Demo: Night Vision Effect

Chris Klaus, TAMU
Course Instructor: Professor Deepa Kundur

Introduction

After enduring the persistent pleas of your kids to make a home movie, you decide to get out your video camera and help them film an espionage movie. The kids want to have a scene where the hero is sneaking into the villain's hideout and the viewers see what he sees through his night vision goggles. Most parents wouldn't know how to do this and would then just give up, but you've taken ECEN 448 and know a thing or two about video and image processing. You understand how images are constructed, how colors are combined, and what operations you can perform on images. This demo will illustrate how one could create a simple night vision effect with the help of Simulink and a DSP board. First, however, a quick overview of some image processing basics and color spaces will be conducted.

Images and Video

An image can be thought of as a two-dimensional signal dependent where the intensity of the color at each location (called pixels) in the image is a function of its x and y coordinates. From this model, a video can then be thought of as a three-dimensional signal, where the display accounts for the two spatial dimensions and time for the third. A video is essentially a sequence of images. The color mapping for each image can be determined in several ways, which are explained below.

Color Mapping

Several different color mappings are used in different applications, but the most common ones are the RGB and the YCbCr mappings. The RGB mapping is the simplest in that each image has a Red, Green and Blue component. Each of these components are added together to give the resulting image. The YCbCr mapping is the default scheme for many television sets, and consists of two main components. The Y, or Luma, component gives the grayscale image, while the Cb (blue difference) and Cr (red difference) components combine to give the color to each pixel. The color determination for this mapping is based on the RGB scheme but maps the colors using a more complex set of equations. The overall effect of using the YCbCr mapping is that some of the redundancies of the simple RGB mapping are removed, which makes storage and transmission more efficient. This demo will utilize the color mappings to introduce a green tint to the camera output, which will approximate the night vision effect (YCbCr).

Design and Implementation

This section will provide step-by-step instructions in how to build the Simulink models for the software demo and the hardware implementation.

Buildling the Simulink Model

The following instructions will explain what blocks to insert into the model and what their corresponding parameters should be. The image at the bottom will indicate the connections amongst them.

1. Insert a From Multimedia File block from the Video and Image Processing blockset. Set the filename to "vipmem_Y.avi", check the "Inherit sample time from file" box, enter "inf" into the "Number of times to play the file" box, uncheck "Output end-of-file indicator", set the Image Color Space to Intensity, and set the video output type to "single".
2. Add a 2-D FIR Filter block from Video and Image Processing -> Filtering. Under the Main tab of the Function Block Parameters window, change the values in the coefficient box to [0.15 0.15; 0.15 0.15].
3. Add two To Video Displays located in the Video and Image Processing blockset. Double click on one of them to open the display. Under Settings, go Image Signal and select Separate Color Signals. This will create the R, G, and B inputs for the To Video Display block.
4. After connecting the blocks together, go to Simulation -> Configuration Parameters and under Solver Options -> Type, select Variable-step.
5. Run the model to view the results.

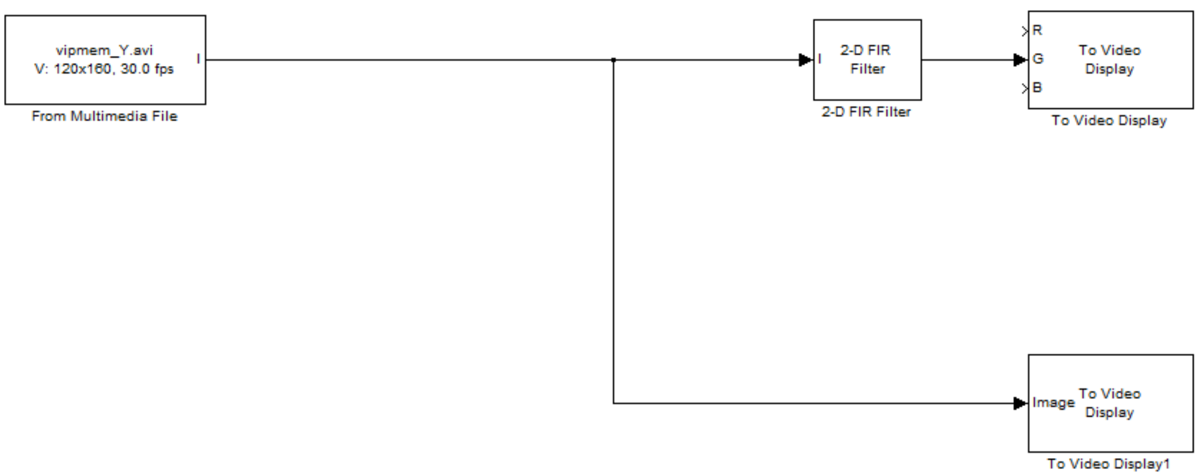

Figure 1: Simulink model of night vision effect

Digital Video Hardware Implementation

The following instructions will explain what blocks to insert into the model and what their corresponding parameters should be. The image at the bottom will indicate the connections amongst them.

1. Insert a Video Capture block from Target Support Package→Supported Processors→TI C6000→Board Support→DM6437 EVM. Set the Sample Time to -1.

2. Add a Deinterleave block under Target Support Package -> Supported Processors -> Texas Instruments C6000 -> Board Support -> DM6437 EVM.
3. Add three Gain blocks under Simulink -> Commonly Used Blocks. Set two of them to 0 and leave the remaining one at 1.
4. Add three Image Data Type Conversion blocks located in the Video and Image Processing blockset -> Conversions. Set each one's Output Data Type to "uint8".
5. Add two Interleave blocks from the Target Support Package (Target Support Package will be equivalent to the path given in Steps 1 and 2). Add two Video Display blocks from the. For the one to be connected to the Interleave block that is directly connected to the Deinterleave block (the one on the bottom in Figure 3), set the Video Window to "Video1", Video Window Position to [0, 240, 360, 240], Horizontal Zoom to 2x, and Vertical Zoom to 2x. For the other Video Display block, set Video Window to "Video 0", Video Window Position to [180, 0, 360, 480], Horizontal Zoom to 2x, and Vertical Zoom to 2x.
6. Add a DM6437 EVM block under Target Support Package -> TI C6000 -> Target Preferences.
7. At this point, the Simulink model is complete and the remaining steps are to set up the board and camera. The following are the necessary steps to set up the board and build the model:
    a. Connect the power cable to the board to turn it on.
    b. Connect the USB input to the board to connect to Simulink
    c. Connect a video cable between one of the three Video-Out ports of the board and one of the TV sets in the lab.
    d. Connect a video cable between the Video-In port of the board and the yellow cable on the camera (there's a yellow, red, and white cable attached to the camera).
    e. Plug a power cable into the red cable of the camera to turn it on.
8. The board is now set up and the only remaining task is to set the simulation parameters for the model. The following steps indicate what must be done to set up the simulation correctly.
    a. Go to Simulation -> Configuration Parameters.
    b. Under the Solver tab, make the following changes:
        i. Change the Stop Time to "inf"
        ii. Change the Solver Options type to "Fixed-Step"
        iii. Change the Solver Options Solver box to "discrete"
        iv. Tasking mode for periodic samples to "SingleTasking"
        v. Fixed step size to 1/8000
    c. Under the Real-Time Workshop tab
        i. Click Browse next to the System Target File box
        ii. Select "ccslink_grt.tlc" from the list and select OK
    d. Under the Embedded IDE Link tab
        i. Change Build Action to Build_and_execute

ii. Type "-o2 –q" in the Compiler Options String box
iii. Type 8192 in the System Stack Size

e. Select OK for Configuration Parameters window
f. Press Ctrl+B while in the model window to build the model.

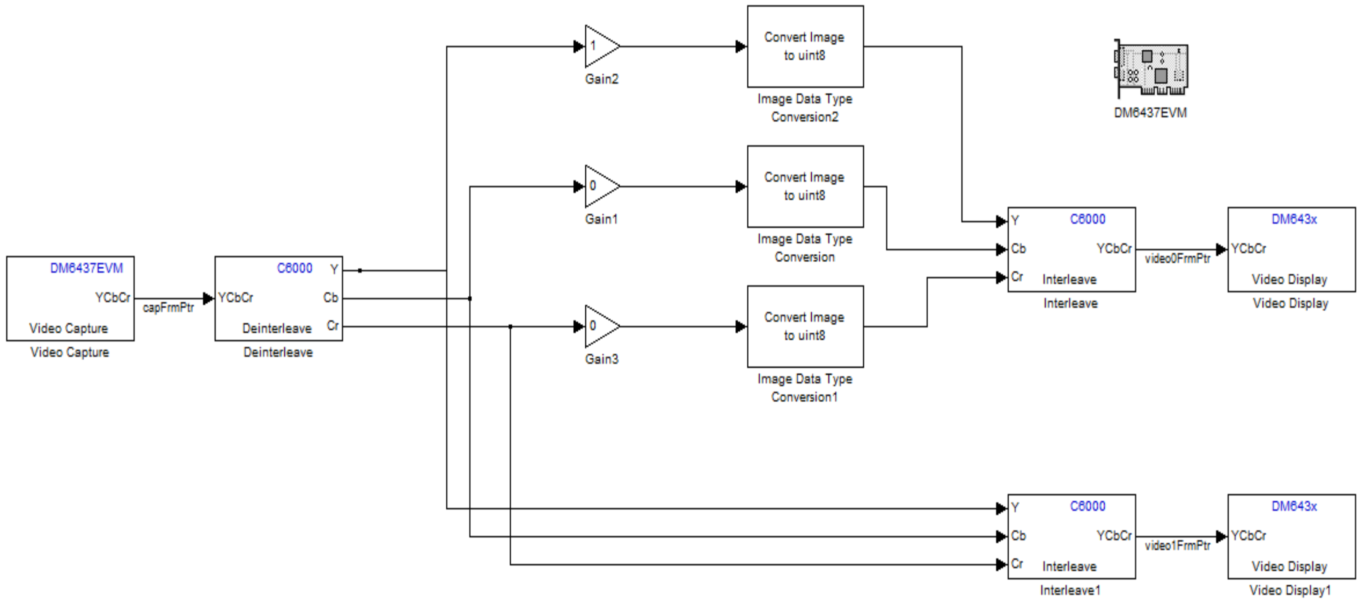

Figure 2**:** Hardware implementation of night vision effect.

Results

This section showcases the results of the software Simulink model and the hardware implementation.

Figure 3: Unaltered test image in grayscale format

Figure 4: Image with Night Vision effect

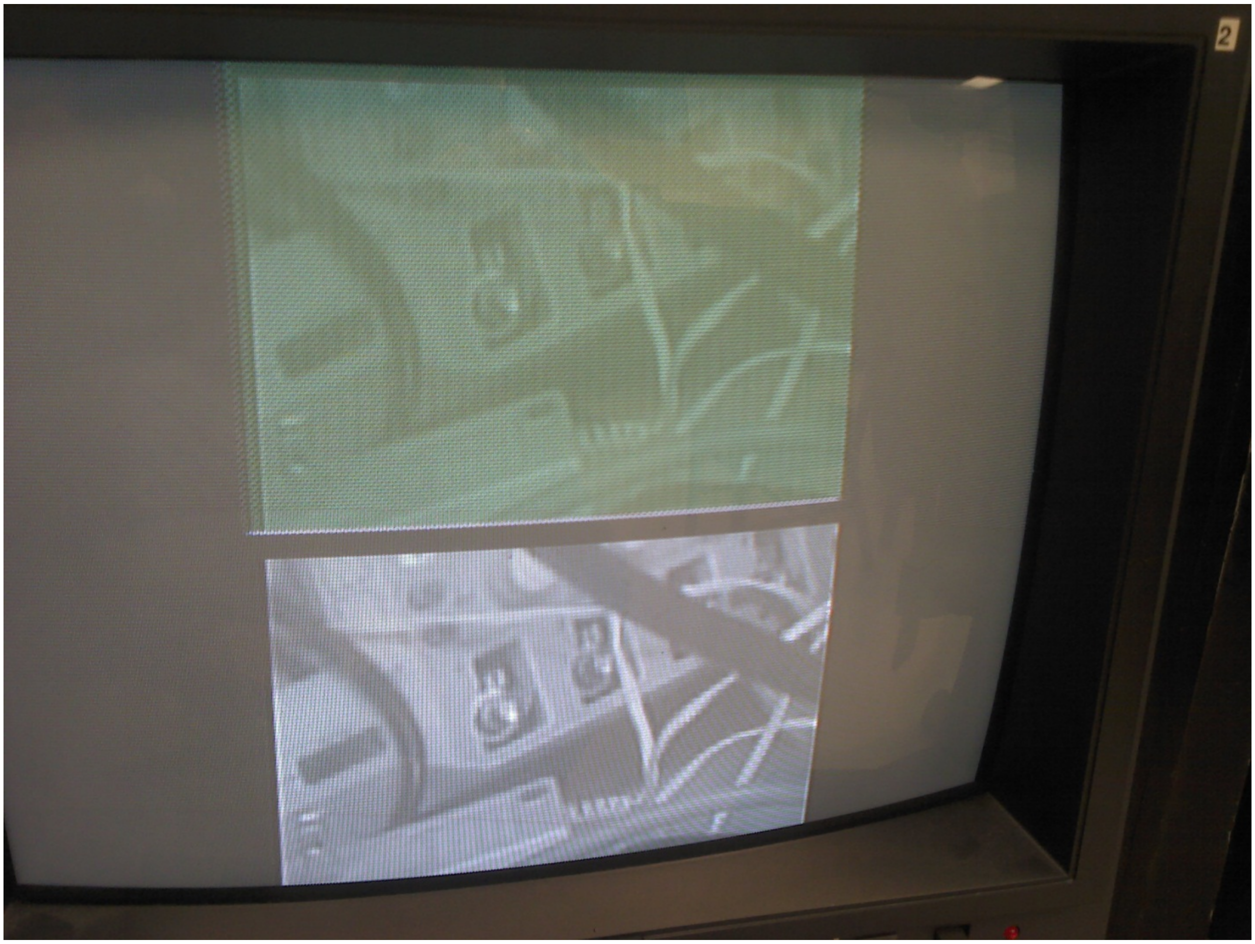

Figure 5: Picture of the display of the camera output for the hardware implementation. The bottom screen is unaltered and the top screen has the night vision effect.

References

Wikipedia. 2011. 12 Dec 2011 < http://en.wikipedia.org/wiki/YCbCr >