The Sense8 Partner Development Program Presents:

3D Visualization in MATLAB Using VRo

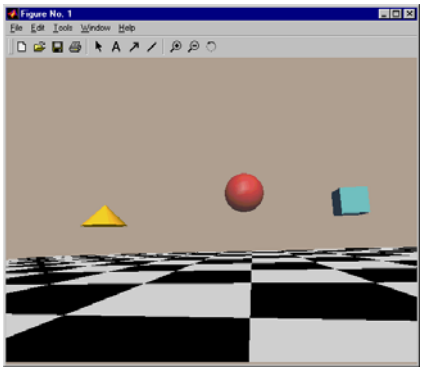


Figure 1*- Illustration of a WorldUp ActiveX Control embedded within a MATLAB figure window.*

Overview

In this tutorial you will use Sense8's ***worldup*** together with MathWorks' MATLAB to visualize a bouncing ball. The purpose of this tutorial is to illustrate just how easy it is to add the power of 3D visualization to a traditional MATLAB environment. For examples of more complex 3D visualizations using this method, please visit the Sense8 website at www.sense8.com.

Requirements

In order to do this tutorial, you will need the following:

- Sense8's ***worldup*** *R5* development environment and ActiveX Control
- Sense8's MATLAB IDE Plugin
- MATLAB (5.3 or higher)
- Terasoft's VRo

This is an introductory level tutorial, which means no specific knowledge of ***worldup*** or MATLAB is required.

Introduction

This tutorial is comprised of 2 parts :

Part 1 – Setting the Scene The first part uses Sense8's ***world****up* and shows you how to quickly create a 3D scene for use in Simulink.

Part 2 – Visualizing the Simulation The second part uses MATLAB and shows you how to work with VRo m-files to control the VR objects created with ***world****up* in Part 1.

Part 1 - Setting the Scene

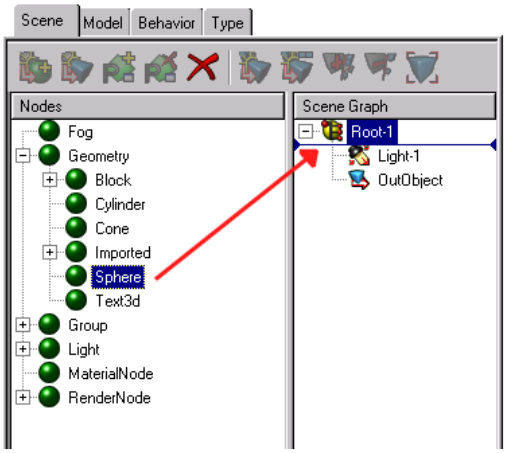
1) Create a 3D Scene in *world*up

a) Start WorldUp.

b) Create a ground plane for reference. The easiest way to do this is either by creating a block and changing it's width, height, and depth values to resemble a floor plane, or else simply import the model "*Benchf.nff*" from the ***world****up* 'Models' directory.

NOTE *If you're not familiar with **world**up, refer to the Model Workview section for how to import models, or simply switch to the **Model WorkView** tab and hit the "Import New Model" icon.*

c) Create a sphere that will be used to represent the bouncing ball. You do this simply by dragging the sphere type (green ball) from the type pane to the scene graph pane in the ***Scene WorkView*** :



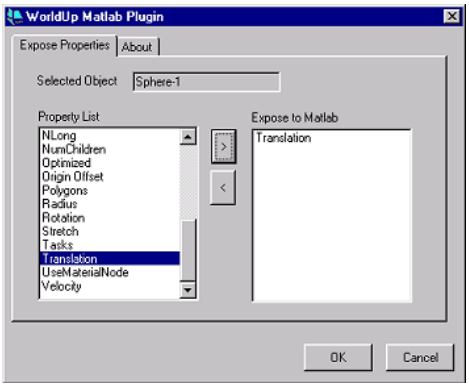
*Creating a Sphere in **world**up*

2) Expose the Sphere's Properties to Simulink

a) Select the Sphere-1 object you created.

b) Select *File -> Plugins -> Expose to Matlab* from the ***world****up* main menu.

c) Add Sphere-1's 'Translation' property from the Property List box and hit OK. The translation property will now be picked up by the Simulink parser and an input created for it when you load your project in Simulink.

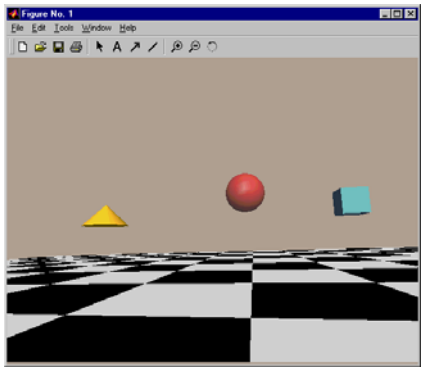


Exposing A Property to Simulink

3) Finally, Save Your Project

a) Select *File -> Save* from the ***world****up* Main Menu. This will save your 3D scene as a ***world****up* project file (an UP file) which you will load in Simulink.

Part 2 – Visualizing The Simulation



1) Load your 3D Visualization scene into a MATLAB figure

a) Use `vroload` to open your UP file in MATLAB:

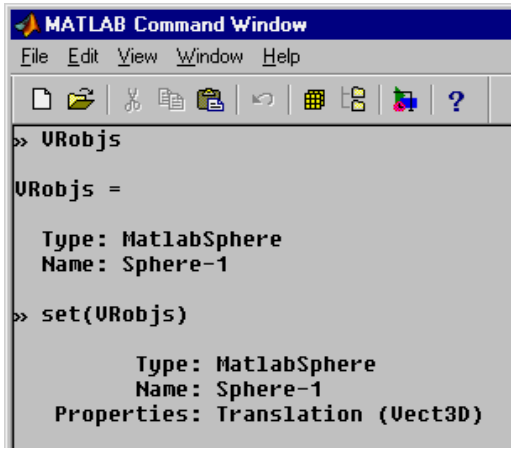
```
>> VRobjs = vroload('yourupfile.up');
```

2) Inspect our VR objects by doing a number of operations

a) By typing our object variable name at the prompt

```
>> VRobjs
```

Here, the display method displays both the object type and name.



b) By using the SET command

```
>> set(VRobjs)
```

The overloaded SET command displays all of our exposed properties. In this case, it is just Translation.

3) Generate 10 seconds of data to control our sphere

```
>> x = 0:0.1:10;
>> y = -abs(6*sin(pi/6*x));
```

We used a minus sign here because in the VR scene, -Y is up and +Y is down.

4) Make our ball bounce

```
>> for k = 1:length(y),
set(VRobjs(1),'Translation',[0, y(k), 0]);
pause(0.1);
end
```

5) Record our bouncing ball into a compressed AVI movie at 10 FPS with a frame skip index of 2.

Before executing these commands, make sure the figure window has focus.

```
>>vrocapture('Open', 'ballmovie.avi', '', 10, 2, 'cvid');
for k = 1:2:length(y),
set(VRobjs(1),'Translation',[0, y(k), 0]);
vrocapture('Snap');
end
vrocapture('Close');
```

Use your media player to view your newly created AVI file.

That's it!

Conclusion

Considering more brain matter is dedicated to visual information processing than all other senses combined, it is easy to see why 3D is well poised as a medium to take specific advantage of our advanced analytic processes.

While using an interactive 3D environment to convey the concept of a bouncing ball could certainly be construed as overkill, there are a plethora of real world simulations that present such voluminous multi-variant datasets to us as to defy comprehension. These complex, multifaceted simulations overload conventional human analysis strategies. In these situations, 3D can provide breakthrough understanding of complex systems that previously could not be contained within the human working memory. In addition, as evidenced in our simple bouncing ball tutorial, 3D provides an intuitive mapping medium that allows for quick and decisive design or behavior communication and validation.

To help you get started, you can fund a vast array of free models, textures, projects, source code, and other 3D simulation examples you can quickly use in your own simulations at ***www.3DRocketFuel.com***

For more information on ***world****up* or any of the other visual tools and solutions offered by Sense8, please visit our website at ***www.sense8.com***, or contact a salesperson directly at 415-339-3223.