might be any kind of data acquisition equipment or computer generated content. We introduce a new format to store this Base Layer information which we call acel (Atomic sCene ELements). Those acels represent the physical content of a scene. Each acel itself is consistent in all its dimensions, but independent from other acels. Coherency information regarding only the data of a single acel is already part of this acel. An acels size can range from a single data value to a full multidimensional object.

The Scene Layer uses the Base Layer information to generate a whole scene. The independent acels stored in the Base Layer are positioned in a scene, and further information is added. Among this is lighting information and coherency information. The coherency information creates a structure between different acels exceeding their positioning in the defined dimensions. Those coherencies provide important information for physical plausibility during post processing and user interaction.

The Directors Layer is the interface between the consumer and the SRA. On the one hand camera information is provided in this layer, through which a user experiences the scene. Related to the camera are all kinds of information which make a movie production a piece of art, like different sorts of filters, blur or other effects. Finally, the Directors Layer can allow user interaction with a scene. By defining interaction rules the directors layer may provide specific options how a scene or the experience of this scene can be modified by the consumer.

With the present document we provide a conceptual description of the SRA including the newly defined acel format.

2 Introduction

SCENE sets out to develop a novel scene representation for digital media that goes beyond the ability of either sample based (video) or model based (CGI) methods to create and deliver richer media experiences. This document provides a conceptual description of a new Scene Representation Architecture. The following section presents an overview of the Scene Representation Architecture and explains how the different elements of this architecture are connected. Afterwards the individual parts of the proposed architecture are discussed in detail. Those parts of the layered architecture we propose are a Base Layer which contains the most primitive information, a Scene Layer establishing a scenario and a Directors Layer which contains all kinds of directors information, from camera settings to allowed user interaction. Special focus will be on a new data representation, the smallest scene element available, which we propose as the most primitive data type in the Base Layer. This document closes with a conclusion on the conceptual description.

3 Scene Representation Architecture - Overview

The Scene Representation Architecture (SRA) is a novel architecture aimed specifically at merging real and computer generated content. Merging those two worlds at the lowest possible level is possible through a layered approach which is based on real world movie production.

We introduce three layers: A Base Layer, a Scene Layer, and a Directors Layer. The Base Layer contains all elements which may be contained in an image or video. Those elements can be contributed by real image acquisition, processed data or computer generated content. Furthermore, all additional processing steps which enhance Base Layer Data are stored in the Base Layer. Thus the Base Layer provides all objects which constitute a scene, and assures that no information is lost during processing.

The Scene Layer combines those elements in a setting, positions lights and defines which elements of the setting are coherent to each other. Finally, the Directors Layer introduces the camera, with the artistic elements introduced by the camera operator. Additionally, the Directors Layer may allow user interaction with the scene. All layers together represent the new scene file format.

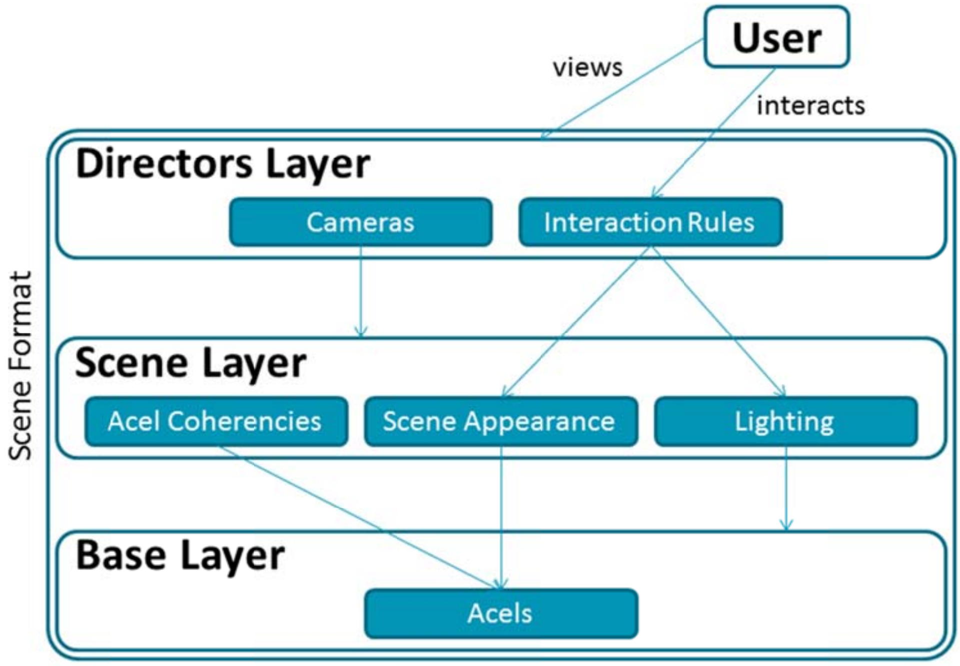


Figure 1: Scene Representation Architecture

Figure 1 represents an overview of the SRA. The Base Layer provides all object information constituting a scene in the form of acels. Those acels are positioned in a scene according to the scene appearance, which is contained in the Scene Layer. This block defines a scene for all dimensions of occurring acels and positions the acels accordingly. Furthermore, coherency information is provided in the Scene Layer. This coherency information directly relates to the Acels and creates relations between sets of those. Lighting information is also included in a Scene Layer. Other than coherency and appearance information lighting does not affect individual acels but all information provided in the Base Layer. The Directors Layer provides one or many cameras which observe the scene as created in the Scene Layer. Using interaction rules a director may allow user interaction with the scene appearance, lighting or camera information. Coherency information however cannot be modified by the user, as physical plausibility depends on coherency information. Finally, the user can observe a scene through the Directors Layer or make use of the defined interaction rules to modify scene content or its appearance.

4 Base Layer Description

The Base Layer comprises an arbitrary number of acels of a scene. All data contributing to the Base Layer needs to be located such that it can be non-ambiguously assigned to one scene (e.g., all files located in a folder). Acels in the Base Layer are addressed by unique identifiers (e.g. consecutive IDs). The naming convention should be such that acels can be easily added and removed from the base layer (e.g. hash values). The Base Layer needs to be consistent ("lowest level consistency checks"), with respect to acel dimensions, IDs and filenames; which means that acel IDs must be unique, all defined dimensions per acel must be specified and links between files must exist.

In addition to the "physically" captured data the Base Layer provides functionality to store additional metadata. This metadata can be used to reproduce the processing steps of the acel information (provenance information) and recover the original data (as recorded by an acquisition device). Provenance information is stored in a tree structure which is linked to the file header of an acel, thus providing the means to undo or redo processing steps. Notice that for correct provenance information all possible processing steps of the processing pipeline need to be known and identified. In order to ensure lossless storage the original raw data can be linked to the acel by naming the respective source file in the acel header.

Example:

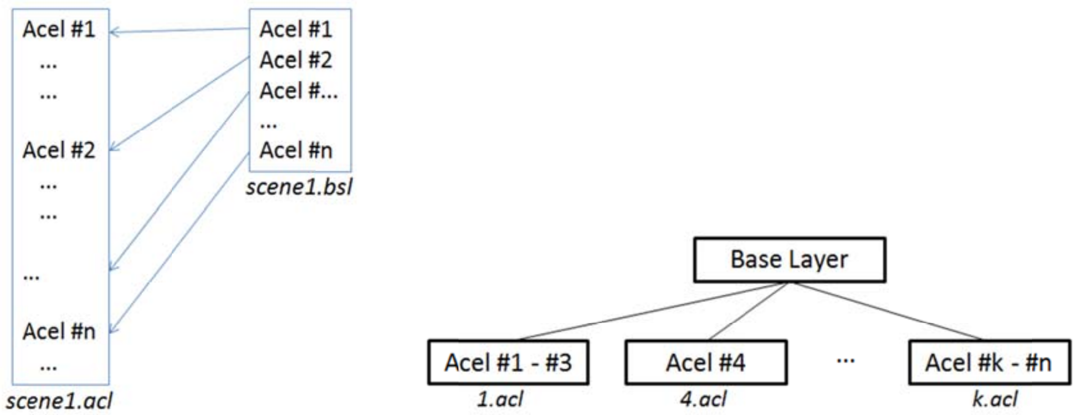


Figure 2: Possible Base Layer Representations

Figure 2 represents possible representations of the Base Layer. The Base Layer keeps track of the different acels constituting the Base Layer. Those acels can be stored in individual files, one large file, or even the same file as the Base Layer.

4.1 Acel

An acel is an "**a**tomic s**c**ene **el**ement", meaning the smallest elements a scene is composed of. Smallest, however, is defined by the encoder. A plain consumer camera for example could consider each pixel of a 2D image as an individual pixel, a "smarter" camera with the ability to subdivide a texture into superpixels could store each superpixel as an acel, and even continuous objects (like computer generated 3D objects) could be stored as an acel.

It is important to consider all dimensions (different spatial dimensions, time, color, reflectance, …) as equal. For example, depth from one camera view corresponds to width from a different perspective; a timeline could exist without any spatial location, or a reflectance value could exist without a color – the acel file does not specify any hierarchy.

Acels are completely contained in a scene. They can be reused for a new scene, but with the beginning and end of a scene (e.g scene-cuts) an acel is closed in all dimensions. More specifically, this means that while an acel can have an arbitrary number of dimensions (space, time, color, reflectance, …) all of these dimensions are limited by the borders of a scene.

Certain file types will occur more often than others. Keeping all options the acel format reserves open for limited types will blow up the file size (i.e. full flexibility for static images). If possible, this should be avoided.

Some dimensions are related (e.g. spatial dimensions, color, …). A default aspect ratio can be assumed, but it should be possible to identify different aspect ratios for reasons of efficiency. The dimensions of an acel need to be consistent in the acel, but not with a "global coordinate system", a "global color setting", etc. Global values are introduced for each acel in a higher level.

It is important that the acel representation does not enforce any transformation if this transformation would lead to a loss of information. This means, if for example a plain consumer camera has no further knowledge exceeding the color information of an image, it should not be forced to store it in any representation which would reduce the rendered quality.

From this description the following characteristics of acels are derived:

4.1.1 Characteristics of Acels:

- Arbitrary number of dimensions
- Arbitrary shape in all dimensions
- (Usually) sparse in these dimensions
- Can be discrete (if sampled, i.e. captured from real world)
- Can be continuous (i.e. if computer generated)
- Quantization steps are known and considered in advance
- Bounded in every dimension
- (Often) belong to a certain standard object type (e.g. image, mesh, …)
- Possibly different aspect ratios in related dimensions
- Needs to have ‘fallback’ possibilities for every means of data acquisition without processing capabilities

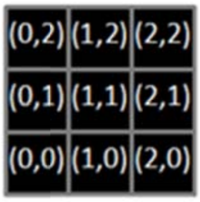
4.1.2 Proposed approach:

- Start and end acel with tag and ID
- Specify acel dimensions in header
- Each element of the acel needs to be specified in all of the given dimensions
- After header, list discrete or continuous acel elements, separate by ;
- For continuous acel elements the dimension variable is set by $X where X is number of dimension and has a default range from 0 to 1
- Similar to continuous acel elements, variables can be other discrete or continuous acels. Acels dimensions are then referenced by #X:$Y where X is the acel and Y its dimension
- Predefine certain “standard” types (i.e. 2D image, 3D mesh, …) and give acel type in header
- Set aspect ratio between related dimensions in header if ratio is not 1:1

The following examples are written in pseudo-code. There is no intention to actually store information text based – names and text are merely used to enhance readability.

4.1.3 Examples:

4.1.3.1 Color Image - black square



Option 1: Representation as 9 discrete acels; each acel contains one pixel, of which position and color are given.

begin acel 1
begin header
position x
position y
color
end header
0; 0; black;
end acel

begin acel 2
begin header
position x
position y
color
end header
0; 1; black;
end acel

…

Option 2: Representation as 1 discrete acel; one acel contains the discrete representation of 9 points of which location and color are given.

begin acel 1
begin header
position x
position y
color
end header
0; 0; black;
0; 1; black;
0; 2; black;
1; 0; black;
…
end acel

Option 3: Representation as 1 continuous acel; the two spatial dimensions are given as variables varying from 0 to 1, multiplied by 2, this way spanning a square of which the color is given as a fixed discrete value.

begin acel 1
begin header
position x
position y
color
end header
*$1*2; $2*2; black;*
end acel

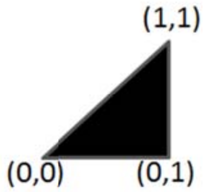
Option 4: Define type ‘image (x,y)’ as an acel with width x and height y, where each new acel line gives values of the remaining defined dimensions consecutively.

begin acel 1
begin header
type image(2,2)
color
end header
black;
black;
black;
black;
end acel

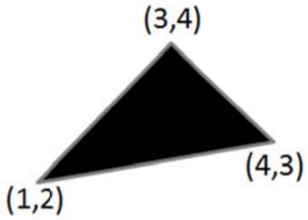
4.1.3.2 Example 2: Triangulations

A triangle (from triangulations); by giving one dimension as the multiple of the other dimension, a triangle is spanned (e.g., when $1 = 0, then $1*$2 = 0, when $1 = 0.5, then $1*$2 = 0 … 0.5, and when $1 = 1, then $1*$2 = 0 … 1).

begin acel 1
begin header
position x
position y
color
end header
*$1; $1*$2; black;*
end acel



Note that through transformations such as scaling, sheering, rotating and shifting this triangle can be arbitrarily modified in space. Consider the following example:



which we can rotate by -45° to

$$\begin{bmatrix} \cos(-45°) & -\sin(-45°) \\ \sin(-45°) & \cos(-45°) \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \end{bmatrix}$$

with the representation:

$$\begin{pmatrix} 2.1 + 2.8 \cdot \$1 \\ 0.7 - 1.4 \cdot \$1 \cdot \$2 \end{pmatrix}$$

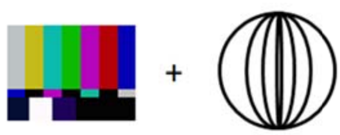
Rotation by 45° gives:

$$\begin{bmatrix} \cos(-45°) & -\sin(-45°) \\ \sin(-45°) & \cos(-45°) \end{bmatrix} \cdot \begin{bmatrix} 2.1 + 2.8 \cdot \$1 \\ 0.7 - 1.4 \cdot \$1 \cdot \$2 \end{bmatrix} = \begin{bmatrix} 1 + 2 \cdot \$1 + \$1 \cdot \$2 \\ 2 + 2 \cdot \$1 - \$1 \cdot \$2 \end{bmatrix}$$

which is the representation of the first triangle.

4.1.3.3 Example 3: Projected Texture

A discrete texture projected on a continuous sphere. The assumption here is that if a discrete acel is mapped as one dimension on a continuous acel, it is sufficient to evaluate the continuous object at those positions which have a value in the discrete acel. Acel 1 and 2 in the following example share the color dimension, and acel 2 is evaluated for the spatial dimensions of acel 1, additionally introducing a 3rd dimension.

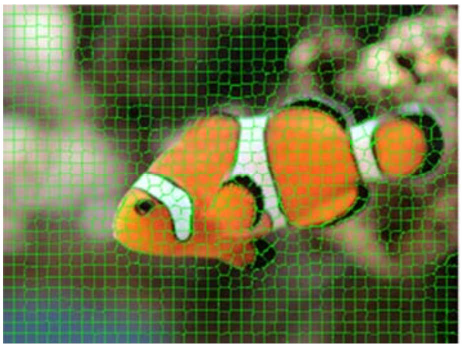


begin acel 1
begin header
position x
position y
color
end header
0; 0; color;
0; 1; color;
0; 2; color;
…
end acel

begin acel 2
begin header
position x
position y
position z
color
end header
$1; $2; sqrt(r^2-$1^2-$2^2); #1:($1,$2);
end acel

4.1.3.4 Example 4: Superpixels

A 2D image segmented into superpixels



Option 1: One acel of type image, where the 'superpixel'-characteristic is introduced as a new dimension

begin acel 1
begin header
type image (1024,768)
color
superpixel-id
end header
color1;id1;
color2;id1;
color3;id1;
…
color4; id2;
…
end acel

Option 2: Each superpixel is stored as one acel of type image where non-rectangular boundaries are filled with zeros.

begin acel 1
begin header
type image (100,100)
color
end header
blank;
blank;
color3;
color4;
…
end acel
begin acel 2
begin header
type image (120,140)
color
end header
blank;
blank;
color1;
color2;
…
end acel

…

Option 3: Each superpixel is an acel, x and y dimension of each superpixel are explicitly given.

begin acel 1
begin header
position x
position y
color
end header
15;20;red;
16;20;green;
15;21;red;
…
end acel

begin acel 2
begin header
position x
position y
color
end header
18;30;blue;
18;31;blue;
18;32;blue;
…
end acel
…

4.1.3.5 Example 5: Stereoscopic Images

Stereoscopic images are images acquired from two cameras, with almost the same content. There are, however, occlusions in both images.

Option 1: The camera has no processing power at all – simply store as two images.

begin acel 1
begin header
type image(X,Y)
color
end header
color1;
color2;
…
end acel

begin acel 2
begin header
type image(X,Y)
color
end header
color1;
color2;
…
end acel

Option 2: The camera has some processing power and can compute the depth at each position – add the depth dimension to the image.

begin acel 1
begin header
type image(X,Y)
color
position z
end header
color1; depth1;
color2; depth2;
…
end acel
begin acel 2
begin header
type image(X,Y)
color
position z
end header
color1; depth1;
color2; depth2;
…
end acel

Option 3: The camera has sufficient knowledge to realize both images contain the same elements and can calculate a 3D scene of both - store scene information in one or many acels.

begin acel 1
begin header
position x
position y
position z
color
end header
x1;y1;z1;color1;
x2;y2;z2;color2;
…
end acel

4.2 Comparison to existing representations

Many possible data types which could form the base layer information of the SRA already exist and are well known and tested. However, none of the existing data types can express all the features required by SCENE, as the SRA should make it possible to capture and lossless store (3D) video and computer generated imagery (CGI), combine video and CGI seamlessly, manipulate and deliver either 2D or 3D content in a linear or interactive form.We therefore propose to introduce acels as described above. The following list presents the required features and shows which of them are fulfilled by existing data types. In acels we try to combine the advantages of the existing file types while overcoming the restrictions imposed by each of them.

Features / Data Types	Lossless storage of 2D / 3D Video	Lossless storage of CGI	Seamless combination of video and CGI	Content manipulation	Linear delivery	Interactive delivery
Raster Graphics	√	X	X	√	√	X
Vector Graphics	X	√	X	√	√	X
Mesh Representation	√	X	X	√	X	√
Point Clouds	√	X	X	√	X	√
Spline / NURB / Bézier	X	√	X	√	X	X
Acels	√	√	√	√	(√)*	√

*Linear and interactive delivery are mutually exclusive, since interaction requires nonlinear content. However, acels can be linearized at any processing step, thus allowing interactive delivery, any kind of hybrid form between linear and interactive delivery (partially interactive) or fully linear delivery.

Conclusion

The three data types which are closest to the representation requirements are continuous curves (Spline, Bézier or NURBs), meshes and bitmaps. However, all three come from very different worlds (image acquisition, depth acquisition and computer generated content) and fail to provide the means to integrate one into the other. While most data acquisition tools will provide only one or another, any hybrid representation might result out of processing steps which bring the different worlds closer together. It is therefore not only an addition but a requirement to have a common representation for all possible data formats. By keeping the representation as open as possible, future representations which are not specified now but which might become valuable as acquisition and processing evolve can be easily integrated in our new acel format.

We therefore propose the use of acels as introduced above with the goal to merge the best of the existing worlds.

5 Scene Layer Description

The Scene Layer manages the different physical components of a scene, which are stored in the Base Layer. It is responsible for laying out a multidimensional scene. In addition, the scene layer manages coherencies between acels. The scene layer thereby enhances commonly used scene graphs by coherency information. While the scene appearance block (see Figure 1) fulfils mainly the same purpose as a scene graph, coherency information in this layer is beneficial for assignment of semantics, user interaction and facilitation of processing steps. The coherency information also eliminates the necessity of a graph structure, as it imposes its own hierarchy and dependencies upon the underlying Base Layer information.

5.1 Scene Management

Each scene is uniquely identified (e.g. by an ID). All dimensions used in the scene which are required to be a superset of the acel dimensions need to be specified in the header of the scene. Acels are placed in a scene by giving the unique acel identifier and a specific position in all dimensions the scene is defined with. The following two cases are differentiated:

- An acel is **defined** for the given dimension: the first entry of that acel is placed at the position defined in the scene layer, all other entries are considered relative to this first acel
- An acel is **not defined** for the given dimension: the acel is constant in the given dimension with the value defined in the scene layer

The scene layer can transform the whole acel, but not entries of an acel. All kinds of affine transformations (e.g. translation, rotation, sheering, expansion, reflection, …) on arbitrary acel dimensions are allowed.

Acel transitions which belong to the “physically” acquired data are stored in the Base Layer. However, explicit transition or transformation rules are described in the Scene Layer. The transition from one acel at time t to a different acel representing the same object at time t+1 can be given as an explicit rule in the Scene Layer.

5.2 Coherence Management

Acels can be coherent to other acels. In addition, acels can be likely to be coherent to other acels. Coherencies are managed per dimension and assigned for each pair of acels. Assigning coherencies is not a requirement; the default value for all coherencies is 'not coherent' until specified differently. The coherency value is assigned on a scale from 0 to 1, where 0 designates no coherency and 1 rigid coherency (rigid coherency corresponds to being stored in the same acel). By assigning coherency values to groups of acels we can introduce a possibility to assign semantics to groups of acels or identify a common behaviour to a group of coherent acels. Furthermore, coherency imposes constraints on the acel modification. Whenever the appearance of an acel is modified the appearance of all acels coherent to this one will need to be adjusted accordingly.

5.3 Confidence Information

When acels are placed in a scene confidence information can be assigned to a whole dimension of an acel. If individual confidence values are assigned to the entries of an acel, this can be done as a further dimension of this acel containing the confidence information. Confidence can be used as a measurement to assign acquisition imprecision. In general, Base Layer data is assumed to represent the physical truth. However, due to imperfect acquisition tools a confidence measure in this data can either be assigned at acquisition time or during later processing steps.

5.4 Lighting

There are several ways to express light in a scene. A light emitting object can exist as an acel. In this case, the light can be adjusted like all other acel dimensions from the scene layer. In addition to that, ambient lighting can be specified in the scene layer. If ambient light is used, this property is set in the scene header. If no light is specified, the default scene lighting is ambient lighting of a fixed strength. This default is only valid if no acel contained in the scene specifies a different light source and if no ambient light is specified in the scene.

5.5 Metadata

The scene layer allows the storage of additional metadata for each Scene element, if available semantic information can be provided for the objects contained in a scene either manually or automatically. In addition, developers information can be stored in the scene layer to facilitate postproduction.

Examples

Example 1: Two images of a plane are combined into a larger scene. Assume two acels of type image with *color1* as the color of entry 0,0 of acel1 and *color2* as the color of acel2 at position 0,0. Further, assume an offset of 250 in dimension y.

begin scene 1
begin header
position x
position y
color
end header
acel1: 0,0,color1
acel2: 0,250,color2
end scene

Example 2: Scenario as above. Now due to lighting the color dimension of the first acel is brighter by 50 as the second acel. (Assume no information is lost due to clipping of too bright values and for simplicity assume color is stored in a single channel.)

begin scene 1
begin header
position x
position y

color
end header
acel1: 0,0,color1+50
acel2: 0,250,color2
end scene

Example 3: A 3D object is added to a 2D image. Note that while only 2 spatial dimensions are defined for the image acel, the scene has 3 spatial dimensions and a 3rd fixed dimension is set for the z dimension. Acel1 is the image, acel2 the 3D object.

begin scene 1
begin header
position x
position y
position z
color
end header
acel1: x1,y1,z1,color1
acel2: x2,y2,z2,color2
end scene

Example 4: Assume the scenario from above, with the 3D object positioned at a specific height of the image (e.g. a 3D object on a table). The 3D object might be allowed to move to all sides, but needs to stay at a fixed height with respect to the image. We use a coherency table to express this relation.

begin scene 1
begin header
position x
position y
position z coherencyTable1
color
coherencyTable1
end header
acel1: x1,y1,z1,color1
acel2: x2,y2,z2,color2
acel3: …
…
end scene
begin coherencyTable1
acel 1:acel 2:1
acel1:acel3:…
…
end scene

6 Directors Layer Description

The Directors Layer adds those elements that can be influenced by the director to a scene. Here one or many cameras (with position in a specific scene, field of view, capture time, focus, etc.) are defined, (lights are positioned) (filters are defined) and rules are given which define further interaction with the scene layer.

6.1 Cameras

One or multiple cameras can be defined. Each camera is defined by a set of parameters, which are set as explicit values. The set of parameters can be differentiated into intrinsic and extrinsic parameters. Cameras used to observe a scene do not become part of that scene, so another camera looking at the position of the first camera does not observe any object there.

Extrinsic camera parameters include

- Position X
- Position Y
- Position Z
- Time

Intrinsic camera parameters include

- Focal Length
- Image Format
- Principal Point
- Filters

6.2 User Interaction Rules

Per default, no user interaction is allowed. If the director wants to specifically allow user interaction, a rule needs to describe the allowed interaction. Rules may allow any changes to the scene layer, e.g. affine transforms on all dimensions of acels or groups of acels. User interaction cannot alter the acels themselves contained in the base layer. Therefore, a user may be permitted by rules to change the appearance of a scene, but he may not change the physical content of a scene. A rule specifies a scene, an acel, a dimension and gives the range of allowed interaction. All acels which are coherent to the changed acel are affected by the change!

6.2.1 Example:

Assume an acel of type image which is acel no. 25 in scene 1. The third dimension of this acel is the color channel, which a user shall be allowed to adjust from 0 to 255.

6.3 User Role Definition

Along with user interaction rules comes the definition of separate user roles. In a movie production the director could, for example, wish to assign different interaction possibilities to broadcasters and viewers (example: broadcaster updates movie-internal advertising, viewer interacts with viewpoint). User role definitions are set by defining user groups with IDs and relating to these IDs when the rules are defined.

6.4 Metadata

In addition to the metadata information stored in the base layer and scene layer the directors layer provides the option to store metadata as well. This metadata could contain interface information for the user interaction (like a help file), and other information relevant to be linked to the whole production.

7 Conclusion

In this document we have presented the concept of a novel Scene Representation Architecture. The layer based architecture goes beyond the current ability of either sample based or model based methods. By merging those two worlds on the lowest possible level the presented SRA facilitates post processing, enables seamless integration and allows user interaction and content delivery in 2D or 3D. The atomic scene element (acel) we specify enables this merging process on the lowest level.

The three layers of the described Scene Representation Architecture carry this acel information in the lowest layer, provide scene and coherency information in the so called Scene Layer and store directors decisions, artistic elements and user interaction rules in the so called Directors Layer.

Thus the presented SRA provides a base for SCENE tools to facilitate production, reduce expenses and enrich the media experience.