

# Declarative Specification of Ambiance in VRML Landscapes

Vincent Jolivet, Dimitri Plemenos, and Patrick Poulingeas

Laboratoire MSI.  
83, rue d'Isle. 87000 Limoges. France.  
{jolivet,plemenos}@unilim.fr  
poulinge@msi.unilim.fr

**Abstract.** Nowadays, VRML has grown to a Web3D standard. But there is still a lack of high-level tools to help a VRML designer in his/her conception of a virtual world (in the same way as declarative modellers make easier the conception than classical modellers). The purpose of our work is to allow a VRML designer to specify an ambiance in a declarative manner. The method described in this paper takes into account imprecision and uncertainty in ambiance descriptions with the use of fuzzy subsets theory. With this framework, the designer can introduce some modifiers (like: “very little” or “extremely”) in his/her objectives in order to refine the wished ambiance. Two characteristics have been studied: the foggy aspect and natural lighting. Natural lighting is introduced with the aim of a natural description: the lived time.

## 1 Introduction

VRML [12] has become a popular standard for building virtual worlds. Its success is likely due to its open text-based and powerful format. Unlike other 3D scene files, VRML files can be treated with simple text-processing tools (such as Unix shell commands) or any script language (such as Perl or Python). Moreover, VRML allows the scene designer to model his virtual world with high-level components (called “nodes” in VRML) ordered in a scene graph. VRML users don’t have to bother with a low-level description model such as classical B-rep [7]. Since VRML97 standard, some nodes can even add a dynamic aspect to artificial worlds (reactions to user’s actions, temporal changes in the scene, scripts written in ECMAScript or Java, etc.)

There are two kinds of tools useful for a VRML designer in his conception work:

1. Geometric modellers (for the static aspect of a world).
2. Behavioral modellers (for the dynamic aspect of a world).

Our aim is to propose an improvement for geometric modellers in the case where the virtual world is a landscape. Classical VRML geometric modellers do not allow to specify in an intuitive and declarative manner the ambiance wished for a scene. The user has to choose numerical values in order to obtain a wished property in an empirical manner. The purpose of this method is the one of declarative modelling [11,1]: give to the designer some means to create several scenes with a set of high-level properties. These properties are the foggy aspect and the natural lighting of the VRML landscape.

In section 2, declarative modelling will be presented and we will locate our work in its field. We will then introduce a process for managing declarative specifications of an ambiance in a landscape, and a new method for introducing uncertainty in descriptions will be presented. In section 3, we will explain how to process the density of fog in a scene with the paradigm developed in the previous section. Then, in section 4, the natural lighting of a scene will be studied. A discussion on the drawbacks and the future developments of our works will be presented in section 5, and we will finally conclude in section 6.

## 2 Declarative Modelling and Declarative Specifications with Fuzzy Subsets Theory

Declarative modelling [11][1] is a useful technique allowing the designer to describe in an intuitive manner his/her idea of a scene. This kind of high-level modellers generates then the scenes corresponding to the eventually imprecise description given by the user. A lot of work usually done by the designer is then done by the declarative modeller; so the designer can spend more time on the conception phase.

There are two kinds of declarative modellers: general purpose declarative modellers which can build almost every scene, and dedicated declarative modellers which have a specific modelling domain. As we are studying ambiance in a VRML landscape, our researches belong to the area of dedicated declarative modellers.

[3][4][5][6] have proposed to represent a property not just as a standard interval, but as a fuzzy interval. This is to manage the imprecision of words like “big” or “weak” in declarative specifications given by the user to a declarative modeller. [3] introduced a taxonomy for properties and some types of linguistic modifiers that can be applied to basic properties.

A new type of linguistic modifiers will be introduced here. They allow the introduction of uncertainty in descriptions. The designer will be able to use such descriptions: “It’s rather certain that the weather is foggy”.

### 2.1 Simple Properties

[3] has started with a simple property associated to a concept (e.g. the property “foggy” associated to the concept “weather” in the following description: “The weather is foggy”). A simple property is represented by a fuzzy subset of the domain  $D$  associated to its concept, i.e. a membership function  $f$ . To define this membership function, [3] uses a quadruple  $(\alpha, a, b, \beta)$  and two functions  $L$  and  $R$  called form functions.

To check a property, the  $\alpha$ -support set  $A_v$  associated to  $f$  must be computed ( $v$  is a fixed threshold value and  $v \in ]0, 1[$ ). The  $\alpha$ -support set  $A_v$  is given by the formula:

$$A_v = [a - \alpha L^{-1}(v) ; b + \beta R^{-1}(v)]. \quad (1)$$

To easily calculate the inverse functions of L and R, in [8], L and R are trapezoidal functions. To generate numeric values for a property, a discretization of the  $\alpha$ -support set is finally done.

## 2.2 Parametric Properties

[3] introduced parametric properties, properties depending on a numerical value (e.g. “The level of snow is 2 meters”). There is still a little vagueness with these properties, so the previous fuzzy representation based on L-R membership functions can be applied to them.

## 2.3 Modifiers

Modifiers are operators that act on simple properties. They produce a new simple property, and they sometimes can be applied once again. The following set of modifiers has been selected: {“extremely little”, “very little”, “rather little”,  $\emptyset$ , “rather”, “very”, “extremely”}.  $\emptyset$  means that no modifier is used. With the simple property: “The weather is foggy”, it's then possible to build a new property with a modifier (e.g. “The weather is *very* foggy”).

Modifiers can only be used with simple properties and not with parametric properties.

[3] and [4] have developed some techniques to determine the membership function  $f'$  associated to a property like “x is *m* P” where *m* is a modifier and P a simple property. The function  $f'$  depends on the membership function  $f$  associated to P and the modifier *m*, but also on other semantic parameters of the property P.

In practice, the application of a modifier makes a translation and a contraction (or a dilatation) on the membership function. For our work, the most important thing is the facility (or difficulty) to compute quickly an  $\alpha$ -support set with the new membership function  $f'$ . The membership function  $f'$  associated to the property *m*P and the  $\alpha$ -support set are given in [8].

## 2.4 Fuzzy Operators

Fuzzy operators act both on simple and parametric properties. They produce a new property, and they can't be applied once again. The following set of fuzzy operators have been chosen: {“exactly”, “really”,  $\emptyset$ , “neighboring”, “more or less”, “vaguely”}.  $\emptyset$  means that no modifier is used. With the parametric property: “The visibility limit is 70 meters”, it is then possible to build a new property with a fuzzy operator (e.g. “The visibility limit is *really* 70 meters”).

Let's consider a property like “X is o P” where o is a fuzzy operator. A contraction (or a dilatation) will be applied to the membership function  $f$  of P. However, the kernel of the membership function will not change.

A fuzzy operator is characterized by a fuzzy coefficient  $j \in \mathbf{R}^{+*}$  which does not depend on the semantic of the property P.

The membership function  $f$  representing the property oP and its associated  $\alpha$ -support set is given in [8].

### 2.5 Uncertainty in Descriptions

The previous works have focused on the management of imprecision in declarative descriptions. We will now introduce a new kind of descriptions: descriptions with uncertainty. These descriptions take the form of: "It's *rather* certain that the size of X is 3 meters". The word "rather" acts as an uncertainty operator which increases or decreases the uncertainty of the proposition.

We suggest the following set of uncertainty operators:  $\{\emptyset, \text{"rather"}, \text{"enough"}, \text{"little"}, \text{"very little"}\}$ .  $\emptyset$  means that the description is certain (but there is still a degree of imprecision). Uncertainty operators can be used with every type of properties.

To model a proposition with an uncertainty, fuzzy intervals can still be used. The action of an uncertainty operator on the membership function will be a reduction of the size of the kernel and a contraction of the function.

An uncertainty operator is characterized by a couple of reals  $(j,k) \in ]0,1] \times \mathbf{R}^+$ .  $j$  acts on the form of the new membership function and  $k$  on the size of the kernel. The case  $j=1$  corresponds to the operator  $\emptyset$ .

The parameters of the new membership function  $f'$  are given by the formulas:

$$\alpha' = \begin{cases} \alpha \left(1 + \frac{c}{j}\right) & \text{if } j \in ]0,1[ \\ \alpha & \text{if } j = 1 \end{cases}$$

$$\beta' = \begin{cases} \beta \left(1 + \frac{c}{j}\right) & \text{if } j \in ]0,1[ \\ \beta & \text{if } j = 1 \end{cases}$$

$$a' = a + \frac{kc(b-a)}{2}$$

$$b' = b - \frac{kc(b-a)}{2}$$

$$t' = t$$

$$L' = L^j$$

$$R' = R^j$$

Remark: In fuzzy sets theory, the usual value for the coefficient  $c$  is 10%.

As the form functions  $L$  and  $R$  have changed, the  $\alpha$ -support set  $A_v$  associated to the new function  $f'$  will be:

$$A_v = [a' - \alpha' L'^{-1}(v) ; b' + \beta' R'^{-1}(v)] \tag{2}$$

with:

$$L'^{-1}(v) = R'^{-1}(v) = 1 - v^{1/j} \tag{3}$$

### 3 Foggy Scenes

VRML allows us to change the foggy aspect of a scene with the Fog node [12]. This node has a visibility field which determines the visibility limit. Two types of properties are proposed to the designer: a simple property and a parametric property.

The simple property is expressed as: “The weather is foggy”. The parametric property introduces the notion of visibility limit. These parametric properties can be expressed as: “The visibility limit is 50 meters”.

For every property, a solution interval is computed using the method exposed in section 3, with  $\alpha=\beta=10$ . Wishing to center the segment  $[a,b]$  on the value  $d$  specified in a parametric property (like “The visibility limit is  $d$  meters”), a heuristic has been introduced.

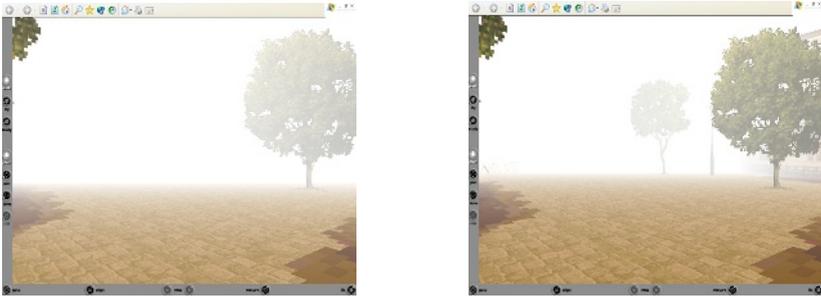
The values  $a = d-e$  and  $b = d+e$  have been chosen, with:

$$e = \frac{M - m}{10} \quad (4)$$

where:  $[m,M]$  is the domain of the membership function.

In figures 1 and 2 are shown some results found by our method.

First, we will show how an operator can modify a property. The processed parametric property is: “The visibility limit is 70 meters”, and we will then apply the fuzzy operator “exactly”. There are several solutions for each description. The minimum of the interval found has been chosen to generate the pictures in figure 1.

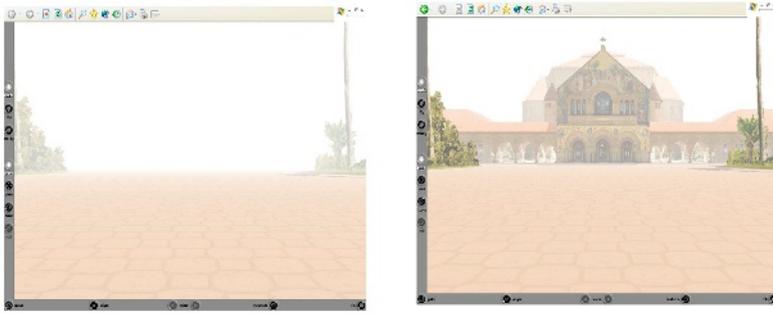


**Fig. 1.** First image: “The visibility limit is 70 meters”. Second image: “The visibility limit is exactly 70 meters”

For a property, our method generates a lot of solutions (depending on the discretization). The scenes in figure 2 correspond to the extreme values computed for the same description as in figure 1. The solution interval computed is:  $[50,90]$ .

Remark:

The scenes generated by the discretization of the interval corresponding to the user's description are often very similar. It seems not obvious to find the fine parameters for the membership function in order to produce very dissimilar scenes. A solution to this problem would be a tool for classifying the different scenes. This tool could select only a few scenes with a significant difference between them.



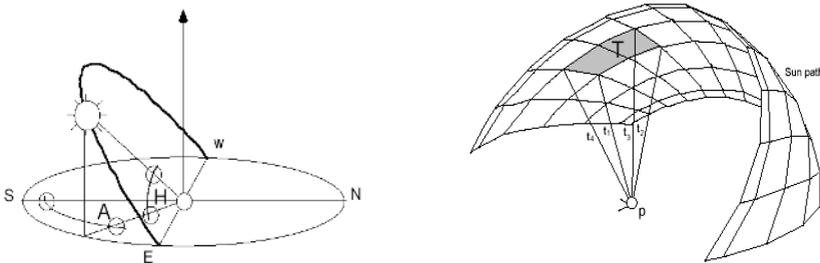
**Fig. 2.** The first image comes from the first scene found for the property: “The visibility limit is 70 meters”. The second image comes from the last scene found for the same property

## 4 Natural Lighting in a Scene

To model natural lighting, VRML proposes a node called `DirectionalLight`. This node allows the designer to introduce a directional light in his/her scene, light corresponding to the sun. Our method can find the sun direction satisfying a declarative specification of sunlighting. For coherence, our method changes the `skyAngle` and `skyColor` fields of the `Background` node to create an appropriate sky.

The sunlighting property is expressed with the concept of lived time introduced by [13]. This concept allows the use of most of semantic descriptions of time intervals. The designer can specify sunlighting with terms like: “the end of morning in spring”.

[14] proposes some formulas (coming from astronomical studies) to translate these descriptions into numerical intervals giving the direction of the sun (The direction is specified with the azimuthal angle  $A$  and the height  $H$  of the sun, as shown in figure 3). These intervals correspond to a part of the surrounding hemisphere containing the entire VRML world. Figure 3 shows the part of the hemisphere for a lived time  $T$  (the point  $p$  is the centre of the VRML world in our case).



**Fig. 3.** Left image: Determination of the position of the sun [14]. Right image: Intervals describing the lived time [13]

The possibilities of an imprecise description of the time come from the composition of an element of {“beginning”, “the middle of”, “the end of”} with an element of {“the day”, “the morning”, “the afternoon”}, from the special word

“midday” and from the direct specification of the time. The possibilities of an imprecise description of a day come from the composition of an element of {“beginning”, “the middle of”, “the end of”} with a month, from the seasons: {“Summer”, etc.} and from the special words: {“equinoxes”, “solstices”}.

In figure 4 are shown scenes with different kinds of natural lighting of the same world. The minima of the intervals corresponding to the description of the lived time have been taken for generating the following pictures:

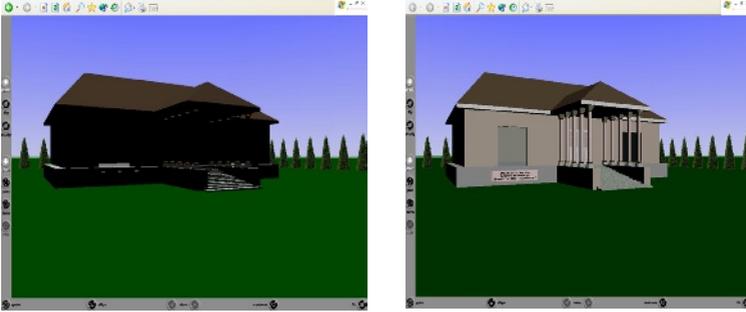


Fig. 4. Examples of natural lighting: “midday” (left) and “the middle of the afternoon” (right)

## 5 Future Researches

Our method presents several drawbacks most due to VRML limitations. The difficulties resulting from VRML are:

- The absence of shadows, which damages the realism of solar lighting.
- The lighting included in some textures may be inconsistent when the solar direction changes.
- The impossibility to take into account some complex physical parameters influencing the atmosphere in a landscape. This impossibility comes from the VRML format limitations and the lack of control in the rendering process.

To overcome these drawbacks, there are several possibilities:

- The rendering of VRML viewers is often based on OpenGL. The new version of OpenGL (OpenGL 1.5 [10]) will introduce shadow functions. So, in the future, VRML may be able to manage shadows in a scene.
- Lighting in textures can be recovered and changed with the method exposed in [9]. This method may be used in our framework as a pre-process stage and with a VRML parser (to locate the texture fields in the VRML file).
- The emergence of X3D will normally offer a better rendering than VRML. The extensions of VRML proposed by Contact [2] seem to be very useful too.

In the future, we will try to develop the scene understanding phase (which is still reduced to the simple display of the VRML world). A classification of the different scenes obtained from a description would be a powerful tool. It could avoid generating a lot of similar scenes with the presentation of some reference scenes.

## 6 Conclusion

In this paper, a tool for creating ambiance effects in VRML landscapes has been proposed. The spirit of this tool follows the ideas of declarative modelling: the designer can specify in a declarative manner his/her wishes with high-level properties.

To model and manipulate these properties, a fuzzy subsets-based representation has been introduced. This representation can take into account the necessary imprecision of the specifications and can manage uncertainty in the user's propositions. One of the key concepts is to introduce several modifiers and operators that can build different properties starting from a basic property.

Two aspects of VRML worlds have been studied: foggy appearance and sunlighting. The first one is a direct application of the concepts and ideas associated to the fuzzy subset representation of a property. The second aspect uses a natural description of sunlighting: the lived time.

The lack of realism in VRML scenes limits dramatically the extension of our method to other ambiance effects. Some extensions of VRML like [2] could allow us to treat other ambiance properties in a similar context.

## References

1. P.-F. Bonnefoi. Techniques de Satisfaction de Contraintes pour la Modélisation Déclarative. Application à la Génération Concurrente de Scènes. Ph.D. Thesis (in french). Université de Limoges, June 1999.
2. Contact, a VRML plug-in of Blaxxun Technologies. <http://developer.blaxxun.com>
3. E. Desmontils. Formalisation des Propriétés en Modélisation déclarative à l'Aide des Ensembles flous. Rapport de Recherche IRIN 106. Nantes, December 1995.
4. E. Desmontils, D. Pacholczyk. A qualitative Approach to fuzzy Properties in Scene Description. CISST'97. Las Vegas, USA, 1997, p.139-148.
5. E. Desmontils, J.-Y. Martin. Properties Taxonomy in Declarative Modeling. CISST'97. Las Vegas, USA, 1997, p.130-138.
6. E. Desmontils. Expressing Constraint Satisfaction Problems in Declarative Modeling using natural Language and fuzzy Sets. Computer & Graphics 24 (2000). p.555-568.
7. J. Foley, A. van Dam, S. Feiner, J. Hughes. Introduction to Computer Graphics. Addison Wesley Longman, Inc, 1994 .
8. V. Jolivet, D. Plemenos, P. Poulingeas. Declarative Approach of Inverse Direct Lighting Problems. 3IA'2002. Limoges, 14-15 of May, 2002.
9. S. Marschner, D. Greenberg. Inverse Lighting for Photography. IS&T/SID Fifth Color Imaging Conference. November 1997.
10. OpenGL 1.5 Specifications: <http://www.opengl.org>
11. D. Plemenos. Declarative Modelling by Hierarchical Decomposition. The Actual State of the MultiFormes Project. GraphiCon'95, St Petersburg, 1-5 of July 1995.
12. VRML 97. ISO/IEC 14772:1997. <http://www.web3d.org>
13. D. Siret. Sunlighting Design: an Inverse Approach of Simulation for CAD Tools. Advances in Computer-Aided-Design, CADEX'96. Hagenberg, September 1996.
14. D. Siret. Propositions pour une Approche Déclarative des Ambiances dans le Projet Architectural. Application à l'Ensoleillement. Thèse de Doctorat. Université de Nantes, June 1997.