18th International Conference
ENGINEERING MECHANICS 2012
Svratka, Czech Republic, May 14 – 17, 2012

pp. 112–113
Paper #49

UTILIZATION OF INSTANT REALITY SOFTWARE FOR MOTION VISUALIZATION

J. Hrbáček*, T. Novotný*

Abstract: *The article deals with the ways of virtual models motion visualization in Instant Reality software based on variable input signals. Further use of this visualization is in virtual reality imaging. Paper includes the possibility of using internal software tools such as events and sensors for simultaneous changes of transformations based on differently generated signals. The output of the work is to verify the acquired knowledge on a simple application task.*

Keywords: ***Instant reality, events, sensors, signals.***

1. Introduction

Many companies spend a lot of money by moving heavy or huge devices across the whole country, in order to present their products to potential buyers. They should consider using a stereoscopic presentation, because then only things they need to move are notebook and easily moveable presentation device. In order to create a great immersion in the virtual projection, the object motion should be realistic as well. To accomplish this, the dynamic simulation should be involved.

Perhaps the most optimized solution for stereoscopic projection is use of Virtual Reality Modeling Language (VRML) as a graphical output of any mathematical software which can create dynamic simulation. However not every mathematical software has its own VRML viewer and because of this, the Instant Reality software is a great solution as an external graphics viewer. This paper refers to the ways of exporting data from mathematical software, such as MatLab, to the Instant Reality software and creating the real motion directly in Instant Reality.

2.1. Background

The stereoscopic presentation is commonly used, but motion visualization has its faults. The main fault is that the motion is only an animation, so the movement is uniform.

There is software for animating virtual objects, such as Autodesk 3D Max. It is great for animating complex motion, for example animation of the whole manufacturing process. But it also has many disadvantages, e.g. it generates a redundant amount of values in the interpolators, which has a negative influence on computing power of the work station.

3. Materials and Methods

In VRML code there is one powerful tool which allows an object to move and this tool is called an interpolator. Interpolators have two input parameters. The first parameter (key) contains the significant time points. The second parameter (keyValue) contains the transformation data. The interpolator itself computes all the values between the defined points (Žára, 1999), as it is shown in Fig.1.

However, in order to make a non-uniform motion there have to be much more defined points and even then it is hard to compute these values by yourself. That is why the dynamic simulation software is used. It can compute all values needed to make an acceleration or deceleration.

* Ing. Jakub Hrbáček, Ing. Tomáš Novotný, ING-PAED IGIP: Institute of Production Machines, Systems and Robotics, Faculty of Mechanical Engineering, Brno University of Technology, Technická 2896/2; 616 69, Brno; CZ, emails: y101623@stud.fme.vutbr.cz, novotny.t@fme.vutbr.cz

Syntax Example of Position Interpolator:

```
DEF PI PositionInterpolator {
    key [0, 0.3, 0.8]
    keyValue [0 0 0, 0 3 1, -1 5 0]
}
```

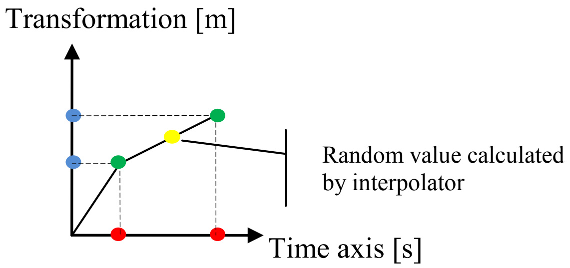

Fig. 1: Linear Interpolation

There are three simple ways to export data calculated in dynamic simulation software. This software should have the ability to save calculated data into simple txt file or similar file format. However this way of exporting could be a little time consuming so in order to automate the export a simple program should be written.

The second way of exporting is quite conditioned by obtaining MatLab software and its 3D animation tool box, which can display simulation itself in VRML integrated viewer (Humusoft s.r.o., 2000). The MatLab is also able to create VRML code with data obtained in dynamic simulation inserted directly into the interpolators so simulations can be viewed in extern VRML viewer. This solution is much faster than the first one.

The third solution uses a timer which is generated by VRML itself. This solution also requires using a Script node. The Script node allows using scripting languages. One of them is ECMAScript, which is appropriate for simple tasks, like converting data type and creating mathematical functions. And they are mathematical functions which define the object movement.

4. Results

The first way can be used for dynamic simulation software without any graphical display. But export automation or manual export could be very time consuming process.

The second solution is conditioned by obtaining MatLab software what could be the main problem for smaller companies which cannot afford it. But this solution is much faster and it also allows viewing the simulations in extern VRML viewer. This VRML code is complete therefore there is no need of data formatting.

The third solution is the most optimized solution, because it does not burden the computation power of workstation. Nevertheless to accomplish this it is necessary to have certain knowledge of programming ECMAScript or Java and to solve the differential equations analytically in order to create motion equations as a mathematical function.

5. Conclusion

This text deals with a few possibilities of data export from dynamic simulation software to a VRML code. Every way has its advantages and disadvantages which are not easy to compare, so each way should be used for a specific need.

There is another way of data export, but it has not been established yet. The VRML itself did not have any interface for an external signal until 1999, after that the Extern Authoring Interface (EAI) was added. Thanks to EAI and Java programming language the virtual model should be able to connect directly to a data stream. So the next step of the research is to establish a connection to MatLab data stream in order to create a real-time control.

References

Humusoft s. r. o. Virtual Reality Toolbox: User’s Manual. Prague (2000). ISBN 80-7080-373-8.

Žára, Jiří. (1999) Laskavý průvodce virtuálními světy. Computer Press.