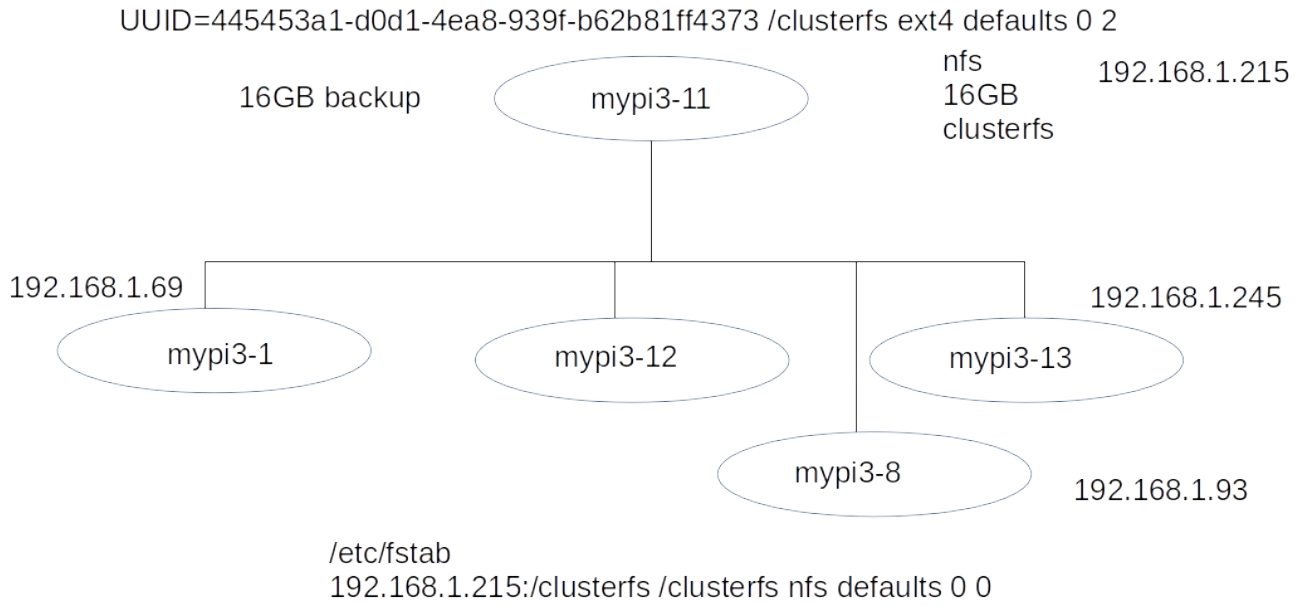
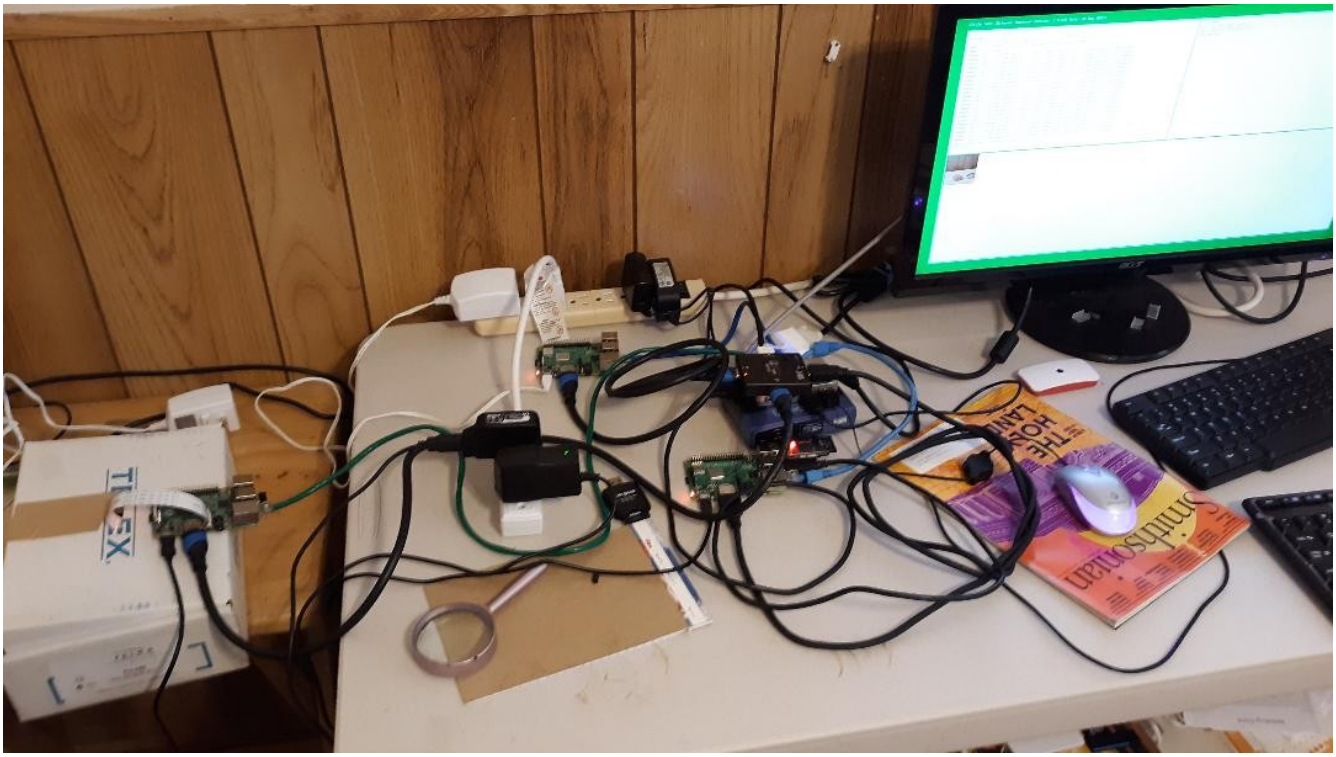
Draft 08/12/19

Met with Dr. Watson and tested the openmv-cam.
The GUI is QT based for Windows the exe contains everything needed.
Not good for embedded operation. The box on left is the Rpi2B with camera mypi3-8 RaspBian buster . The Rpi3B+ near the power strip is the Ultibo Bare Metal. To left of magzine is a Rpi3B+ mypi3-11 RasBian buster. The Rpi3B+ mypi3-11 provides a NFS file system to the lab.

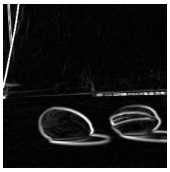


In the area above Rpi3B+ is a HDMI 3 in 1out switch a USB Hub, and a 4 port Ethernet hub.



Started looking at Rpi2
Was able to get a number of frames collected in bmp format.

Edge



541 frames in 5 min 300 sec 1.8 sec /frame Is this fast enough?

frame0000.bmp to framexxxxx.bmp
The tested the image obtained on Raspbian using Ultibo Bare Metal JPEG2000 Openjpeg which is made up 22 C files which is quite a bit of processing.

/home/devel/Ultibo_Projects/jpeg2000/src/bio.c
/home/devel/Ultibo_Projects/jpeg2000/src/cio.c
/home/devel/Ultibo_Projects/jpeg2000/src/dwt.c
/home/devel/Ultibo_Projects/jpeg2000/src/event.c
/home/devel/Ultibo_Projects/jpeg2000/src/function_list.c
/home/devel/Ultibo_Projects/jpeg2000/src/image.c
/home/devel/Ultibo_Projects/jpeg2000/src/invert.c
/home/devel/Ultibo_Projects/jpeg2000/src/j2k.c
/home/devel/Ultibo_Projects/jpeg2000/src/jp2.c
/home/devel/Ultibo_Projects/jpeg2000/src/mct.c
/home/devel/Ultibo_Projects/jpeg2000/src/mqc.c
/home/devel/Ultibo_Projects/jpeg2000/src/openjpeg.c
/home/devel/Ultibo_Projects/jpeg2000/src/opj_clock.c
/home/devel/Ultibo_Projects/jpeg2000/src/opj_malloc.c
/home/devel/Ultibo_Projects/jpeg2000/src/pi.c
/home/devel/Ultibo_Projects/jpeg2000/src/raw.c
/home/devel/Ultibo_Projects/jpeg2000/src/t1.c
/home/devel/Ultibo_Projects/jpeg2000/src/t2.c
/home/devel/Ultibo_Projects/jpeg2000/src/tcd.c
/home/devel/Ultibo_Projects/jpeg2000/src/tgt.c
/home/devel/Ultibo_Projects/jpeg2000/src/thread.c

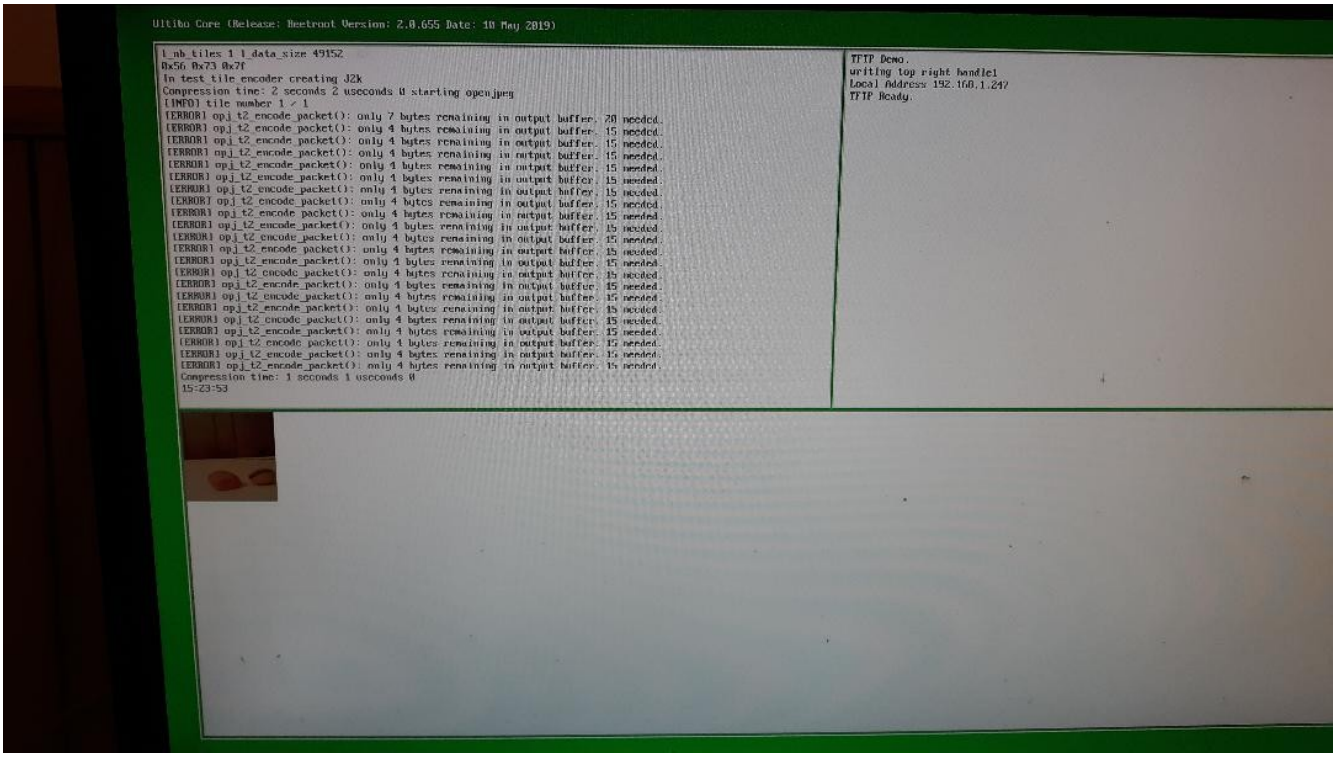
15-5-19 22:24:28 3197924 kernel7.img
firmware
devel@mypi3-8:/clusterfs/firmwar_for_ultibo $ ls 042519/
bootcode.bin fixup.dat fixup_x.dat start.elf start_x.elf

13-8-19 06:13:04 49206 frame0000.bmp
Input to openjeg program
13-8-19 06:13:04 49206 MyBitmap.bmp
output
3-8-19 15:23:52 409 test.j2k
devel@mypi3-8:~/camerawatson/timelapse $ tftp 192.168.1.247
tftp> binary
tftp> get test.j2k
Received 409 bytes in 0.0 seconds
tftp> quit

devel@mypi3-8:~/camerawatson/timelapse $ /clusterfs/t_ultibo/build/bin/opj_decompress -i test.j2k -o tt.bmp

[INFO] Start to read j2k main header (3481168).
[INFO] Main header has been correctly decoded.
[INFO] No decoded area parameters, set the decoded area to the whole image
[INFO] Header of tile 1 / 1 has been read.
[INFO] Generated Outfile tt.bmp
decode time: 8 ms
The decompressed image

bare metal



Items unknown
time between frames
type of processing Singular Value Decomposition SVD on 128 x 128 image is plan
The SVD currently uses a rgb_pack.bin which has 3 values for the image is a single 32 bit word used by FPGA processing & th_svd
https://github.com/develone/Ultibo_Projects/blob/master/th_svd/doc/threaded_svd_ex.pdf
These also use octave for visualization of the images the RaspBian version buster is not currently working the RaspBian stretch version is okay which is on some of the older Rpi3B+ mypi3-1.

This currently executes on RaspBian or Ultibo Bare Metal.
This requires modify to read bitmap or conversion of bitmap to rgb_packed.bin a single 32 bit word.
type of jig for camera
    requires externel light
    mirrors
processing capability
    current tests with a Raspberry Pi 2 B V1.1 Quad Core running RaspBian (buster) Linux with camera.

The Ultibo Bare metal supports NFS I have not done any testing at this time.
Is NFS needed?
If all work done on Rpi4B maybe will not need a NFS file system. Maybe just RaspBian Linux.
I have not tested the camera on a Rpi4B.