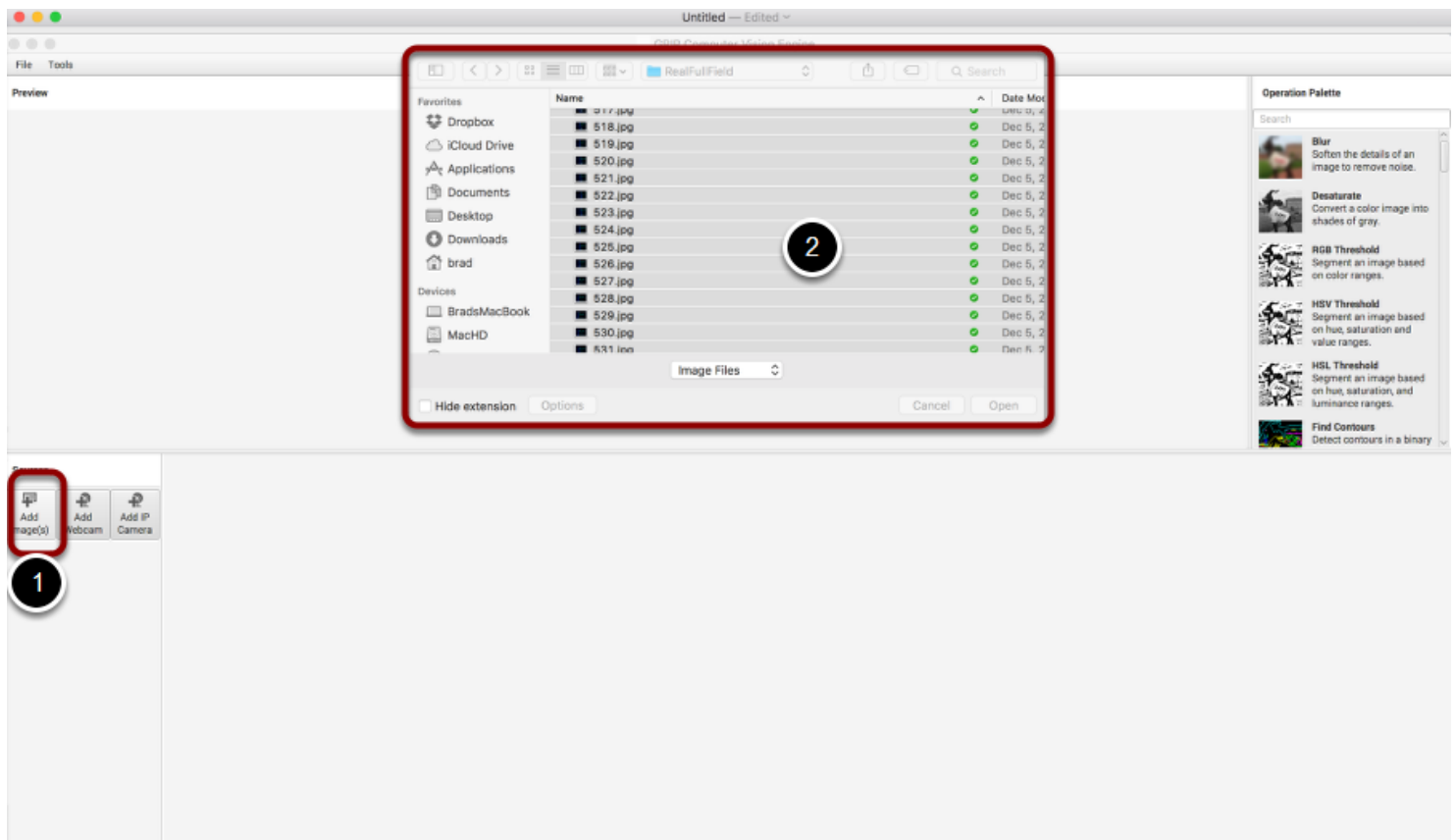


Using GRIP for the 2016 game

GRIP can be used to locate goals for the FIRST Stronghold by using a series of thresholding and finding contours. This page describes the algorithm that was used. You should download the set of sample images from the WPILib project on <http://usfirst.collab.net>. The idea is to load up all the images into a multi image source and test your algorithms by cycling through the pictures.

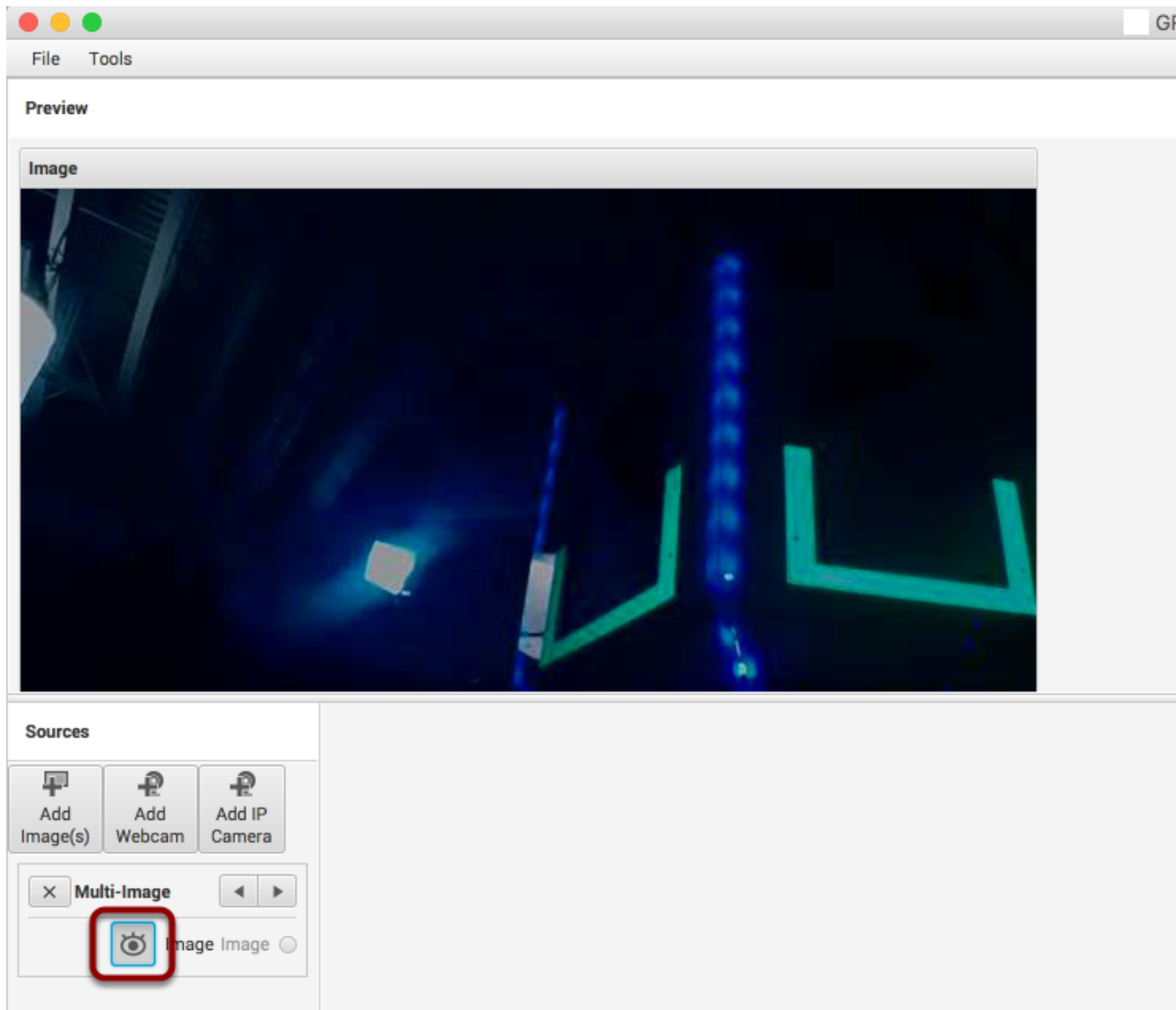
Select all the vision samples as multi image source



Click the Add images button to create the multi-image source, then select all the images from the directory where they were unzipped.

Using GRIP for the 2016 game

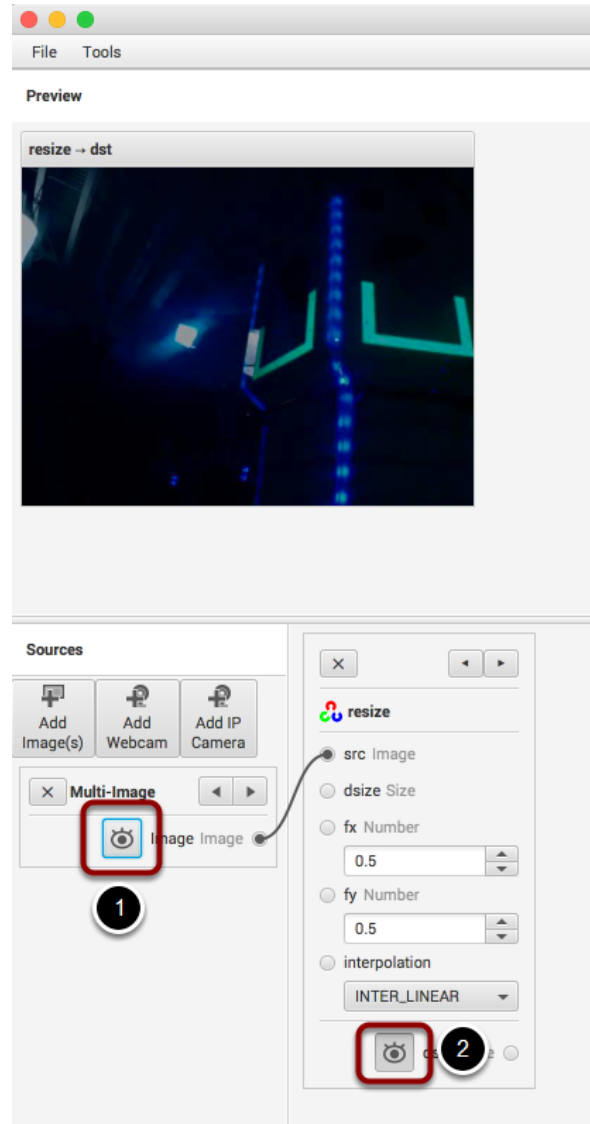
Images are too big (maybe)



I decided that in this example the images were too big and would work just as well if they were resized. That may or may not be true, but it made the pictures small enough to capture easily for this set of examples. You can decide based on your processor and experience whether you want to resize them or not. The images are previewed by pressing the preview button shown above.

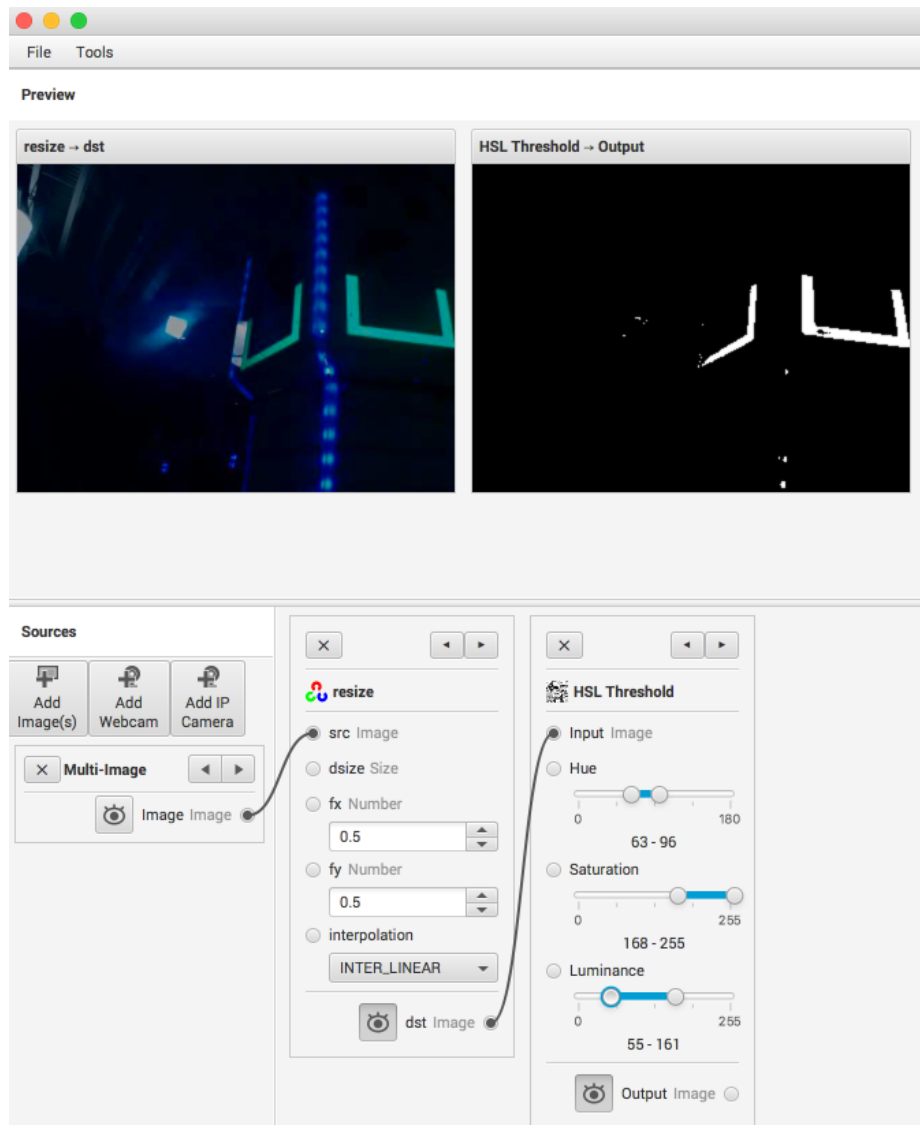
Using GRIP for the 2016 game

Resized images



Changing the scale factor by 0.5 reduces the images to 1/4 the number of pixels and makes processing go much faster, and as said before, fits much better for this tutorial. In this example, the full size image preview is turned off and the smaller preview image is turned on. And, the output of the image source is sent to the resize operation.

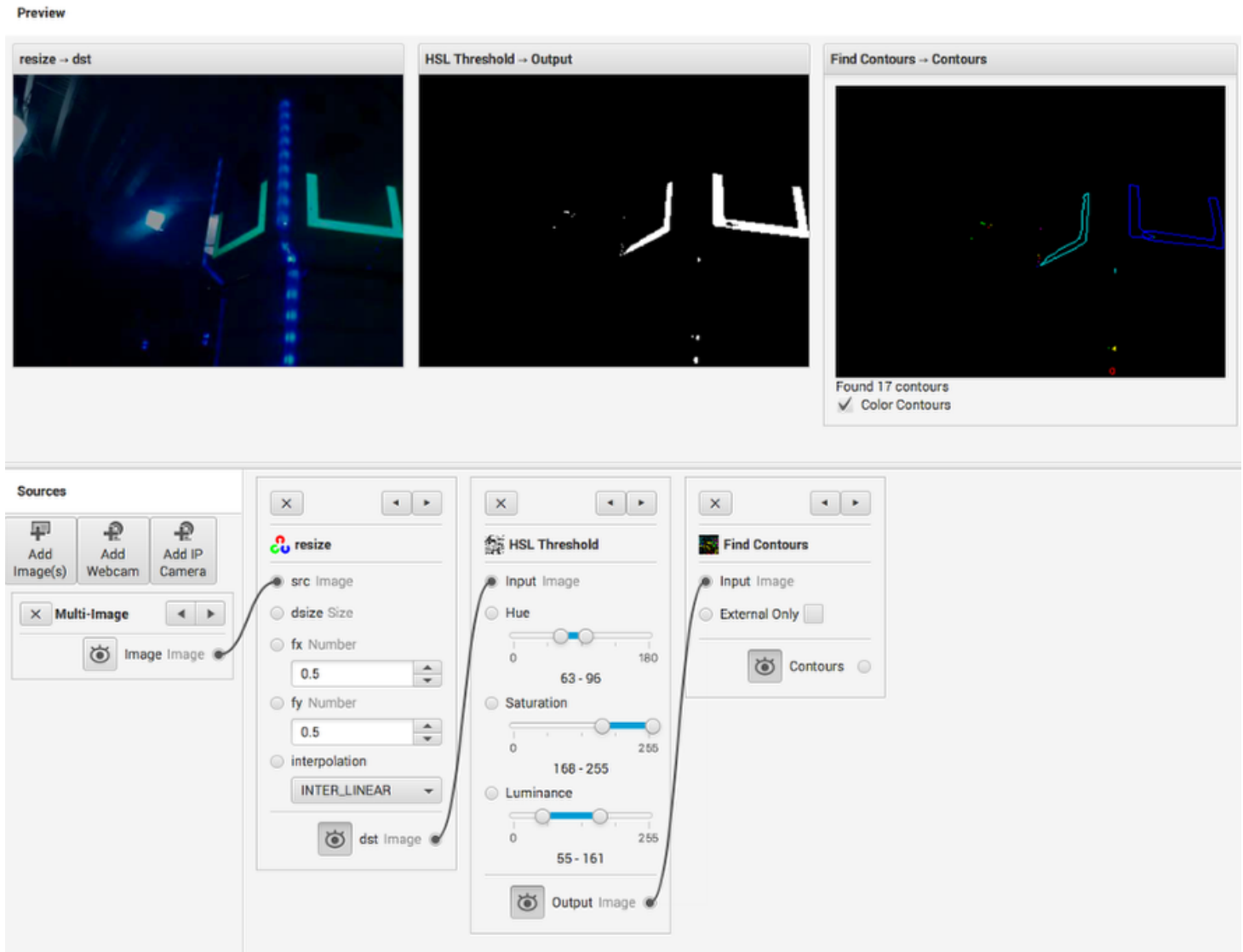
Use HSV Threshold operation to detect the targets



Since the targets were illuminated with a constant color ringlight, it's possible to do a threshold operation to detect only the pixels that are that color. What's left is a binary image, that is an image with 1's where the color matched the parameters and 0's where it didn't. By narrowing the H, S, and L parameters, only the target is included in the output image. You can see that there are some small artifacts left over, but mostly, it is just the vision targets in the frame.

Using GRIP for the 2016 game

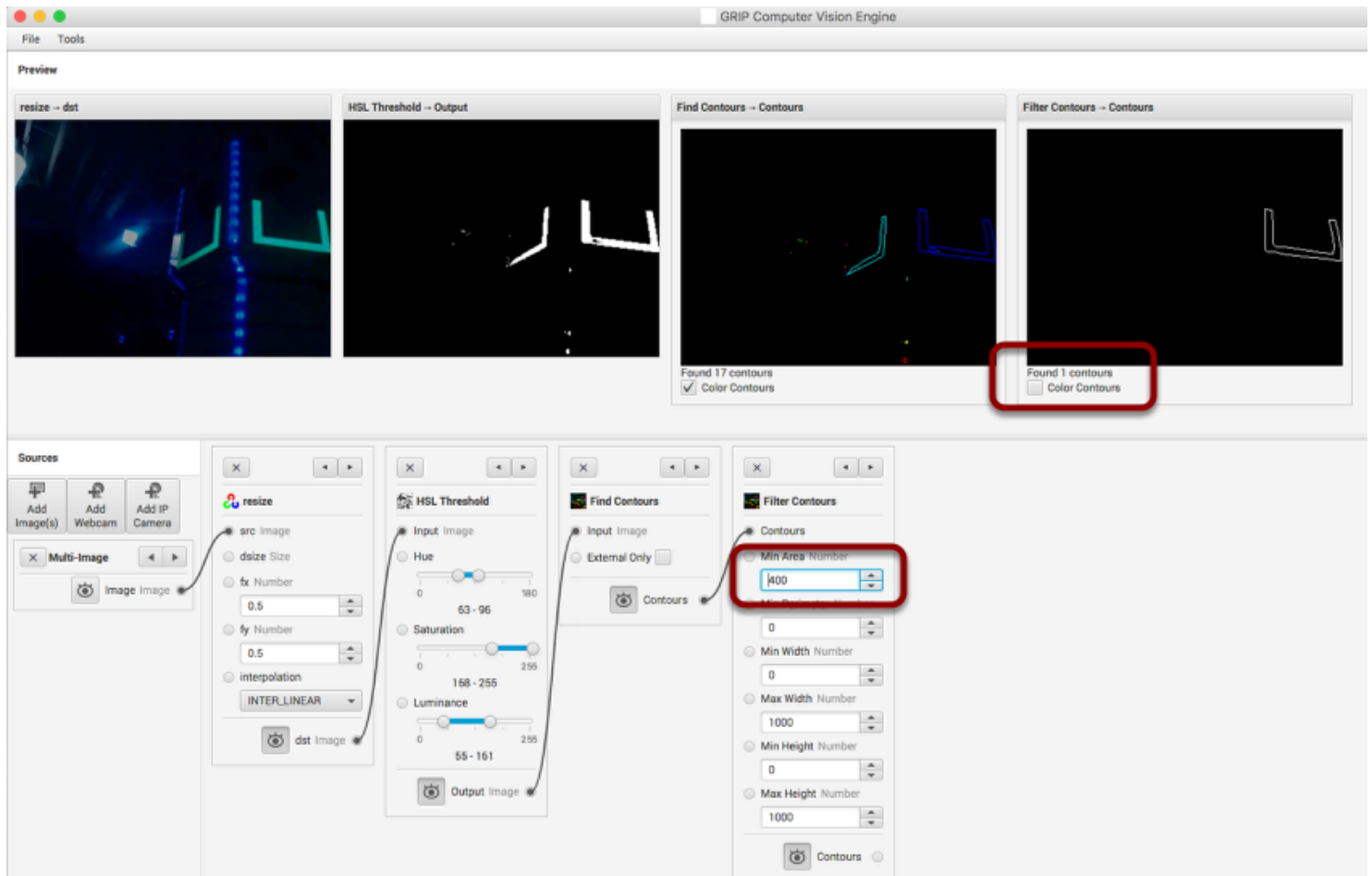
Find contours in the image



Now we can find contours in the image - that is looked for connected sets of pixels and the surrounding box. Notice that 17 contours were found that included all the smaller artifacts in the image. It's helpful to select the Color Contours checkbox to more distinctly color them to see exactly what was detected.

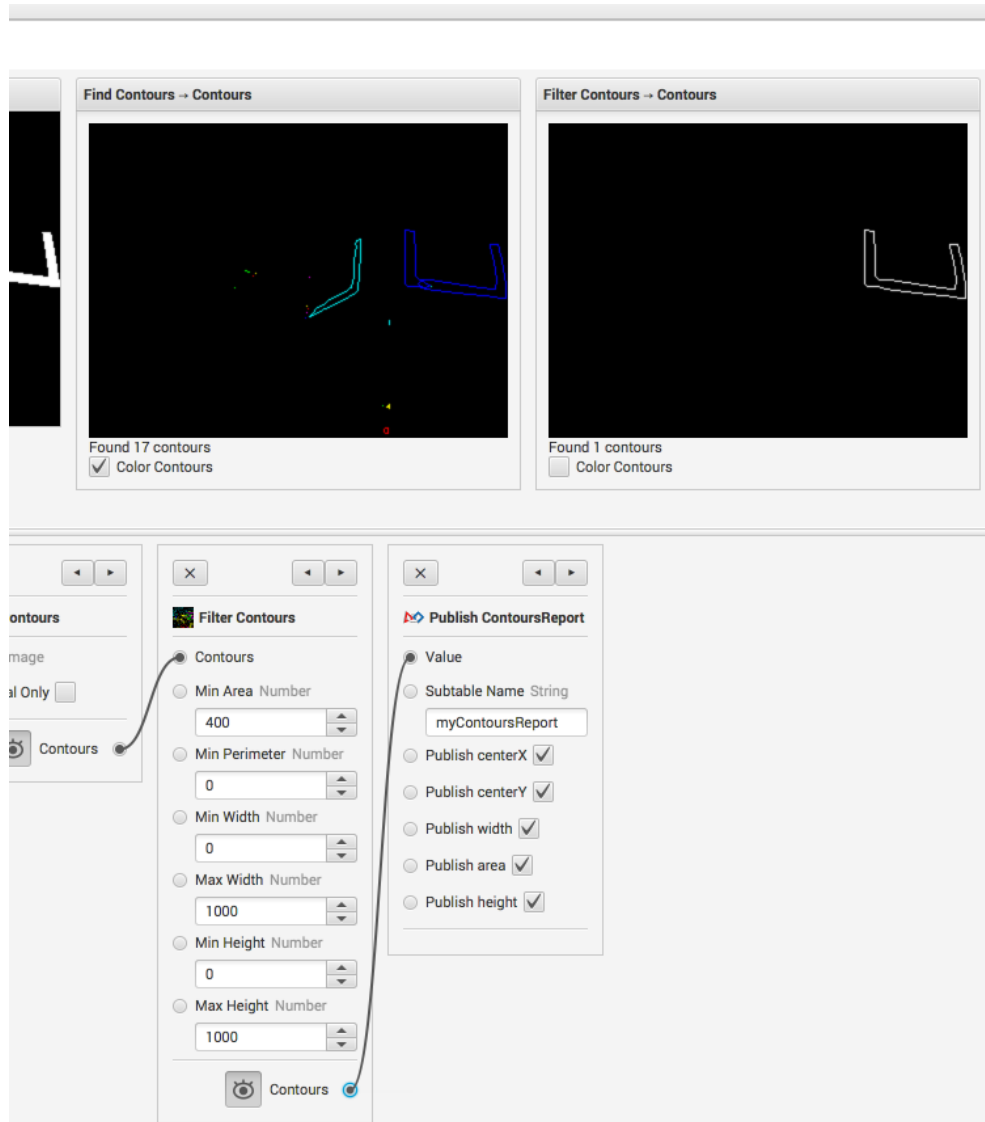
Using GRIP for the 2016 game

Filtering the discovered contours



Filtering the contours with a minimum area of 400 pixels gets down to a single contour found. Setting these filters can reduce the number of artifacts that the robot program needs to deal with.

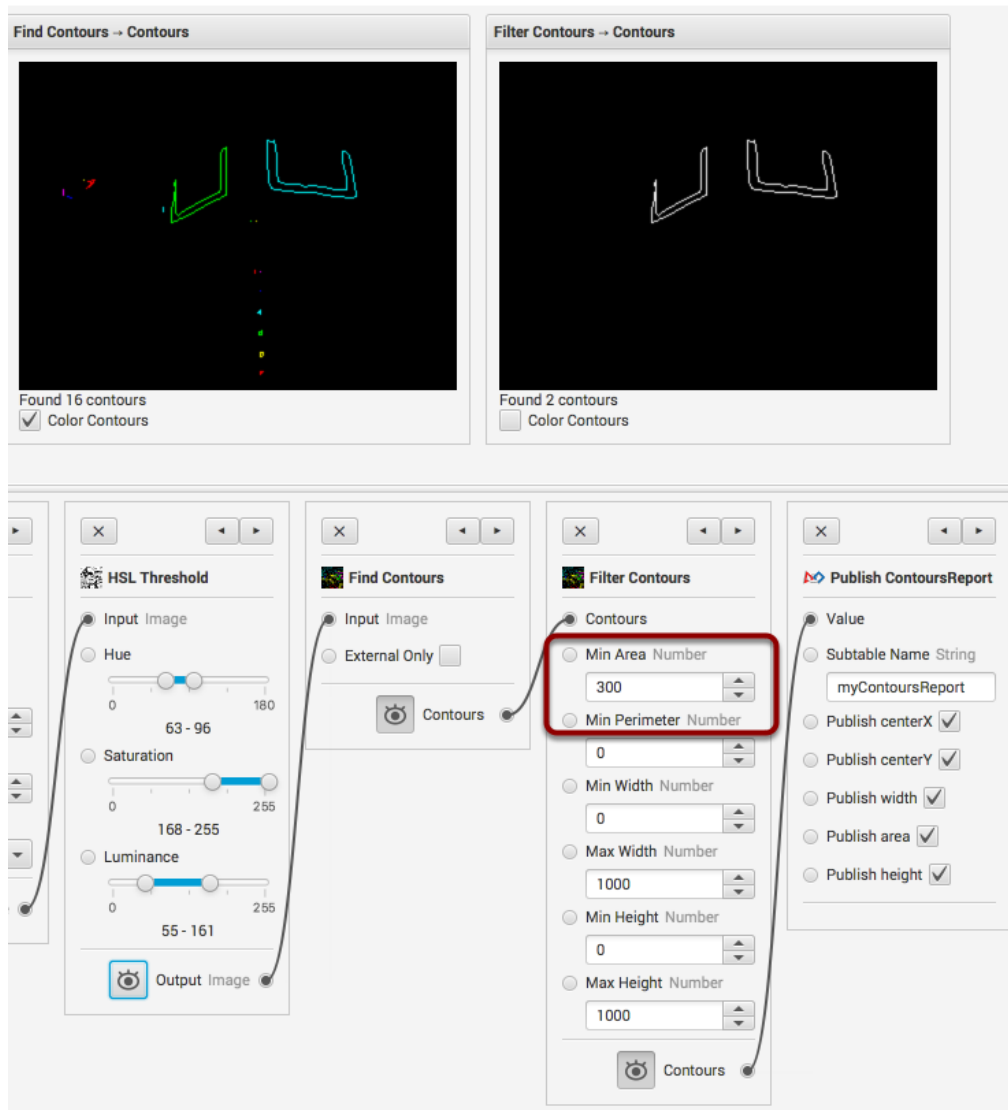
Publishing the results to network tables



The Publish ContoursReport step will publish the selected values to network tables. A good way of determining exactly what's published is to run the Outline Viewer program that's one of the tools in the <username>/wpilib/tools directory after you installed the eclipse plugins.

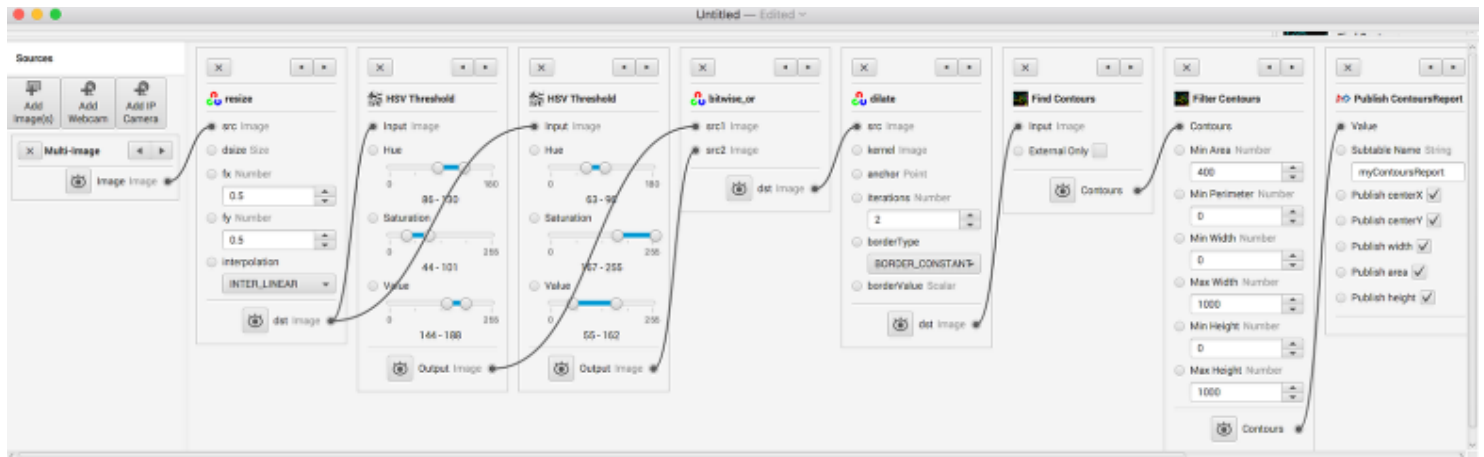
Using GRIP for the 2016 game

Changing filter values



Reducing the minimum area from 400 to 300 got both sides of the tower targets, but for other images it might let in artifacts that are undesirable.

Allowing other colors



In some of the images, the two faces were slightly different colors. One way of handling this case is to open up the parameters enough to accept both, but that has the downside of accepting everything in between. Another way to solve this is to have two separate filters, one for the first color and one for the second color. Then two binary images can be created and they can be OR'd together to make a single composite binary image that includes both. That is shown in this step. The final pipeline is shown here. A dilate operation was added to improve the quality of the results which adds additional pixels around broken images to try to make them appear as a single object rather than multiple objects.