VU-Kinect: User's Guide

Version 1.0 by James Peyton Jones, Tyler Young, Joshua Fabian, Garrett Clayton, Sarvesh Kulkarni
Center for Nonlinear Dynamics & Control, Villanova University, Villanova PA 19085, USA

Contents

VU-Kinect

1. Overview

The Villanova University Kinect (VU-Kinect) block for Simulink is designed to integrate the color and depth video signals from the Microsoft Kinect sensor into the Simulink environment so that non-developers can access the capabilities of the sensor and leverage Simulink's powerful tools for image processing and analysis. The VU-Kinect block utilizes the open-source libfreenect driver, encapsulated in an S-Function wrapper, to interface to the Kinect device and outputs the color and depth videos at VGA resolution.

Please contact the authors directly at james.peyton-jones@villanova.edu to resolve any technical issues that may arise.

2. Legal Disclaimers and Licenses

- The terms of use of the MathWorks MATLAB Central website where the VU-Kinect block is posted may be found here: http://www.mathworks.com/matlabcentral/disclaimer.html
- The VU-Kinect block is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.
- The VU-Kinect block is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public Licensefor more details.
- The **libfreenect** driver on which the project is based is licensed under a dual Apache v2/GPL v2 license. It remains the user's responsibility to inspect, agree and abide by the license agreements associated with this tool. Further details may be found in the Section 3.

3. System Requirements

- Operating system: Linux (tested under Ubuntu 10.04 LTS and 12.04 LTS).
- Required Programs: MATLAB and Simulink (tested under MATLAB R2011b).
  - It is recommended to install the Simulink Video and Image Processing blockset (now known as the Computer Vision System Toolbox Version 4.1), which provides many capabilities for processing and managing the video signals generated by the VU-Kinect block.
- Required Drivers: libfreenect is an open-source driver for the Kinect device provided on the Openkinect.org website. The driver will be downloaded and installed automatically by the installer program included in the VU-Kinect distribution – see section 5 below.

4. Superuser Privileges

It is important to run MATLAB with superuser privileges when using or installing the VU-Kinect toolbox. Running MATLAB with normal privileges will cause many of the installation steps to fail. Furthermore, the VU-Kinect block needs to access the USB port, and this also requires superuser privileges on a Linux system unless USB port access has been specifically granted to the current user.

1. To start MATLAB with superuser privileges open an xterminal window (console) and type '**sudo**' followed by the full path to the MATLAB executable program. For typical installations, the command would therefore appear as: **sudo/usr/local/MATLAB/R2012a/bin/matlab**

5. Download and Installation

Step 0: Install and 'Select' a C Mex Compiler for Matlab and Simulink

Simulink requires a C compiler in order to compile MATLAB Executable (Mex) files. This compiler must be installed and 'selected' prior to installing VU-Kinect.

1. In order to determine if you have a compiler installed, start MATLAB in superuser mode (see section 4) and type **mex -setup** at the Matlab prompt.
2. A list of option files specific to the installed compilers will be displayed.
   - Note that on Linux, no C compiler is supplied with MATLAB. However, the GNU compiler (gcc) is included with many Linux distributions.
   - If your Linux distribution does not come with a C compiler pre-installed, then openy your distribution's package manager, search for, and select the gcc compiler for installation.
   - If no compiler is found, a list of other supported C compilers can be found here.
3. Enter the number corresponding to the options file for your desired compiler and follow the prompts to install.

Step 1: Download the VU-Kinect Distribution

You have probably already completed this step, since this document is itself part of the distribution.

1. Download **VU-Kinect.zip** from http://www.mathworks.com/matlabcentral/fileexchange/
   Extract the zip-file contents directory to any location on your hard drive **provided the pathname contains no spaces**. Note that although any directory will work, It is recommended that the files be extracted to a user-created directory called **VU-Kinect**, placed under the root MATLAB directory. (The MATLAB command **matlabroot** returns the complete path to the root MATLAB directory for your installation).

Step 2: Setup VU-Kinect

1. Open MATLAB and navigate to the directory in which VU-Kinect was unzipped.
2. Type **installKinect** at the MATLAB command prompt. (If you are prompted for a password, then you are probably not running MATLAB in superuser mode. You must exit MATLAB and restart in superuser mode as described in Section 4). The installer then automatically:
   - Downloads the required software for the libfreenect driver – files will be placed in various locations in order to facilitate the driver's use.
   - Compile (mex) the VU-Kinect.cpp S-Function file into a 'mex' file that can be executed within a Simulink block or MATLAB m-code – This file is placed in the same directory to which VU-Kinect was unzipped.
   - Add the VU-Kinect directory to the MATLAB search path.

6. Getting Started

6.1 General Notes for Using the VU-Kinect block in a Simulink model

1. Make sure the Kinect is connected to your machine via the USB cable!
2. You may need to run MATLAB with superuser privileges in order to access the USB port (see section 4).
3. To use the VU-Kinect block within a Simulink model open the kinect.mdl example (eg. By typing **kinect** at the MATLAB command line) and copy the kinect block into a new model.
4. Under the model's Simulation menu, select Configuration Parameters/Solver Pane, and set:
   i. Stop time: inf
   ii. Solver type: Fixed Step
   iii. Solver: Discrete (no continuous states)
   iv. Fixed-step size: 0.03 - gives approx. 30 frames per second. If you choose Auto, it will run as fast as it can, but will give a warning that the step size has not been specified.
5. The Kinect cannot sense objects closer than approximately 50 cm or further away than 10m. It will output the maximum value of 670 (which appears white in a Video Viewer block) if either of these conditions is met.
6. It is often useful to display results using **Video Viewer** blocks. However, these require the Computer Vision System Toolbox. Note also, that when displaying depth images in a Video Viewer block, the **Tools/Colormap** option should be set to '**Specify Range of displayed pixel values**' with minimum and maximum of **0** and **670** respectively.

6.2 A first example – kinect.mdl

In this example, two Simulink models, **kinect.mdl** and **kinect_no_video.mdl** will be used to test whether the VU-Kinect has been properly installed. The **kinect.mdl** example streams the color camera and depth images to two Video Viewer blocks. The **kinect_no_video.mdl** example is provided for users who do not have access to the Video Viewer blocks in the Computer Vision System Toolbox. It simply outputs the averaged measured distance to a numeric display block.

Using kinect.mdl

1. Start MATLAB in superuser mode in order to be able to access the USB port (see section 4).
2. Copy the kinect.mdl Simulink model into a working MATLAB directory and open the model.
3. Connect the Kinect to your computer via USB and to a power outlet.
4. In Simulink, click on the Simulation menu and select 'Start.'
5. If the installation was successful, you should see streaming depth and color video in the 2 video viewers.

Using kinect_no_video.mdl

1. Follow the same steps as for kinect.mdl (replacing kinect.mdl with kinect_no_video.mdl).
2. If the installation was successful, you should be able to change the value in the display box (which is displaying the average of the depth signal) by walking towards the Kinect.

6.3 Running the kinect.mdl Simulink model as Real Time compiled code

The Simulink model containing the VU-Kinect block can be run as real time embedded code on a hardware target such as the host PC itself, or some other target such as the Pandaboard or Beagleboard. In this example the host PC itself will be used as the target.

1. Copy the kinect.mdl Simulink model into a working MATLAB directory and open the model.
2. Under the model's **Simulation** Menu, select **external**, as opposed to **normal**.
3. Again, under the **Simulation** menu, select '**Configuration Parameters**' and confirm that the following parameters have been set correctly:
   a. **Code Generation** (formerly Real-Time Workshop) tab:
      i. System target file: grt.tlc
      ii. Language: C++
      iii. Template makefile: grt_unix.tmf
   b. Code Generation / Interface tab
      i. Interface: External mode
      ii. Transport layer: tcpip
      iii. MEX-file arguments: 'localhost' (in single quotes!)
   c. Code Generation / Custom Code
      i. Include list of additional …libraries: **/usr/lib/libfreenect.so**
   d. Click '**OK**'
4. In the model window, press CTRL-B to build an executable of the model.
5. Now start the executable running on the target hardware: In this case, open an Xterm terminal window, and navigate to the current working directory. The executable has the same name as the original model (**kinect** in this case). At the terminal window prompt, type **sudo ./kinect** The program starts running, and now waits for a connection to be initiated from the host.
6. Within the Simulink model, select **Simulation/Connect to target**. The real-time code model executing on the target should then start streaming data back to the host model, and the results should be visible in the two video viewers.
7. To terminate the process, select **Simulation/Stop Real-Time Code**.

7. An illustrative example: edge_detection.mdl

This model illustrates the power of integrating the Kinect device with the Simulink environment as described in reference [2]. The model makes use of high-level **Edge Detection**, **Geometric Transformation** and **Compositing** blocks within the Computer Vision System Toolbox in order to a) detect edges in the depth image, b) apply a simple transformation from the depth camera to the color camera and c) overlay the depth edges on the color video.

To run the model, simply copy the **edge_detection.mdl** Simulink model into a working MATLAB directory, open the model, and within the **Simulation** menu, select **Start Simulation**.

Note:

1. The Computer Vision System Toolbox is required for this model.
2. The geometric transformation applied in this example is only for illustration purposes. In order to accurately fuse information from the 2 cameras, a more thorough camera calibration must be performed.

References

1. OpenKinect Community website: http://openkinect.org.

2. Fabian, J. Young, T., Peyton Jones, J. C., Clayton G.M., Low-cost, High-Capability, Embedded Systems for Education and Research: A Toolbox for the Microsoft Kinect, (2012), *ASEE Annual Conference and Exposition*, San Antonia, TX, web link