On Indexing of 3D Scenes Using MPEG-7

Ioan Marius BILASCO
LSR IMAG
681 rue de la Passerelle
St. Martin d'Hères
FRANCE
+33 4 76 82 72 11
bilasco@imag.fr

Jérôme GENSEL
LSR IMAG
681 rue de la Passerelle
St. Martin d'Hères
FRANCE
+33 4 76 82 72 80
gensel@imag.fr

Marlène VILLANOVA-OLIVER
LSR IMAG
681 rue de la Passerelle
St. Martin d'Hères
FRANCE
+33 4 76 82 72 80
villanov@imag.fr

Hervé MARTIN
LSR IMAG
681 rue de la Passerelle
St. Martin d'Hères
FRANCE
+33 4 76 82 72 80
martin@imag.fr

ABSTRACT

The evolving desktop computer capacities and the emergence of the X3D standard offer a new boost to 3D domain. Giving sense to 3D content becomes a major issue specially for reusing such a content extracted from existing 3D scenes. In this paper, we address this issue by proposing a generic semantic annotation model for 3D called 3DSEAM (3D SEmantics Annotation Model). 3DSEAM aims at indexing 3D content considering visual, geometric and semantic aspects. 3DSEAM is instantiated using MPEG-7 extended with 3D specific locators. These locators link the visual, geometric and semantic features of a 3D content to the corresponding X3D fragment.

Categories and Subject Descriptors

H.3.1. [**Information Storage and Retrieval**]: Content Analysis and indexing – *indexing methods*

General Terms

Management, Documentation, Standardization, Languages

Keywords

3D content, semantic annotations, multimedia indexing

1. Introduction

The evolution of the desktop computers facilitates the large deployment of 3D contents. With the emergence of widely accepted standards such as X3D [1], 3D contents are progressively covering larger areas: from spatial planning, transports, defence to tourism. The possibility of creating, modifying or running simulation on *virtual* representation of real world environments is more attractive than working with linear or 2D data.

Efforts have been made in order to facilitate the work of 3D model designers, providing them with powerful tools (Maya, 3DS Max…). Besides facilities for creating 3D objects and scenes, the ability of reusing 3D scenes is very important for the multimedia community.

Usually, in a 3D scene, one models only the geometric features of the scene paying very little attention to semantic information that should guide and help their reuse. A first step in adding semantic to the scene corresponds to the identification of interesting objects. Their granularity can vary from a simple geometric element (e.g. a cube) to a full scene (e.g. a building). In order to facilitate the reuse of 3D objects, some semantic information should be added. Semantic queries on annotated objects would then yield the most appropriate result according to the intent of reuse.

Sustained research efforts are engaged in the characterisation of audio, image or video objects. For instance, MPEG-7 [2] descriptors and description schemas are largely accepted as multimedia description standard tools. However, the current state of MPEG-7 does not cover 3D media objects.

In this paper, we propose a solution that permits to integrate 3D contents into MPEG-7 by means of specific localizing tools. The integration of 3D into MPEG-7 description tools opens the way to complex but more meaningful 3D content annotations.

The article is organized as follows. The next section gives a sketch of the 3D world representation, focusing on the X3D standard. The section 3 presents issues related to the localisation of objects in a 3D scene. In section 4, after a brief introduction to MPEG-7, extensions to the standard are proposed in order to reference 3D contents. An overview of 3DSEAM, our 3D content management model, and an MPEG-7 instantiation of 3DSEAM are presented in section 5 before we conclude.

2. Design of 3D scenes using X3D

The Extensible 3D (X3D) standard, proposed by the Web3D consortium, defines a runtime environment and a delivery mechanism for 3D content and applications running over a network. It combines geometry descriptions, runtime behavioural descriptions and control features. It proposes many types of encodings including an XML encoding.

An X3D document represents a 3D scene as an n-ary tree. Among the most important nodes are found: geometric primitives (Cube, Box, IndexLineSet, etc), geometric transformations (Transform dealing with translations and rotations), composite objects (Group), alternate content (Switch), multi level representation (LOD), etc. The tree also contains some environmental elements: lights, viewpoints, etc., and a meta-data node (WorldInfo). The WorldInfo node contains some textual information related to the parent node.

In order to illustrate the use of X3D language, we propose a trivial construction of a scene representing the office room of a researcher (see Figure 1). This office contains a desk and a chair. On the desk, there are two stacks of books and papers. An excerpt

MM'05, November 6–11, 2005, Singapore.

of the X3D code corresponding to the materialisation of the chair is shown in Figure 2.

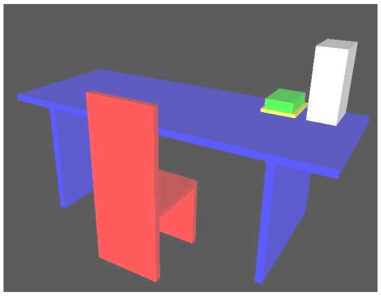

Figure 1 A Researcher's Office modelled using X3D.

As illustrated in Figure 1, the desk is built up using three boxes: one for the desk top and two for the desk legs. The chair is also composed of three boxes: one for the back-side (including the back-side legs), one for the front legs and one for the seat. Three boxes model the books on the desk: two small ones for the books on the left and a bigger one for the stack of papers. The boxes are given an adequate size using the `scale` attribute of `Transform` nodes as shown in Figure 3 (line 6 for the desk top, line 15 for the left leg…). They are grouped together into `Group` nodes (line 3 for the whole scene, line 4 for the desk...), put at the right position using the `translation` attribute (line 6 for the desk top, line 15 for the left leg…) and are given a fixed orientation using the `rotation` attribute (line 24 for the chair).

```
 1: <X3D profile="Core">
 2:  <Scene>
 3:   <Group>
 4:    <Group id ="desk">
 5:     <!--desk top-->
 6:     <Transform scale="25 0.5 10"
             translation="0 9 -50" >
 7:      <Shape>
 8:       <Box />
 9:       <Appearance>
10:        <Material
             diffuseColor="0 0 1" />
11:       </Appearance>
12:      </Shape>
13:     </Transform>
14:     <!--left leg-->
15:     <Transform scale="0.5 9 10"
             translation="-17.5 0 -50">
16:      …
17:     </Transform>
18:     <!--right leg-->
19:     <Transform scale="0.5 9 10"
             translation="17.5 0 -50" >
20:      …
21:     </Transform>
22:    </Group>
23:    <!- chair -->
24:    <Transform
             translation = "50 2 -40"
             rotation="0 1 0 1.57">
25:      <Group id="chair">
26:       ...
27:      </Group>
28:     </Transform>
29:     <!-- books -->
30:     <Transform
             translation="20 10 -5">
31:           ….
32:      <!-- stack of papers -->
33:      <Transform scale="2 5 3" id="papers"
             translation="0 5 -50" >
34:       <Shape>
35:        <Box />
36:        ...
37:       </Shape>
38:       <WorldInfo>some papers..
          </WorldInfo>
39:     </Transform>
40:    </Transform>
41:   </Group>
42:  </Scene>
43: </X3D>
```

Figure 2 Excerpt of the X3D file associated with the 3D scene of Figure 1.

The main scope of an X3D scene model is to give a precise and rich representation of the scene. The semantic aspect is scarcely addressed. The language does not provide any specific tools for capturing the semantic information related to the object contained in the scene. This choice is deliberate for the nature of the semantics varies from one application domain to another. For instance, in a virtual museum the age and the value of the cultural artefacts (modelled as 3D objects) are valuable information. While, for a find path algorithm inside a virtual museum, the value of objects is not of major importance. However, textual information can be included in the scene using the `WorldInfo` node (see line 38 in Figure 2).

Even though the XML encoding of an X3D file facilitates the retrieval of some attributary information (position, colour, appearance), different kinds of queries still remain unanswered (e.g. content-based retrieval). A complementary description, by means of annotations, would help taking into account some related semantic aspects. Firstly, annotations allow to localise the geometric elements that correspond to real world object (the chair is modelled by the node with the id *chair*). Secondly, semantic can be associated with the related object (the chair is made out of wood).

3. Localising real world objects in 3D scenes

The localisation process assumes that some (geometrics) elements that constitutes the representation of a 3D object are identified. In the case of a 2D image, objects are associated to sub-regions defined by sets of 2D polygons. In 3D spaces, one can localise objects by selecting 3D surfaces and volumes used to build up the scene. For instance, Funkhouser presents in [3] a *magic scissor* tool that help designers to cut parts of 3D models by defining volumes following separation lines and surfaces.

The nature of localised elements can vary from simple geometric elements (surfaces, volumes) to complex document entries (e.g. a cluster of geometric elements). We can notice two types of localisation: spatial (dedicated to geometric elements) and structural (for document entries).

The Structural Localisation uses the structure of a document in order to point out at the document entries that correspond to the object. For instance, if the document was encoded using XML, a structural localisation would be composed of a series of XPATH expressions. Let us consider the scene described in Figure 2. The localisation of the desk legs can be expressed as the combination of the two following XPATH expressions: /X3D/Scene/Group/Group [@id='desk']/Transform[position()=2] and //Group[@id='desk']/ Transform[position()=3].

It may happen that the structure of the document is not as fine-grained as necessary. In our example, we could imagine that the top of the stack of papers contains papers published in the last month while the bottom of the stack contains papers about multimedia indexing. The structural localisation cannot reference the papers situated at the bottom of the stack since no structural element corresponds to it. The whole stack is defined by only one box. As a consequence, the same structural element is used to represent two different objects. A solution for identifying the object is to complete the structural localisation using a spatial localisation. In our example, we can isolate the multimedia indexing related papers with a volume: a cube having the same basis as the stack and a fixed height. The coordinates of the cube are defined relatively to the box. In this case, there is a mix between one structural localisation (//Group[@id="papers"]) and a spatial one (Box(C(0,-2.5,-1,5), L(2), H(2.5),W(3)).

This situation presents similarities with documents that do not have any internal structure. This is often the case in the presence of simples Digital Terrain Models [4], a widely used form of *pseudo* 3D model. For this kind of document, the spatial localisation is the only manner to attach real world objects to a 3D content.

In the next section, we present the MPEG-7 indexing possibilities and the way we integrate 3D content descriptions throughout specific object locators.

4. Indexing 3D content with MPEG-7

MPEG-7 has stemmed from research efforts led by The Moving Picture Experts Group (MPEG) working group. It has become a standard that addresses the semantic description of media resources. Even if MPEG-7 has been proposed in the context of digital audio and video data, it is highly extensible and could cover other areas. Due to its high capability of evolution, we consider MPEG-7 as a valuable candidate for fulfilling the requirements needed in order to associate semantic annotations with a 3D scene.

MPEG-7 offers useful multimedia content description utilities for audio and visual contents. This standard provides a set of Descriptors (D), Descriptors Schemes (DS) and a Description Definition Language (DDL). Descriptors are indexation units describing the visual, audio and semantic features of objects. Description Schemes group several D and other DS into structured, semantic units. The DDL defines the syntax for creating new DS. Derived from the XMLSchema, it ensures the extensibility of the standard.

The existing DS cover the following areas: visual description (VDS), audio description (ADS) and multimedia content description (MDS). The VDS and ADS are linked to the physical, logical or semantic organisations of the document described by MDS. MDS offers DS for characterising both the physical and the logical structures of a multimedia content. It ensures also the semantic description. The basic structural element is called a *segment*. A segment corresponds to a spatial, temporal or spatio-temporal decomposition of the content. A segment (Audio Segment, Visual Segment…) can be decomposed in smaller segments leading to a hierarchical segmentation of the media content. Then, each segment is indexed using the available set of (visual/audio/...) descriptors and description schemes.

In order to facilitate the integration of 3D content description into MPEG-7, we have analysed the predefined locators. MPEG-7 supports two main types of localisation descriptors: the MediaLocatorType and the RegionLocatorType. The MediaLocatorType is a generic structural locator.

The MediaLocatorType could be used directly in order to structurally localise 3D objects. However, the fact that the MediaLocator can only contain a unique URI is too restrictive in the case of 3D. A quite common technique consists in decomposing large 3D spaces into several X3D files. Objects can then be represented over several files. Even for smaller scenes, one object can be represented over several document entries (e.g. the legs of the desk are introduced using two Transform nodes lines 15 and 19 in Figure 2). Several URIs are necessary to point out to this kind of objects. We introduce the `StructuralLocatorType` as an extension of the MediaLocatorType in order to support the localisation of objects represented in several files. The Figure 3 presents the DDL definition of the newly created locator.

```
<complexType name="StructuralLocatorType">
    <complexContent >
      <extension base="mpeg7:MediaLocatorType">
        <element name=”MediaLocator” " type="anyURI"
        minOccurs="1" maxOccurs=”unbounded”/>
      </extension>
    </complexContent >
</complexType >
```

Figure 3 Definition of 3DobjectStructLocatorType

The MPEG-7 RegionLocatorType introduces some geometric localisations. Building blocs are assembled from rectangular boxes and polygons. We propose `3DRegionLocatorType` as a 3D specific region definition. In this proposition, we propose two types of 3D regions: 3D surfaces (e.g. polygons) and volumes (e.g. 3D box, ellipsoid) - not detailed here due to lack of space.

The localisation of 3D objects can now be obtained integrating 3D region locators with structural locators into a generic `3DLocatorType`. The definition of the `3DLocatorType` is presented in Figure 4. The `SpatialLocator` guarantees the geometric localisation, while the `MediaLocator` inherited from the `StructuralLocatorType` is concerned with the structural selection of content units.

```
<complexType name="_3DLocatorType">
    <complexContent>
      <extension base="StructuralLocatorType">
        <sequence>
            <element name="SpatialLocator" minOccurs=”0”
                 type="3DRegionTypeLocatorType"/>
        </sequence>
      </extension>
    <complexContent>
</complexType>
```

Figure 4 Definition of 3DLocatorType

In order to deal with the indexation of 3D media objects, we extend the `MultimediaSegmentType`. The `3DObjectType` corresponds to a segment in a 3D scene and is defined by at least one `3DLocator`.

```
<complexType name="_3DObjectType">
    <complexContent>
      <extension base="mpeg7:MultimediaType">
        <sequence>
            <element name="Locator" minOccurs=”1”
         maxOccurs=”unbounded” type="3DLocatorType"/>
        </sequence>
      ….
      </extension>
    <complexContent>
</complexType>
```

Figure 5 Definition of 3DObjectType

The definition presented in Figure 5 is intentionally left incomplete. A discussion on candidate elements for allowing a complete description of a 3D object is presented in the next section.

5. Toward a generic model for 3D content management

In order to introduce a link between the annotations and the localization of 3D content inside a scene, we propose a generic model called 3DSEAM (3D Semantics Annotation Model).

The central part of the model is the concept of `Entity`, which defines a real-world object. As illustrated in Figure 6, an entity is characterized by the semantic information associated with it, as well as by the way it is materialized in the document.

The demand in terms of semantic information associated with the entity can vary according to application needs. The semantic information is organized in `semantic profiles`. A semantic profile contains semantic properties together with semantic relations that the entity shares with its environment.

A real-world object can be found in different documents. Each materialisation (`MultimediaFragment`) of the object can vary in terms of: the nature of the used medium (text, image, video, 3D object), the encoding (primitives-based or polygons-based 3D scenes, etc), size, costs, etc. These data together with management information correspond to the `MediaProfile` associated with the fragment. The `MediaLocator` contains means of referencing a multimedia fragment inside a multimedia document

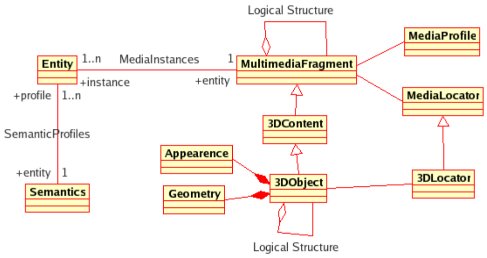

Figure 6 A partial view of the 3DSEM model

A `3DObject` refers to a unit of 3D content. A unit corresponds to a fragment (a *sub-scene*) of a 3D scene, including geometry, appearance, and any other environmental aspect related to it (light, script, etc.). Depending on the decomposition level a `3DObject` can be associated with a whole scene, a cluster of objects, or even a simple 3D geometric primitive (surface, volume). In order to be able to respond to other search criteria (geometry, appearance, spatial organisation), the model includes descriptions of the appearance (visual features indexed using MPEG-7) and of the geometry (shape, position, size, …).

The Figure 7 presents an instantiation of the model using the proposed extension for MPEG-7. The example is concerned with the indexation of the *stack of papers* from Figure 1. A first description of the partial scene (lines 3 to 30) delimits the multimedia fragments (`3DObjectType`): the entire stack (5-28), the bottom `mm_papers`(15-23) and the top (24-29). The entire stack is referred to at using a simple structural locator (7-9), while the bottom is identified with a mixed locator: a structural one (17-19) and a geometric one (20). The `LogicalStructure` is introduced using the `MediaSourceDecomposition` element (14). The `Appearance`, the `Geometry` and the `MediaProfile` are directly stored within the `3DObjectType` description. The link with the entity is established throughout a second independent semantic decomposition (31-49). Three entities are introduced: *papers_entity* (34), *mmpapers_entity* (40), *lastmonthpapers_entity* (not presented in the excerpt*)*. The link between the entity and the 3D units is materialised by the *id* associated with each unit *(line 37 for papers_entity)*. Semantics can then be associated with each of the defined entities.

6. Conclusion

In this paper, we have recalled some problems that are inherent to the process of indexing of 3D scenes. Widely accepted standards exist for the modelling of 3D scenes (X3D) and multimedia indexing (MPEG-7). They enhance the deployment of 3D data and the management of multimedia content. However, to our knowledge, no research project aims at solving interoperability issues between the two standards in order to support the management - and specially the reuse - of 3D multimedia content.

We have presented a way of including 3D content descriptions inside an MPEG-7 document. We propose a set of specific 3D region/object locators. A general 3D indexation model has been provided. The semantic added to the pure X3D geometric modelling of the scene enhances the management process of 3D objects. Depending on the type of semantics included, queries can be formulated in order to extract the most appropriate 3D content.

Future research efforts will concern the validation of the model for specific application domains (Geographic Information System, etc.). A framework for the indexation and the management of 3D content should form the basis for sharing and reusing 3D contents.

7. References

[1] Web3D Consortium, “Information technology — Computer graphics and image processing — Extensible 3D (X3D) — Part 1: Architecture and base components”, ISO/IEC FDIS 19775-1:200x, 2001

[2] J. M. Martinez and R. Koenen, “MPEG-7: The Generic Multimedia Content Description Standard, Part 1”, IEEE Multimedia, IEEE Computer Society Press, Volume 9, Issue 2, April 2002, pp. 78-87

[3] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and David Dobkin, “Modelling by Examples”, ACM Transactions on Graphics (SIGGRAPH 2004), Los Angeles, CA, August 2004, pp. 652 – 663

[4] R. Weibel and M. Heller, “Digital terrain modelling”, In Maguire, D.J., Goodchild, M.F., Rhind, D.W. (Eds.) Geographical Information Systems: principles and applications. Longman: London, 1991, pp. 269-297.

```
 1: <Mpeg7
 2:  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

 3: <Description xsi:type="ContentEntityType">
 4: <MultimediaContent xsi:type="MultimediaType">
 5:  <Multimedia xsi:type="3DObjectType«
           id="papers_3Dsegment">
 6:   <Locator xsi:type="StructuralLocatorType">
 7:    <MediaLocator>
 8:     xpath://document('office.x3d')//child::*[@id='Papers']
 9:    </MediaLocator>
10:   </Locator>
11:   <MediaProfile>...</MediaProfile>
12:   <Geometry>...</Geometry>
13:   <Apperance>...</Appearance>
14:   <MediaSourceDecomposition> <!--logical structure -->
15:     <Multimedia xsi:type="3DObjectType
             id="mmpapers_3Dsegment">
16:      <Locator xsi:type="3DLocatorType">
17:       <MediaLocator>
18:         xpath://document('office.x3d')//child::*[@id='Papers']
19:       </MediaLocator>
20:       <SpatialLocator xsi:type="Box3D"
             x="0" y="-2.5" z="-50" l="2" h="2.5" w="3"/>
21:      </Locator>
22:      ...
23:     </Multimedia>
24:    <Multimedia xsi:type="3DObjectType"
               id="lastmonth">
25:      ...
26:     </Multimedia>
27:    </MediaSourceDecomposition>
28:   </Multimedia>
29:  </MultimediaContent>
30: </Description>
31: <Description xsi:type="SemanticDescriptionType">
32:  <Semantics>
33:   <SemanticBase xsi:type="ObjectType">
34:    <Object id="papers_entity">
35:     <MediaOccurrence>
36:      <MediaInformationRef
37:          idref="papers_3Dsegment"/>
38:     </MediaOccurence>
39:     ...
40:     <Object id="mmpapers_entity">
41:      <MediaOccurence>
42:       <MediaInformationRef
43:           idref="papers_3Dsegment"/>
44:      </MediaOccurence>
45:      ...
46:     </Object>
47:    </SemanticBase>
48:   </Semantics>
49:  </Description>
50: </Mpeg7>
```

Figure 7 MPEG-7 instantiation of the 3DSEM model for the stack of papers shown in Figure 1.