Imaging Whiteboard
User's Guide

Contents

Introduction

The Imaging Whiteboard is designed to allow the user to quickly whiteboard real-time imaging algorithms. As the algorithms are designed they will be actively implemented allowing the user to see the results and modify the algorithm.

The Imaging Whiteboard is designed to work on a desktop computer running Windows 7 or later. Windows 8 and Windows 10 were used during development. It will not run on Windows XP, Linux, or iOS. The Musician's Workbench will run on a Windows tablet. The user interface is not designed to be finger friendly so a mouse is recommended.

The Imaging Whiteboard is a product of Sound Analysis.

Installation

Before installing the Imaging Whiteboard; make sure that you have version 4.6.2 (or later) of the .NET framework installed. If not, you can get it from Microsoft at:

https://www.microsoft.com/net/download/dotnet-framework-runtime/net462

If you have a previous version or the Imaging Whiteboard installed uninstall that first.

The installation files are delivered as a zip file. Extract the file to any location on your computer, and then run setup.exe.

The Whiteboard Metaphor

When the imaging whiteboard first starts it presents the user with a blank background.

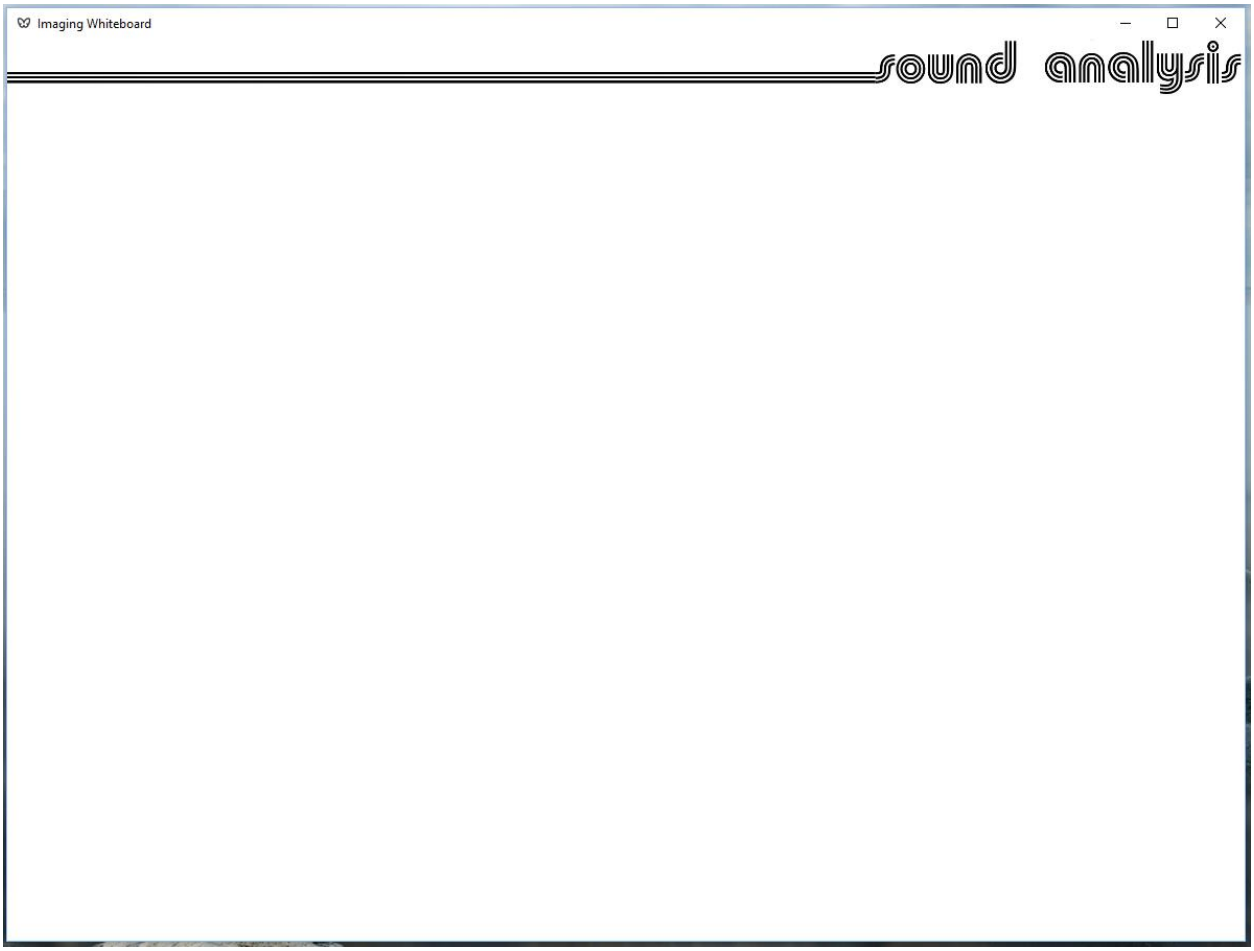


Figure 1 : Imaging Whiteboard initial state.

To draw on the whiteboard right click on the background and select imaging components from the menu. These components may be moved using the left or right mouse buttons, and connected to each other using the left mouse button.

Components may be configured by opening the settings expander.

Clicking on a control will give it a red outline. This has no effect on the operation of the control. This feature may be useful when discussing the algorithm with others, or in a teaching situation.

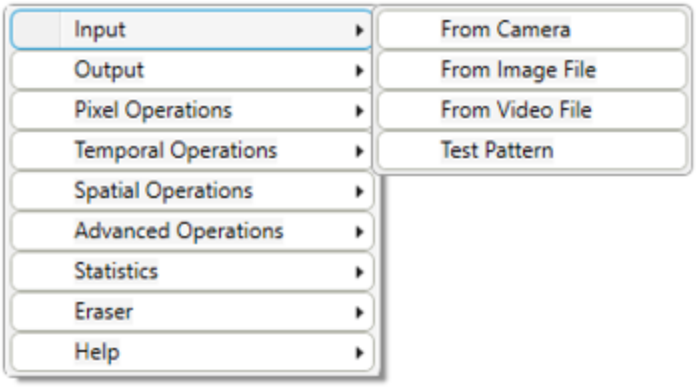


Figure 2 : Input menu selection

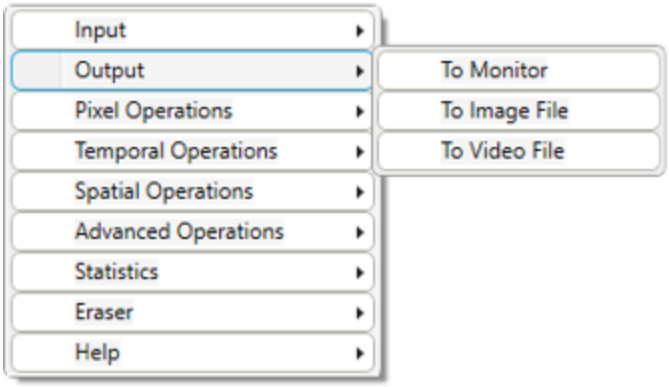


Figure 3 : Output menu selection

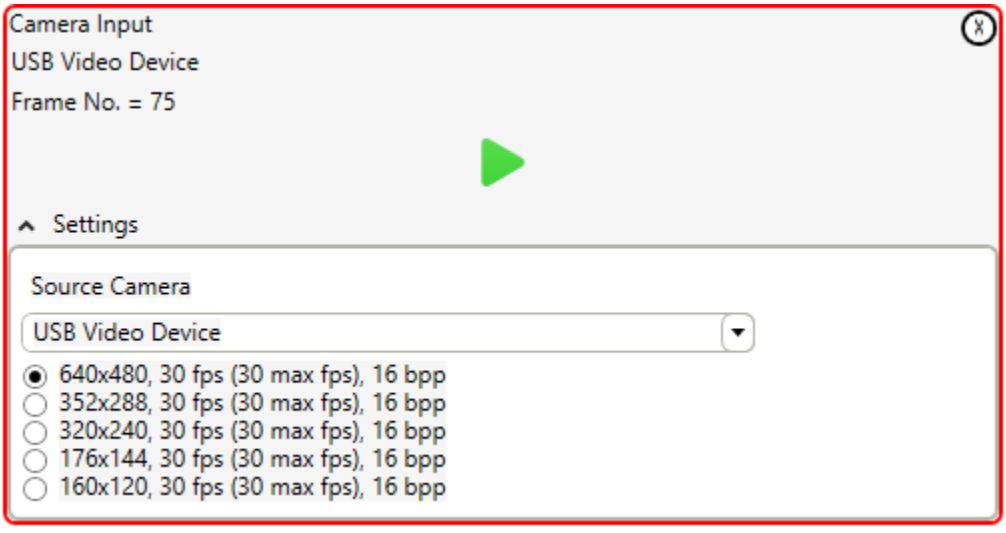


Figure 4 : Camera input settings

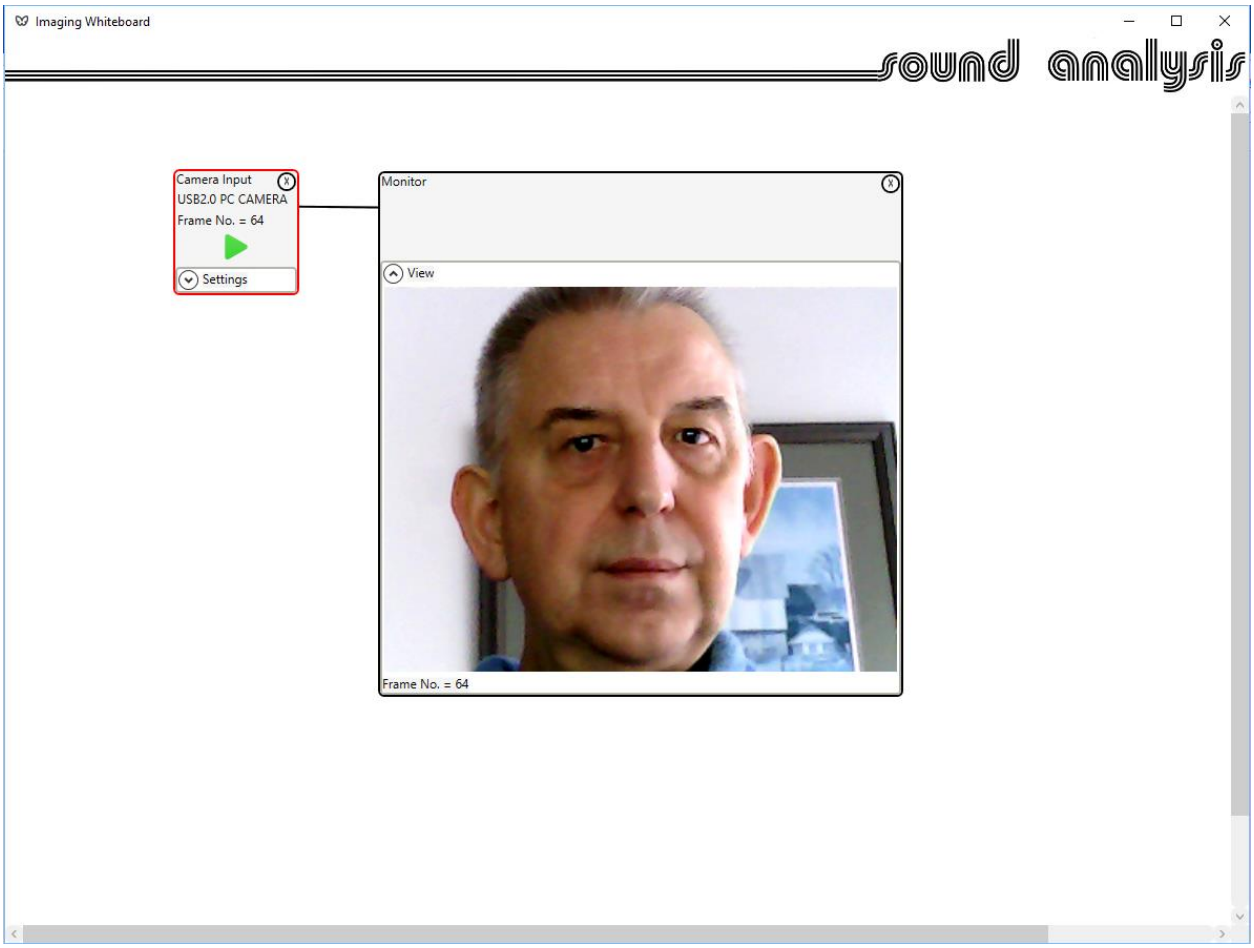


Figure 5 : Simple video viewer.

To add processing pick a control from one of the operations menus.

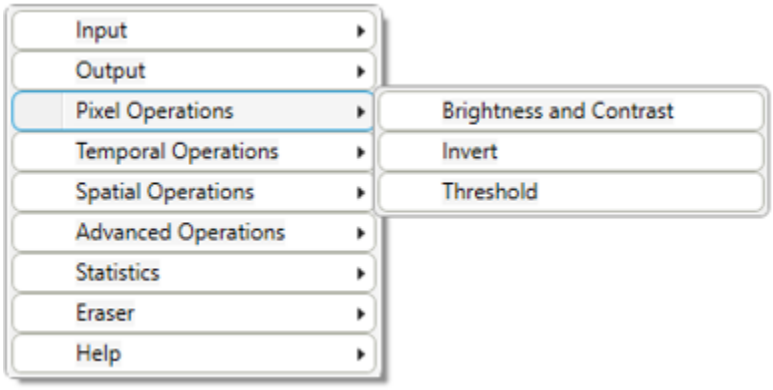


Figure 6 : Pixel operations menu.

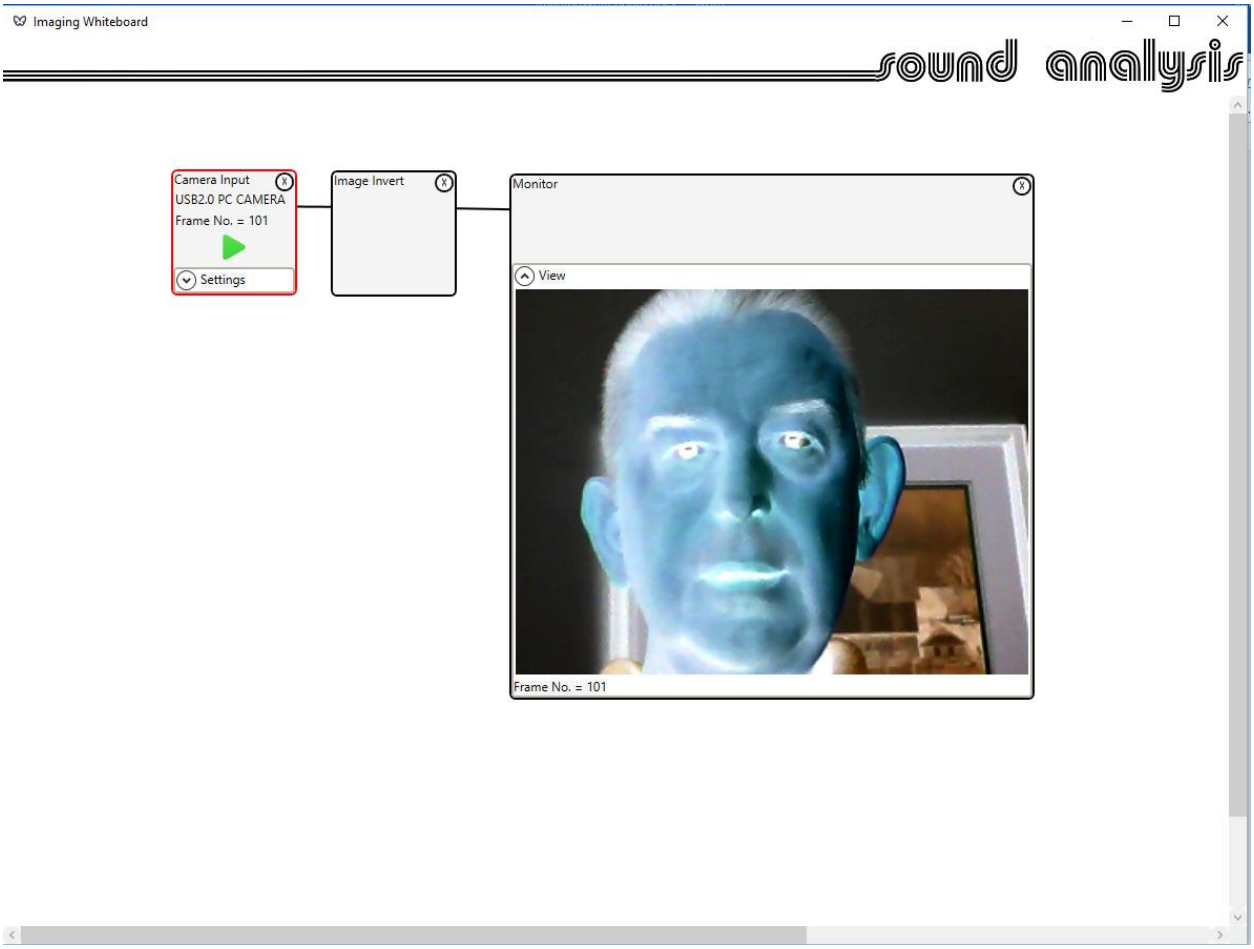


Figure 7 : Image inversion.

String multiple operations together for more complex algorithms.

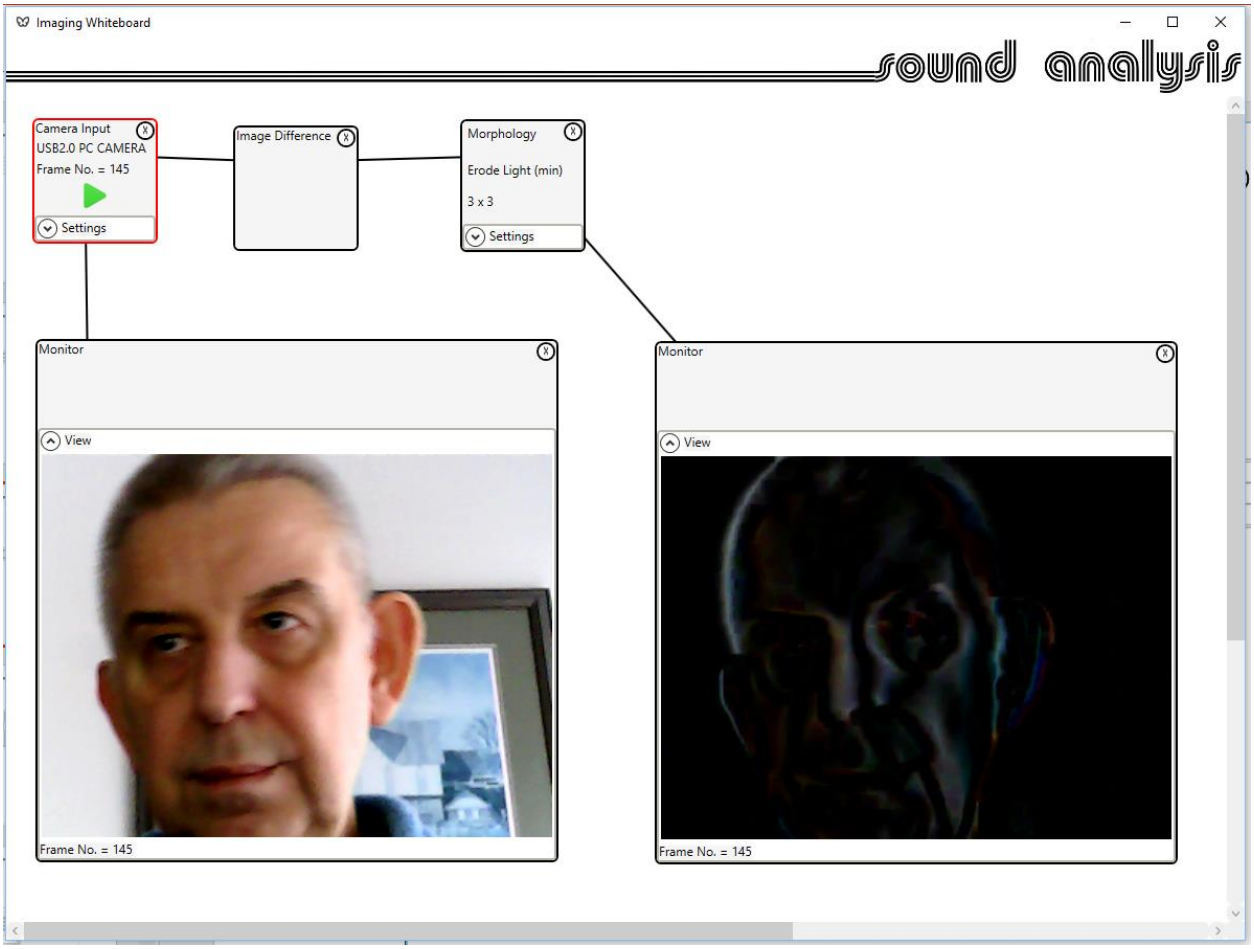
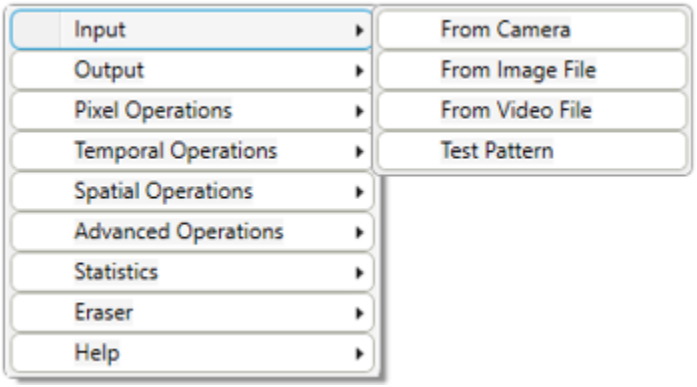


Figure 8 : Motion detection with false alarm reduction.

Controls in detail

Input controls



Camera Input

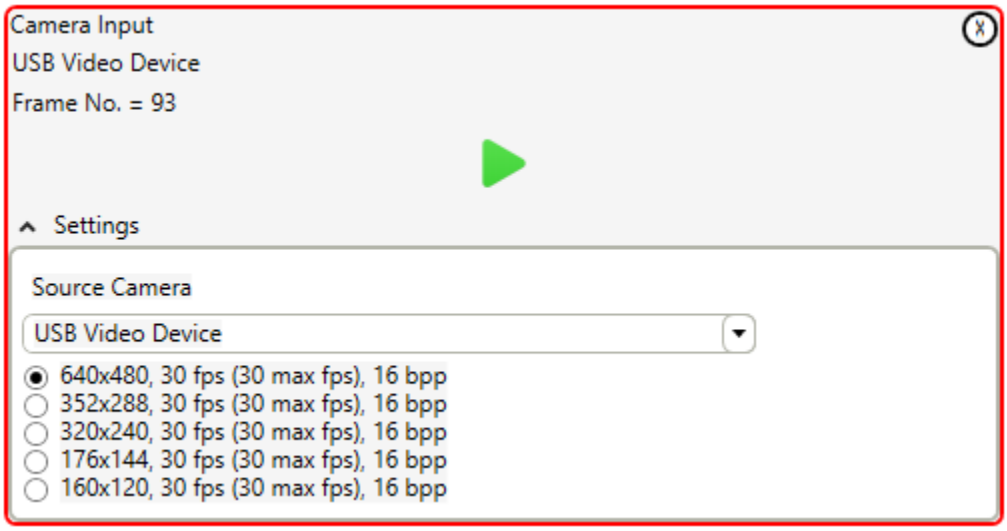


Figure 9 : Camera input with settings expanded

The camera input setting will allow the user to select a camera. A tablet such as the surface will have forward and rear facing cameras, either will be available. If there is only one camera that will be the default. Once the camera is selected the resolutions available from that camera will be displayed. Selecting a resolution will enable the camera and display the play icon.

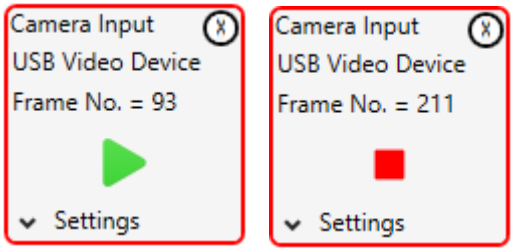


Figure 10 : Camera input with settings collapsed

With setting collapsed (after selecting the camera and resolution) will only display the device selected and the current frame number. A play / stop button will be provided.

Image File Input

Figure 11 : Image file Input with settings expanded.

The image input control will allow the user to select an image file from disc. Files with .bmp, ,jpg, or .gif may be selected.

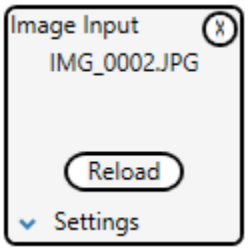


Figure 12 : Image File Input with settings collapsed

Once a file is selected and the settings are collapsed the file name will be displayed. A reload button will be available to reload the image, after changing the pipeline settings for example.

Video File Input

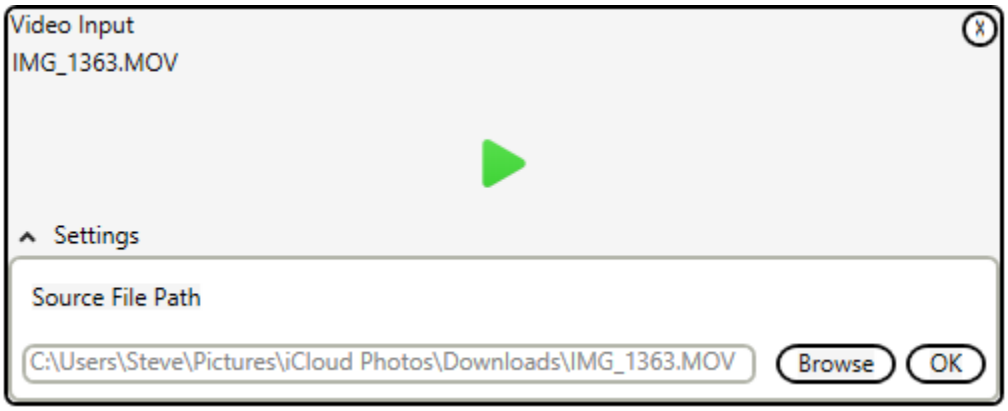


Figure 13 : Video File Input with settings expanded

The image input control will allow the user to select a video file from disc. Files with .wmv, .mov, .avi, .mp4, or .mpeg may be selected.

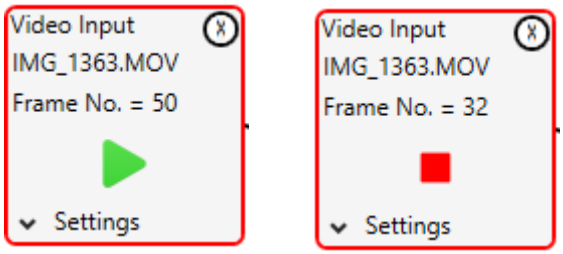


Figure 14 : Video File Input with settings collapsed

With setting collapsed (after selecting the video file) will only display the file selected and the current frame number. A play / stop button will be provided.

Test Pattern Control

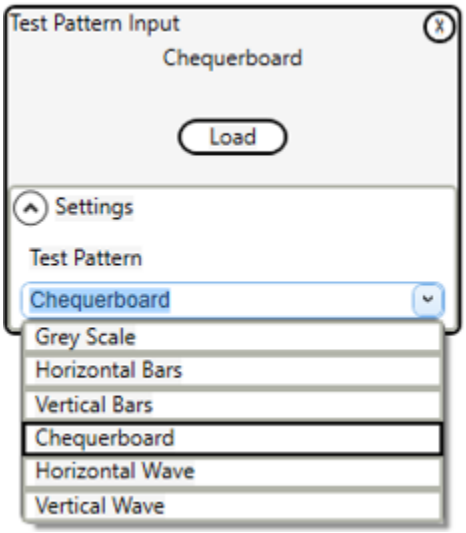


Figure 15 : Test Pattern Control with settings expanded

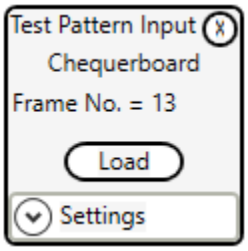


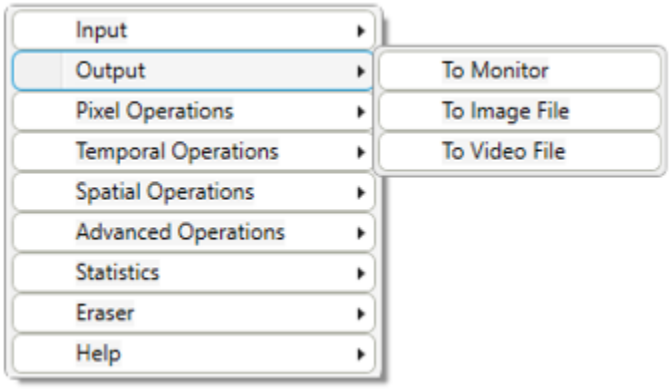
Figure 16 : Test pattern Control with settings collapsed

The test pattern control will generate test patterns to simplify the testing of algorithms.

When expanded the drop down will allow the use to select from a list of test patterns.

Test patterns are generated and sent when the load button is clicked.

Output Controls



Output to Monitor

This control will show the video. This is used to provide instant feedback. Multiple monitor controls may be opened at once to show the various stages of algorithms.

The view may be collapsed to save space on the whiteboard.

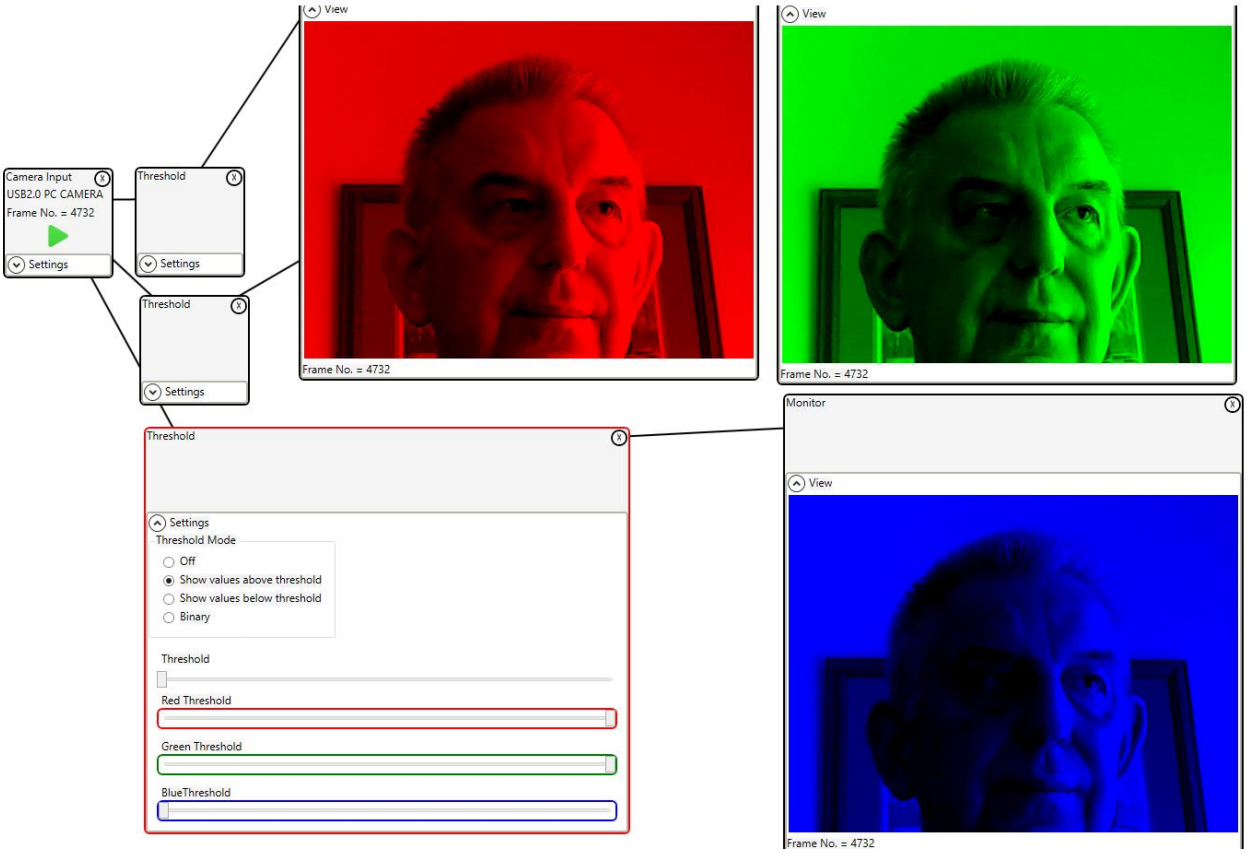


Figure 17 : Color component separation using the threshold control

Output to File

Figure 18 : Output to Image File with setting expanded

The image output control will allow the user to save an image file to disc. Files with .bmp, ,jpg, or .gif may be used.

Figure 19 : Output to Image File with settings collapsed

Once a file is selected and the settings are collapsed the file name will be displayed. The file name will auto increment if more than one image is saved, so if the input is from camera, for example, a sequence of still images will be saved.

Output to Video File

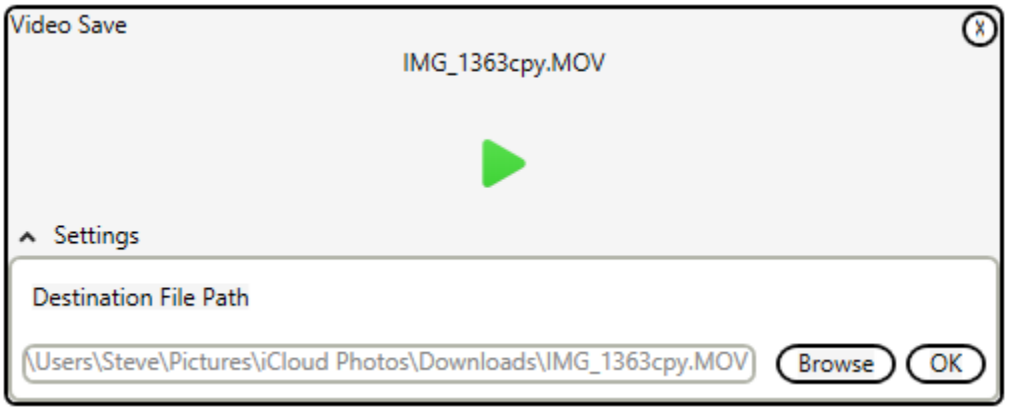


Figure 20 : Video File Output with settings expanded

The video output control will allow the user to save a video file to disc. Files with .wmv, .mov, .avi, .mp4, or .mpeg extensions may be created.

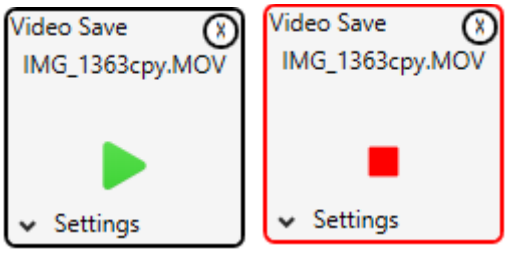


Figure 21 : Video File Output with settings collapsed

With setting collapsed (after selecting the video file) will only display the file selected.

A play / stop button will be provided to start and stop the recording. The file name will auto increment if the recording is stopped and started.

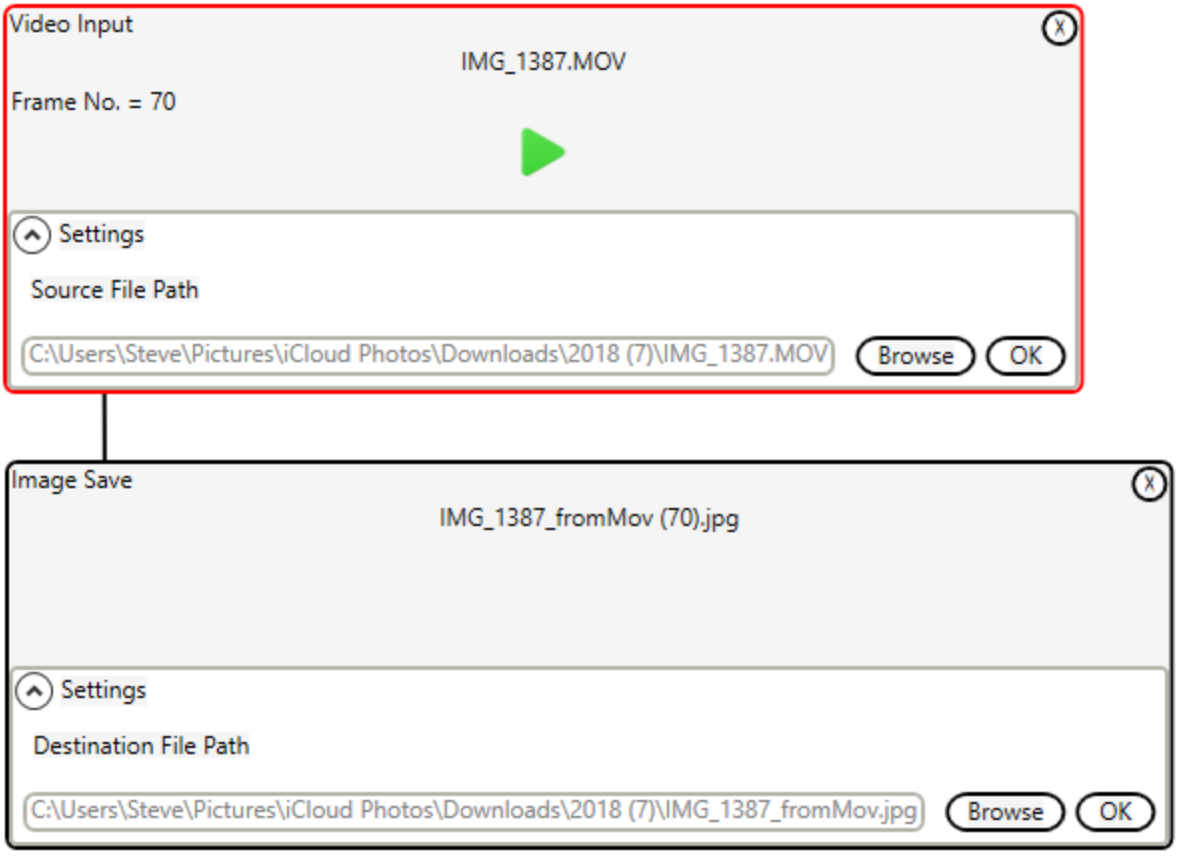
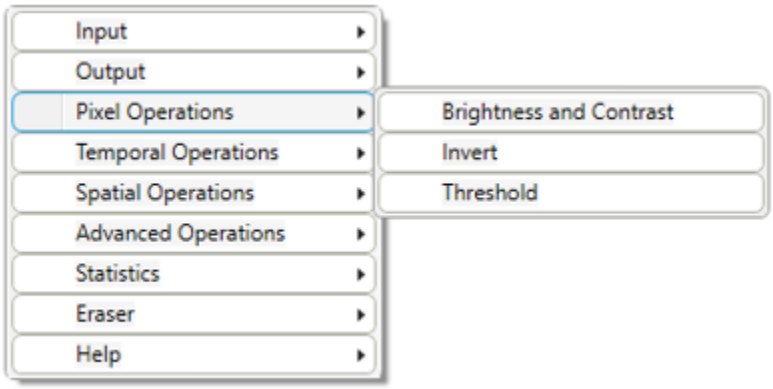


Figure 22 : Splitting a video file into separate images

A video file can be separated into a separate image files for each frame. Useful for those movies that an iPhone creates when you think you took a picture.

Pixel Operations



Pixel operations are those which perform the same operation on every pixel without regard to neighboring pixels or preceding pixels.

Brightness and Contrast

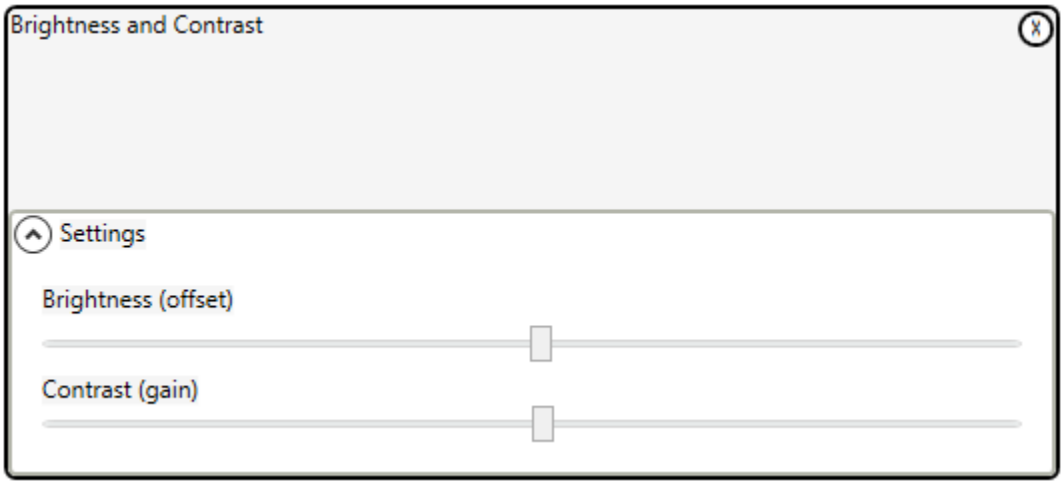


Figure 23 : Brightness and Contrast Control with settings expanded

Figure 24 : Brightness and Contrast Control with settings collapsed

Brightness will go from full black on the left to full white on the right.

Contrast will go from zero (mid grey) on the left to times 2 on the right.

In the center position the image will not be changed.

Image Invert

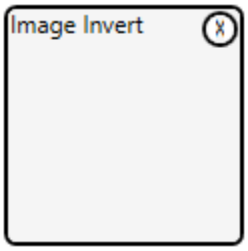


Figure 25 : Image Inversion Control

The image inversion control will simply invert the brightness of every pixel. There are no settings.

Threshold Control

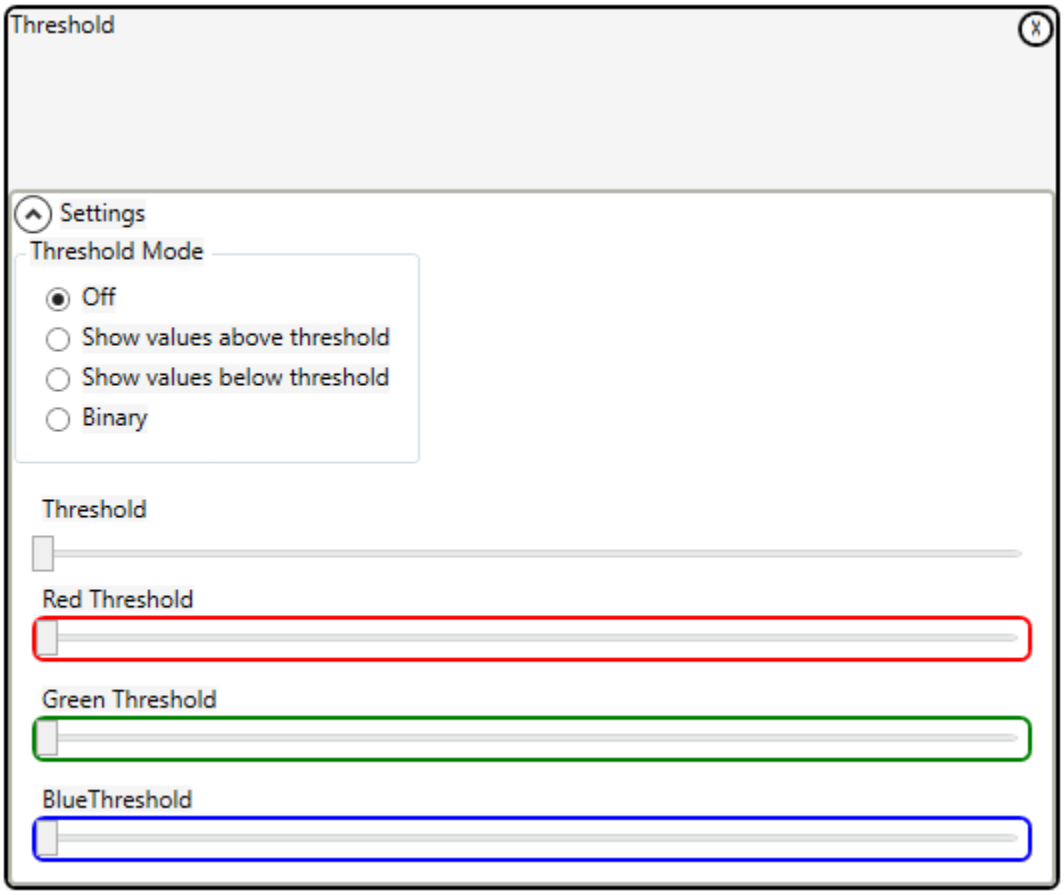


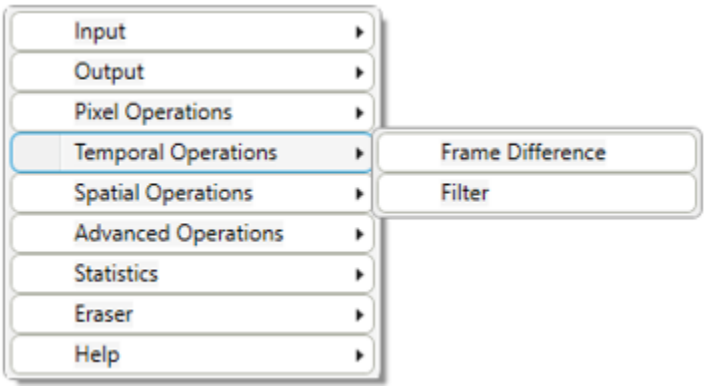
Figure 26 : Threshold Control with settings expanded

Figure 27 : Threshold Control with settings collapsed

Threshold values range from 0 (Black) on the left to 255 (White) on the right. Each color's threshold may be set independently or moved simultaneously by moving the top threshold slider. There are 4 modes:

Off – the image is not changed.
Show values above threshold – Pixels with a value greater than the threshold are displayed unchanged, others are black.
Show values below threshold – Pixels with a value less than the threshold are displayed unchanged, others are white.
Binary – Pixels with a value less than the threshold are displayed black, others are white.

Temporal Operations



Temporal operations are those that set the value of a pixel based on its current value and its value in previous frames.

Frame Difference

Figure 28 : Frame Difference Control

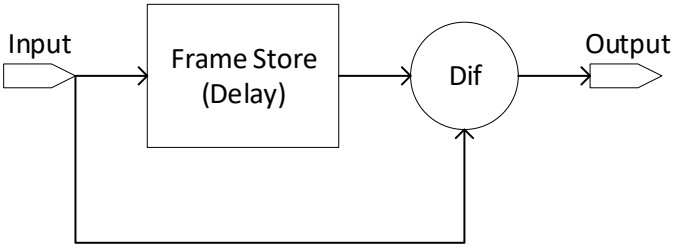


Figure 29 : Frame Difference Operation

The frame difference control will output the difference between the current frame and the previous frame on a pixel by pixel basis.

The are no settings.

Temporal Filter

Figure 30 : Temporal Filter control with settings expanded

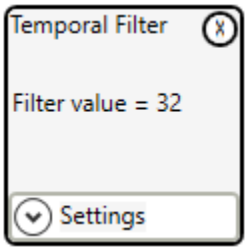


Figure 31 : Temporal Filter Control with settings collapsed

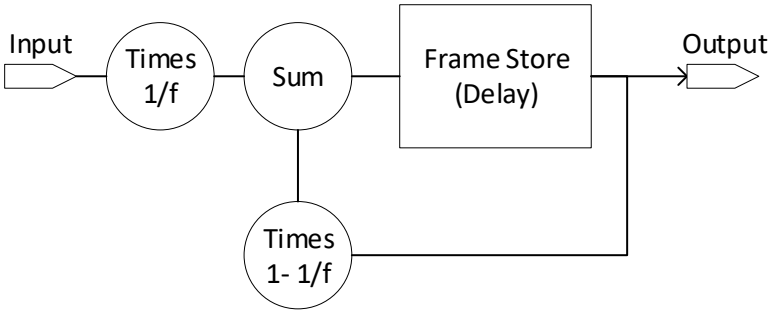


Figure 32 : Temporal Filter operation

The temporal filter will add a fraction of the incoming frame to a complimentary fraction of the output frame. For example, 1/10 input + 9/10 output.

The slider range is from 0 to 100 and is the percent of the outgoing image that is added.
A value of 0 means that all of the result comes from the input and no filtering is applied.
A value of 100 means that all of the result comes from the output freezing the image.
Values in between will apply filtering.
Generally, the value should be set high enough to remove noise, but not so high as to blur moving objects.

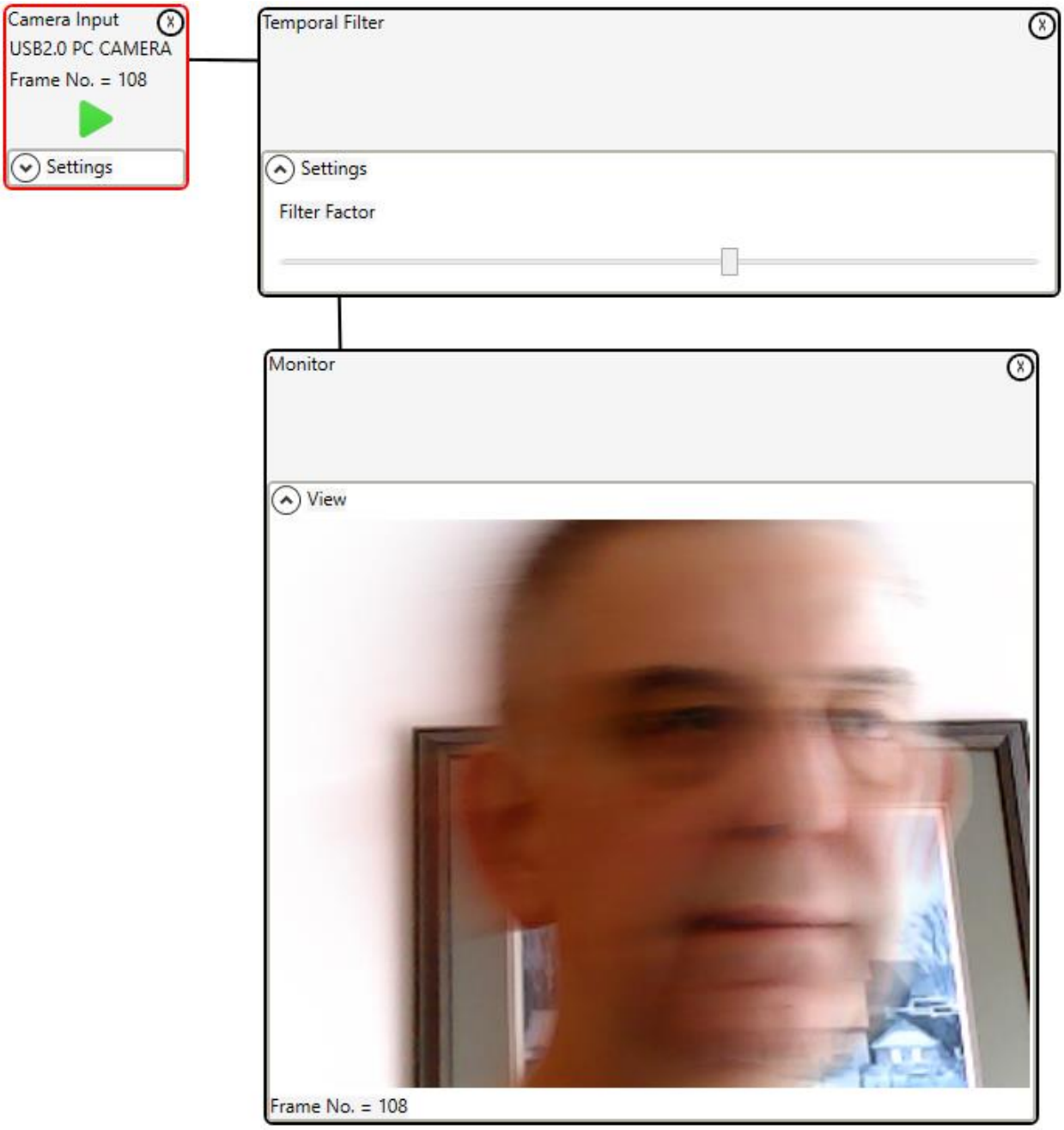
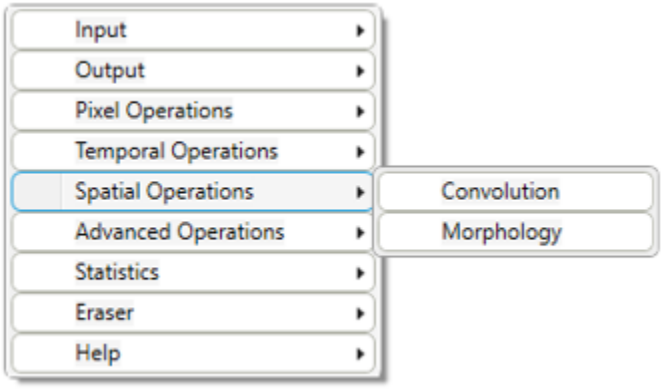


Figure 33 : Temporal Filter applied to a moving image

Spatial Operations



Spatial operations are those that set the value of a pixel based on its current value and the values of neighboring pixels in the same frame.

Convolution

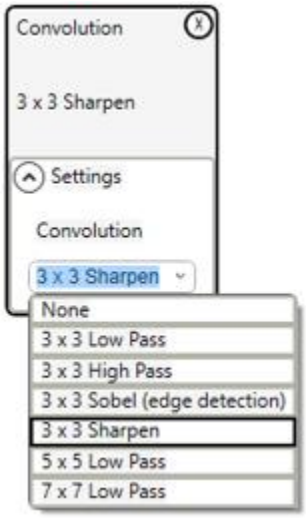


Figure 34 : Convolution Control with settings expanded

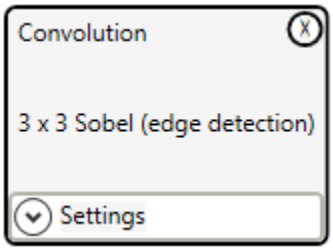


Figure 35 : Convolution Control with settings collapsed

A number of commonly use convolutions are provided.

3 x 3, 5 x 5, and 7 x 7 low pass:
These are used to remove high frequency components from the image

3 x 3 high pass:
This is used to remove low frequency components from the image

3 x 3 Sobel:
This is a combination of 2 convolutions, horizontal and vertical edge detection, resulting in edges of all orientations being detected.

3 x 3 Sharpen
This is used to enhance the high frequencies and to sharpen edges in the image.

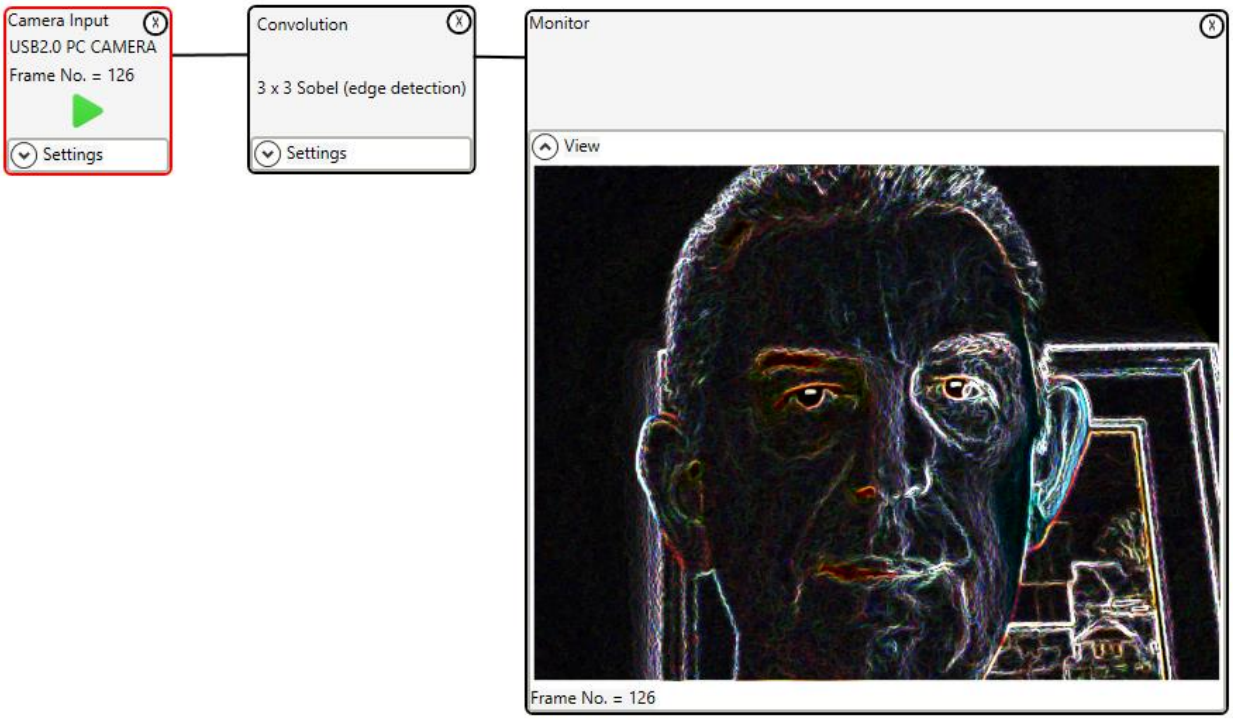


Figure 36 : Sobel filter applied to camera input

Morphological Operations

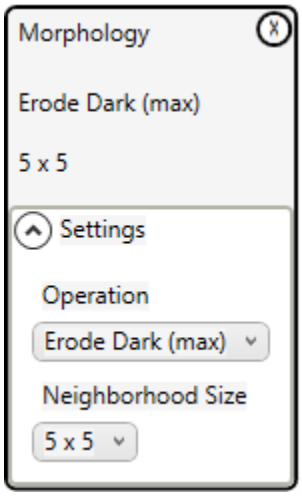


Figure 37 : Morphology Control with settings expanded

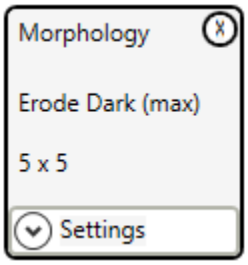


Figure 38 : Morphology Control with settings collapsed

Two kinds of morphological operations are provided:

Erode dark (max):

This takes the maximum valued pixel from the neighborhood.

Erode light (min):

This takes the minimum valued pixel from the neighborhood.

Three sizes of neighborhood are provided:

3 x 3, 5 x 5, and 7 x 7.

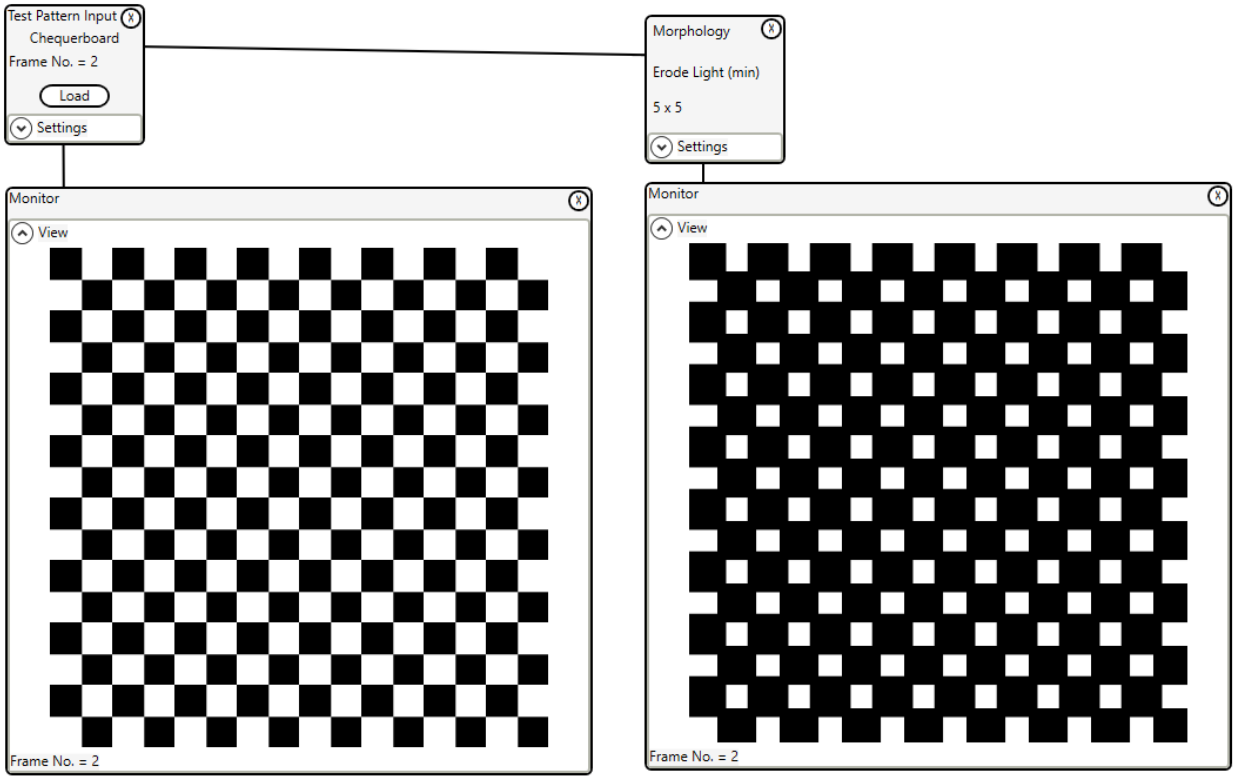
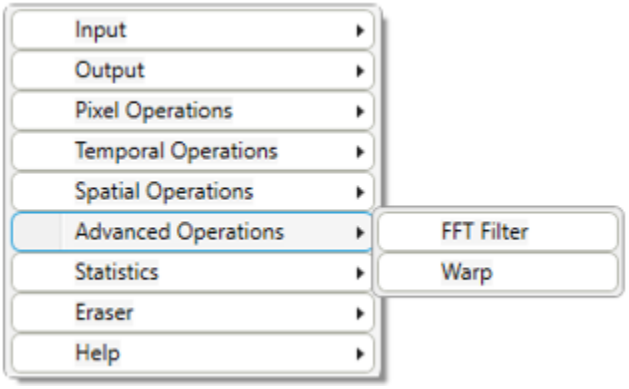


Figure 39 : 5 x 5 Erosion applied to checkerboard test pattern

Advanced Operations



FFT Filter

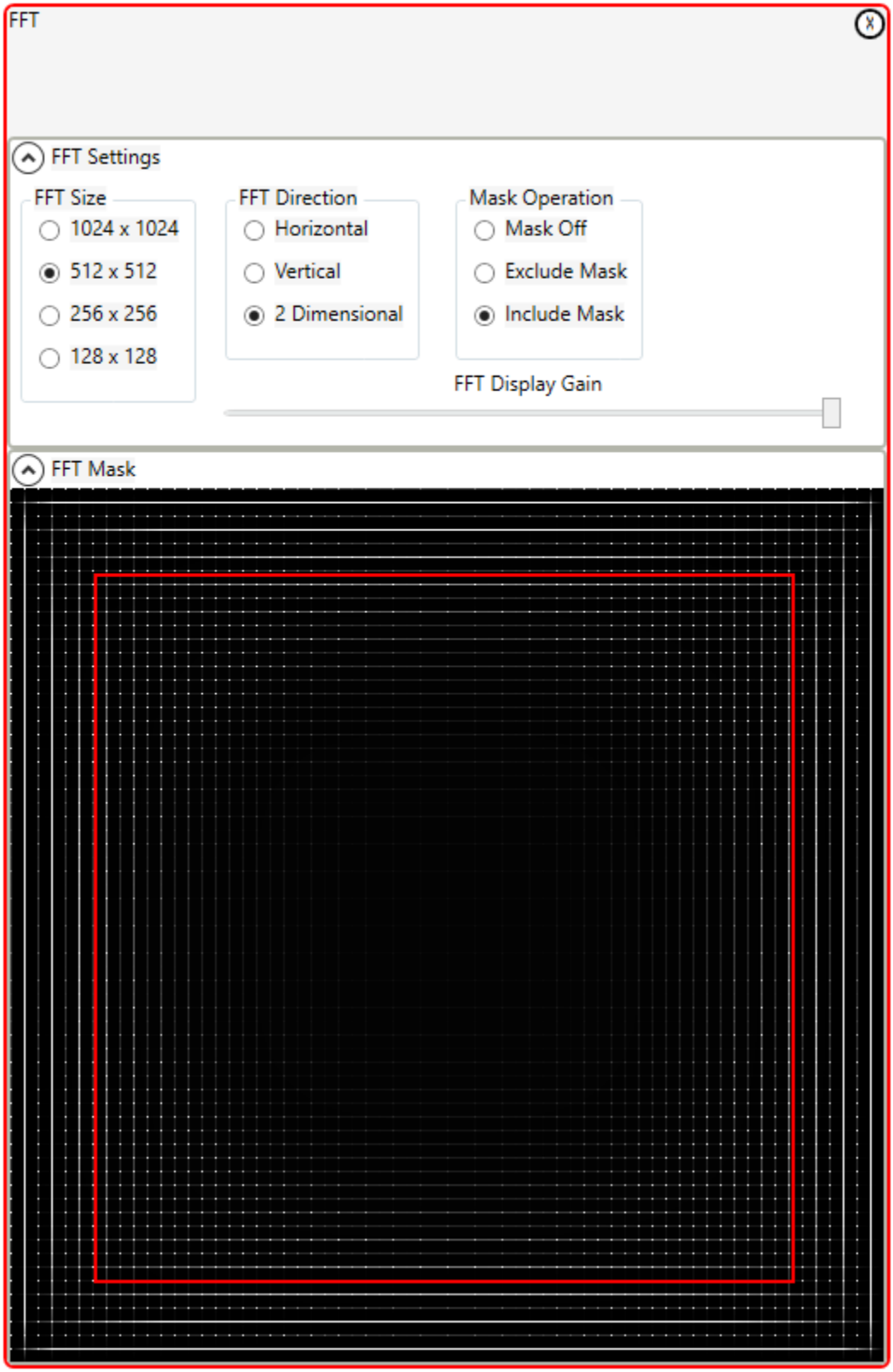


Figure 40 : FFT Filter Control with mask and settings expanded

Figure 41 : FFT Filter Control with mask and settings collapsed

The FFT filter control will provide FFT sizes of 128 x 128 to 1024 x 1024. The FFT will be performed on the center region of the input image. Should the user select an FFT size which is too large for the input image the FFT size will be reduced to fit under the covers (a warning will be logged). The output from the filter will be an image of the same size as the FFT performed.

FFTs may be performed in just the horizontal direction, or just the vertical direction, more commonly in both directions 2 Dimensional.

The mask operations offered are:
Off which will use all frequencies to reconstitute the original image.
Exclude which will use all frequencies not defined by the mask to reconstitute the image.
Include which will use all frequencies defined by the mask to reconstitute the image.

The output from the FFT is displayed in the mask expander and the user may use the left mouse button to position the mask. Note that the output from the FFT is symmetrical with high frequencies in the center of the mask area.

The output from the FFT has a large dynamic range which makes it difficult to show all of the high and low values. The control will dynamically adjust the output to best show the peak values, this may result in some low values not being visible. The FFT gain control may be used to adjust the FFT output gain if necessary.

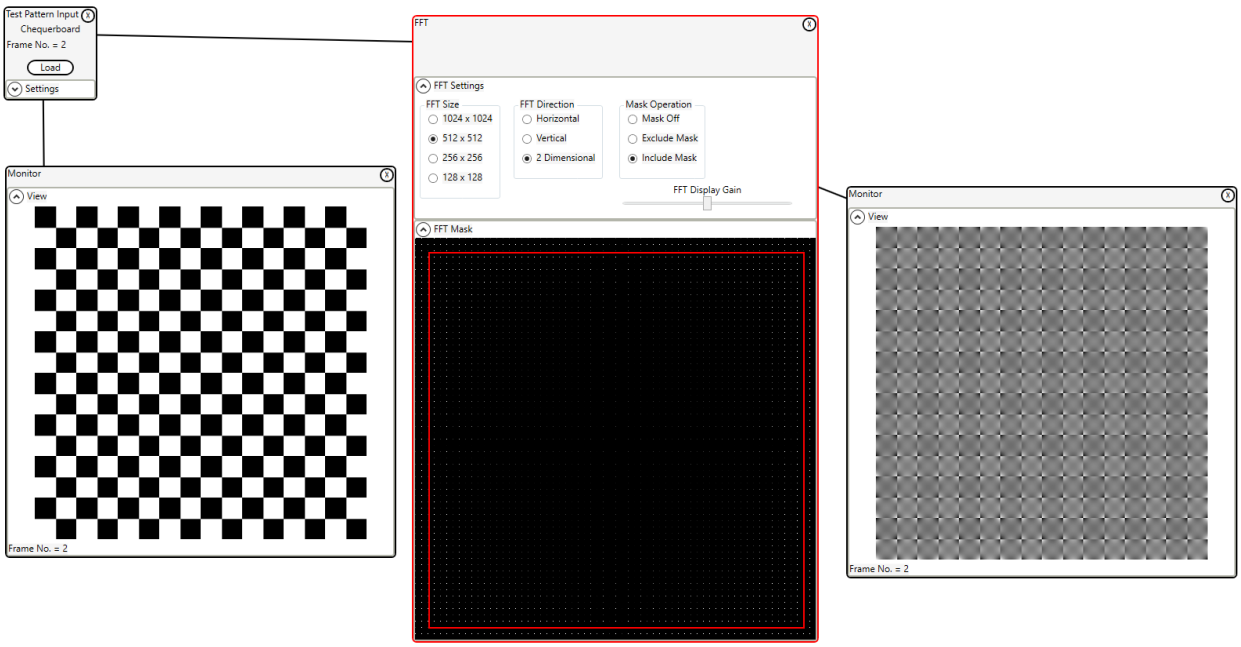


Figure 42 : FFT Include high frequencies mask applied to checkerboard test pattern

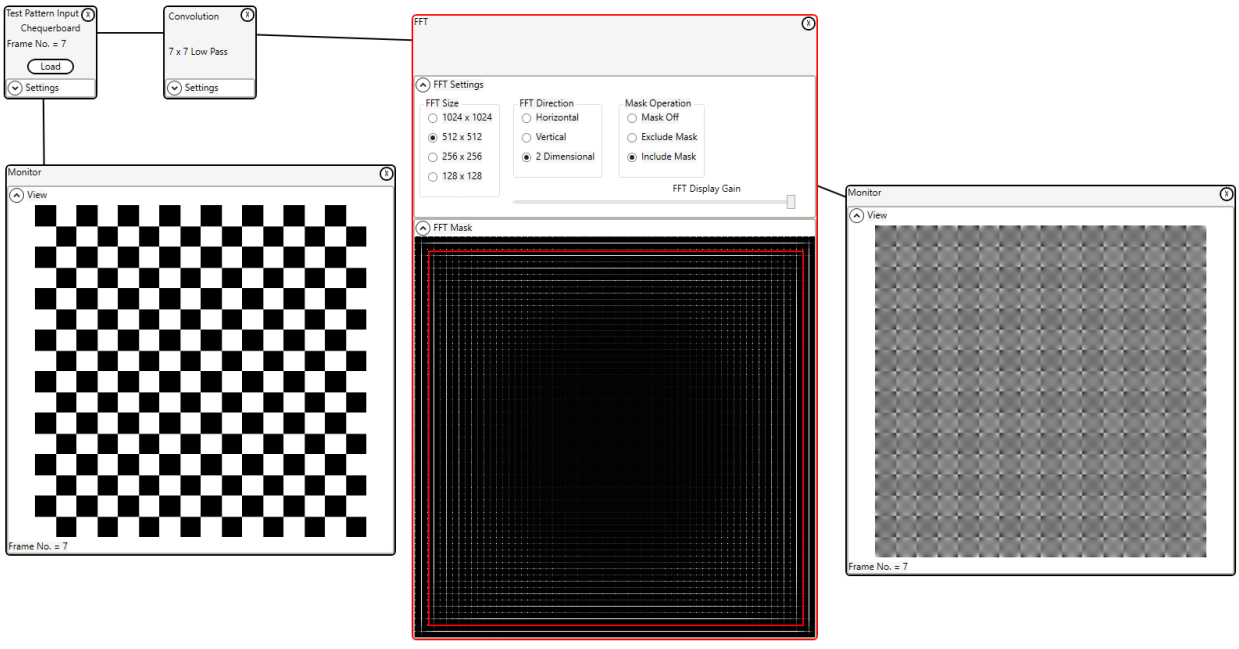


Figure 43 : FFT Include high frequencies mask applied to blurred checkerboard test pattern

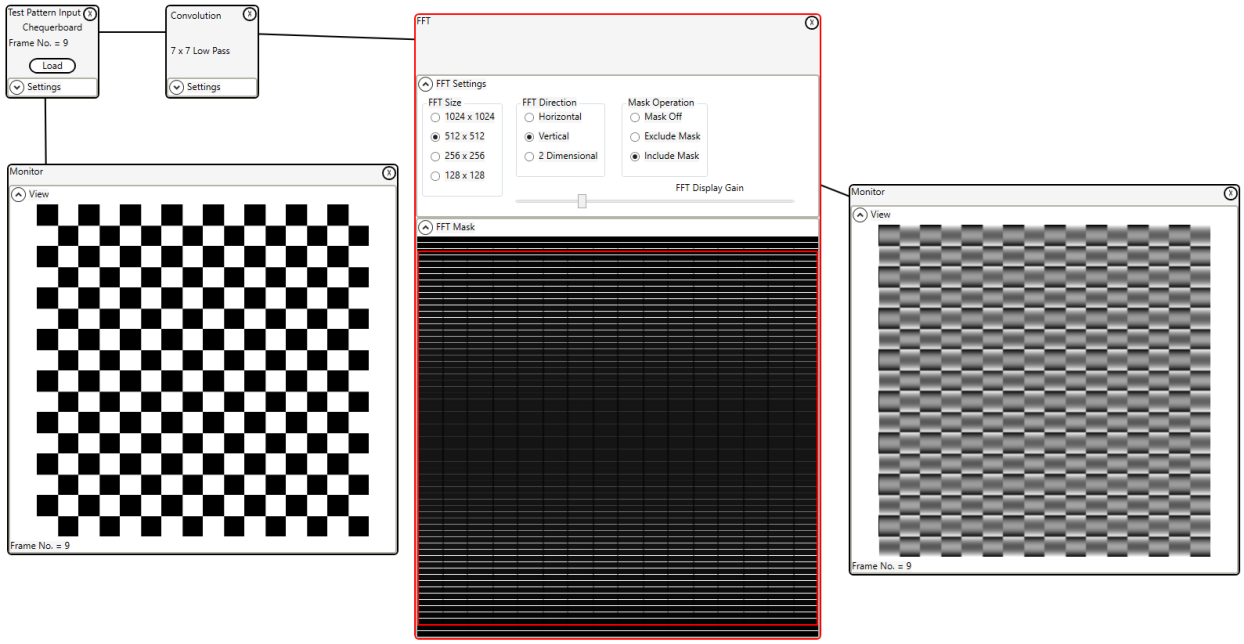


Figure 44 : Vertical FFT Include high frequencies mask applied to blurred checkerboard test pattern

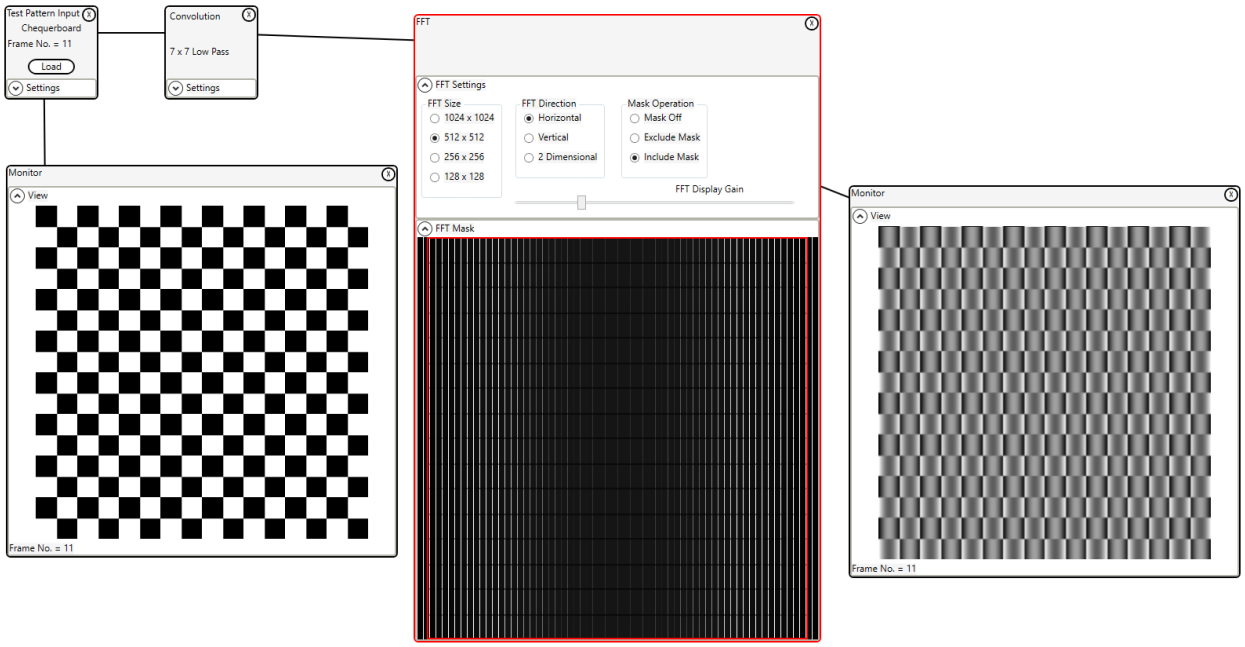


Figure 45 : Horizontal FFT Include high frequencies mask applied to blurred checkerboard test pattern

Warp Control

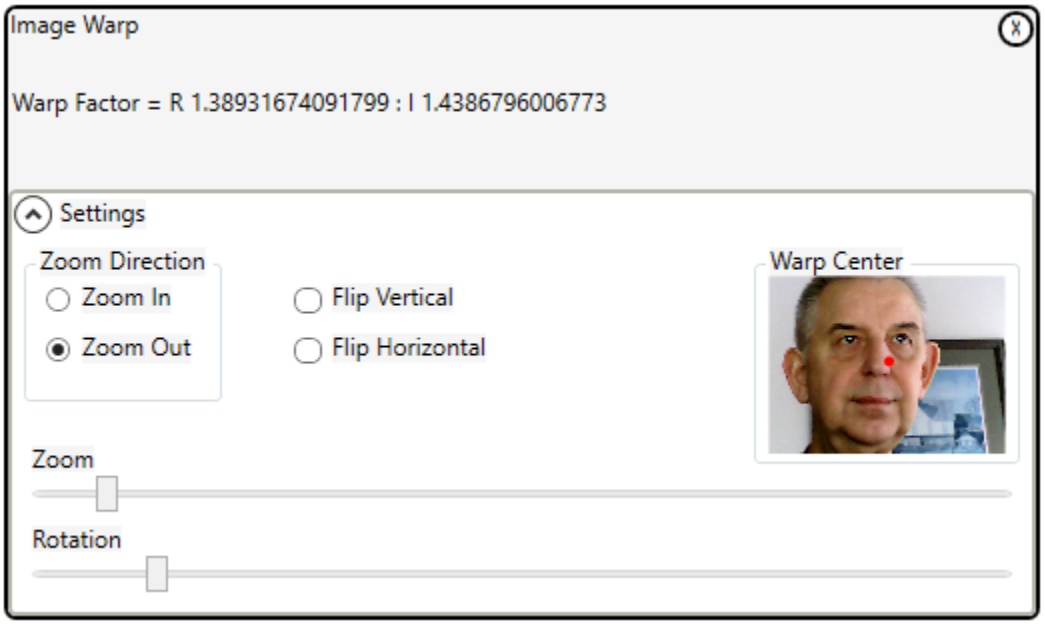


Figure 46 : Warp Control with settings expanded

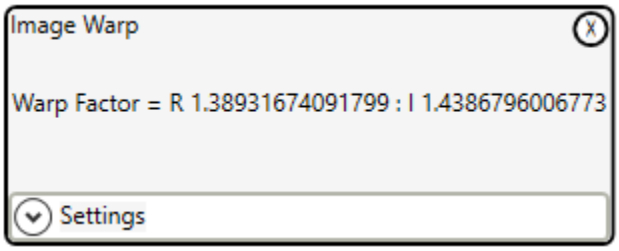


Figure 47 : Warp Control with settings collapsed

The warp control will perform basic warp functions, zoom, rotate and flip. All warps are performed with interpolation to avoid pixilation.

Zoom direction may be in or out. Flips may on or off, both may be used simultaneously.

Rotation will be from 0 – 360 degrees, 0 to the left of the slider 360 to the right.

The warp center is initially set at the center of the image, by clicking on the image the warp center can be repositioned. All warps are performed around the warp center. This is useful if for example you wish to zoom in on a particular point in the image.

The displayed warp factor is the complex number that is used to perform the warp, it is calculated from the zoom and rotation settings. Usually warping is an exercise in co-ordinate geometry, this warper uses complex number arithmetic. A single complex number is used to

calculate the source pixel from the target pixel. The result is the same but the code is simpler and faster. At some point I will blog about the details.

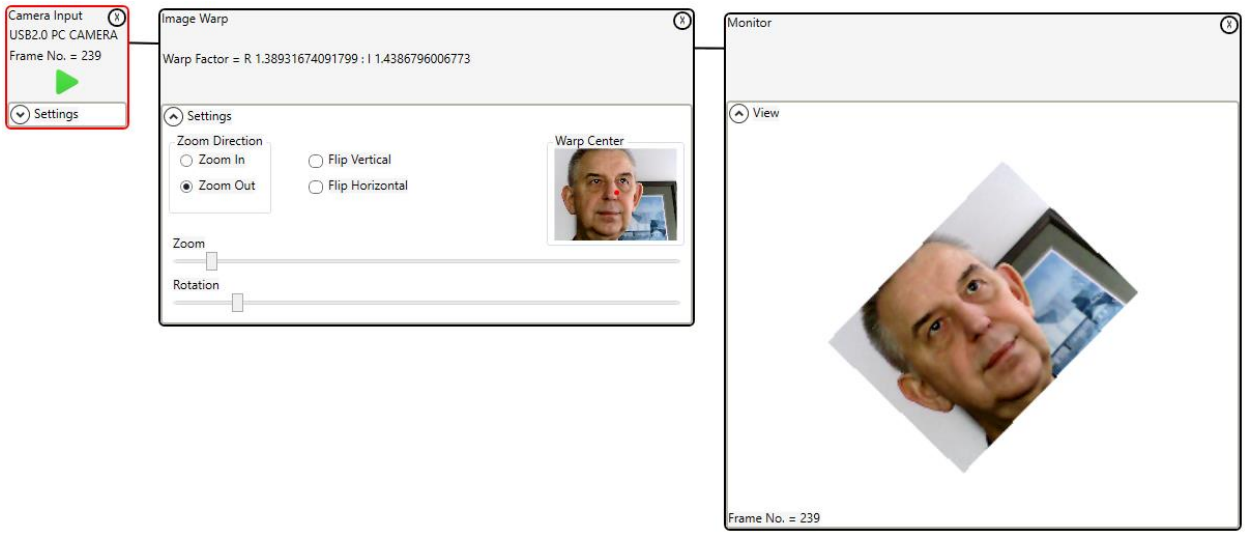


Figure 48 : Zoom and rotation of live video

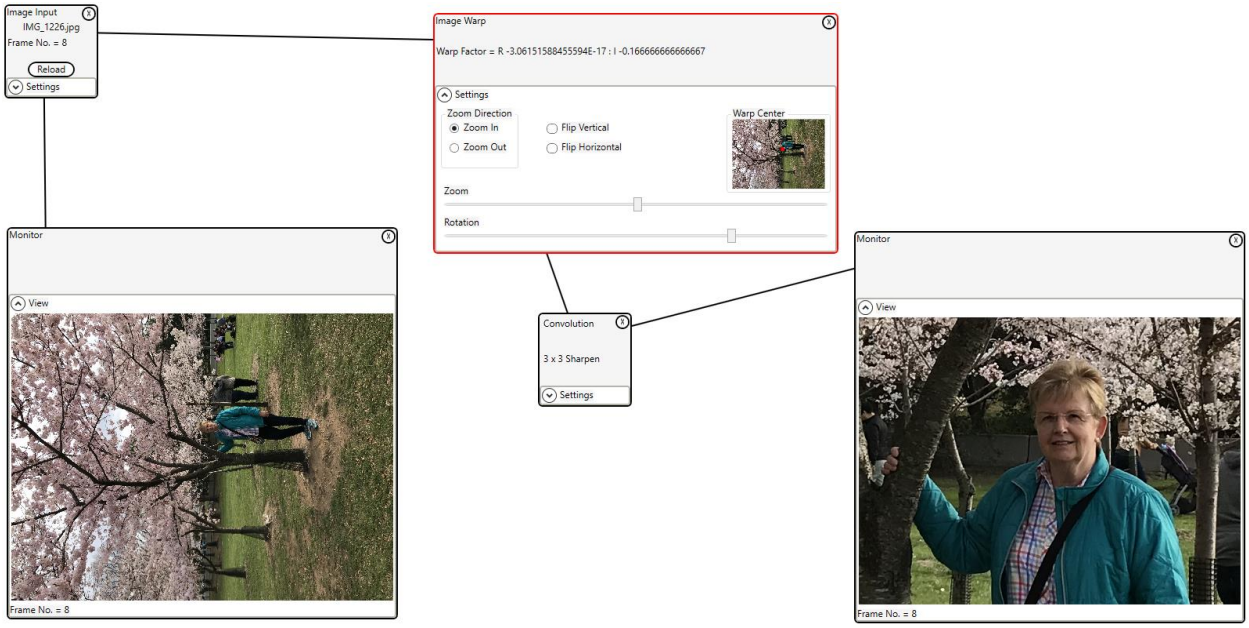
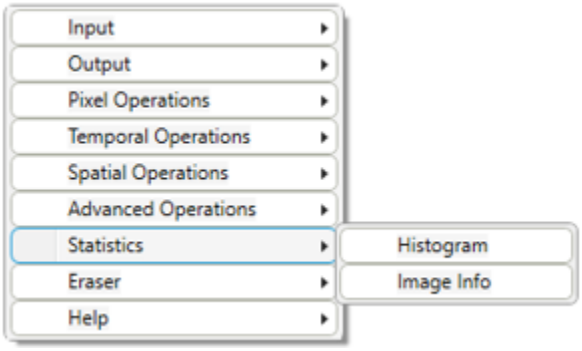


Figure 49 : Extracting a detail and using sharpening to reduce blurring

Statistics



Statistical functions provide information about the image.

Histogram Control

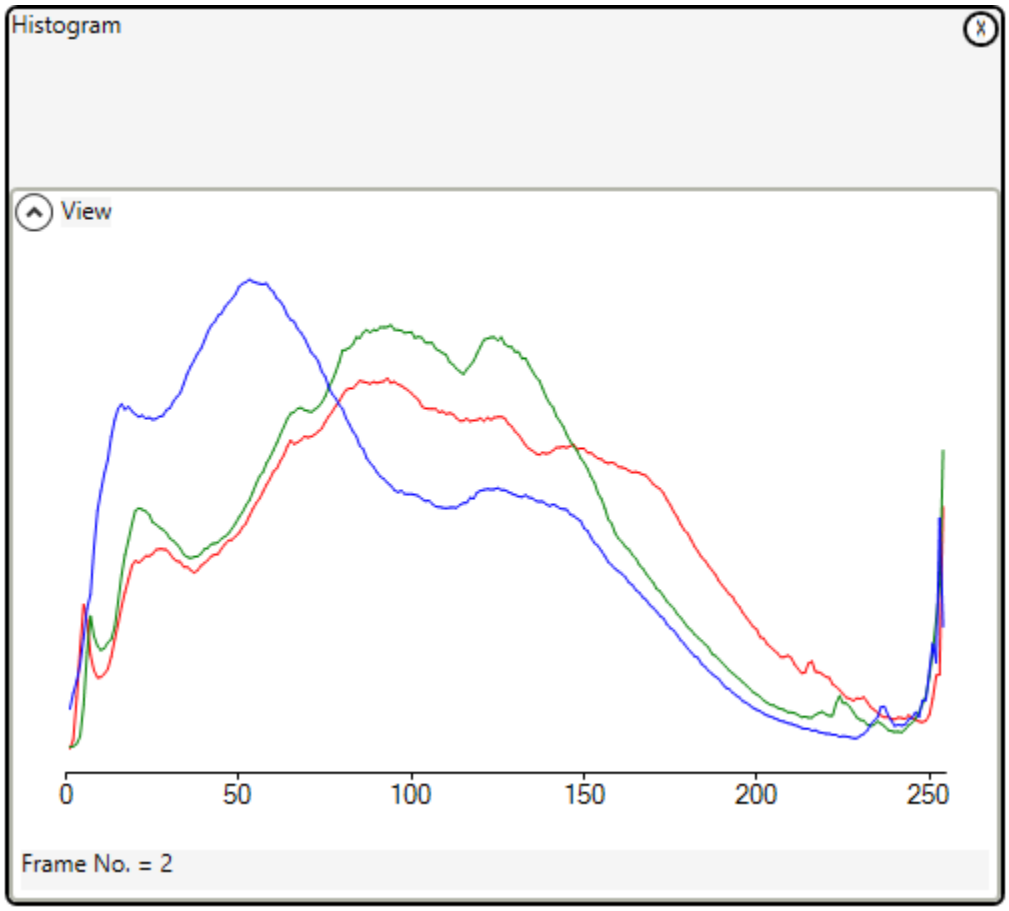


Figure 50 : Histogram control with view expanded

The histogram control will show graphically, for each color, the number of pixels of each value in the image. The view may be collapsed to save space on the whiteboard.

Image Info Control

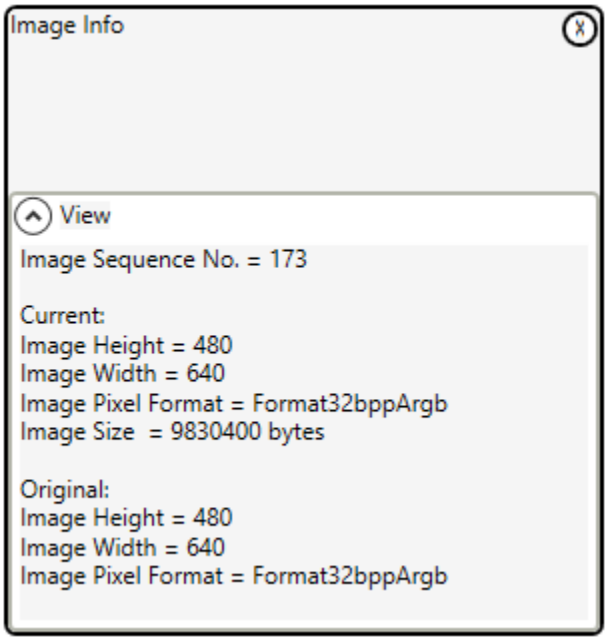
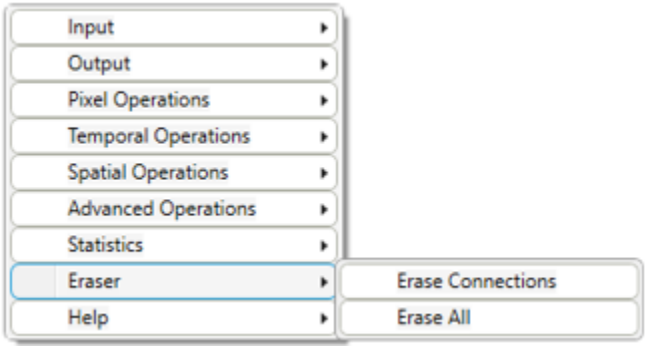


Figure 51 : Image Info Control with view expanded

The image info control will display basic information about the image. The current image is the double precision floating point image that is used for processing, the original image is the bitmap image that was originally acquired by the camera or from disc.

Eraser Functions

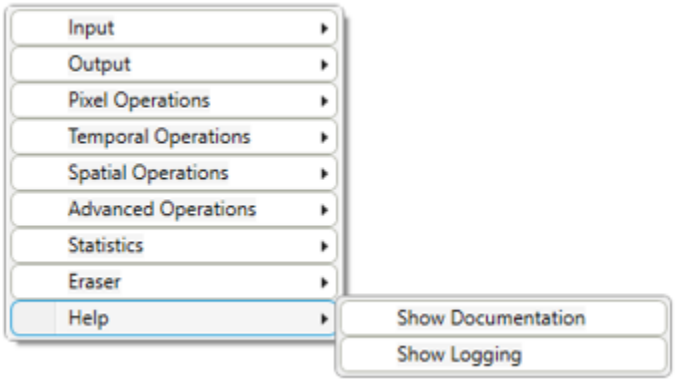


To remove a control and its connections the delete button on the control should be used.

Erase connections will just erase all connection on the whiteboard and leave the controls in place.

Erase all will clear the whiteboard and return it to its original blank state.

Help Functions



Show documentation will launch this document.

Show logging will launch the log viewer to display debug logging.

Performance

Real-time image processing has historically (1970s – 1990s) been performed by specialized hardware https://en.wikipedia.org/wiki/Datacube_Inc. Real-time performance was guaranteed by using as much hardware as was required for a given algorithm, for example if your algorithm required a convolution you would add a convolver to the hardware. Always there were people predicting that one day soon a PC would be powerful enough to match the performance of pipelined hardware. In 2018 is this true yet? Part of the motivation for this project was to find out.

There has been some partial success. Temporal operations require real-time as they require processing to be performed on successive frames; these work well with lower resolution images. Even image warping runs real-time at low resolutions. The limitations are not so much the processing required, certainly with modern multi-core processors, floating point performance is adequate for most algorithms. The issue is with memory, once local memory is used virtual memory has to be employed. The other issue is that as your algorithm increases in complexity you cannot add more hardware. FFT filtering does not run real-time, but, even with pipeline processors FFTs never did run real time.

Compromises

The Imaging Whiteboard handles performance issues in various ways, depending on the input used.

Image file, and test patterns are single shot and are not required to run real-time.

Video Files are usually of too high a resolution to run real-time. The Video Input control will read the file as fast as possible and all frames will be processed.

Camera Input cannot be slowed so here frames that cannot be processed will be dropped.

What helps?

More memory is the most useful enhancement. The machine used for development had 32GB.

A SSD (solid state drive) is useful, performance on a Surface one seems better than on a more powerful desktop with a hard drive.

Use lower resolution if possible, how low depends on your hardware.

Contact

If you have any questions, suggestions, or feedback email steve@sound-analysis.com