

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

A Virtual Reality Simulation System for Flying Shears

Fuwen Hu and Jiajun Li

College of Mechanical and Electrical Engineering, North China University of Technology,
100144, Beijing, China

Abstract: Flying shears in rolling lines are used to cut rolled products into predefined lengths. A virtual reality simulation system was developed for two typical flying shears: rotary-crank type and rotary-drum type. Its main function modules consist of motion simulation, parameters calculation and a 3D visual knowledge base for the design and utilization of flying shears. Firstly, the development process of the virtual reality simulation system was put forward. Then the implementations of core function modules were discussed in detail based on the Virtools' behavior technologies. Finally, the last section concludes the presented work and describes the future work.

Key words: Virtual reality, flying shears, animation production, virtual manufacturing

INTRODUCTION

Virtual Reality (VR) is a computer-simulated artificial environment that can simulate physical presence of the real world or our imaging world. As an emerging technology and a powerful tool, virtual reality has been successfully used in diverse fields including entertainment, tourism, surgery, training as well as manufacturing. For example, an overview was provided on the virtual reality applications in manufacturing processes: Product design, operations management, machining, assembly and inspection (Mujber *et al.*, 2004). And Guttentag (2010) explored the primary uses for VR within tourism industry. Hu (2013) studied the development methods of virtual simulation experiment platforms for practice teaching using VR technology as well as other simulation methodologies.

The prominent feature of VR is that the participants can interact with the simulated environment in real time via the sensor devices like head-mounted display, data glove, precision position tracking system and so on (Zhou and Deng, 2009). Actually, due largely to technical limitations on processing power, image resolution and communication bandwidth, a fully high-fidelity virtual environment is currently very difficult to develop. Therefore, according to the degree of immersive interaction, the VR systems may be grouped into three levels: Non-immersive system, partially-immersive system and fully-immersive system (Fourqueta *et al.*, 2013). Generally, one basic way to create a virtual reality system is to use the most popular computer graphic libraries, e.g., OpenGL, Direct 3D, Java 3D and VRML. Another

convenient approach is to employ the VR software packages available on the market, such as 3D VIA Virtools, Quest 3D and Vega Prime. Relatively speaking, the first way has higher requirements for developers' programming abilities.

This study presented a virtual reality simulation system tailored for flying shears which are used to cut rolled products into predefined lengths. Its three main function modules included motion simulation, parameters calculation and the knowledge base for flying design and utilization. The following section discusses the development procedure of the virtual reality simulation system. And Section 3 will introduce implementations of core function modules via the Virtools software package. Some conclusions and future work will be described in the last section.

DEVELOPMENT PROCEDURE

Although 3DVIA Virtools of Dassault Systems has provided a ground-breaking solution for interactive 3D content creation, it is just an authoring platform, used to integrate media and add interactivity. Normally, the media is created in some other modeling softwares and then saved to or exported to a Virtools readable format. Hence, a long and complicated development procedure still needs to be suffered before creating an authoring application.

Figure 1 illustrates the common procedure to develop a virtual world called a composition in the Virtools. First, since Virtools is not a modeling application, 3D models should be built via 3D modeling softwares, such as 3DS Max, Solidworks, Inventor and CATIA. For instance,

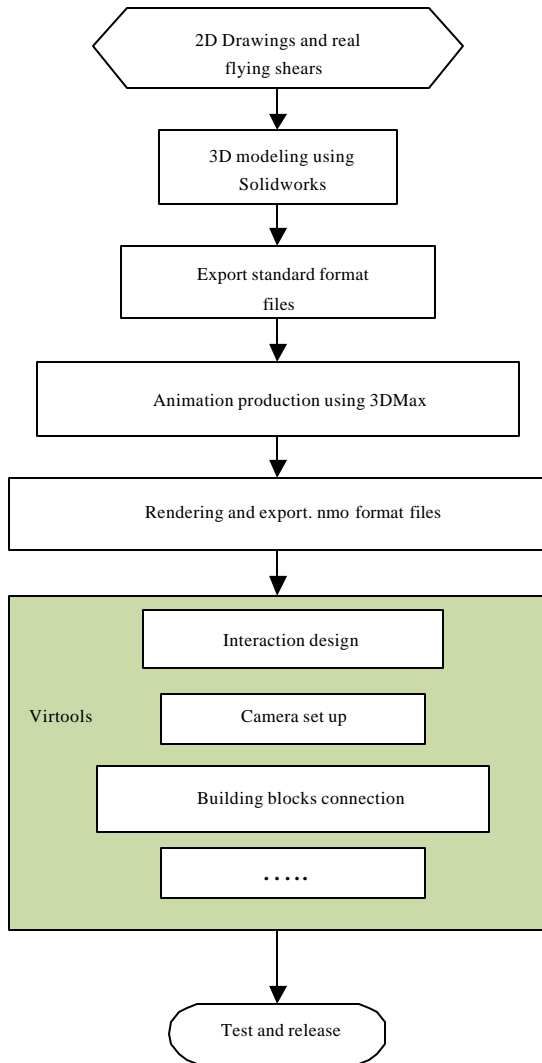


Fig. 1: Flowchart of the development for a virtual reality simulation system

Fig. 2 and 3 separately present the 3D models of rotary-crank type flying shear and rotary-drum type flying shear which are built in the Solidworks system. Next, the established 3D models are exported into STL format files or .wrl files and other standard format files. After reading these intermediate files, the animation production softwares like 3DS Max and Maya are employed to fabricate 3D scenes, characters and animations. Afterwards these standard media should be transferred into Virtools readable format files, i.e. .nmo format files. Then the behavioral building blocks provided by the Virtools can be connected by drag-and-drop to make the contents interactive. A library of over 450 ready-to-use Behavior Building blocks (BBs) that can be can be directly

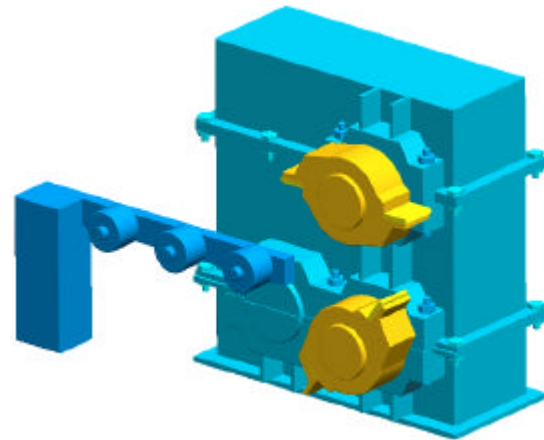


Fig. 2: 3D model of the rotary-crank type flying shear

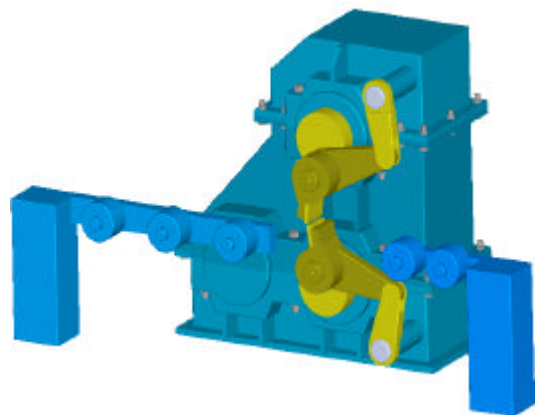


Fig. 3: 3D model of the rotary-drum type flying shear

attributed to objects, characters, cameras and any other prototype components. Undoubtedly, this step is the core of the whole process. If there is no ready-made building block for some functionality, the developer can create new building blocks with the Virtools Scripting Language (VSL) or the Software Development Kit (SDK). As another aspect for an interactive composition, the graphical user interface is also an indispensable part of the whole system. The necessary materials should be prepared by Photoshop. Lastly, the established composition can be released after testing.

In short, industry standard media such as 3D models, character models, 3D animations, images, videos and sounds can be brought to life by Virtools' behavior technologies. Thereby a high-fidelity environment or an imaging world is constructed that can interact with the human.

CORE FUNCTION MODULES

Framework of the developed system: The framework of the virtual reality simulation system for flying shears was designed as illustrated in Fig. 4. The foremost function module is motion simulations for the rotary-crank type and rotary-drum type flying shears. This module would help the users to visually understand the construction and operating principle of each type of flying shear. And the second function module is parameters design for shearing speed, maximum shearing force, shearing cycle and so on. Thirdly, a 3D knowledge base is built for the design and maintenance of flying shears. The system interface is shown in Fig. 5. When the user pushes the button, the application will launch the corresponding scene. In Virtools, a scene is a temporal device used to describe what objects present at a certain moment and how they should act. Generally, an application can be divided into several scenes. The following subsection would introduce the development principles for every function part.

Virtools process loop: Virtools is a real-time engine and it allows any behavior to react constantly and consistently

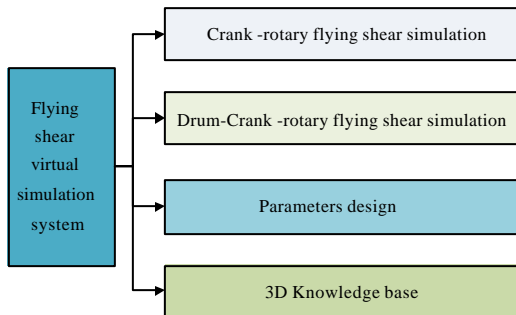


Fig. 4: Framework of the developed system

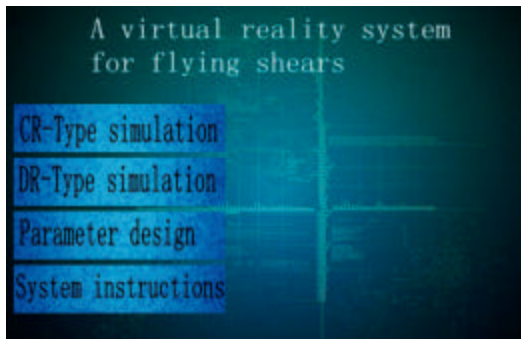


Fig. 5: Interface of the developed system

to its environment, including the use or player. Each Virtools context runs in a single system process and thread. When a Virtools composition (a context) is played, Virtools repeatedly performs the process loop until a composition is stopped or reset. The process loop is also referred as a frame or a rendering frame. Simplistically, as shown in Fig. 6, there are two phases of the process loop: processing behaviors and rendering the scene. Behavioral processing creates the interactivity within your composition. And rendering displays the composition and is performed by a separate render engine. Each cycle of the loop is a succession of phases which always happen in the same order. It might also be noted that the messages processing. As a means of transferring information between elements or between scripts, messages always have a delay of one pass through the process loop. In other words, a message is sent in the current frame but the message is not received until the next frame (process loop). A deep understanding of the process loop will help to create efficient scripts and smooth running compositions with a satisfactory frame rate.

Behavior and script: In Virtools there are three types of behaviors: Building block, behavior graphs and script. A script is the visual representation of a behavior which is composed of two parts: A header and a body. As shown in Fig. 7, the script header displays the name of the script and the owner of the script. The script body is composed of the start and one or more behavior building blocks, behavior graphs, parameters, behavior links, parameter links, comments, etc.

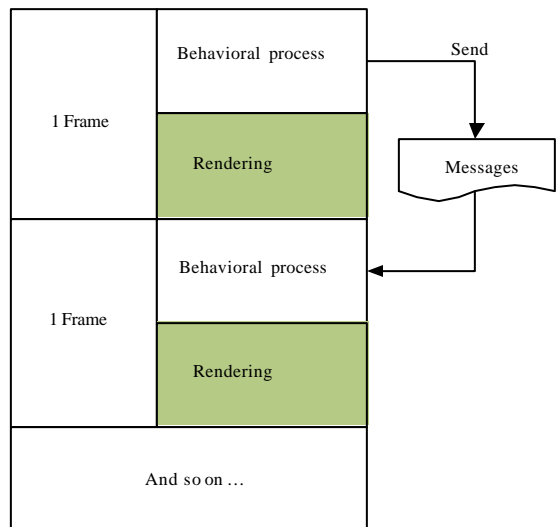


Fig. 6: Virtools process loop

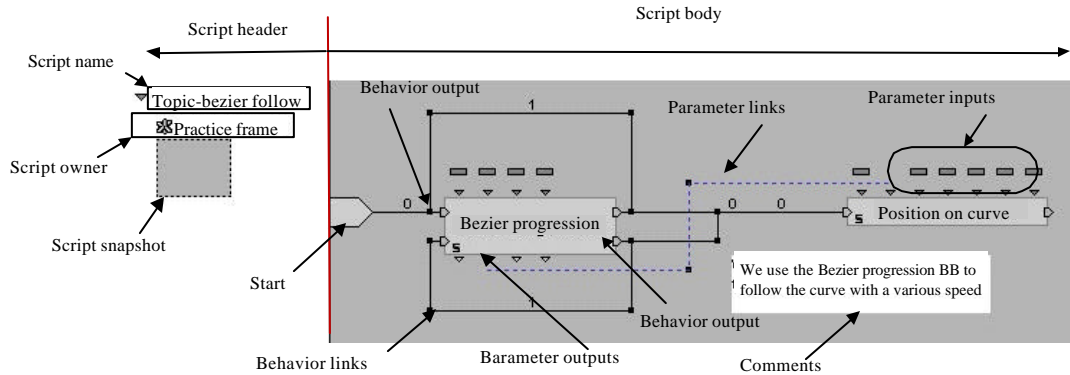


Fig. 7: Behavior and script

The Behavior Building block (BB) is a visual representation of a software element known as a function, a ready-to-use solution to a known task. There are three kinds of BBs that are processed at run-time: single action, internally looped and externally looped. The single action type of BBs completes processing within the current frame. The internally looped BB is turned On and the BB is activated every frame until the BB is turned Off. An externally looped BB completes one step in the BB's process loop within the current frame. A BB typically has at least one behavior input (bIn) and one behavior output (bOut). Horizontally, when the BB receives activation from a bIn, a BB starts processing. As soon as the processing is finished, the bOut will activate. Then activation flow follows any attached behavior links which propagate activation flow from one BB to the next. Behavior links can also point from a behavior to itself. In such a case, looped link delays of 1 and 0 separately indicate the time-splitting and time-intensive algorithms. Vertically, the BB carries out calculation according to the parameter input (pIn) and produces the parameter output (pOut). A behavior passes data to another behavior through parameter links which propagate a value from a pOut or a local parameter to a pIn.

A Behavior Graph (BG) is an author-defined behavior composed of one or more of BBs, Parameter Operations, etc. BGs encapsulate a behavior in such a manner that the BG can be saved and reused. Virtools treats a BG exactly the same as a BB.

Besides scripting graphically with the existing BBs, two other advanced scripting methods are also provided: Virtools scripting language and SDK level scripting. For programmers, VSL is an interface to the SDK from within Virtools. With no need to create a C++ project, we can easily and quickly attempt new ideas and implement

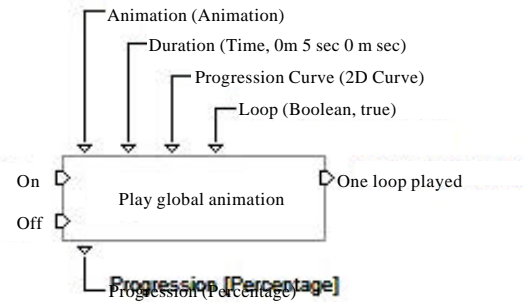


Fig. 8: Information of play global animation BB

custom code without implementing custom DLLs. However, there are certain things that VSL cannot do, e.g., no class creation in VSL. The SDK is primarily designed to define new behaviors in the form of Building Blocks encapsulated inside DLLs. Hence, the developers can create more powerful BBs by using C++ or Lua language access to the Virtools SDK, such as the creation of managers, file importers, new parameter types, etc.

Flying shear motion simulation: To simulate the motions of flying shears, one way is to directly use the 3D transformations BBs to realize them within Virtools. This way may be relatively complex for the rotary-crank type flying shear. Another approach, as previously mentioned, is to fabricate the motion animations of flying shears in the 3DS Max and export them into Virtools. Then the BB of play global animation was used to perform the ready-made animations. As presented in Fig. 8, the BB of play global animation has four input parameters and one output parameter. The first pIn referred to the imported animations. As soon as the BB of play global animation was activated by interactions, the animation would play. Perhaps some troubles may be encountered while

exporting animations from 3DS Max to Virtools. First of all, the unit of modeling is best to set up as the metric system. Besides, the topology of the animation should be proper to prevent excessive distortion. Moreover, animation keys will be removed if they don't change the animation more than a threshold value.

Parameters calculation: In Virtools, a behavior graph is composed of one or more of the following components: BBs, parameter operations, parameters, behavior links, parameter links, etc. In a case of calculating the maximal shear force, a special Behavior Graph (BG) was encapsulated as shown in Fig. 9.

The maximum shearing force can be solved by the formula:

$$F_s = K\tau_{max}bh \quad (1)$$

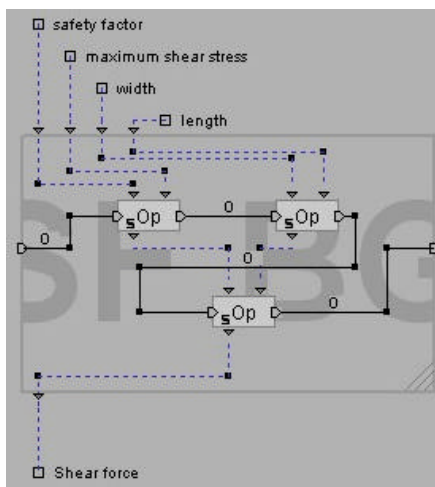


Fig. 9: Behavior graph for parameters calculation

where, τ_{max} denotes the maximum shearing stress and K is the safety factor. b and h represent the width and length of the rolled workpiece. Therefore, three parameter operation BBs were linked by blink and plink. For the sake of good readability and conciseness, the BG approach was further employed.

3D knowledge base: A knowledge base is very important for the design, fabrication, use and maintenance of the flying shears. Unlike the general knowledge base that organizes information via the invisible “black box”, we present the knowledge in the 3D virtual environment. When the user clicks any component of the flying shears with the mouse key, the related information will be presented automatically, such as its name, function, material, source and other information. In Virtools, attributes are viewed as a means of appending information to objects. Therefore, we regard the knowledge of a component as a new definition of an attribute. Therefore the Has Attribute BB can be used to retrieve the value of the attribute. Fig. 10 indicates the script of the 3D visual knowledge base.

Interactive operation: To create the buttons, labels and other interactive interfaces is very convenient in Virtools. However, the message processing between different commands has certain difficulty. Messages include the messages sent from the mouse and keyboard as well as the messages of our own definitions. There are seven ready-to-use Bbs to handle message, as illustrated in Fig. 11. Additionally, some other logics BBs were often used to deal with the complex information flows, such as Sequencer BB, Identity BB and Trigger event BB. As previously mentioned, messages are sent at the end of behavioral processing and are received in the following frame. Therefore, interactions driven by messages always

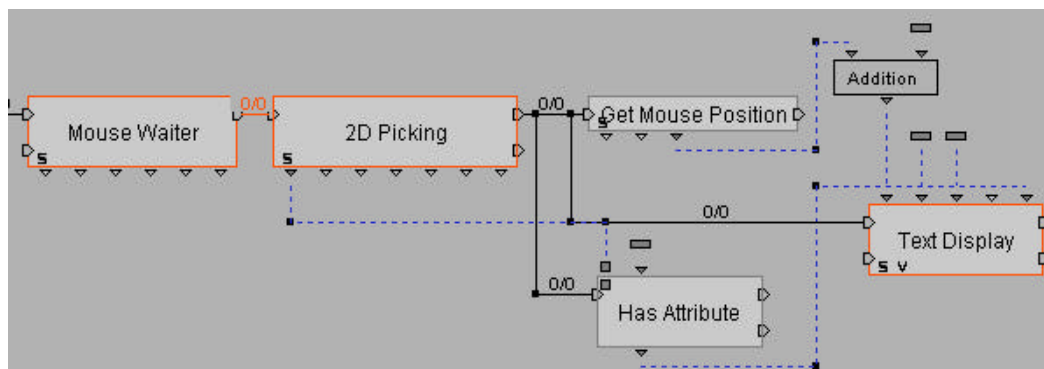


Fig. 10: Script for the 3D visual knowledge base

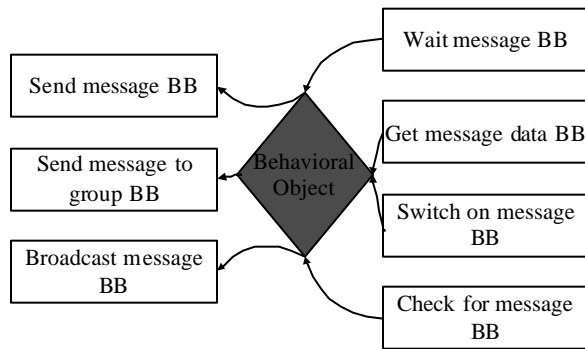


Fig. 11: Message processing BBs

experience a one frame delay between the frame in which the message is sent and the frame in which the message is received.

CONCLUSION

A virtual reality simulation system tailored for flying shears was developed using 3D VIA Virtools' behavior technologies. This system provided a visual, interactive and three-dimensional platform for operation training, product development, maintenance and sales promotion of flying shears. The development procedure and principles were amplified and can be referred as a quick-start tutorial for newcomers. Now 3D VIA Studio Pro has been released to replace Virtools and there are a lot of common concepts as well as appreciable differences between them. Therefore, this work still has a certain reference value. Certainly, the online reference of Virtools provided more help information systematically.

Future research will investigate how to build a fully-immersive virtual reality system for the rolling line. And the augmented reality (Raajan *et al.*, 2012 and Makris *et al.*, 2013) technology is also considered to add virtual elements and extra data to the real rolling line, thereby to provide more enriched experiences for participants.

REFERENCES

- Fourqueta, J.Y., R. Le Bolloc'ha, S. Cailhola, A. Datas and B. Puelb, 2013. Using virtual reality and 3D industrial numerical models for immersive interactive checklists. *Comput. Ind.*, 64: 1253-1262.
- Guttentag, D.A., 2010. Virtual reality: Applications and implications for tourism. *Tourism Manage.*, 31: 637-651.
- Hu, F.W., 2013. Development approaches for the virtual simulation experiments. *Proceedings of the Symposium on Education Reform of North China University of Technology*. October 12-13, 2013, North China University of Technology, Beijing, pp: 1-4.
- Makris, S., G. Pintzos, L. Rentzos, G. Chryssolouris, 2013. Assembly support using AR technology based on automatic sequence generation. *CIRP Annals-Manu. Tech.*, 62: 9-12.
- Mujber, T.S., T. Szecsi and M.S.J. Hashmi, 2004. Virtual reality applications in manufacturing process simulation. *J. Mater. Process. Tech.*, 155: 1834-1838.
- Raajan, N.R., S. Suganya, M.V. Priya, S.V. Ramanan and S. Janani *et al.*, 2012. Augmented reality based virtual reality. *Procedia Eng.*, 38: 1559-1565.
- Zhou, N.N. and Y.L. Deng, 2009. Virtual reality: A state-of-the-art survey. *Int. J. Auto. Comput.*, 6: 319-325.