

Demo: Object Tracking through Background Estimation

Cheng Cheng and Bradley Reid, TAMU
Course Instructor: Professor Deepa Kundur

Introduction

Real time object tracking has many practical uses for every type of implementation. For example, a moving camera mounted on the hood of a car could track still traffic signs and report their distance and message to an on-board computer. However, we will be implementing the opposite: a still camera that tracks moving objects. This could be used to highlight a moving object in a news video or in security footage.

Our method of object detection involves finding the background and tracking what is not a part of it. To do this, we take a median over time of each pixel of the video. This ensures that any large changes are pushed to either side of the median and we are left with the background. We then subtract the background from the image to get what remains in the foreground. Technically, what we get is not the image of the foreground, but the difference between the background and the foreground. This unfortunately means that if an object in the foreground is visually close enough to the background, this method will not detect it.

Once we have the difference between the whole picture and the background, we take the absolute value of it to eliminate negatives. We then establish a threshold – anything above it is in the foreground and becomes white, anything below it is in the background and becomes black. Next we must close holes in the white images to make whole objects.

Luckily for us, Simulink can take care of the next part all by itself. Simulink can perform blob detection on the white objects and determine the points needed to draw a rectangle around them. We can then take those points and draw rectangles on the original image to produce an output where objects that are not a part of the background have rectangles drawn around them, even as they move around.

Design and Implementation

Building the Simulink Model

1. Insert a From Multimedia File block. Change the image color space to RGB and the signal to one multidimensional signal. Under the Data Types tab, make sure the video output data type is single.
2. Insert a Color Space Conversion block. Change the conversion to RGB to intensity and the signal to one multidimensional signal. Connect the input of this block to the output of the From Multimedia File block.
3. Insert a Reshape block and change the output dimensionality to Row Vector (2-D). Connect the input of this block to the output of the Color Space Conversion block.
4. Insert a Buffer block. Change the buffer size to 30 and the buffer overlap to 29. Connect the input of this block to the output of the Reshape block.

5. Insert a Median block. Change the median value to over each column. Under the Data Type Attributes tab, change the accumulator data type to `fixdt([],32,0)`. Connect the input of this block to the output of the Buffer block.
6. Insert a Frame Conversion block and change the sampling mode to sample based. Connect the input to the output of the Median block.
7. Insert another Reshape block. Customize the output dimensionality to the dimensions of your video. Connect the input to the output of the Frame Conversion block.
8. Insert an Add block and change the list of signs to `+-`. Connect the `-` input to the output of the second Reshape block and the `+` input to the output of the Color Space Conversion block.
9. Highlight all the blocks between and including the two Reshape blocks (Reshape, Buffer, Median, Frame Conversion, Reshape). Right click and select Make Subsystem. Open this subsystem and insert an Enable block.
10. Back in the general block diagram, insert an Abs block and connect the input to the output of the Add block.
11. Insert an Autothreshold block and connect the input to the output of the Abs block. All the parameters for this block should already be correct.
12. Insert a Closing block. Change the neighboring or structuring element source to specify via dialog and change the element to `strel('square',5)`. Connect the input of this block to the output of the Autothreshold block.
13. Insert a Blob Analysis block. Make sure only Bounding Box is selected under Statistics and the output data type is double. Under Blob Properties, change the maximum number of blobs to 80, the minimum blob area to 200, and the maximum blob area to 2000. Connect the input of this block to the output of the Closing block.
14. Insert a Draw Shapes block. Change the shape to rectangles, the border color source to specify via dialog, and the border color to whatever you want. Make sure to draw shapes in the entire image and that the image signal is one multidimensional signal. Connect the Pts input to the output of the Blob Analysis block and the Image input to the output of the From Multimedia File block.
15. Insert a To Video Display block and connect the input to the output of the Draw Shapes block.
16. Highlight the Abs, Autothreshold, Closing, Blob Analysis, Draw Shapes, and To Video Display blocks. Right click and select Make Subsystem. Open this subsystem and insert an Enable block.
17. Back in the general block diagram, insert an N-Sample Enable block. Change the Trigger Count to 30 and the Active Level to Low. Connect the output to the enable of the first subsystem.
18. Insert a Logical Operator block and change the operator to NOT. Connect the input to the output of the N-Sample Enable block. Connect the output to the enable of the second subsystem.

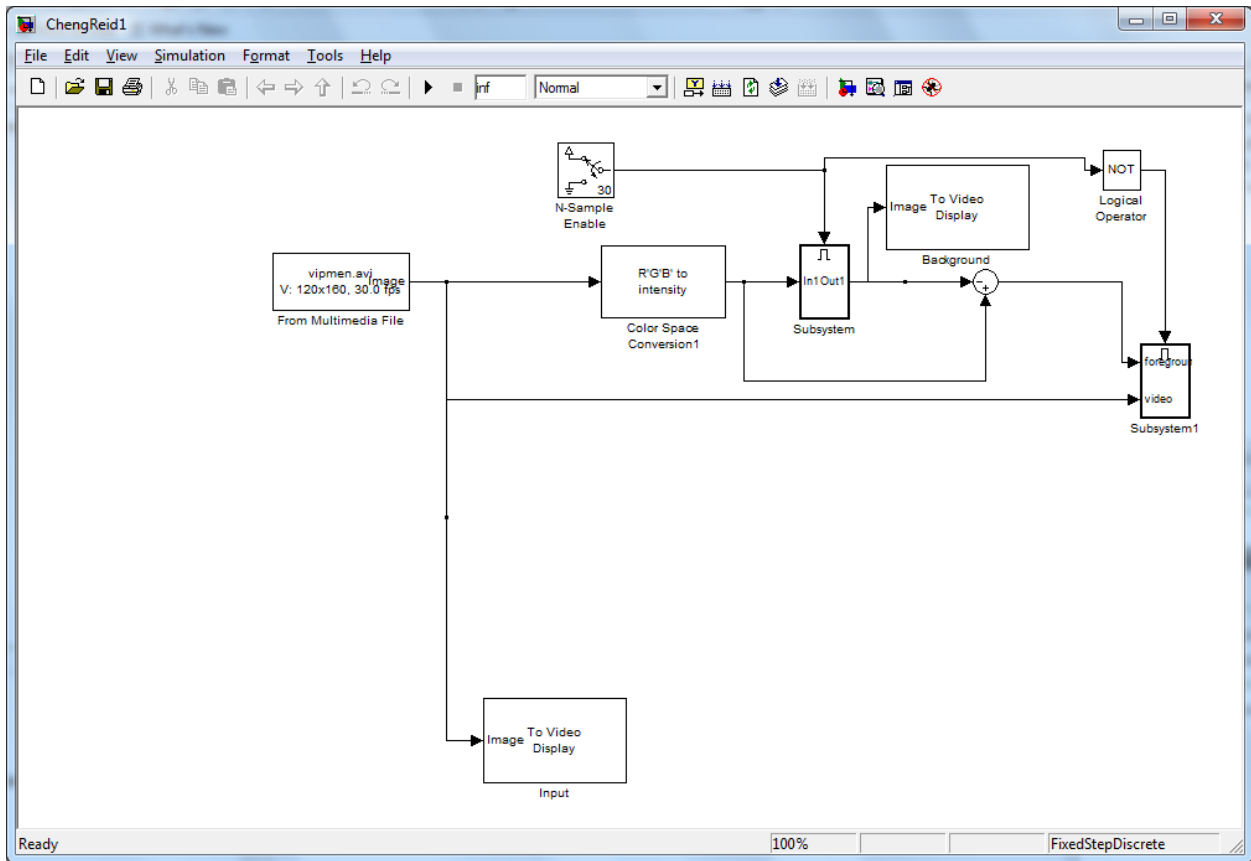
Digital Video Hardware Implementation

1. Insert a DM6437EVM block.
2. Insert a Video Capture block.
3. Insert a Deinterleave block and connect the input to the output of the Video Capture block.
4. Insert two Matrix Concatenate blocks. Change the concatenate dimension for both to 1. Connect both inputs of one to the Cb output of the Deinterleave block and both outputs of the other to the Cr output.
5. Insert a Color Space Conversion block, change the conversion to YCbCr to RGB, and make sure the image signal is separate color signals. Connect the Y input to the Y output of the Deinterleave block, the Cb input to the output of the Matrix Concatenate block with double Cb inputs, and the Cr input to the output of the Matrix Concatenate block with double Cr inputs.

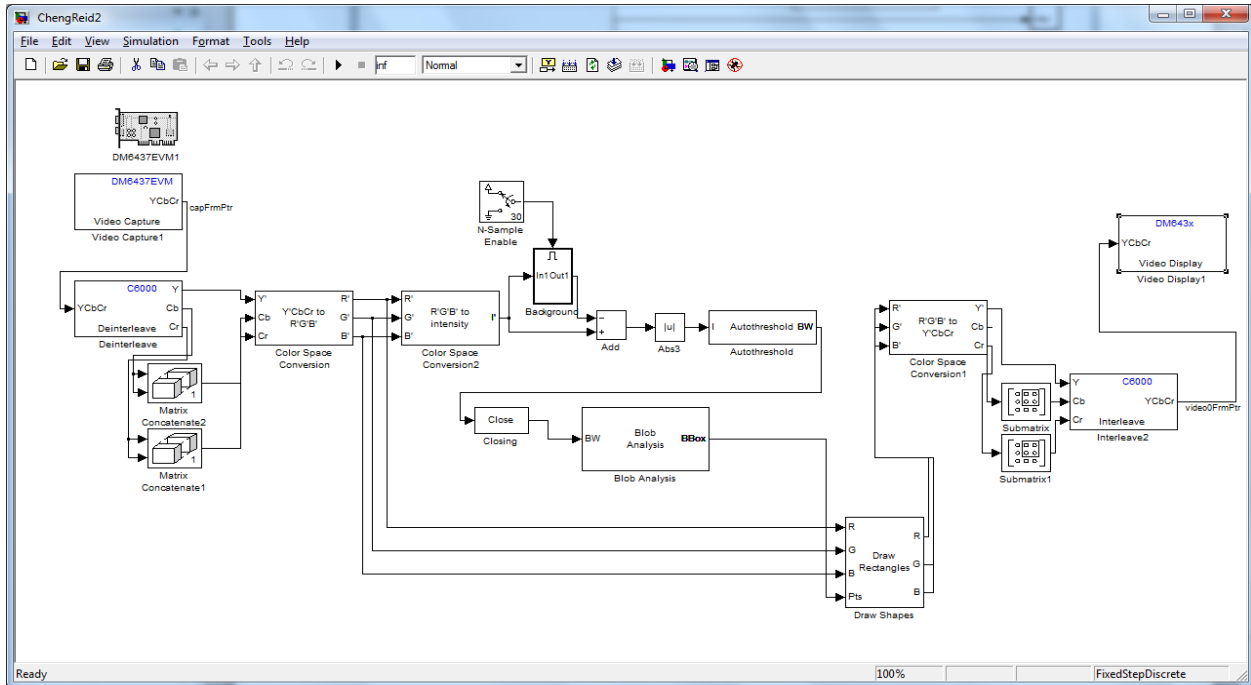
6. Insert another Color Space Conversion block, change the conversion to RGB to intensity, and make sure the image signal is separate color signals. Connect the R, G, and B inputs to the R, G, and B outputs of the previous Color Space Conversion block.
7. Insert a Reshape block and change the output dimensionality to Row Vector (2-D). Connect the input to the output of the previous Color Space Conversion block.
8. Insert a Buffer block and change the buffer size to 30 and the buffer overlap to 29. Connect the input to the output of the Reshape block.
9. Insert a Median block. Change the median value to over each column. Under the Data Type Attributes tab, change the accumulator data type to `fixdt([],32,0)`. Connect the input of this block to the output of the Buffer block.
10. Insert a Frame Conversion block and change the sampling mode to sample based. Connect the input to the output of the Median block.
11. Insert another Reshape block. Customize the output dimensionality to the dimensions of your video. Connect the input to the output of the Frame Conversion block.
12. Insert an Add block and change the list of signs to `+-`. Connect the `-` input to the output of the second Reshape block and the `+` input to the output of the second Color Space Conversion block.
13. Highlight all the blocks between and including the two Reshape blocks (Reshape, Buffer, Median, Frame Conversion, Reshape). Right click and select Make Subsystem. Open this subsystem and insert an Enable block.
14. Back in the general block diagram, insert an N-Sample Enable block. Change the Trigger Count to 30 and the Active Level to Low. Connect the output to the enable of the subsystem.
15. Insert an Abs block and connect the input to the output of the Add block.
16. Insert an Autothreshold block and connect the input to the output of the Abs block.
17. Insert a Closing block. Change the neighboring or structuring element source to specify via dialog and change the element to `strel('square',5)`. Connect the input of this block to the output of the Autothreshold block.
18. Insert a Blob Analysis block. Make sure only Bounding Box is selected under Statistics and the output data type is double. Under Blob Properties, change the maximum number of blobs to 80 and check Fill empty spaces in outputs. Everything else under Blob Properties should be unchecked. Connect the input of this block to the output of the Closing block.
19. Insert a Draw Shapes block. Change the shape to rectangles, the border color source to specify via dialog, and the border color to whatever you want. Make sure to draw shapes in the entire image and that the image signal is separate color signals. Connect the Pts input to the output of the Blob Analysis block and the R, G, and B inputs to the R, G, and B outputs of the first Color Space Conversion block.
20. Insert another Color Space Conversion block and change the conversion to RGB to YCbCr. Connect the R, G, and B inputs to the R, G, and B outputs of the Draw Shapes block.
21. Insert two Submatrix blocks. Change the row span for both to range of rows, the starting row to the first row, and the ending row to the height of your video divided by two. Make sure the column span for both is all columns. Connect the input of one to the Cb output of the last Color Space Conversion block and the input of the other to the Cr output.
22. Insert an Interleave block. Connect the Y input to the Y output of the last Color Space Conversion block, the Cb input to the output of the Submatrix block with the Cb input, and the Cr input to the output of the Submatrix block with the Cr input.

23. Insert a Video Display block. Change the video window position to [0, 0, 480, 480] and the horizontal and vertical zoom to 1x.

Results



Simulink block diagram for software implementation



Simulink block diagram for hardware implementation

Unfortunately, our implementation does not work perfectly. Our main problem in the hardware implementation is lag. The system takes almost a full second to get a new frame and analyze it for background and object detection. This results in a loss of accuracy and frustrating testing. In both implementations, we also ran into a problem with the closer block not doing enough – multiple rectangles end up being drawn around one object because the blob detection reads many white areas in the same object. We believe that, given more time, we could address these problems.

Despite these issues, the Simulink models works as intended. Figure 1 a) and b) show screenshots featuring an input and corresponding output of the software implementation system. Admittedly, these were lucky shots. Generally, throughout the video, the system would draw multiple rectangles over the people as they passed through the picture.

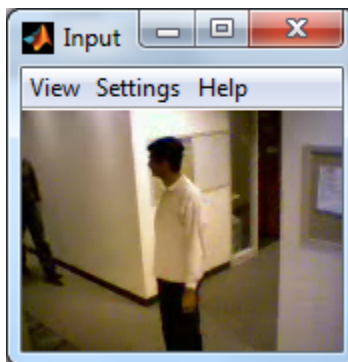
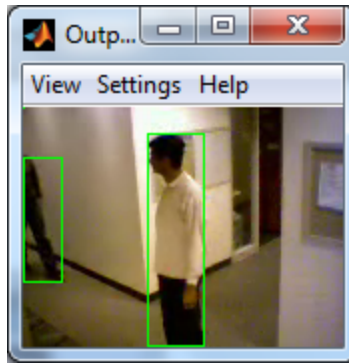


Figure 1a) Input



a) Output

References

“mathworks.com” [MathWorks – MATLAB and Simulink for Technical Computing](http://www.mathworks.com/products/demos/shipping/vision/viptraffic.html?product=VP)

12

December

2011

<http://www.mathworks.com/products/demos/shipping/vision/viptraffic.html?product=VP>