Introduction to Game Programming and Robotics

Unit # 9

Simulation Engine Overview

- The Simulation Environment is composed of the following components:
  - The **Simulation Engine Service** - is responsible for rendering entities and progressing the simulation time for the physics engine. It tracks of the entire simulation world state and provides the service/distributed front end to the simulation.
  - The **Managed Physics Engine Wrapper** - abstracts the user from the low level physics engine API, provides a more concise, managed interface to the physics simulation.
  - The **NVIDIA™ PhysX™ Technology** - enables hardware acceleration through the NVIDIA™ PhysX™ capable NVIDIA™ graphics processing units (GPUs).
  - **Entities** - represent hardware and physical objects in the simulation world. A number of entities come predefined with the RDS and enable users to quickly assemble them and build rich simulated robot platforms in various virtual environments.

Tutorial # 1

- This tutorial is designed for use with Microsoft DSS Manifest Editor (DSSMe), which is shipped as part of RDS.
- Open the manifest named "SimulationIntroduction.manifest.xml" from the "File" -> "Open" menu.
- The file is located in ".....\samples\Config".

Keyboard based Control of the Simulation Environment

- **F2** - Toggles display/rendering of the physics primitive visualization. Useful for debugging your physics geometry.
- Camera Movement:
  - **A / Left Arrow / Num Pad 4** - Strafe Left (lateral move left)
  - **D / Right Arrow / Num Pad 6** - Strafe Right (lateral move right)
  - **W / Up Arrow / Num Pad 8** - Move Forward
  - **S / Down Arrow / Num Pad 2** - Move Backward
  - **Q / Page Up / Num Pad 9** - Move Up on the Y-axis (vertically)
  - **E / Page Down / Num Pad 3** - Move Down on the Y-axis (vertically)
  - **Left shift or right shift** - Hold down either of these keys to move the camera faster
  - **Home** - Reset camera to initial position

Saving and Recording the Scene Options

- Try moving the camera around and then selecting the "File" -> "Save Scene As..." option.
- Now try moving the camera again and then open the scene you just saved using the "File" -> "Open Scene..." option. You will see that the simulation camera is restored to the position and orientation when you saved the scene.
- Explore the recording and playback features.

Tutorial # 3:

- This tutorial covers how to create new entities in the Simulation Editor.
- After DSSMe starts, select "File" -> "Open" and open "SimulationEditorOverview.Manifest.xml"

Selecting and Adding Entities

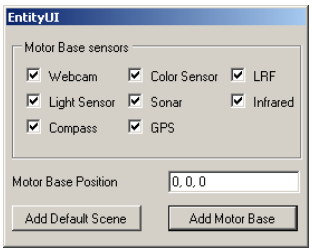
- Picking an entity can be performed by right clicking on the entity. Pressing “Control” key will allow you to see the entity that has been picked.
- Adding a new entity
  - Select "Entity" -> "New..."
  - Add a Lego Tribot.
  - Enter (9, 0, 13) for its position.

Built-in Simulation Entities

- The built-in entities can be divided into the following four categories
  - Sky and ground entities
  - Light entities
  - General Purpose Entities
  - Robot and Sensors
- Explore Lego NXT Tribot Simulation

Tutorial # 4

- Open DSSMe and open "EntityUI".
- Run the manifest.
- Add a basic scene by clicking the "Add Default Scene" button.
- Uncheck all the sensors except the Webcam checkbox.
- Click "Add Motor Base" to add a motor base with a simulated differential drive and a webcam to your scene.
- Save this scene to some file.



Tutorial # 4 (Cont'd)

- Open and run the saved manifest.
- Within the simulation window, click File -> Open Manifest and select SimpleDashboard.manifest.xml.

Simple Vision Service

- The **SimpleVision** sample service shows you how to write a service that implements image processing functions using a webcam.
- This service performs a **color object**, a **simplified face** and **hand gestures detections**.
- Other services can get the detection results by subscribing to a SimpleVision service.

Simple Vision Service (Cont'd)

- The SimpleVision sample service supports a real webcam and a simulated webcam.
- The default configuration file for this service is called "simplevision.config.xml", and it specifies the webcam polling interval, the normalized RGB color components of a specfied target oject, a color similarity threshold value and area threshold values that will be used in image processing.
- Use DSSMe to open "simplevision.manifest" or "simplevisionsim.manifest" and run it.

Simple Vision Service (Cont'd)

- While the service is running, it can detect a color object, a face and hand gestures.
- The service provides four types of notifications. One for a color registration, the others, for sending detection results.
- In the service, a color object is considered as a user's colored shirt. This assumption is also used to detect a face and hand gestures. To detect a specified color object, the service uses a normalized color and a similarity measure between current pixel vectors and a registered color vectors and segmentations.

Simple Vision Service (Cont'd)

- To detect a face, the service uses a skin image filtered by a predefined skin color model.
- After attaining a skin image, the service performs segmentation over the skin image.
- To find a face region among the segmented face candidate regions, the service uses an ellipse validation and two simple geometric constrains.
- The first constrain is a face on a color object and the second constrain is a face that should be in the upper image plane.

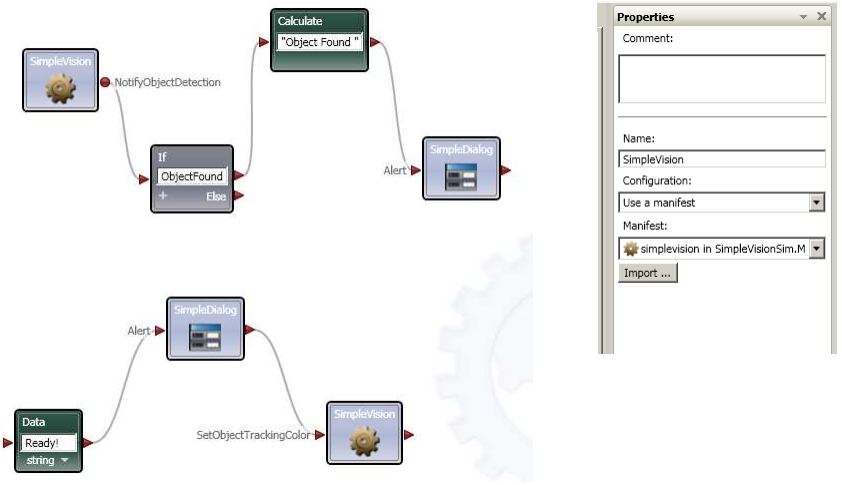
Simple Vision Service (Cont'd)

- To detect hand gestures, the service uses geometric information over a skin image and foreground image.
- It also uses results of a detected color object and a face.
- The foreground image is calculated by subtracting the current image from the background image.
- The background image is grabbed when a motion is not detected between the current camera frame and the previous camera frame for a specified time in a different image.

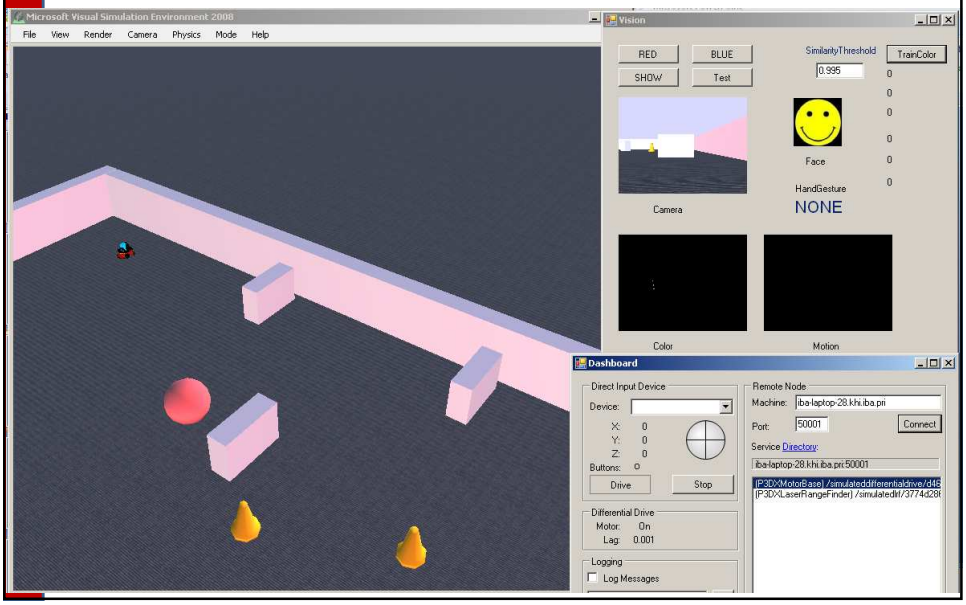
Simple Vision Service (Cont'd)

- The service provides the window form to register a specified color and to show the image processing outputs.
- You can register a color using mouse. First, make a specified circle by pressing the left mouse button in a given position and dragging to enlarge the size in the 'Camera' image on the form. Finally, press 'TrainColor' button to register an average color in a circle region.
- The registered color is saved automatically in the configuration file.

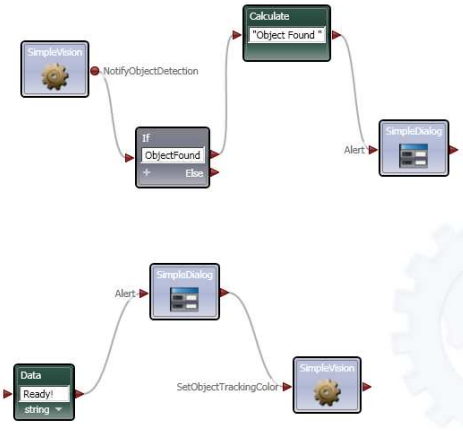
Simple Vision in VPL

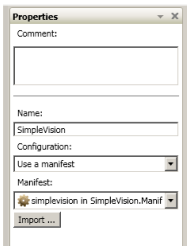


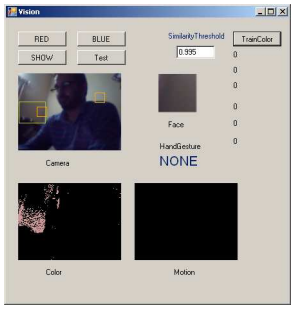
Simple Vision using Simulated Webcam



Simple Vision using Actual Webcam







Sajjad Haider Fall 2009