Embedded Machine Vision and Intelligent Automation

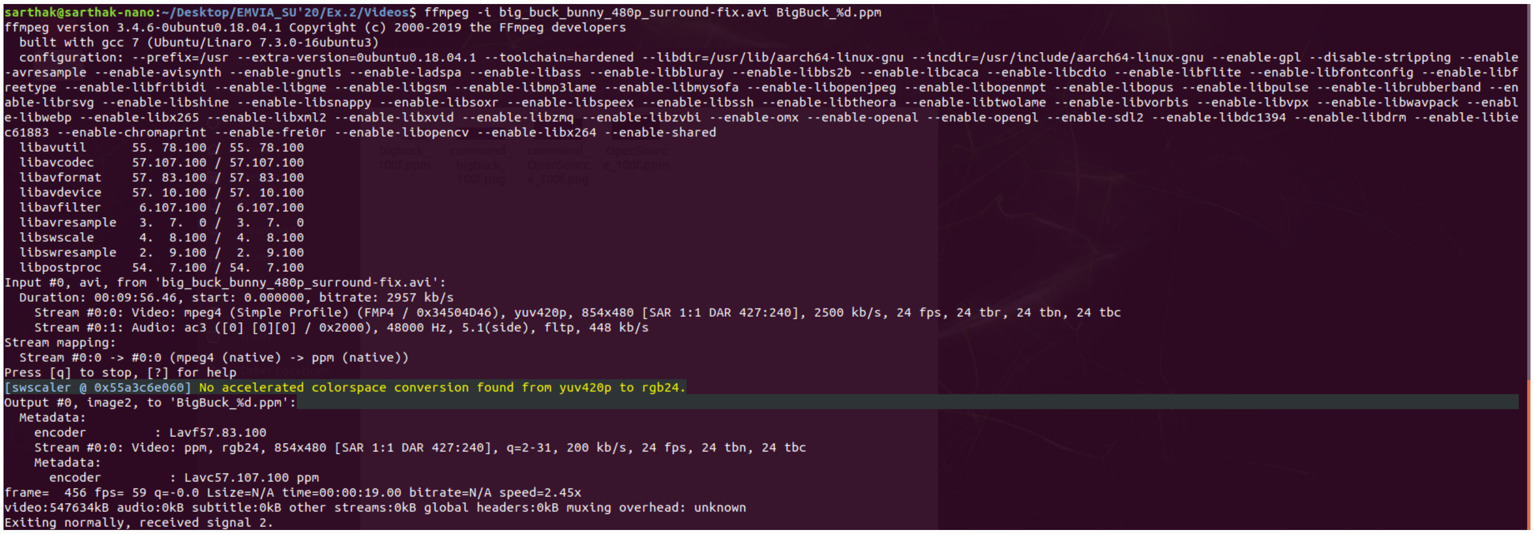
Exercise 2

Sarthak Jain

June 21, 2019

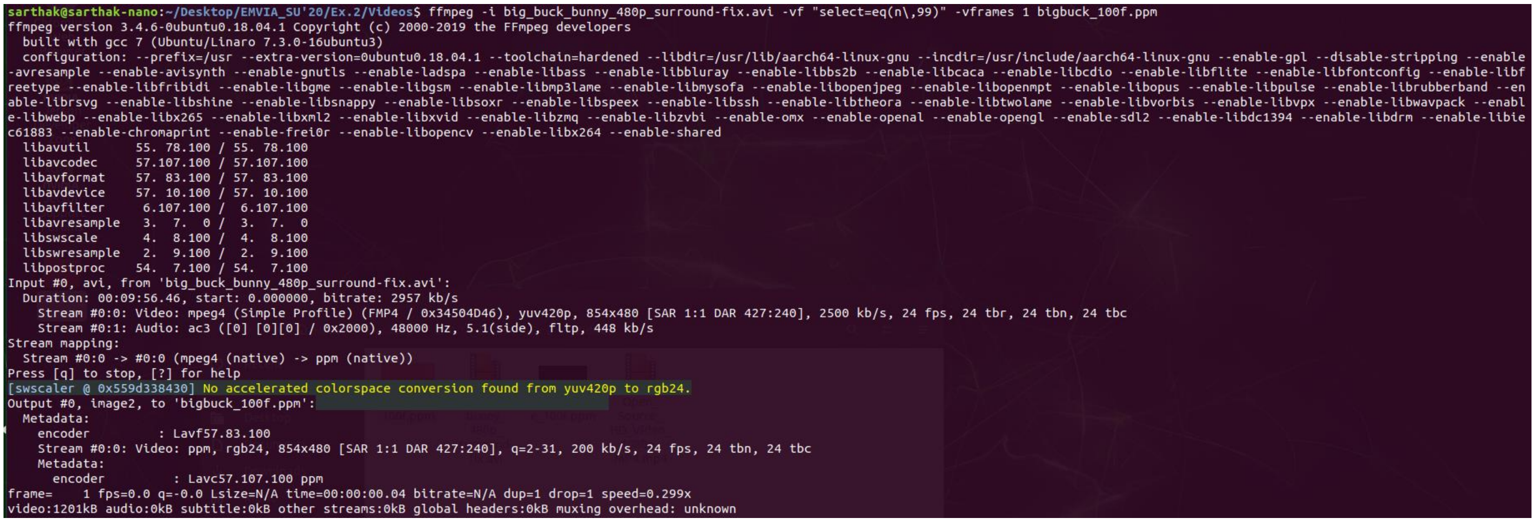
Performed on Jetson Nano

Question 1



```
sarthak@sarthak-nano:~/Desktop/EMVIA_SU'20/Ex.2/Videos$ ffmpeg -i big_buck_bunny_480p_surround-fix.avi BigBuck_%d.ppm
ffmpeg version 3.4.6-0ubuntu0.18.04.1 Copyright (c) 2000-2019 the FFmpeg developers
  built with gcc 7 (Ubuntu/Linaro 7.3.0-16ubuntu3)
  configuration: --prefix=/usr --extra-version=0ubuntu0.18.04.1 --toolchain=hardened --libdir=/usr/lib/aarch64-linux-gnu --incdir=/usr/include/aarch64-linux-gnu --enable-gpl --disable-stripping --enable
-avresample --enable-avisynth --enable-gnutls --enable-ladspa --enable-libass --enable-libbluray --enable-libbs2b --enable-libcaca --enable-libcdio --enable-libflite --enable-libfontconfig --enable-libf
reetype --enable-libfribidi --enable-libgme --enable-libgsm --enable-libmp3lame --enable-libmysofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus --enable-libpulse --enable-librubberband --en
able-librsvg --enable-libshine --enable-libsnappy --enable-libsoxr --enable-libspeex --enable-libssh --enable-libtheora --enable-libtwolame --enable-libvorbis --enable-libvpx --enable-libwavpack --enabl
e-libwebp --enable-libx265 --enable-libxml2 --enable-libxvid --enable-libzmq --enable-libzvbi --enable-omx --enable-openal --enable-opengl --enable-sdl2 --enable-libdc1394 --enable-libdrm --enable-libie
c61883 --enable-chromaprint --enable-frei0r --enable-libopencv --enable-libx264 --enable-shared
  libavutil      55. 78.100 / 55. 78.100
  libavcodec     57.107.100 / 57.107.100
  libavformat    57. 83.100 / 57. 83.100
  libavdevice    57. 10.100 / 57. 10.100
  libavfilter     6.107.100 /  6.107.100
  libavresample   3.  7.  0 /  3.  7.  0
  libswscale      4.  8.100 /  4.  8.100
  libswresample   2.  9.100 /  2.  9.100
  libpostproc    54.  7.100 / 54.  7.100
Input #0, avi, from 'big_buck_bunny_480p_surround-fix.avi':
  Duration: 00:09:56.46, start: 0.000000, bitrate: 2957 kb/s
    Stream #0:0: Video: mpeg4 (Simple Profile) (FMP4 / 0x34504D46), yuv420p, 854x480 [SAR 1:1 DAR 427:240], 2500 kb/s, 24 fps, 24 tbr, 24 tbn, 24 tbc
    Stream #0:1: Audio: ac3 ([0] [0][0] / 0x2000), 48000 Hz, 5.1(side), fltp, 448 kb/s
Stream mapping:
  Stream #0:0 -> #0:0 (mpeg4 (native) -> ppm (native))
Press [q] to stop, [?] for help
[swscaler @ 0x55a3c6e060] No accelerated colorspace conversion found from yuv420p to rgb24.
Output #0, image2, to 'BigBuck_%d.ppm':
  Metadata:
    encoder         : Lavf57.83.100
    Stream #0:0: Video: ppm, rgb24, 854x480 [SAR 1:1 DAR 427:240], q=2-31, 200 kb/s, 24 fps, 24 tbn, 24 tbc
    Metadata:
      encoder         : Lavc57.107.100 ppm
frame=  456 fps= 59 q=-0.0 Lsize=N/A time=00:00:19.00 bitrate=N/A speed=2.45x
video:547634kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: unknown
Exiting normally, received signal 2.
```

Command used to extract all frames of video BigBuck

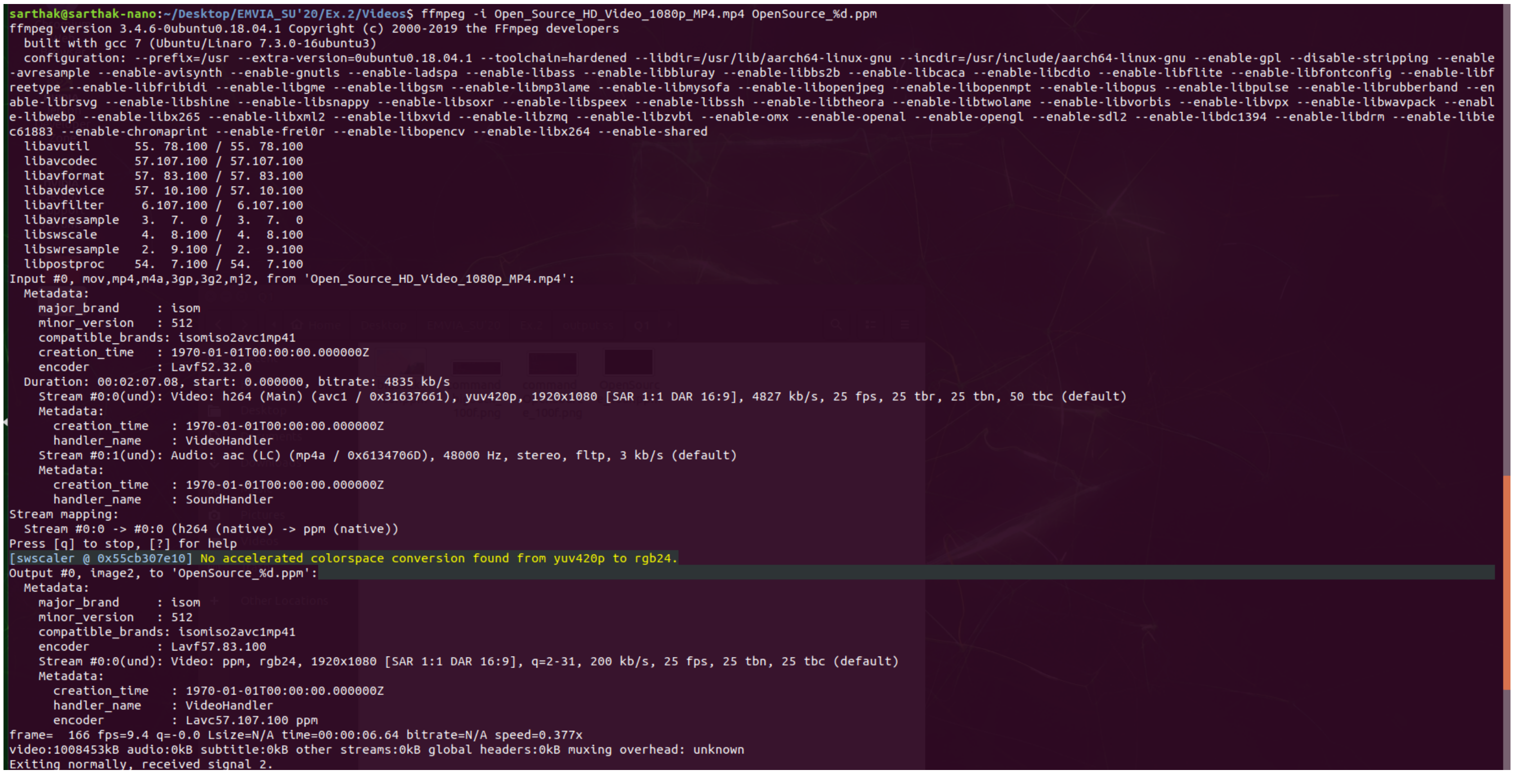


```
sarthak@sarthak-nano:~/Desktop/EMVIA_SU'20/Ex.2/Videos$ ffmpeg -i big_buck_bunny_480p_surround-fix.avi -vf "select=eq(n\,99)" -vframes 1 bigbuck_100f.ppm
ffmpeg version 3.4.6-0ubuntu0.18.04.1 Copyright (c) 2000-2019 the FFmpeg developers
  built with gcc 7 (Ubuntu/Linaro 7.3.0-16ubuntu3)
  configuration: --prefix=/usr --extra-version=0ubuntu0.18.04.1 --toolchain=hardened --libdir=/usr/lib/aarch64-linux-gnu --incdir=/usr/include/aarch64-linux-gnu --enable-gpl --disable-stripping --enable
-avresample --enable-avisynth --enable-gnutls --enable-ladspa --enable-libass --enable-libbluray --enable-libbs2b --enable-libcaca --enable-libcdio --enable-libflite --enable-libfontconfig --enable-libf
reetype --enable-libfribidi --enable-libgme --enable-libgsm --enable-libmp3lame --enable-libmysofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus --enable-libpulse --enable-librubberband --en
able-librsvg --enable-libshine --enable-libsnappy --enable-libsoxr --enable-libspeex --enable-libssh --enable-libtheora --enable-libtwolame --enable-libvorbis --enable-libvpx --enable-libwavpack --enabl
e-libwebp --enable-libx265 --enable-libxml2 --enable-libxvid --enable-libzmq --enable-libzvbi --enable-omx --enable-openal --enable-opengl --enable-sdl2 --enable-libdc1394 --enable-libdrm --enable-libie
c61883 --enable-chromaprint --enable-frei0r --enable-libopencv --enable-libx264 --enable-shared
  libavutil      55. 78.100 / 55. 78.100
  libavcodec     57.107.100 / 57.107.100
  libavformat    57. 83.100 / 57. 83.100
  libavdevice    57. 10.100 / 57. 10.100
  libavfilter     6.107.100 /  6.107.100
  libavresample   3.  7.  0 /  3.  7.  0
  libswscale      4.  8.100 /  4.  8.100
  libswresample   2.  9.100 /  2.  9.100
  libpostproc    54.  7.100 / 54.  7.100
Input #0, avi, from 'big_buck_bunny_480p_surround-fix.avi':
  Duration: 00:09:56.46, start: 0.000000, bitrate: 2957 kb/s
    Stream #0:0: Video: mpeg4 (Simple Profile) (FMP4 / 0x34504D46), yuv420p, 854x480 [SAR 1:1 DAR 427:240], 2500 kb/s, 24 fps, 24 tbr, 24 tbn, 24 tbc
    Stream #0:1: Audio: ac3 ([0] [0][0] / 0x2000), 48000 Hz, 5.1(side), fltp, 448 kb/s
Stream mapping:
  Stream #0:0 -> #0:0 (mpeg4 (native) -> ppm (native))
Press [q] to stop, [?] for help
[swscaler @ 0x559d338430] No accelerated colorspace conversion found from yuv420p to rgb24.
Output #0, image2, to 'bigbuck_100f.ppm':
  Metadata:
    encoder         : Lavf57.83.100
    Stream #0:0: Video: ppm, rgb24, 854x480 [SAR 1:1 DAR 427:240], q=2-31, 200 kb/s, 24 fps, 24 tbn, 24 tbc
    Metadata:
      encoder         : Lavc57.107.100 ppm
frame=    1 fps=0.0 q=-0.0 Lsize=N/A time=00:00:00.04 bitrate=N/A dup=1 drop=1 speed=0.299x
video:1201kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: unknown
```

Command used to extract 100th frame from video BigBuck

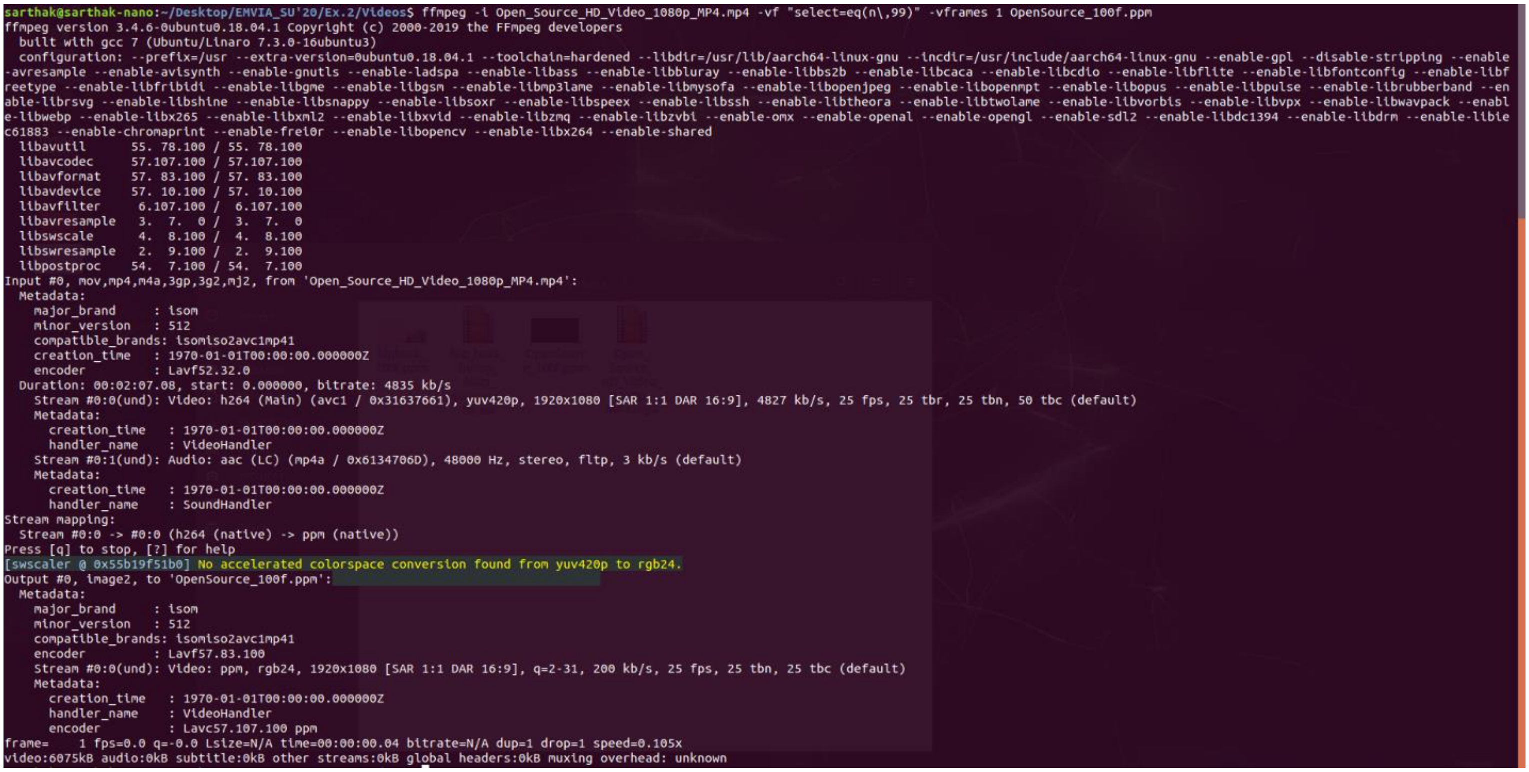
100th frame of BigBuck video

For doing so, I have used the select filter of ffmpeg to extract frame equal to 100th frame.



```
sarthak@sarthak-nano:~/Desktop/EMVIA_SU'20/Ex.2/Videos$ ffmpeg -i Open_Source_HD_Video_1080p_MP4.mp4 OpenSource_%d.ppm
ffmpeg version 3.4.6-0ubuntu0.18.04.1 Copyright (c) 2000-2019 the FFmpeg developers
  built with gcc 7 (Ubuntu/Linaro 7.3.0-16ubuntu3)
  configuration: --prefix=/usr --extra-version=0ubuntu0.18.04.1 --toolchain=hardened --libdir=/usr/lib/aarch64-linux-gnu --incdir=/usr/include/aarch64-linux-gnu --enable-gpl --disable-stripping --enable
-avresample --enable-avisynth --enable-gnutls --enable-ladspa --enable-libass --enable-libbluray --enable-libbs2b --enable-libcaca --enable-libcdio --enable-libflite --enable-libfontconfig --enable-libf
reetype --enable-libfribidi --enable-libgme --enable-libgsm --enable-libmp3lame --enable-libmysofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus --enable-libpulse --enable-librubberband --en
able-librsvg --enable-libshine --enable-libsnappy --enable-libsoxr --enable-libspeex --enable-libssh --enable-libtheora --enable-libtwolame --enable-libvorbis --enable-libvpx --enable-libwavpack --enabl
e-libwebp --enable-libx265 --enable-libxml2 --enable-libxvid --enable-libzmq --enable-libzvbi --enable-omx --enable-openal --enable-opengl --enable-sdl2 --enable-libdc1394 --enable-libdrm --enable-libie
c61883 --enable-chromaprint --enable-frei0r --enable-libopencv --enable-libx264 --enable-shared
  libavutil      55. 78.100 / 55. 78.100
  libavcodec     57.107.100 / 57.107.100
  libavformat    57. 83.100 / 57. 83.100
  libavdevice    57. 10.100 / 57. 10.100
  libavfilter     6.107.100 /  6.107.100
  libavresample   3.  7.  0 /  3.  7.  0
  libswscale      4.  8.100 /  4.  8.100
  libswresample   2.  9.100 /  2.  9.100
  libpostproc    54.  7.100 / 54.  7.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'Open_Source_HD_Video_1080p_MP4.mp4':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    creation_time   : 1970-01-01T00:00:00.000000Z
    encoder         : Lavf52.32.0
  Duration: 00:02:07.08, start: 0.000000, bitrate: 4835 kb/s
    Stream #0:0(und): Video: h264 (Main) (avc1 / 0x31637661), yuv420p, 1920x1080 [SAR 1:1 DAR 16:9], 4827 kb/s, 25 fps, 25 tbr, 25 tbn, 50 tbc (default)
    Metadata:
      creation_time   : 1970-01-01T00:00:00.000000Z
      handler_name    : VideoHandler
    Stream #0:1(und): Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz, stereo, fltp, 3 kb/s (default)
    Metadata:
      creation_time   : 1970-01-01T00:00:00.000000Z
      handler_name    : SoundHandler
Stream mapping:
  Stream #0:0 -> #0:0 (h264 (native) -> ppm (native))
Press [q] to stop, [?] for help
[swscaler @ 0x55cb307e10] No accelerated colorspace conversion found from yuv420p to rgb24.
Output #0, image2, to 'OpenSource_%d.ppm':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf57.83.100
    Stream #0:0(und): Video: ppm, rgb24, 1920x1080 [SAR 1:1 DAR 16:9], q=2-31, 200 kb/s, 25 fps, 25 tbn, 25 tbc (default)
    Metadata:
      creation_time   : 1970-01-01T00:00:00.000000Z
      handler_name    : VideoHandler
      encoder         : Lavc57.107.100 ppm
frame=  166 fps=9.4 q=-0.0 Lsize=N/A time=00:00:06.64 bitrate=N/A speed=0.377x
video:1008453kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: unknown
Exiting normally, received signal 2.
```

Command used to extract all frames of video OpenSource

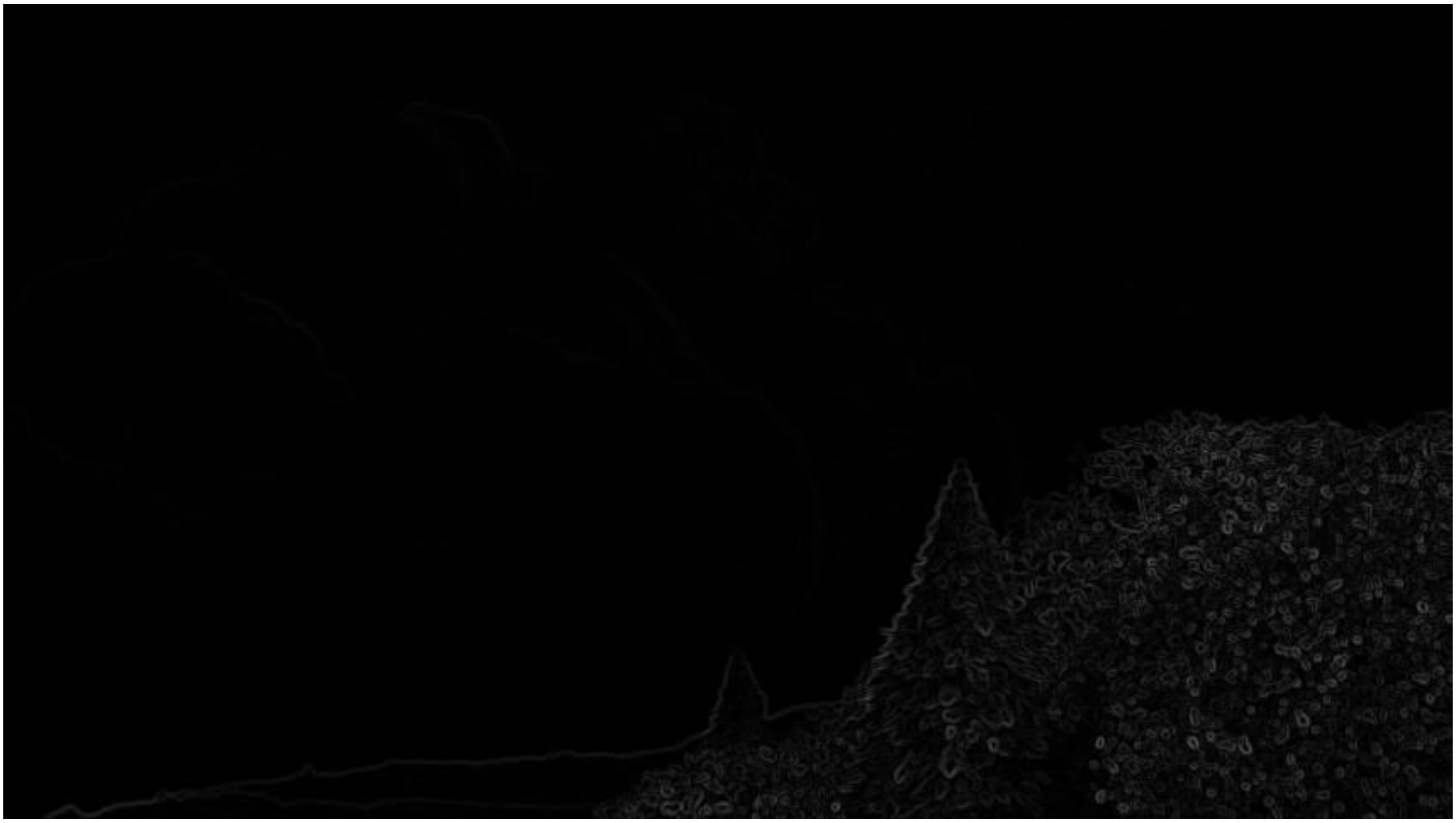


```
sarthak@sarthak-nano:~/Desktop/EMVIA_SU'20/Ex.2/Videos$ ffmpeg -i Open_Source_HD_Video_1080p_MP4.mp4 -vf "select=eq(n\,99)" -vframes 1 OpenSource_100f.ppm
ffmpeg version 3.4.6-0ubuntu0.18.04.1 Copyright (c) 2000-2019 the FFmpeg developers
  built with gcc 7 (Ubuntu/Linaro 7.3.0-16ubuntu3)
  configuration: --prefix=/usr --extra-version=0ubuntu0.18.04.1 --toolchain=hardened --libdir=/usr/lib/aarch64-linux-gnu --incdir=/usr/include/aarch64-linux-gnu --enable-gpl --disable-stripping --enable
-avresample --enable-avisynth --enable-gnutls --enable-ladspa --enable-libass --enable-libbluray --enable-libbs2b --enable-libcaca --enable-libcdio --enable-libflite --enable-libfontconfig --enable-libf
reetype --enable-libfribidi --enable-libgme --enable-libgsm --enable-libmp3lame --enable-libmysofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus --enable-libpulse --enable-librubberband --en
able-librsvg --enable-libshine --enable-libsnappy --enable-libsoxr --enable-libspeex --enable-libssh --enable-libtheora --enable-libtwolame --enable-libvorbis --enable-libvpx --enable-libwavpack --enabl
e-libwebp --enable-libx265 --enable-libxml2 --enable-libxvid --enable-libzmq --enable-libzvbi --enable-omx --enable-openal --enable-opengl --enable-sdl2 --enable-libdc1394 --enable-libdrm --enable-libie
c61883 --enable-chromaprint --enable-frei0r --enable-libopencv --enable-libx264 --enable-shared
  libavutil      55. 78.100 / 55. 78.100
  libavcodec     57.107.100 / 57.107.100
  libavformat    57. 83.100 / 57. 83.100
  libavdevice    57. 10.100 / 57. 10.100
  libavfilter     6.107.100 /  6.107.100
  libavresample   3.  7.  0 /  3.  7.  0
  libswscale      4.  8.100 /  4.  8.100
  libswresample   2.  9.100 /  2.  9.100
  libpostproc    54.  7.100 / 54.  7.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'Open_Source_HD_Video_1080p_MP4.mp4':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    creation_time   : 1970-01-01T00:00:00.000000Z
    encoder         : Lavf52.32.0
  Duration: 00:02:07.08, start: 0.000000, bitrate: 4835 kb/s
    Stream #0:0(und): Video: h264 (Main) (avc1 / 0x31637661), yuv420p, 1920x1080 [SAR 1:1 DAR 16:9], 4827 kb/s, 25 fps, 25 tbr, 25 tbn, 50 tbc (default)
    Metadata:
      creation_time   : 1970-01-01T00:00:00.000000Z
      handler_name    : VideoHandler
    Stream #0:1(und): Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz, stereo, fltp, 3 kb/s (default)
    Metadata:
      creation_time   : 1970-01-01T00:00:00.000000Z
      handler_name    : SoundHandler
Stream mapping:
  Stream #0:0 -> #0:0 (h264 (native) -> ppm (native))
Press [q] to stop, [?] for help
[swscaler @ 0x55b19f51b0] No accelerated colorspace conversion found from yuv420p to rgb24.
Output #0, image2, to 'OpenSource_100f.ppm':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf57.83.100
    Stream #0:0(und): Video: ppm, rgb24, 1920x1080 [SAR 1:1 DAR 16:9], q=2-31, 200 kb/s, 25 fps, 25 tbn, 25 tbc (default)
    Metadata:
      creation_time   : 1970-01-01T00:00:00.000000Z
      handler_name    : VideoHandler
      encoder         : Lavc57.107.100 ppm
frame=    1 fps=0.0 q=-0.0 Lsize=N/A time=00:00:00.04 bitrate=N/A dup=1 drop=1 speed=0.105x
video:6075kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: unknown
```

Command used to extract 100th frame from video OpenSource

100th frame of OpenSource video

Question 2



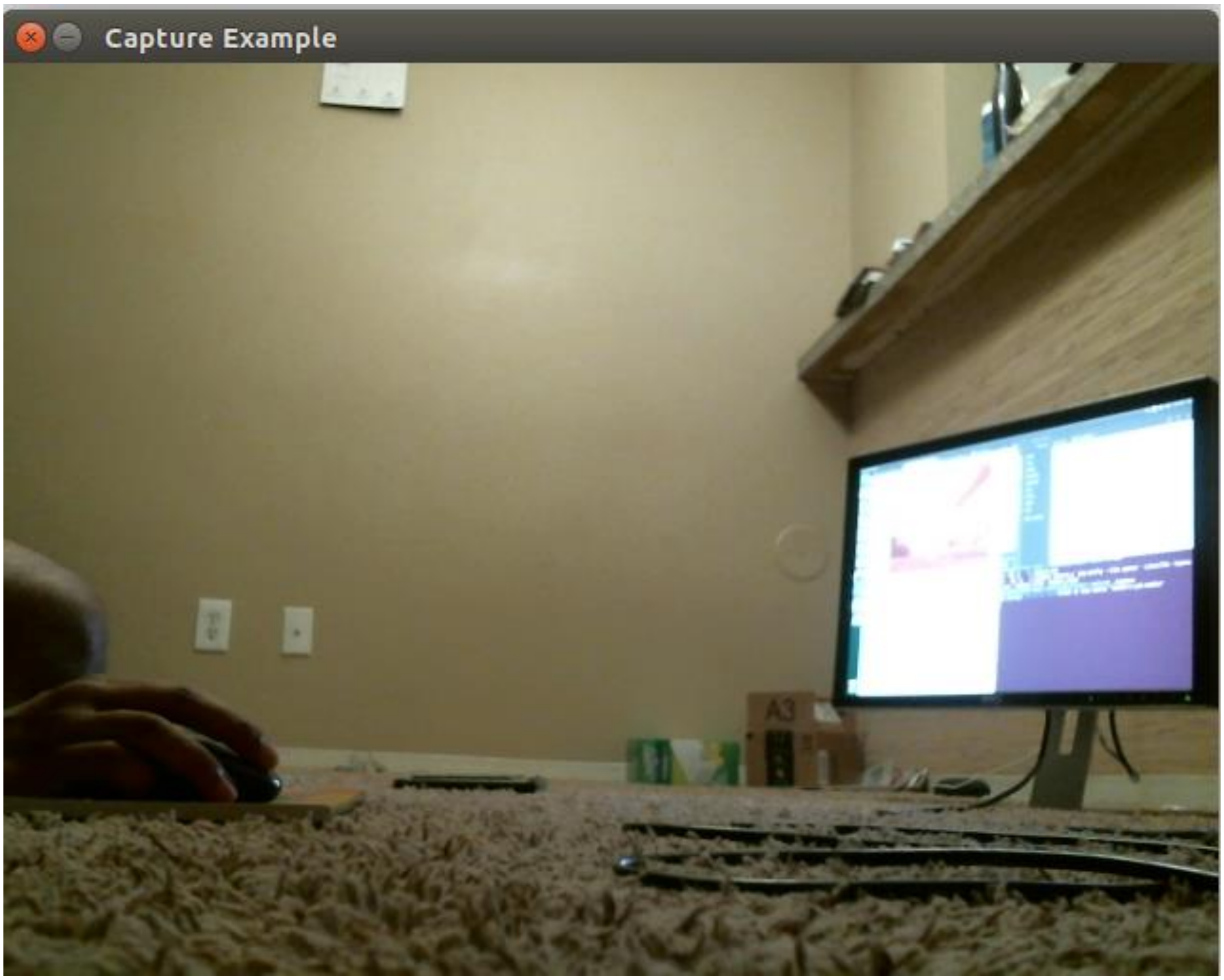
Sobel Transform on 100th frame of BigBuck video

Sobel Transform on 100th frame of OpenSource video

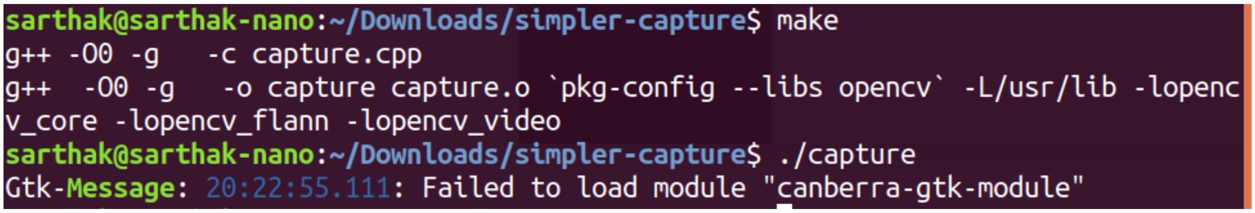
The options I used were twofold; initially I used the Sobel Edge Transform under the Edge Detect options, however this led to a colored Sobel Transform, which I was not sure was desired. To convert the image to grayscale, I used the Colorify filter with the color palette of black and white selected, which has the effect of turning the image into black and white. The same result was achieved using the Grayscale filter, which can be found under image options -> mode.

Question 3

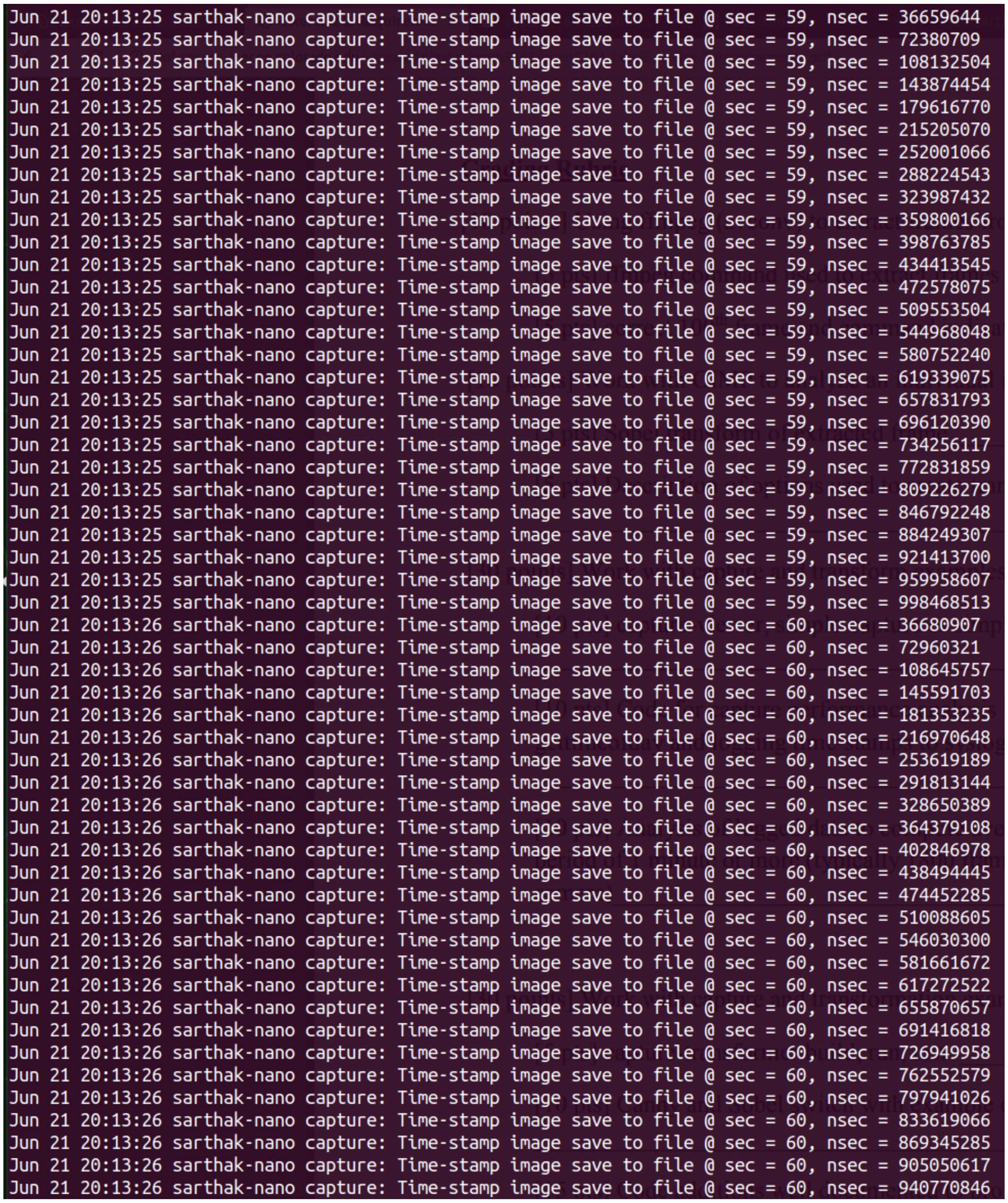
The code I decided to use for the purpose of video capture was the simpler-capture code. Provided below is a screenshot of my home lab, using the Logitech C615 camera:



Provided below is a screenshot of build and run of the simpler-capture code:

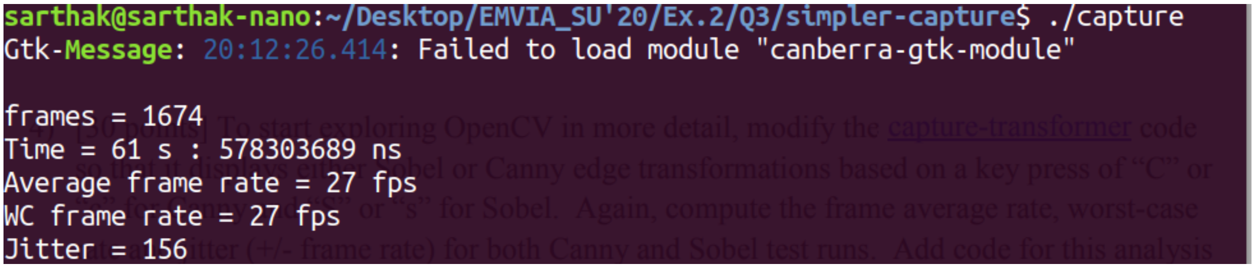


```
sarthak@sarthak-nano:~/Downloads/simpler-capture$ make
g++ -O0 -g   -c capture.cpp
g++  -O0 -g   -o capture capture.o `pkg-config --libs opencv` -L/usr/lib -lopenc
v_core -lopencv_flann -lopencv_video
sarthak@sarthak-nano:~/Downloads/simpler-capture$ ./capture
Gtk-Message: 20:22:55.111: Failed to load module "canberra-gtk-module"
```



```
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 36659644
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 72380709
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 108132504
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 143874454
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 179616770
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 215205070
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 252001066
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 288224543
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 323987432
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 359800166
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 398763785
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 434413545
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 472578075
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 509553504
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 544968048
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 580752240
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 619339075
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 657831793
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 696120390
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 734256117
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 772831859
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 809226279
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 846792248
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 884249307
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 921413700
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 959958607
Jun 21 20:13:25 sarthak-nano capture: Time-stamp image save to file @ sec = 59, nsec = 998468513
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 36680907
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 72960321
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 108645757
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 145591703
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 181353235
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 216970648
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 253619189
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 291813144
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 328650389
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 364379108
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 402846978
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 438494445
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 474452285
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 510088605
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 546030300
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 581661672
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 617272522
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 655870657
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 691416818
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 726949958
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 762552579
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 797941026
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 833619066
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 869345285
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 905050617
Jun 21 20:13:26 sarthak-nano capture: Time-stamp image save to file @ sec = 60, nsec = 940770846
```

Screenshot of timestamps being logged via syslog using clock_gettime



```
sarthak@sarthak-nano:~/Desktop/EMVIA_SU'20/Ex.2/Q3/simpler-capture$ ./capture
Gtk-Message: 20:12:26.414: Failed to load module "canberra-gtk-module"

frames = 1674
Time = 61 s : 578303689 ns
Average frame rate = 27 fps
WC frame rate = 27 fps
Jitter = 156
```

Analysis of average frame rate, worst-case and jitter

Code Description:

The ‘simpler-capture’ code was modified to include a few ‘clock_gettime()’ API calls for obtaining the time. An initial call was made for ‘start_time’. In the while loop, a code logic for computing the worst-case frame rate is included. A logic for monitoring the step of each second is used, and each time a second passes, framerate is computed for that second. If found to be less than that of the previous second, worst-case framerate is updated with the new value. For computing average rate, a ‘stop_time’ variable is used, difference of start and stop times is computed, and division with number of frames gives the average frame rate. Jitter is computed by assuming an ideal framerate of 30 FPS. Multiplied by the amount of time over which the program runs, this gives the ideal number of frames acquired in that time period. Difference with the actual number of frames gives the jitter over that time period.

Analysis of results:

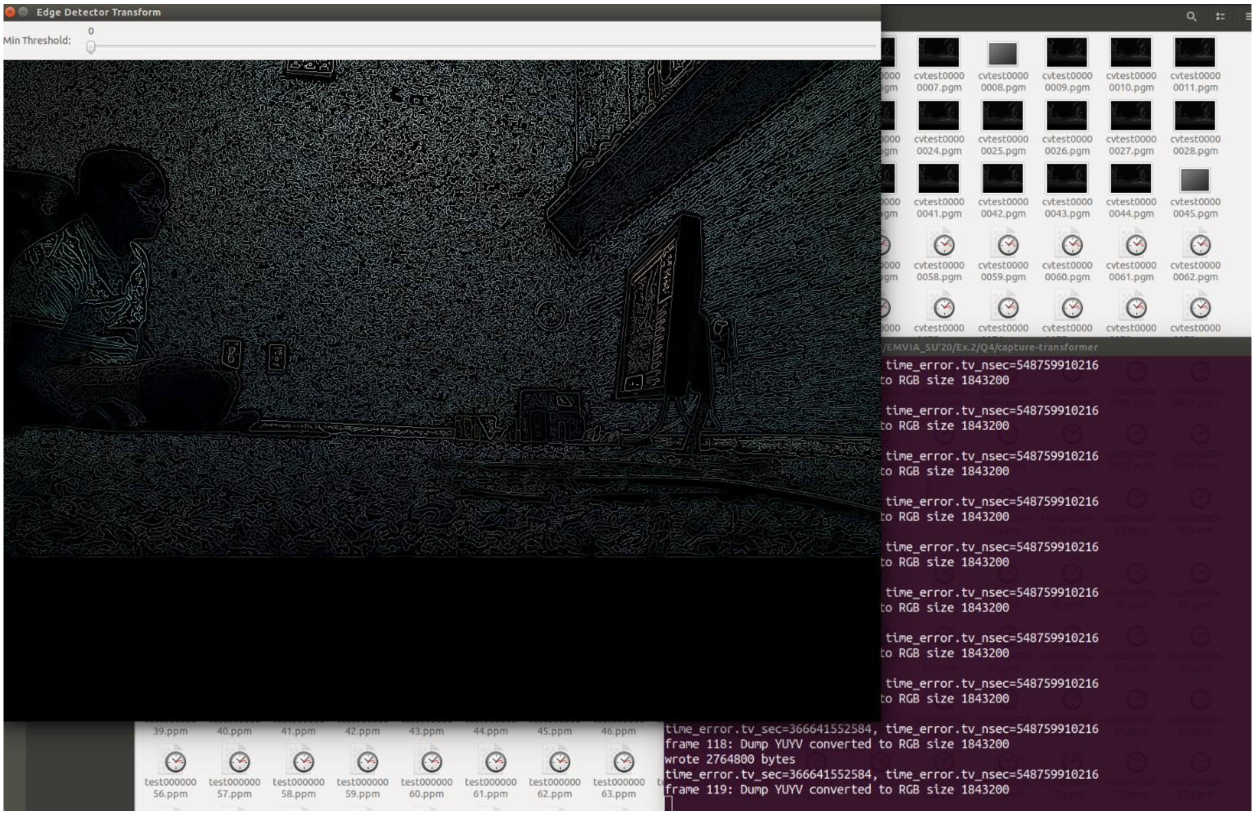
Being the Logitech C615 which has an FPS of 30, the average frame rate was found to be 27 FPS, which is quite decent. The shortage can be attributed to jitter. It could also be due to any other reasons, like latency in capturing frames from the camera itself (I/O latency), code inefficiency (possibly due to clock_gettime(), but this is highly doubtful) or utilization of CPU resources for some other task, rather than actually capturing frames for the video.

The Worst-case Jitter was found to be same as the average framerate, which is quite a stroke of luck. It implies the number of frames per second was consistent at 27, and can be put down to efficiency of the camera’s bus connection to the processor.

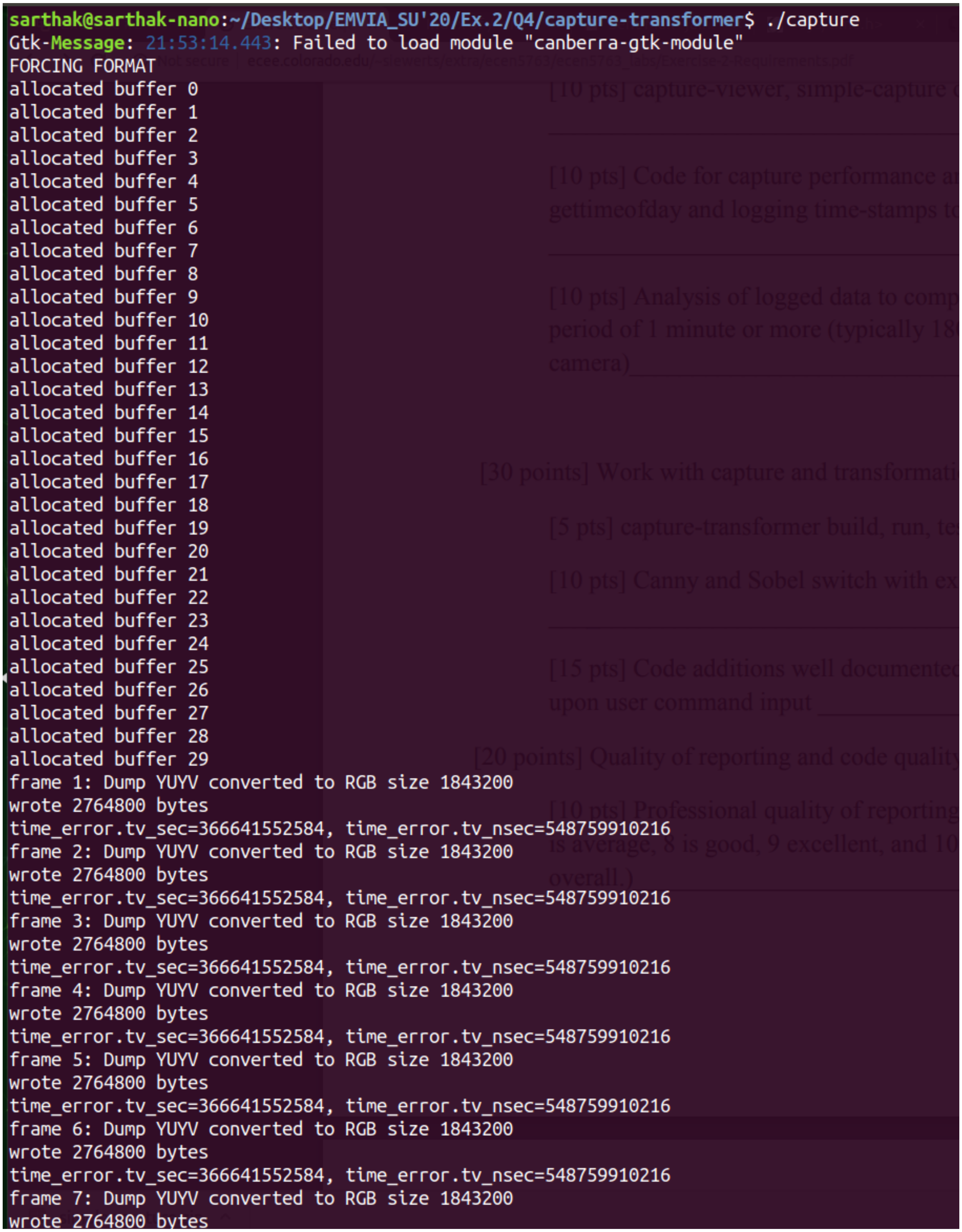
Jitter over a time period of 61 seconds was found to be 156, giving an average of 2.5 frames lost per second. This could be put down to other tasks the kernel is executing at the same time, or code paths the processor takes while executing said code. Giving an exact specification for this jitter would be rather improbable.

Question 4

Build, run and test of capture-transformer code

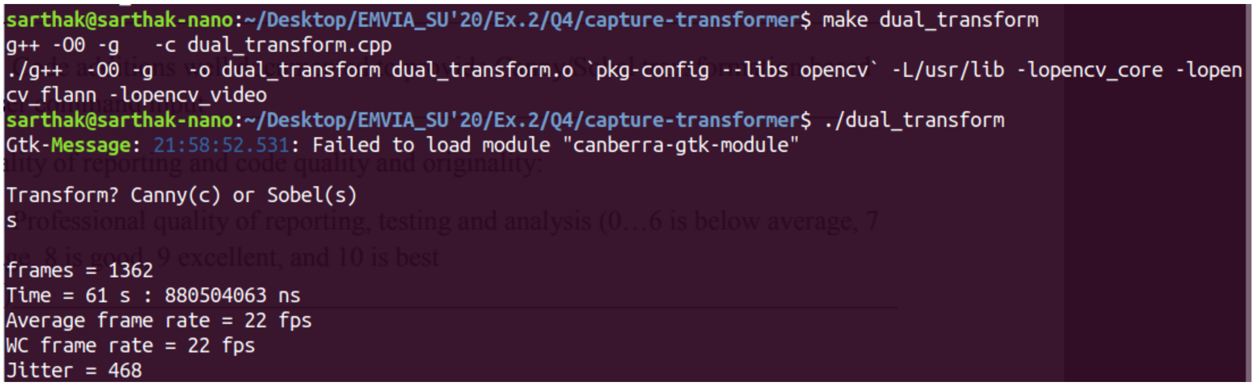


```
 time_error.tv_nsec=548759910216
to RGB size 1843200

 time_error.tv_nsec=548759910216
to RGB size 1843200

 time_error.tv_nsec=548759910216
to RGB size 1843200

 time_error.tv_nsec=548759910216
to RGB size 1843200

 time_error.tv_nsec=548759910216
to RGB size 1843200

 time_error.tv_nsec=548759910216
to RGB size 1843200

 time_error.tv_nsec=548759910216
to RGB size 1843200

 time_error.tv_nsec=548759910216
to RGB size 1843200
time_error.tv_sec=366641552584, time_error.tv_nsec=548759910216
frame 118: Dump YUYV converted to RGB size 1843200
wrote 2764800 bytes
time_error.tv_sec=366641552584, time_error.tv_nsec=548759910216
frame 119: Dump YUYV converted to RGB size 1843200
```

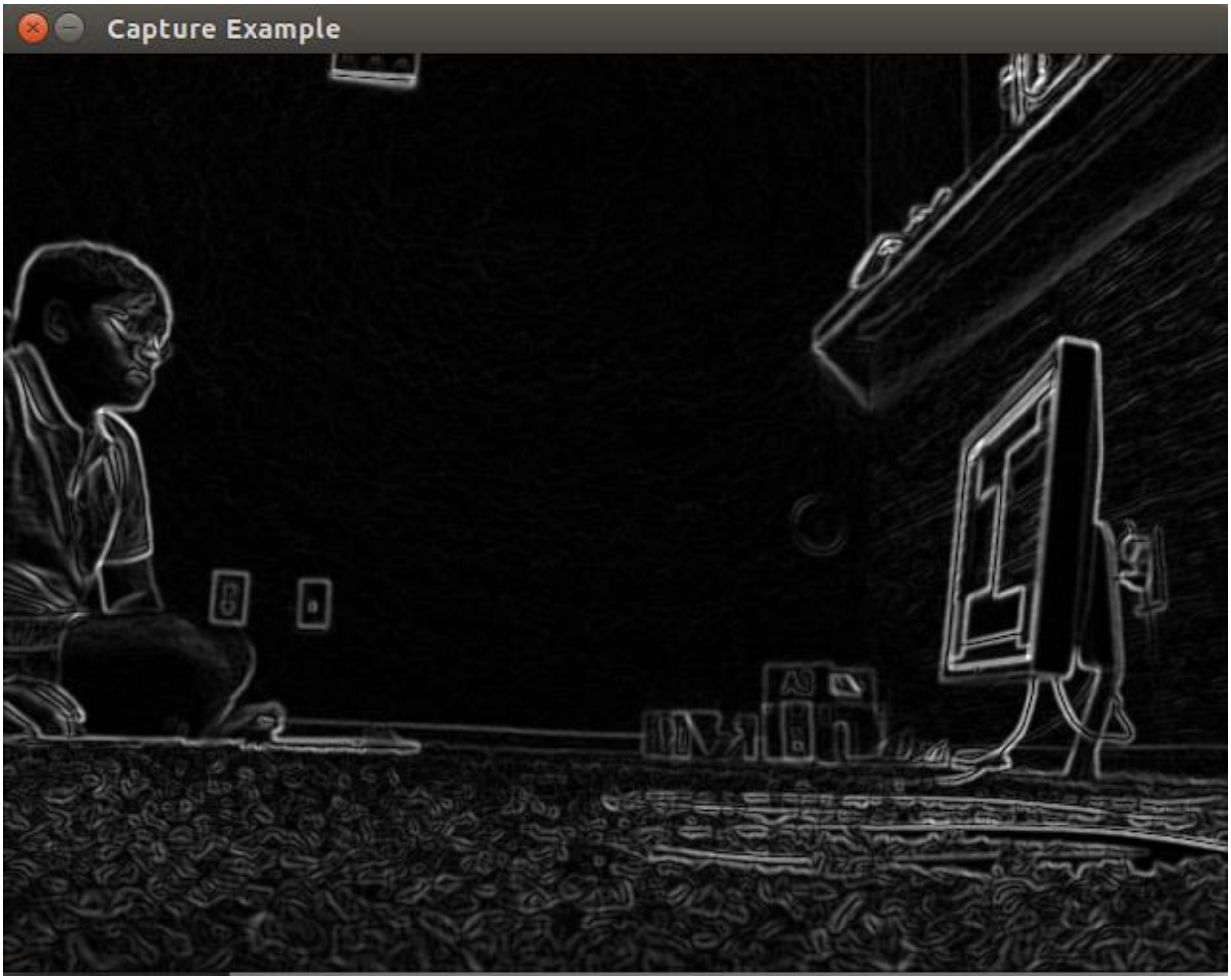


```
sarthak@sarthak-nano:~/Desktop/EMVIA_SU'20/Ex.2/Q4/capture-transformer$ ./capture
Gtk-Message: 21:53:14.443: Failed to load module "canberra-gtk-module"
FORCING FORMAT
allocated buffer 0
allocated buffer 1
allocated buffer 2
allocated buffer 3
allocated buffer 4
allocated buffer 5
allocated buffer 6
allocated buffer 7
allocated buffer 8
allocated buffer 9
allocated buffer 10
allocated buffer 11
allocated buffer 12
allocated buffer 13
allocated buffer 14
allocated buffer 15
allocated buffer 16
allocated buffer 17
allocated buffer 18
allocated buffer 19
allocated buffer 20
allocated buffer 21
allocated buffer 22
allocated buffer 23
allocated buffer 24
allocated buffer 25
allocated buffer 26
allocated buffer 27
allocated buffer 28
allocated buffer 29
frame 1: Dump YUYV converted to RGB size 1843200
wrote 2764800 bytes
time_error.tv_sec=366641552584, time_error.tv_nsec=548759910216
frame 2: Dump YUYV converted to RGB size 1843200
wrote 2764800 bytes
time_error.tv_sec=366641552584, time_error.tv_nsec=548759910216
frame 3: Dump YUYV converted to RGB size 1843200
wrote 2764800 bytes
time_error.tv_sec=366641552584, time_error.tv_nsec=548759910216
frame 4: Dump YUYV converted to RGB size 1843200
wrote 2764800 bytes
time_error.tv_sec=366641552584, time_error.tv_nsec=548759910216
frame 5: Dump YUYV converted to RGB size 1843200
wrote 2764800 bytes
time_error.tv_sec=366641552584, time_error.tv_nsec=548759910216
frame 6: Dump YUYV converted to RGB size 1843200
wrote 2764800 bytes
time_error.tv_sec=366641552584, time_error.tv_nsec=548759910216
frame 7: Dump YUYV converted to RGB size 1843200
wrote 2764800 bytes
```

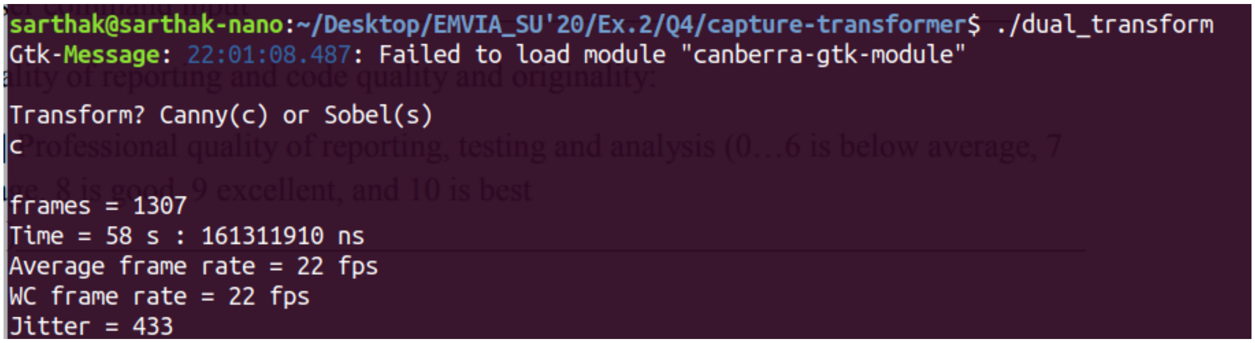
Switch to Sobel transform with option ‘s’ provided by user



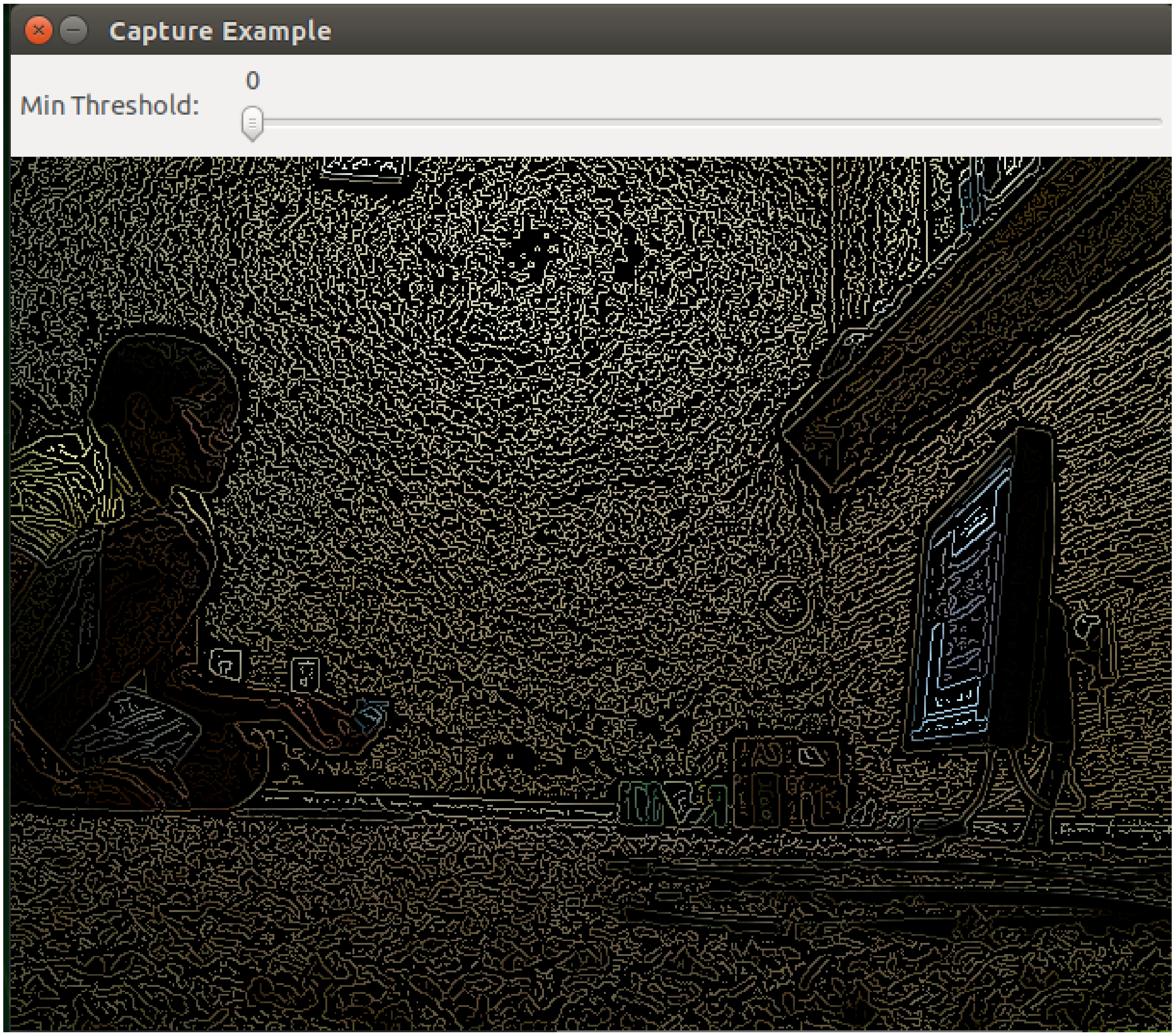
```
sarthak@sarthak-nano:~/Desktop/EMVIA_SU'20/Ex.2/Q4/capture-transformer$ make dual_transform
g++ -O0 -g   -c dual_transform.cpp
./g++  -O0 -g   -o dual_transform dual_transform.o `pkg-config --libs opencv` -L/usr/lib -lopencv_core -lopen
cv_flann -lopencv_video
sarthak@sarthak-nano:~/Desktop/EMVIA_SU'20/Ex.2/Q4/capture-transformer$ ./dual_transform
Gtk-Message: 21:58:52.531: Failed to load module "canberra-gtk-module"

Transform? Canny(c) or Sobel(s)
s

frames = 1362
Time = 61 s : 880504063 ns
Average frame rate = 22 fps
WC frame rate = 22 fps
Jitter = 468
```



Switch to Canny transform with option ‘c’ provided by user



```
sarthak@sarthak-nano:~/Desktop/EMVIA_SU'20/Ex.2/Q4/capture-transformer$ ./dual_transform
Gtk-Message: 22:01:08.487: Failed to load module "canberra-gtk-module"

Transform? Canny(c) or Sobel(s)
c


frames = 1307
Time = 58 s : 161311910 ns
Average frame rate = 22 fps
WC frame rate = 22 fps
Jitter = 433
```



Min Threshold: 0

Code Description:

A set of if-else conditional statements is used to provide the path for code to execute based upon user input, whether 's', or 'c', else a backup of standard video capture. The code from Q.3 is modified to add the Sobel and Canny transform operations. For Sobel, the 'GaussianBlur' API is used as a high-pass filter, which blurs out an image using the Gaussian filter. Subsequent APIs are used to convert the image from color to grayscale. The 'Sobel' API is used to find derivative of each frame to a depth of 2-byte integer values. 'convertScaleAbs' scales the x and y gradients, calculates absolute values, and used the same to find the total weighted values by finding square root of sum of squares of each. 'imshow' is used to display each image as it is being operated on.

For Canny Transform, 'CannyThreshold' API is the logical brain. The 'Canny' API is used to detect the canny edges in the frame basis the low threshold, ratio and kernel matrix size values.

Analysis of results:

The average frame rate comes out to be lower than that for the previous code from Q.3. This is to be expected, as the OpenCV operations of blurring, differentiating and applying the respective Sobel and Canny APIs would naturally cause some delay in querying frames from the camera. The larger code size of the program also means more lines of code for the compiler to traverse through, adding to the overhead. The large jitter value is also reasonably justified for, since the average rate itself is so low.

The jitter value of Sobel is more than that of Canny by about 35 frames, averaging to 1 frame per second. This leads to the conclusion that the operations required for Sobel transform are slightly more CPU-intensive than those required for Canny.

References:

1. Reading and Writing Images and Video - https://docs.opencv.org/2.4/modules/highgui/doc/reading_and_writing_images_and_video.html