Demo: Real-Time Video Focus Identification and Assessment

Edwin Collado and Yessica Saez, TAMU
Course Instructor: Professor Deepa Kundur

Introduction

Video processing is a particular case of signal processing, which often employs video filters and where the input and output signals are video files or video streams. Video processing techniques have several applications. For instance, you could use video data to determine if a car is going to shift to the left or to the right or even to look inside the car and determine if the passenger is an adult or a child (Automotive active safety). The following picture summarizes some of the video processing techniques applications that we may found in our daily life:

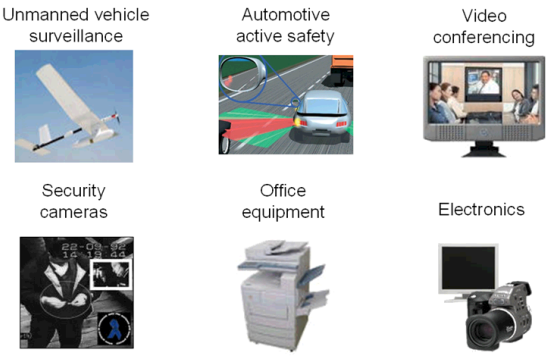

Figure 1: Video Processing Applications [2]

Even though the previous applications (and thus the images) are very different from each other, we are working with images and video data; therefore, we have a lot of commonalities and here is when MATLAB and Simulink come to play. Since the images data are essentially matrices, we can use the power of MATLAB and Simulink to work with images, to analyze them and to develop algorithms.

MATLAB possess a variety of products that we can use to process images and videos. The Image Processing Toolbox can be used to perform image processing, analysis and algorithm development. The Image Acquisition Toolbox acquires images and videos into MATLAB and Simulink and the Image and Video Processing Blockset allows us to design and simulate video and images processing systems.

Video Focus Identification and Assessment

This demo lab uses the Video and Image Processing Blockset to determine whether video frames are in focus using the ratio of the high spatial frequency content to the low spatial frequency content within a region of interest (ROI). When this ratio is high, the video is in focus. When this ratio is low, the video is out of focus.

Region of Interest (ROI)

This demo will show a video sequence (or a video image from a webcam) that is moving in and out of focus. The model uses the Draw Shapes block to highlight an ROI (Region of interest) on the video frames and the Insert Text block to indicate whether or not the video is in focus. The ROI can be determined in a very interactive way by moving the highlighted region until we reach the most interesting part of the image. We will be using constant, slider gain and matrix concatenate blocks to define our ROI.

FFT and Relative Focus

Once we have defined our ROI, we must pass it through a FFT to compute how much high frequency components there are on the video image. The more high frequency data the more in focus the image video will be. On the other hand, if there is a lot of low frequency content then we will have a very blurry image.

The Relative Focus window displays a plot of the ratio of the high spatial frequency content to the low spatial frequency content within the ROI. This ratio is an indication of the relative focus adjustment of the video camera. When this ratio is high, the video is in focus. When this ratio is low, the video is out of focus. Although it is possible to judge the relative focus of a camera with respect to the video using 2-D filters, the approach used in this demo enables you to see the relationship between the high spatial frequency content of the video and its focus.

Design and Implementation

We have described our video processing demo to determine if our video frames are in focus or not. Now it is time for you to design this demo.

First you will design a video focus assessment and test it in Simulink normal mode. You can read up some data from the webcam by using a “From video Device” block, which acquires live image data from an image acquisition device (i.e. a webcam). This block is available in Image Acquisition Tool block. You can also read your data from a video by using a “From Multimedia File” block, which on Windows, reads video frames and/or audio samples from a compressed or uncompressed file. This block is available in Video and Image Processing Blockset→Sources. You can send the output of your system to either a “To Video Display” block or to a “Video Viewer” block. Both blocks will make the output available to you in a separate window and both of these output blocks are available in Video and Image Processing Blockset→Sinks.

Building the Simulink Model

1. Create your New Model. Open Simulink by typing “Simulink” in the command prompt of MATLAB. This will open up the library browser. Then create a new model by going to File→New→Model.
2. Read up data from a webcam. Go to Image Acquisition toolbox and drag a From Video Device block. Double click on it and set the following parameters: In Device select the webcam you are using, set the

Video Format in terms of frame size to 320x240. Set Data Type to double. Go ahead and make sure that your from Video Device is working. Use a Video Viewer or a To Video Display block to observe the data that is coming from the webcam.

3. Look at the information going down from the Video Device to the Video Viewer. Go to Format→Port/Signal Displays→Signal Dimensions. You must see a 320x240x3 size. The 320x240 represents the frame size and the x3 is the red, green and blue components. For our purpose let us work just with gray scale images. To convert this go to Video and Image Processing Blockset→Conversions→Color Space Conversion and place it between the From Video Device and the Video Viewer. Set the Color Space Conversion to go from R'G'B' to intensity. Run the model again and make sure you are seeing a gray input now.

4. Defining the ROI. Go to Video and Image Processing Blockset→Text and Graphics→Draw Shapes and place it between the Color Space conversion and the Video Viewer. Now we need to specify the points that are going to define our Region of Interest (ROI). Go to Simulink→Sources and drag 4 constants blocks, one for row, one for column, one for width and other for height. Set row=1, column=1, width=128, height=128. Connect the output of the row and the column constant blocks to a slider gain each. The slider gain will make you able to select the ROI in a more interactive manner. The slider gain is available in Simulink→ Math operations. Then connect the output of the slider gains, the width and height blocks to a matrix concatenate from Simulink→ Math Operations. Now your Simulink model should look like the following:

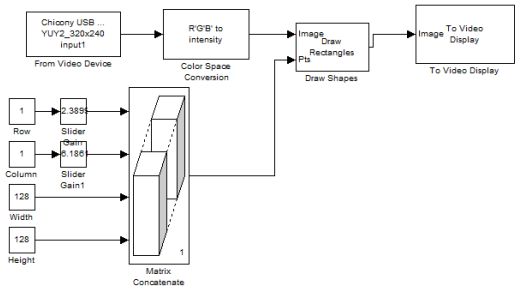

Figure 2: Model so Far

You can take the blocks that define the ROI and create a subsystem to keep the model simpler. Select the blocks→doble click→create subsystem→change the name of the block to ourROI. Now the model should look like this:

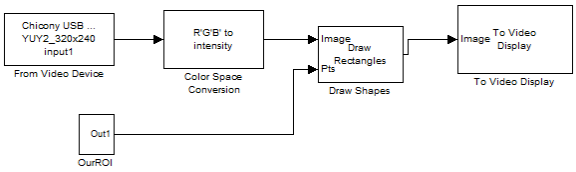

Figure 3: Reduced Model

5. Now is time to pull out the section defined by our ROI that we will be using to take the FFT. Grab a selector block from Simulink → Signal Routing→Selector. Double click on the selector and change the

value of the input dimension to 2. Set the starting indexes to be read from a port and the outputs size to be 128 by 128. Notice that now the selector has two new inputs.

Double click on the ROI subsystem and create two new outputs and connect them to the row and column respectively.

6. Let select the area of the ROI that we will be using to take the FFT. Go ahead and drag a Submatrix block from Signal Processing Blockset →Signal Management→Indexing.

7. Now that you have the cropped section, let us take the FFT of it. Go to Video and Image Processing Blockset→Transform and grab a 2D-FFT block. If you remember, we set the image acquisition block to read from the webcam and output the data type to double. When a variable has a data type double the video and image processing blocks assume that the values should run from 0-1. But the FFT is going to calculate some values grader than 1, which causes the pixels to be saturated and also we have a large DC bias. What we can do is subtract a value of 0.5 from the submatrix output assuring the signal is going to be on top of zero. Grab a sum from Simulink →Math Operations and set the list of sings to +-. Grab a constant block and set it to be 0.5. Connect the output of the constant block to the - input of the sum. Connect the output of the submatrix block to the + sign of the sum block. Finally connect the output of the sum block to the input of the 2D FFT. Also remember that the FFT generates a complex value signal and the video viewer is waiting for a real value signal. Therefore we must place an absolute value block after the FFT block. Grab the math function block from Simulink →Math Operations, set it to be Magnitude ^2. Your Simulink model should now look like the following:

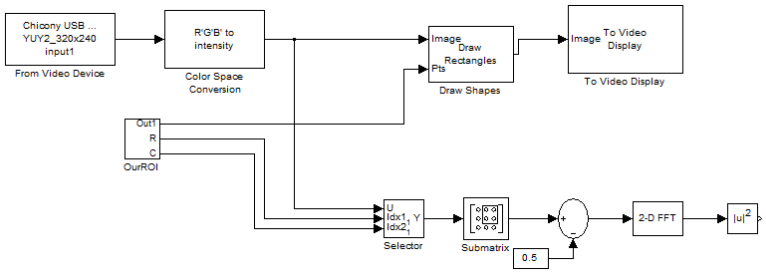

Figure 4: Model so far

Go ahead and create a subsystem of the FFT computation. The model should look like the following:

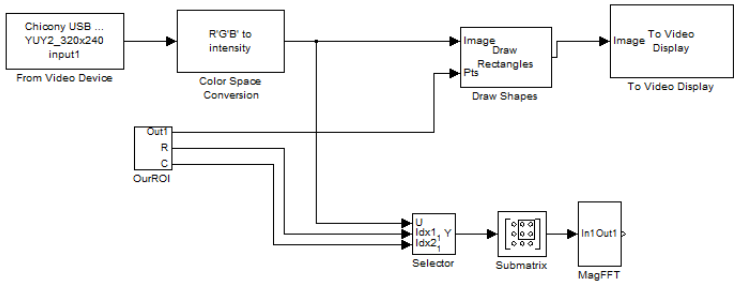

Figure 5: Reduced Model

9. Now is time to select the matrix portion that will contain the frequencies that will be used to compute the ratio of high spatial frequency component to low frequency components and thus to determine if the image is in focus or out of focus. Go ahead and place 3 submatrix blocks in your model. And set up each of them in the following manner:

Submatrix 1: Ending row=index, ending row index=16, Column span=one column.

Submatrix 2: Ending row=index, ending row index=16, start column=index, starting column index=2.

Submatrix 3: Ending row=index, ending row index=16, start column=offset from last, starting column offset=14.

Connect the 3 submatrix blocks to the Magnitude FFT block. Then take the output of submatrix 3 and flip its rows by using a flip block from Signal Processing Blockset →Signal Management→Indexing. Then add the output of the second submatrix to the flipped version of the third one and then concatenate the sum with the output of the first submatrix by using a matrix concatenate block and setting its input number to 2.

Let us take a log plot of the FFT data corresponding to these collected frequencies. Use a math function block (set it to log10), a scaling block (1/8) and a video viewer.

The model should now look like the following:

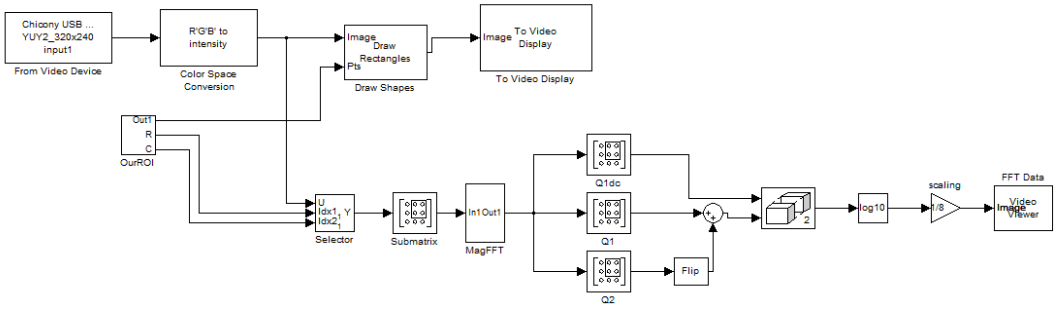

Figure 6: Model so far

Again, try to reduce the complexity of your model by creating subsystems.

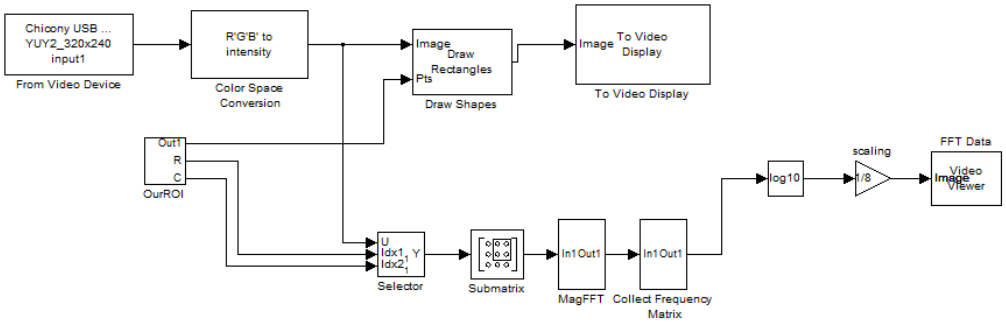

Figure 7: Reduced Model

10. Now is time to compute the high to low frequency components ratio. If you observe the FFT data, you will be able to observe that the low frequency components are toward the corners. So what you can do is cutoff a portion of the upper left corner. To do this place a submatrix block after the collect frequency matrix and set its configuration as the following: Ending row=index, ending row index=5, start column=first, end column index=2. Then change the output of the submatrix to a 2D vector by using a reshape block from Simulink→ Math Operations. Take the output of the reshape block and sum its elements by using a Sum block from Simulink→ Math Operations. Set the list of signs of the sum block to 1. This will compute the low frequency components part.

Now, take the output of the collect frequency matrix subsystem and pass it through a reshape block (without using a submatrix), sum its elements and subtract the low frequency components by using a sum block with list of signs set to -+.

Finally, take the ratio of high to low frequency by using a divide block from Simulink→ Math Operations.

Let us use a Vector Scope block from Signal Processing Blockset→ Signal Processing Sinks to display a plot of the ratio of the high spatial frequency content to the low spatial frequency content within the ROI. This ratio is an indication of the relative focus adjustment of the video camera. When this ratio is high, the video is in focus. When this ratio is low, the video is out of focus.

Your model should look like the following:

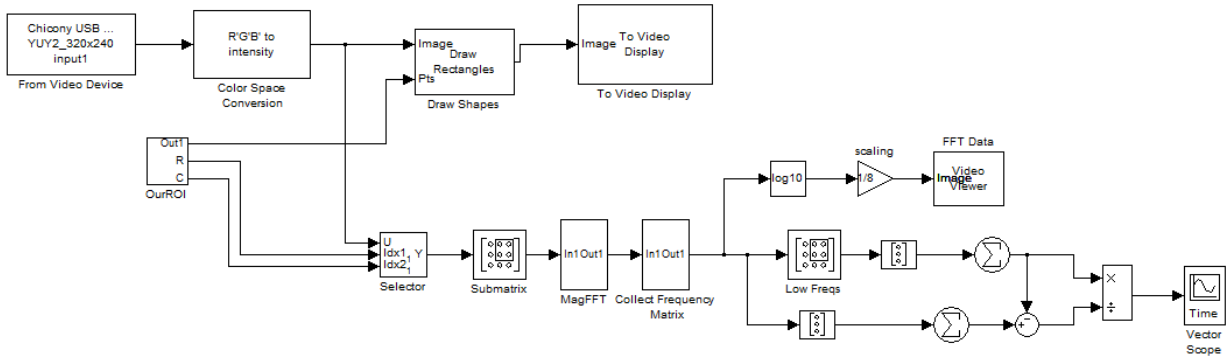

Figure 8: Model so far

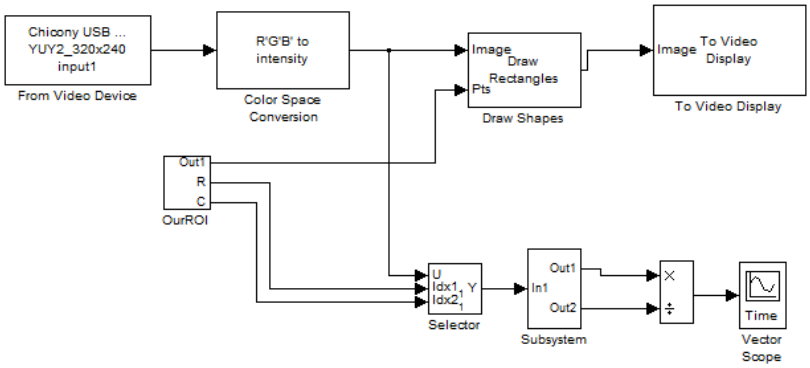

Figure 9: Reduced Model

11. The final step in our model is to Insert Text block to indicate whether or not the video is in focus. Use a relational operator block from Simulink→ Logic and bit Operations to compare the ratios with a constant=0.03. Use an insert text block from Video and Image Processing Blockset→Text and Graphs. Set the test to :{'Out of focus', 'In focus'} and location to [10 100]. In this way if the ratio is greater than 0.03 the text: In Focus will appear in the specified location otherwise the text: Out of focus will appear. Connect the draw shape output to the image input and the ratio to the select input of the Insert Text block and its output to the video viewer (or to video display).

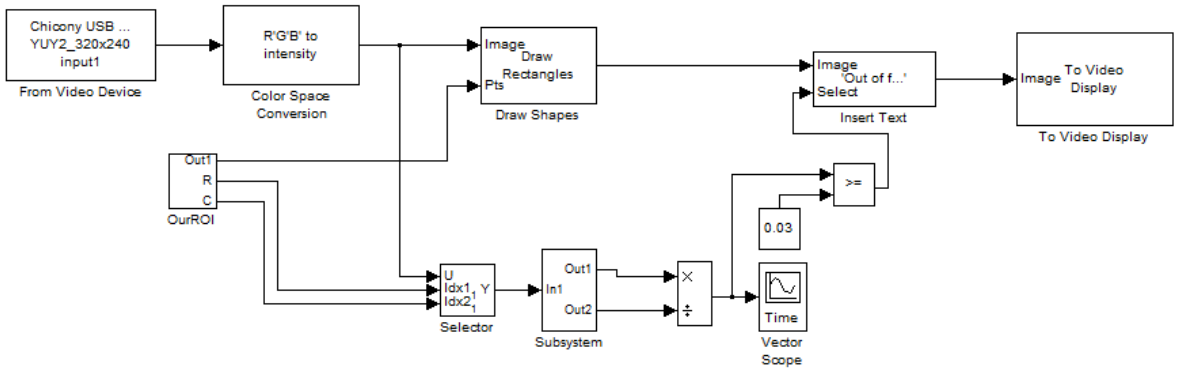

Figure 10: Simulink Model

Again, we can reduce the complexity of the model by creating subsystem. Try to get your model to look like the following, with the same subsystem block names.

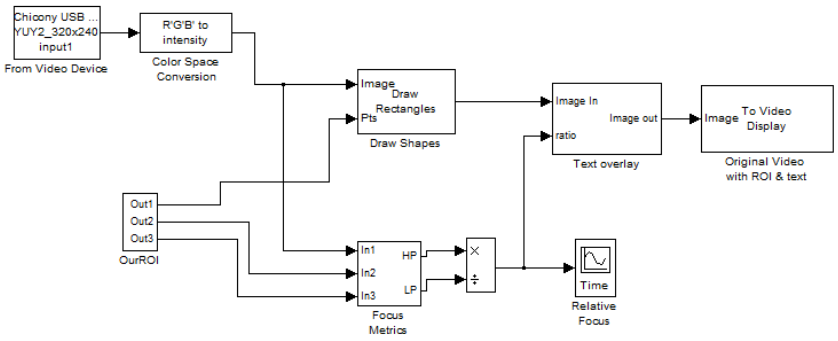

Figure 11: Final Simulink Model Block Diagram

Digital Video Hardware Implementation

1. Go to the Simulation→Configuration Parameters, change the solver options type back to "Fixedstep", solver to "discrete (no continuous states)."

2. Start with your Simulink Model for digital video edge detection and delete all the input and output blocks.

3. Under Target Support Package→Supported Processors→TI C6000→Board Support→DM6437 EVM, found a Video Capture block. Set the Sample Time to -1.

4. Add a Video Display block. For the first one, set Video Window to "Video 0", Video Window Position to [0, 0, 720, 480], Horizontal Zoom to 2x, Vertical Zoom to 2x.

5. Add two Image Data Type Conversion blocks from the Video and Image Processing Blockset →Conversions. Link the input of the first one to the output of Video Capture Block and its output to the input of the Draw Shapes block. Set the output data type to "double". For the other Image Data Type Conversion Block set the output data to "uint8" and link its input to the Text Overlay Subsystem and its input to the Video Display block.

6.Since we are developing a model to run on an embedded target that uses a row-major format and all our video and Image Processing blocks use column-major data organization, we need to select the **Input image is transpose (data order is row major)** check box on our Insert Text block. In that way, the insert text block gives us the option to process data that is stored in row-major format. Also change the location to [180 180] so we can see the text centered on the screen.

7. Adding a DM6437EVM Block under Target Support Package TI C6000→Target Preferences.

8. Connect the video output (the yellow port) of the camera with the video input of the board, and the output of the board with the monitor.

9. Go to Tools->Real Time Workshop -> Build the model.

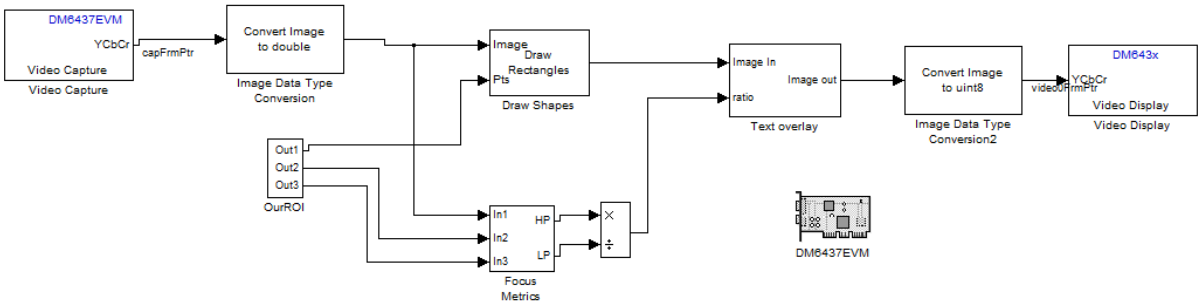

Figure 12: Real-Time Video Focus Identification and Assessment Model to be Implemented.

Simulation Results

The following picture shows the FFT data used to compute how much high frequency components there are on the video image.

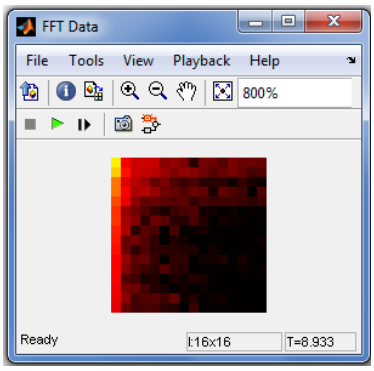

Figure 13: FFT Data

The Relative Focus window presented below displays a plot of the ratio of the high spatial frequency content to the low spatial frequency content within the ROI. This ratio is an indication of the relative focus adjustment of the video camera. When this ratio is high, the video is in focus. When this ratio is low, the video is out of focus.

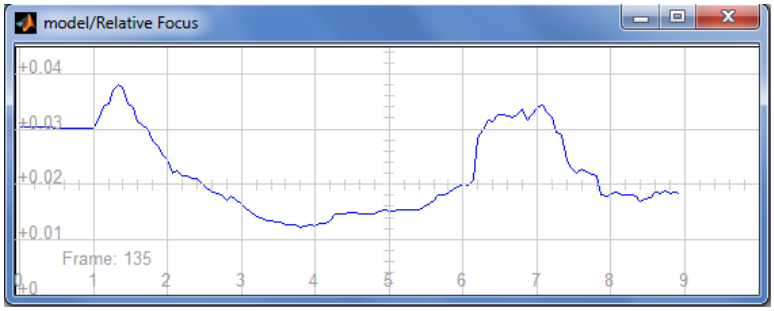

Figure 14: Relative Focus Window

The following windows show the results of our Video Focus Identification and Assessment demo.

We took a video sequence that was moving in and out of focus, we observe that our demo highlights our ROI (Region of interest) on the video frames and then it inserts a Text indicating whether or not the video is in focus.

Figure 15: Video with ROI and Text (Software Implementation)

Hardware Implementation Results

Using the webcams of our DSP laboratory we took a real-time video sequence that was moving in and out of focus.We observe that our demo highlights our ROI (Region of interest) on the video frames and then it inserts a Text indicating whether or not the video is in focus.

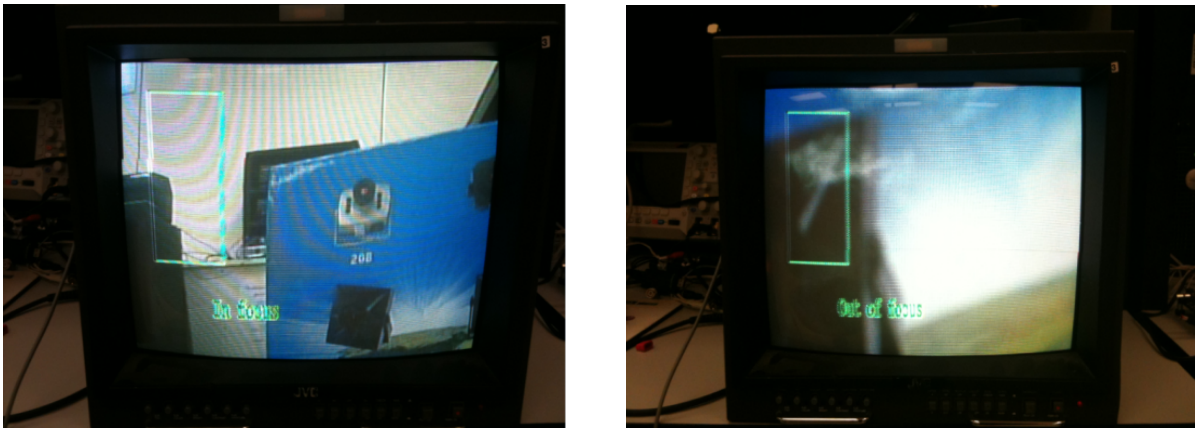

Figure 16: Video with ROI and Text (Hardware Implementation)

References

1. Bovik, Al (ed.). Handbook of Image and Video Processing. San Diego: Academic Press, 2000.
2. The MathWorks, Inc. Webinar recorded on 9/18/2008. Webinar accessed on 10/20/2011 at http://www.mathworks.com/company/events/webinars/wbnr30960.html?id=30960&p1=65234&p2=65236.
3. MATLAB 2009b→ Help→Demos→Video Focus Assestment.