Demo: Real-time Edge-Detection-Based Motion Tracking

Armando Nava and David Yoon, TAMU
Course Instructor: Professor Deepa Kundur

Introduction

This demo is intended to build on Lab 6 which discusses the mathematical formulation of various edge detection algorithms by applying them to a real-time video processing task of tracking the motion of an object. Motion tracking is a mainstay of some of the most important video processing applications such as human-computer interaction, security and surveillance, video communication and compression, augmented reality, traffic control, medical imaging, and video editing [1].

The motion tracking algorithm to be implemented in this demo can be broken up into three main stages: background subtraction, edge detection, and blob analysis.

Background Subtraction

In most videos, the objects of interest reside in the foreground of a scene where things are happening and movements are taking place. Therefore, in order to track the motion of these objects, it is necessary to extract and distinguish them from the static background before any further processing. There are many sophisticated algorithms for subtracting the background of a scene such as Pfinder, Halevy, and Cutler, but these algorithms tend to be computationally expensive [2].

A naïve yet effective and computationally efficient approach is to compute the difference between the current frame and the preceding frame of a video. The intensity of the pixels corresponding to the static background remains largely unchanged between two consecutive frames. By computing the difference, these pixels get canceled out and only those of moving foreground objects are retained. A visual example of this approach is shown in Figure 1 below.

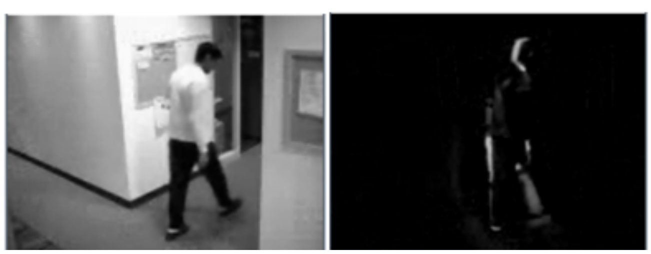

Figure 1: Background subtraction by computing difference between two consecutive frames

Edge Detection

Now that the objects of interest have been extracted from the static background, their edges are detected. Edge detection is the process of identifying sharp changes in the intensity of an image [3]. One approach to detecting the edges of an image is to compute the magnitude of the gradient of an image and compare its value to a threshold. The gradient measures the rate of change of a multidimensional signal such as an image. The edges of an image are located at points where the magnitude of the rate of change in intensity is greater than a certain threshold value. The threshold determines how many edges will be detected in an image. An example of edge detection using the Sobel algorithm which is based on gradient computation is shown in Figure 2 below.

Figure 2: Edge detection using the Sobel algorithm

Blob Analysis

Finally, blob analysis is performed to pinpoint the coordinates of the objects that have been identified by their edges. Blob analysis refers to the process of identifying points or regions of an image that differ from the surrounding area in properties such as color and intensity [4]. The mathematical formulation of blob analysis algorithms is beyond the scope of ECEN 448 and will not be discussed.

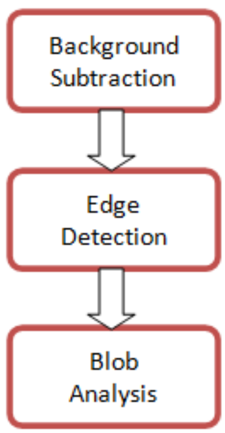

Figure 3: Motion tracking algorithm based on edge detection

Design and Implementation

Buildling the Simulink Model

1. In your new Simulink model, add a From Multimedia File block from Signal Processing Blockset → Signal Processing Sources. Double click on the block. Click on Browse and locate viptraffic.avi, then click Open. Under Outputs, change Image Color Space to Intensity.
2. Add a Difference block from Simulink → Discrete. Connect the output of the From Multimedia File block to the input of the Difference block.
3. Add an Edge Detection block from Video and Image Processing Blockset → Analysis & Enhancement. Double click on the block and choose any method you prefer. Set the Threshold Scale Factor to 4. Under the Fixed-point tab, set Rounding Mode to Floor, Overflow Mode to Wrap, Product Output to Binary point scaling, Word Length to 32, Accumulator to Same as product output. Uncheck Lock scaling against

changes by the auto scaling tool. Click OK. Connect the output of the Difference block to the Edge Detection block.

4. Add a Blob Analysis block from Video and Image Processing Blockset → Statistics. Double click the block. Check Bounding Box and uncheck everything else. Set Statistics output data type to double, Connectivity to 8, and uncheck Output label matrix. Under the Blob properties tab, set the Maximum number of blobs to 5, check Specify minimum blob area in pixels and set the value to 50, check Fill empty spaces in outputs, and set Fill Values to -1 (By specifying the maximum number of blobs and the minimum blob area, spurious motions are ignored). Click OK and connect the output of the Edge Detection block to the input of the Blob Analysis block.
5. Add a Draw Shapes block from Video and Image Processing Blockset → Texts & Graphics. Double click on the block. Set Shape to Rectangles, uncheck Fill shapes, set Border color Source to Specify via Dialog, Border Color to Black, Draw shapes in to Entire Image, Image Signal to One multidimensional signal. Click OK. Connect the output of the Blob Analysis block to the input Pts. Connect the output of the From Multimedia File block to the Image port of the Draw Shapes block.
6. Add four To Video Display blocks from Video and Image Processing Blockset → Sinks. Connect one to the output of the Draw Shapes block, another to the output of the From Multimedia File block, another to the output of the Difference block, and the last one to the output of the Edge Detection block.
7. Press Ctrl-E. Under Solver → Simulation Time, set the Stop time to inf. Click OK.
8. Click on the Play button.

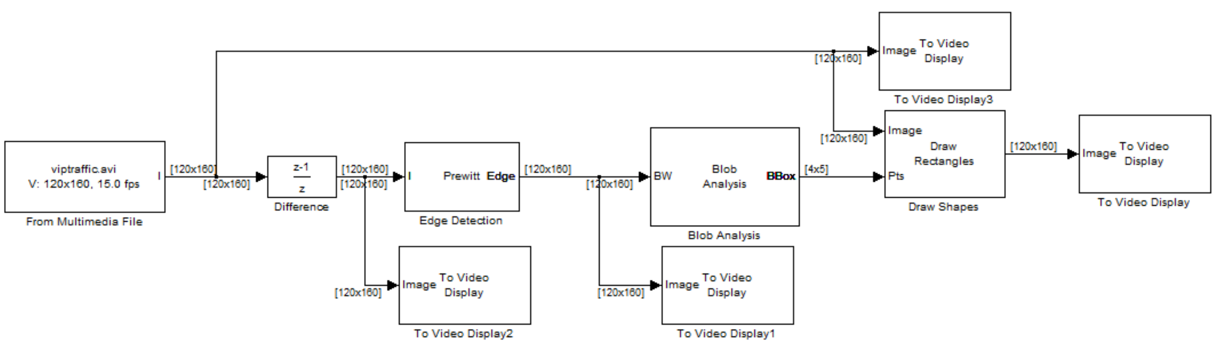

Figure 4: Simulink model of the motion tracking algorithm based on edge detection

Digital Video Hardware Implementation

1. In the previous Simulink model, delete all To Video Display blocks.
2. Replace the From Multimedia File block with a Video Capture block and a Deinterleave block from Target Support Package → Supported Processors → TI C6000 → Board Support → DM6437EVM. Connect the output of the Video Capture block to the input of the Deinterleave block. Set the sample time of the Video Capture block to -1. Connect the Y output of the Deinterleave block to the Difference block and to the Image port of the Draw Rectangles block.
3. Add an Interleave block from Target Support Package → Supported Processors → TI C6000 → Board Support → DM6437EVM. Connect the outputs of the Deinterleave block to the corresponding input ports of the Interleave block.
4. Add a Video Display block from Target Support Package → Supported Processors → TI C6000 → Board Support → DM6437EVM. Set Video Window to Video 0, Video Window Position to [180, 0, 360, 480],

Horizontal Zoom to 2x, and Vertical Zoom to 2x. Connect the output of the Interleave block to the Video Display block.

5. Add a second Interleave block. Connect the output of the Draw Rectangles block to the Y port of the second Interleave block.
6. Add a second Video Display block. Set Video Window to Video 1, Video Window Position to [0, 240, 360, 240], Horizontal Zoom to 2x, and Vertical Zoom to 2x. Connect the output of the second Interleave block to the second Video Display block.
7. Link the Cb and Cr channels between the Deinterleave block and both Interleave blocks.
8. Add a DM6437EVM block from Target Support Package → Supported Processors → TI C6000 → Board Support → DM6437EVM.
9. Connect the video output (the yellow port) of the camera to the video input of the board. Connect the output of the board to the monitor.
10. Press Ctrl-B to compile the code and program the model onto the board.

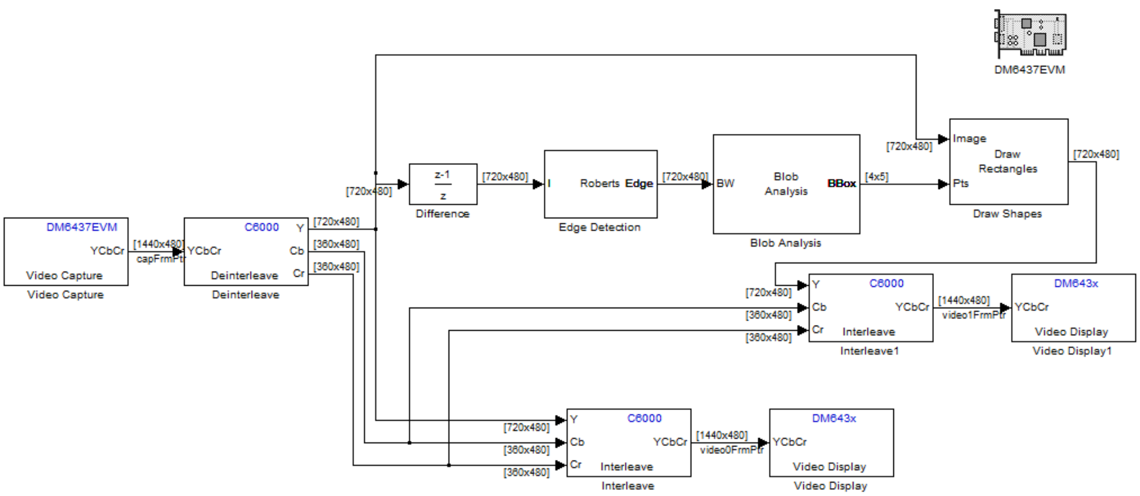

Figure 5: Modified Simulink model of the motion tracking algorithm to be implemented on the board

Results

The results of the simulation are shown in Figures 6, 7, 8, and 9 below. Figure 6 shows the original video frame. Figure 7 shows the frame with its background removed. Figure 8 shows the edges of moving cars. Finally, Figure 9 shows the moving cars being constantly tracked by rectangular boxes.

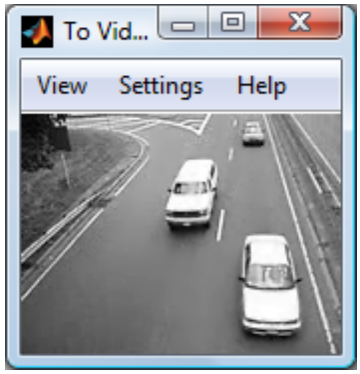

Figure 6: Original video frame

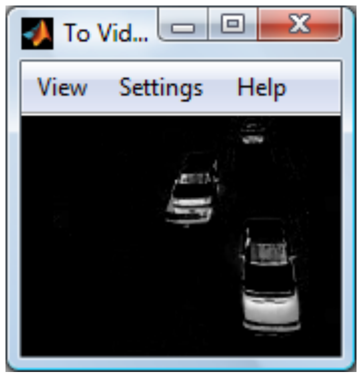

Figure 7: Video frame after background subtraction

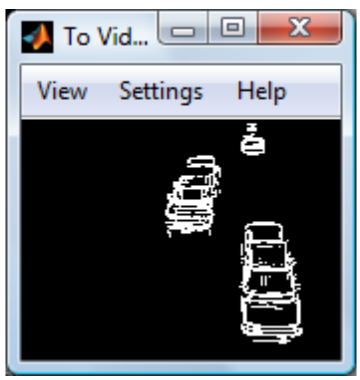

Figure 8: Video frame after edge detection

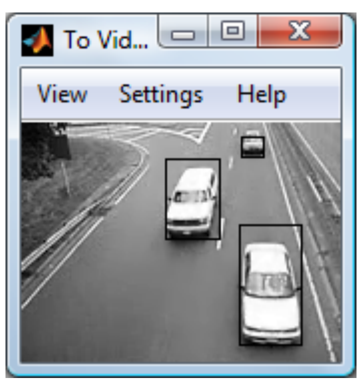

Figure 9: Moving cars being tracked by rectangular boxes

References

[1] “Video Tracking.” *Wikipedia*. 10 Dec. 2011 < http://en.wikipedia.org/wiki/Video_tracking>.

[2] McIvor, Alan. “Background Subtraction Techniques.” *California State University at East Bay*. 10 Dec. 2011 < http://algebra.sci.csueastbay.edu/~tebo/Classes/6825/>.

[3] “Edge Detection.” *Wikipedia*. 8 Dec. 2011 < http://en.wikipedia.org/wiki/Edge_detection>.

[4] “Blob Detection.” *Wikipedia*. 9 Dec. 2011 < http://en.wikipedia.org/wiki/Blob_detection>.