

Code Appendix B

Borescope Viewing / Live Segmentation

Package(s): BorescopeViewer, VideoSegmentation

Description:

BorescopeViewer

This code package presents a hybrid GUI/CLI for viewing the live stream of data coming from the borescope. It allows real-time viewing of two streams: a gamma-corrected full-color feed, and a segmented feed. In addition, it provides the ability to record the full-color feed to file as a CISST-compatible video or as a .AVI.

The code is based off of the StereoVisionTutorial template in the CISST library, and uses a modified segmentation algorithm from Ascari, et. al. ¹

VideoSegmentation

This code package presents the same interface and algorithm as above, but uses a previously recorded .avi file as an input instead of a live feed from the borescope. It does not offer the ability to record to file, since the input is already a stored video.

The borescope we used interfaces with the computer via Firewire 400/800. As of writing, there are no available drivers for Ubuntu that allow for live video streams over Firewire.

Platforms Tested:

OS X 10.6 on a 2011 Macbook Pro (2.53GHz Intel i5 processor and 4GB RAM)
Ubuntu 10.10 (without Firewire support)

Requirements:

The CISST library (at least version 2.4) with cisstStereoVision installed.
Firewire 400/800 port with appropriate drivers

Installation:

Both packages can be compiled via CMake. The .tar.gz files need to be extracted, and then can be compiled.

¹ L. Ascari, U. Bertocchi, C. Laschi, C. Stefanini, A. Starita, and P. Dario. "A segmentation algorithm for a robotic micro-endoscope for exploration of the spinal cord." IEEE International Conference on Robotics and Automation, 2004. Proceedings p. 491-496.
<http://dx.doi.org/10.1109/ROBOT.2004.1307197>

BorescopeViewer User Manual

Files in Package:

BorescopeViewer.cpp	-	Main program
borescopeImageSemgenter.h	-	Parameters for borescope
borescopeImageSemgenter.h	-	Performs the actual segmentation
borescopeImageSegmenterFilter.h	-	Parameters for filter
borescopeImageSegmenterFilter.cpp	-	Creates a filter for use with CISST Stereo Vision, allows for chaining segmentation to other filters
CMakeLists.txt	-	For CMake

Parameter Tweaking

All user-adjustable parameters are found in `borescopeImageSegmenter.h`. These include parameters for the segmentation algorithm; more details on these can be found in the paper by Ascari, et. al. They are currently configured to run on the borescope with external lighting.

Running the Program

The BorescopeViewer program is run from the command line using the command `./BorescopeViewer` once the program has been made.

The terminal will run a scan of all video input devices on the computer, including built-in webcams and the borescope camera. The program will prompt the user to select one, as well as to select the resolution of the input.

Next, the user will be prompted for the file to save the video feed to. Entering a filename / location here does not automatically begin recording, but rather just prepares variables for use later.

Finally, the program will prompt for a desired frame rate, from 1-1000 fps. During testing on the Macbook Pro, we found that 24fps was ideal. On older computers, the rate may need to be lowered to as little as 10fps.

After this, the program will start the stream and display two windows:

“Original”, which displays the full-color feed; and

“Segmented”, which displays the binary segmented image

In the command line window will be the user input interface. This allows the user to start/stop recording, adjust the gamma filtering, take screenshots, and quit the program.

Filter Work Flow

One of the advantages of using the CISST Stereo Vision libraries (as opposed to stock OpenCV) is the ability to daisy-chain filters together. This modularizes the code and makes it easy to incorporate various filters, splitters, and overlays together. For example, in this package, we use splitters to create two feeds from the original (plus a third for writing to file) and gamma filters to make the video stream more clear to the human eye.

Here is a basic diagram depicting the video stream's flow from borescope to the user's computer screen.

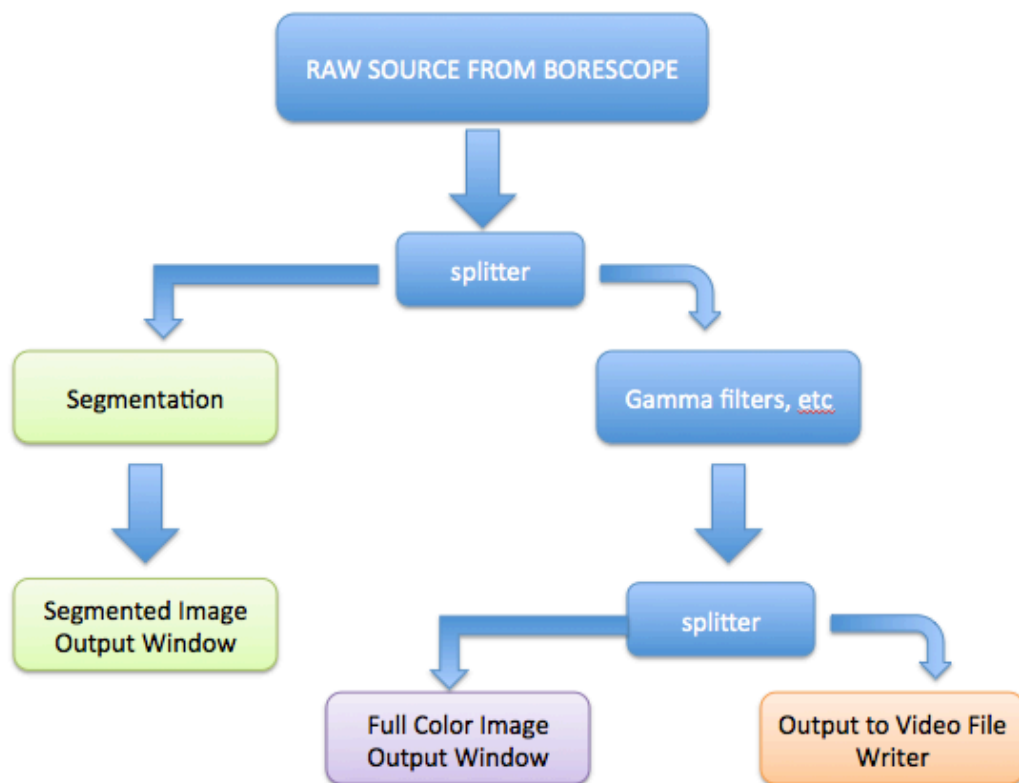


Figure 1 – Workflow diagram of BorescopeViewer filters

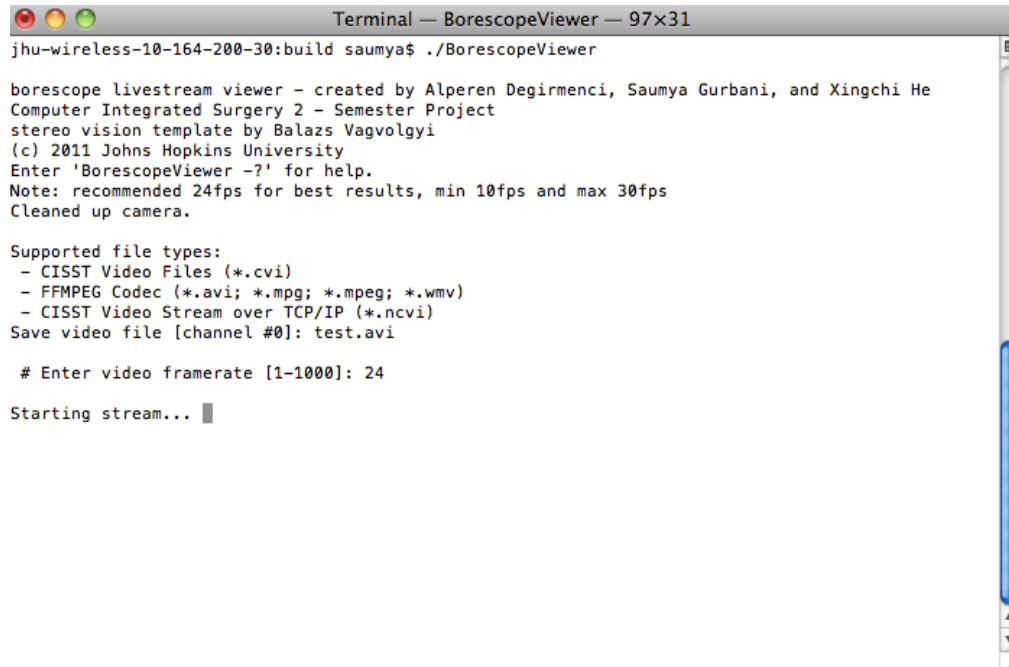
Segmentation Algorithm

Our algorithm's goal is similar to that described in the paper by Ascari, et al. We want to segment the image so that our target, the cochlear round window, is highlighted in one color while all of the bones and tissue are in a different color. The algorithm needs to be especially robust since we are exploring very low light regions, often with images from the borescope stream being so dark that it is difficult to determine the border of the round window. Segmenting would not only make these contours clearer, but also allow the computer to know where the target is in relation to the robot's current position. This would be used in the future for automatic safe path generation.

During each iteration of video stream processing, our algorithm is called. We begin by grabbing the current image frame from the borescope video stream. We convert the image into a vector of luminance values, using standard OpenCV commands. We then create a histogram of luminance values in order to determine the threshold for binarizing the image. Essentially, our goal is to separate all the pixels into two groups: one representing the round window, and the other representing all surrounding tissue. We apply a 5-pixel sliding window to smooth out the histogram, then do a minima search to find the valleys in the luminance histogram. These minima represent the borderline intensity pixels, one of which will be used as the threshold for binarizing the image.

Using the method described in Ascari, we are able to find a dynamic threshold for each image which is robust to changes in light, noise, and sudden movements of the borescope. Furthermore, the algorithm is able to run at 24fps on a modern laptop, allowing for clear, real-time analysis of the borescope video feed.

Screenshots



```
Terminal — BorescopeViewer — 97x31
jhu-wireless-10-164-200-30:build saumya$ ./BorescopeViewer

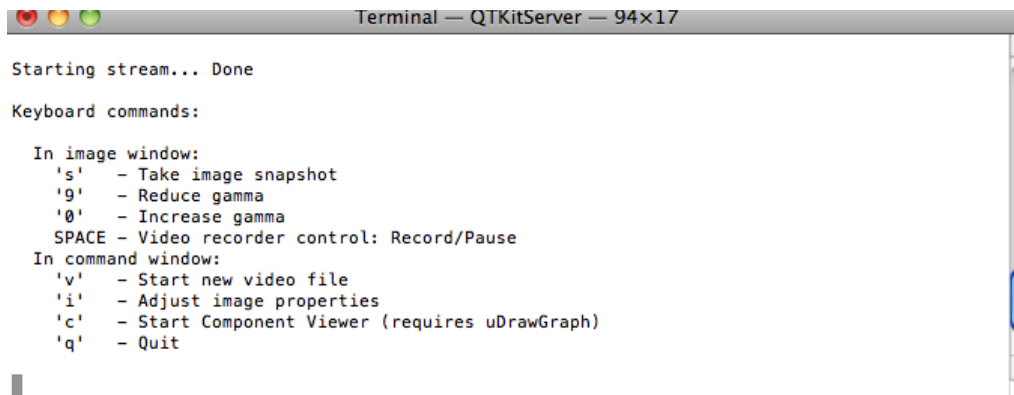
borescope livestream viewer - created by Alperen Degirmenci, Saumya Gurbani, and Xingchi He
Computer Integrated Surgery 2 - Semester Project
stereo vision template by Balazs Vagvolgyi
(c) 2011 Johns Hopkins University
Enter 'BorescopeViewer -?' for help.
Note: recommended 24fps for best results, min 10fps and max 30fps
Cleaned up camera.

Supported file types:
- CISST Video Files (*.cvi)
- FFmpeg Codec (*.avi; *.mpg; *.mpeg; *.wmv)
- CISST Video Stream over TCP/IP (*.ncvi)
Save video file [channel #0]: test.avi

# Enter video framerate [1-1000]: 24

Starting stream... █
```

Figure 2 – Command line options when starting program



```
Terminal — QTKitServer — 94x17
Starting stream... Done

Keyboard commands:

In image window:
's' - Take image snapshot
'g' - Reduce gamma
'0' - Increase gamma
SPACE - Video recorder control: Record/Pause
In command window:
'v' - Start new video file
'i' - Adjust image properties
'c' - Start Component Viewer (requires uDrawGraph)
'q' - Quit

█
```

Figure 3 – Command line options while running program

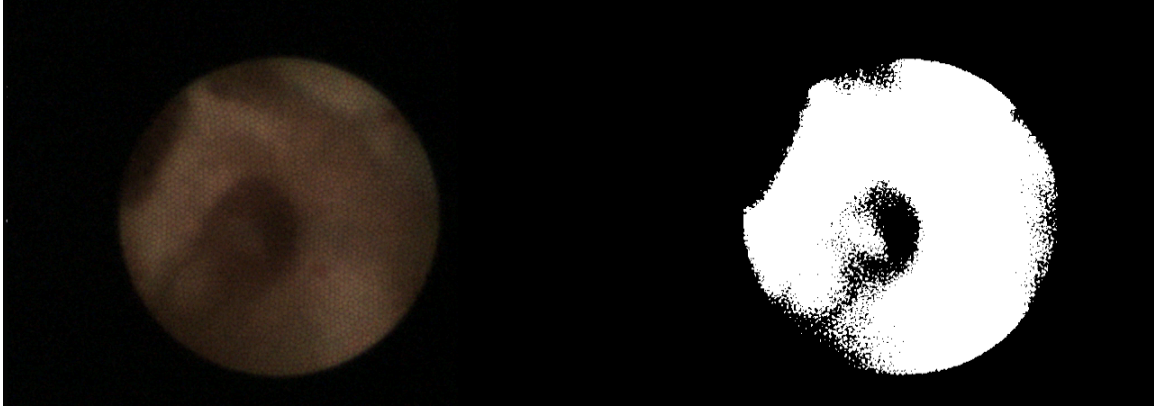


Figure 4 – Screenshot of both live video feeds, depicting the round window of the cochlea. Full color feed on left and segmented feed on right. The feed was running at 24 frames per second.

VideoSegmentation User Manual

Files in Package:

VideoSegmentation.cpp	-	Main program
borescopeImageSemgenter.h	-	Parameters for borescope
borescopeImageSemgenter.h	-	Performs the actual segmentation
borescopeImageSegmenterFilter.h	-	Parameters for filter
borescopeImageSegmenterFilter.cpp	-	Creates a filter for use with CISST Stereo Vision, allows for chaining segmentation to other filters
CMakeLists.txt	-	For CMake

Parameter Tweaking

All segmentation parameters are found in borescopeImageSegmenter.h. These include parameters for the segmentation algorithm; more details on these can be found in the paper by Ascari, et. al. They are currently configured to run on the borescope with external lighting.

Running the Program

The video which is to be segmented needs to be selected in the file VideoSegmentation.cpp.

The VideoSegmentation program is run from the command line using the command “./VideoSegmentation” once the program has been made.

The program will automatically load the video file and begin segmentation. Two windows will be shown, one depicting the original video file and the other the segmented image.

Future Work

This segmentation algorithm is very simple in that all it does is segment a video stream. While this is useful in itself for viewing while guiding the robot (since it can provide distinction when the full color feed is blurry or too dark), there is much more that can be done.

When combined with optical flow for depth perception, segmentation can be used to determine the robot's surroundings in 3D. This can ultimately be used for a basic 3D reconstruction, allowing a surgeon to determine how much room they have on each side. Virtual fixtures can be produced to further help guide the surgeon through the ear to the cochlea.

Source Code for both packages are attached. Since the borescopeImageSegmenter and borescopeImageSegmenterFilter files are identical, they are only presented once.