Preliminary Implementation of Text-to-scene System

Fuping Yang\textsuperscript{a}, Jianchao Sun\textsuperscript{b} and Zhiyong Huang

School of Chongqing University of post and telecommunication, 400065, China
\textsuperscript{a}yangfp@cqupt.edu.cn, \textsuperscript{b}jianchao805@126.com

Keywords: Text-to-scene, natural language processing, visual scene, XML.

Abstract. Text-to-scene is a new technology used in many fields. It provides approaches to convert natural language into virtual scene. People can also use this technology to show text information using the visual method without professional knowledge. This paper attempted to provide a route of text-to-scene implementation. The researches related to text-to-scene were reviewed in brief and then a preliminary implementation of text-to-scene was proposed. The implementation contains lexical, syntactic, semantic library. Furthermore, the composition, function and work mechanism of each sub-modules constituting the text-to-scene implementation were discussed by integrating text processing and graphic processing respectively. In this article, the researches provided a route for applications of text-to-scene.

Introduction

In recent years, text-to-scene has drawn more and more attentions from the researches on artificial intelligence, transportation, education, medical technology and other fields. It is also of great significance for other visualization of the internet of things and artificial intelligence. Text-to-scene contains two types: text-to-picture and text-to-3D scene. Text-to-picture, transforming text description to 2D scene, is generally static picture.\cite{1} It is often used in retrieval and annotation of images. This way of text-to-scene needs large scale tagged image, so it is a difficult and time-consuming task. With the help of more and more modern design tools, text-to-scene for creating 3D virtual scene has been developed. Several 3D graphics-based applications, like Maya, 3D Maxs provide a suite of tools for 3D graphics functions. However, using this kind of software to create a 3D virtual scene is time consuming and requires professional technology because of the complex functions of these tools. Later on, many developed system appeared. There are two tendencies: the first tendency focuses on scene modeling and pays little and even no attention on fuzziness of high level description; \cite{1} In the other tendency, language-based 3D scene generation system takes both parts into account.\cite{2}

Normally text-to-scene has two parts, linguistic part and graphic part. There are some typical systems. WordsEye\cite{3} is a famous system in the world which is developed by AT&T laboratory, Semantic Light Co.Ltd. It contains a large database of linguistic and world knowledge about objects, parts, and other properties. Now it is used for many filed like language learning, directing cooking, cooking, chatting and other entertainment industry. WordsEye contains two parts. The first part is syntactic part. When the text is input, text will be partitioned to many words, and these words contain what message will be analyzed. The second part is interpreting the dependency structure. The semantic expression represents what dependency structure will be analyzed. Finally, use 3D scene to show the structure. WordsEye is used in many filed. Carsim\cite{4} is another famous text-to-scene system. It is a text-to-scene converter applied to traffic accidents description. It transfers the descriptive accidents in a 3D scene. Like WordsEye, Carsim also consists of parts that communicate using a formal representation of the accident. This formalism adopts a template structure similar to that of information extraction systems. CarSim system analyzes descriptions of written car accidents specifically and returns them to the 3D scenes. Just like other text-to-scene conversion applications, CarSim\cite{5} also has two parts: syntactic analyzes the part and 3D scene part. Firstly, CarSim analyzes specifically descriptions of car accidents written and return them to the 3D scenes. It does syntactic
and semantic analysis of the text you input, and then develops a 3D scene depended on the semantic information.

Nowadays, a lot of 3D modeling software is used for people to build 3D models, but it is usually complex and cost much time. Text-to-scene technology speeds up the process of creating 3D models. So many texts in many fields can convert into 3D scene conveniently.

Text-to-scene system usually contains two parts: language analysis and 3D model. The language part analyzes text description. When the texts are input, this part will know what the texts mean, how many objects the scene have, and what properties the objects have. The 3D model part is used to show the 3D scene after the objects in the texts match the 3D models.

Implementation Procedure

A text usually contains many objects, those objects change frequently. So it is hard to describe that scene. In this paper, we propose a framework to transform the texts into 3D scene. The framework is constituted of 3 levels as shown in figure 1. Level 1 contains the input of the texts. Level 2 contains the words separation, the object library and the sentence library. When the sentence which contains objects' information is input, it is separated into words. In this part the words will be transferred into XML files which contain all the properties of the objects. Since the sentences in one field are usually similar, the sentence patterns are always similar to each other. Therefore we use a sentence pattern library to store the sentence patterns to help computer understanding the meaning of the sentence. Level 3 is syntactic analysis. After matching the sentence patterns library and the object library, computer will match the objects in the sentence and the objects in the library. Level 4 is the integration of results of the above three levels and display of the 3D scene. If this part has any problems, the 3D scene will not be created, and the sentence will be analyzed again and again until the computer can match all the objects and their behaviors.

Language Processing. In this section, the information in the text about objects and the properties of the objects can be extracted from the input text descriptions. We will use this information and save them in a library.

When the text is input, it will be separated into words. We will know the objects in the form of words, and the language component analyzes it to extract the information and output as XML a formatted semantic representation. This XML file contains information of objects in the scene and spatial relationships of the objects.[6] Then the properties of the objects are shown and converted into
a visual semantic representation. Finally the XML files match the object library and visualize the scene.

Semantic Library. In this part, the meaning of a sentence will be known. When it is a simple sentence, it is easy to know the meaning. If it is hard for the computer to get the meaning, we can analyze it by hand for the first time, and then we can build a library to store this kind of sentence. There are a great variety of sentences in our life, and it will cost too much work. But if we use the sentence patterns just in one filed, the work will be less and more useful.

In our daily life, our language contains a wide variety of sentence structure and grammar, so completing the daily grammar analysis needs lots of work and requires a very large database to store this grammar and sentence patterns.[7] But if it is used for one field only, the database containing plan in grammar and sentence patterns is relatively small, and we can build a small database to store the frequently used sentence patterns and grammar. This will make it convenient for the computer to understand the statement. When detecting the input sentence can be matched to the database sentence patterns and grammar, the computer can make a quick conversion, and this can save much time. For example, we can use context-aware technology to analyze a sentence, [8] and determine the meaning and then store it in the patterns library. If this kind of sentence appears, we can use the same technology to analyze it rapidly.

Object Library. 3D models are stored in the object library. The objects can be matched in the text and shown intuitionistic by 3D visual scene. There are many objects in our life, which have different behaviors and different properties.[10] These properties will be store as the XML files. The files can match the objects in the library. It contains the objects’ size, height, width, color, life and so on. Different texts have different meaning. Some of the objects differ a lot in many texts. But we must make decision depending on the environment and choose the most convenient way.

An object library is made of many objects. Depending on the properties of these objects, the objects will be sorted separately. Some objects’ properties or location can be changed, storing them together. When some of them changed, it will be easier to change the XML files of the objects.

Graph Processing Module. The visual scene is also one of the important parts of the text-to-scene system. The XML files will be analyzed in this part. After matching the objects library, their behavior, attributes, location and other information are stored. We analyze the sentence objects’ behavior, combined with the attribute changes and their influence on other objects, and integrated into the generated 3D virtual scene. In the text-to-scene system, the types of objects are relatively few, but it is necessary to build a library to store the special behavior of special objects we do not see before.[12] Stored in the library, they will not be special and thus will be easy to control next time when we meet them.

This module still needs to have a secondary processing module for some particular objects. This module can deal with many special properties of objects and some properties changed through the context and even temporary change of objects properties and location.

Case Study

When we have text, we need to find the objects in the text and their properties and store them in the XML files. Take this sentence for example, “There is a sofa on the floor, the table is next to the sofa, a big teapot is on the table.” In this sentence, the objects are sofa, table, teapot and a room. We can store them in a library like Table 1.

Table 1 Record of objects in the library

<table>
<thead>
<tr>
<th>location</th>
<th>size</th>
<th>account</th>
</tr>
</thead>
<tbody>
<tr>
<td>teapot</td>
<td>On the table</td>
<td>big</td>
</tr>
<tr>
<td>sofa</td>
<td>Next to the table</td>
<td>small</td>
</tr>
<tr>
<td>table</td>
<td>Middle of the room</td>
<td>middle</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
The more properties of the object are stored in the library, the more complex the library is. We can handle more kinds of objects with different kinds of behavior. The text we can input can also be more complex.

The property of the teapot is its size, and it is big. We can turn it into an XML file like the following codes.

```xml
<object_frame>
  <transform>
    ...
    <teapot>
      <objectType type="3D" x="0.0" y="0.0" z="0.0" angle="0.0"/>
      <volume w="1.5" h="1.5" d="1.5"/>
      <eventIn method="set_translation" x="0.0" y="h/2" z="0.0"/>
      <eventOut method="translation_changed"/>
      <spatial_value>
        <top> y="h / 2" </top>
        <base> y="-h / 2" </base>
        <left> x="-w / 2" </left>
        <right> x="w / 2" </right>
        <back> z="-d / 2" </back>
        <front> z="d / 2" </front>
      </spatial_value>
    </teapot>
    ...
  </transform>
</object_frame>
```

All the objects’ properties need to be saved in the XML files. We use these files to match the object library and show them in the scene, like Fig. 2.

Fig. 2 Display of 3D scene

**Conclusion**

In this paper, we provide a preliminary implementation of converting the text to 3D scene. The texts are input as sentences, and then the sentences will be separated into words. We use syntactic analysis and semantic analysis to translate the text to XML files. Then the objects in the sentence will match the ones in the object library. Many sentences have fuzziness, so we use semantic analysis to make
computer understand them. Since the sentence patterns of one field are usually similar, it will be easier to achieve. Finally we use the XML files to match the object library and integrative show the objects in the sentences as 3D scene. At present, there are still various aspects of the above implementation process that can be improved. The authors will do a further research on the detail of implementation, especially the fuzziness of texts and the more accurately visualization.

Acknowledgments

Chongqing Municipal Education Commission (No. KJ130532) and Natural Science Foundation of China (No. 40801214), The natural science foundation of Chongqing(cstc2011jjA20005)T, The China postdoctoral science foundation funded project(2011M500648), The postdoctoral science foundation funded project of Chongqing (XM20120019).

References


