University of Colorado Boulder

EXERCISE 3

BY
Amreeta Sengupta

06/28/2019
Question 2,3,4,5,6 have been performed using Jetson Nano

QUESTION 1

- Interactive media systems can be defined as system which incorporate audio, video and rendered graphics to form an application that also has the ability to communicate with the user, thus, providing the user the control of the various digital services. Some tools and method which can be employed while dealing with interactive media system include:

  - Mobile-to-cloud operating system such as Android include Linux servers and mobile Linux devices which enable interaction for collection of user data, analysis of this data in real time and feeding this data back to the devices for improved services. These systems basically employ an application SDK that make the development, growth and distribution of the interactive media services extremely easy.
  - Digital video and audio encoding are important fields while dealing with interactive media systems which includes transportation of streams used for standards like H.264/H.265 for MPEG.
  - Graphical rendering and digital video frame annotation which encompasses complex photo-realistic-ray-trace rendering, simple annotation of video, polygon-based graphics for game engines etc. which allows modification of reality and enhancement of the digital video through the overlay of information.
  - Advanced user interaction and sensing includes gesture recognition and facial expression analysis for an increased user-engaging application.

- Interactive media has a wide array of skills which it exploits for encoding, transport, decoding of data, graphical rendering etc., examples of which are provided below:

  - Camera Capture and Frame Transformation in Linux employs OpenCV which is an API developed by Intel consisting of library of reusable algorithms for performing common image processing and image transformation functions such as annotation of frames, edge transformation etc.
  - Frame transformation for compression to encode and transport to devices can be achieved using MPEG4 which offers higher compression ratios and improved quality of digital video. MPEG I-frame compressions includes:
    - Subsampling color (red and blue) compared with green
    - Division of each frame into 8x8 macroblocks
    - DC transform of each macroblock
    - Scaling

. Weighting and truncation of the DCT macroblock
. Zig-zag lossless compression of each macroblock

These frames are in turn used in a set of images to compress frames based on change in the pixel intensities.

- ❖ Frame transformation for computer vision is usually used to extract key features of a frame for improved understanding of data. This can refer to detection of edges in an image, recognition and tracking of objects etc.
- ❖ Integration of 3D rendering animation with digital video can be done using RenderMan which is a software for description of various scenes for a photorealistic image. It describes the lighting, perspective for viewers, colors, textures etc. for the 3D scene rendered on the 2D screen.
- ❖ Graphical frame annotation allows easy correlation of essential information of a particular scene for improved user understanding. For example, it can be used to in AR goggles for viewing a specific part information while repairing a vehicle.

- Interactive media systems allow us to dream of a future with advanced personalized digital services which provides the user with a more informed view. It enables two-way communication and analytics based on user interest. It can facilitate the user with the control of audio and video services and can also help in creation of augmented reality.

Overall, Interactive Media services is a step towards an advanced and personalized provision of digital services catering to the different requirement and interest of various users. It basically refers to the various audio and video services that engages with the users and gives responses in line with the user specifications. Its application is widespread which includes video games, social media, augmented reality for providing the user with a more informed view. This technology promotes innovation of new system devices suitable for the interactive-media engulfed lifestyle.

Graphical Rendering is one of the integral parts of interactive media systems. One of the methods employed for the same is Ray Tracing which traces the path of light as pixels for generation of an image. It is a widely employed by movie and theatre makers. However, this was perhaps a concern for the computer vision researchers due to its intrusive nature if misused. With todays ever growing technology, anyone can easily use graphical rendering to mislead people and spread incorrect information which can be a major threat to the integrity and security of the original data. Hence, if used wisely, it can create a more stimulating and interactive experience for the user. However, in wrong hands, it can lead to disastrous consequences.

University of Colorado Boulder

QUESTION 2

Median Filter is basically used to de-noise an image. This filter replaces each pixel of an image with a pixel with the value of the median of all the neighboring pixels. For high noise levels, median filter is used. Its disadvantage is that it lacks the capability to distinguish noise from finer detail.

Build and Run

```
amreeta@desktop:~/Downloads/EX3/Q2$ make
g++ -O0 -g   -c q2.cpp
g++  -O0 -g   -o q2 q2.o `pkg-config --libs opencv` -L/usr/lib -lopencv_core -lopencv_flann -lopencv_video
amreeta@desktop:~/Downloads/EX3/Q2$ ./q2
Gtk-Message: 19:55:47.527: Failed to load module "canberra-gtk-module"
```

Extraction of first frame from the video using ffmpeg

```
amreeta@desktop:~/Downloads/EX3/Q2$ ffmpeg -i Dark-Room-Laser-Spot.mpeg -vframes 1 frame%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-libfreetype --enable-libfribidi --enable-libgme --enable-libgsm --enable-libmp3lame --enable-libmysofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus --enable-libpulse --enable-librubberband --enable-librsvg --enable-libshine --enable-libsnappy --enable-libsoxr --enable-libspeex --enable-libssh --enable-libtheora --enable-libtwolame --enable-libvorbis --enable-libvpx --enable-libwavpack --enable-libwebp --enable-libx265 --enable-libxml2 --enable-libxvid --enable-libzmq --enable-libzvbi --enable-omx --enable-openal --enable-opengl --enable-sdl2 --enable-libdc1394 --enable-libdrm --enable-libiec61883 --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, mpeg, from 'Dark-Room-Laser-Spot.mpeg':
  Duration: 00:00:42.14, start: 0.500000, bitrate: 1431 kb/s
    Stream #0:0[0x1e0]: Video: mpeg1video, yuv420p(tv), 1920x1080 [SAR 1:1 DAR 16:9], 104857 kb/s, 29.97 fps, 29.97 tbr, 90k tbn, 29.97 tbc
    Stream #0:1[0x1c0]: Audio: mp2, 44100 Hz, stereo, s16p, 224 kb/s
Stream mapping:
  Stream #0:0 -> #0:0 (mpeg1video (native) -> ppm (native))
Press [q] to stop, [?] for help
[swscaler @ 0x557ab4e9e0] No accelerated colorspace conversion found from yuv420p to rgb24.
Output #0, image2, to 'frame%d.ppm':
  Metadata:
    encoder         : Lavf57.83.100
    Stream #0:0: Video: ppm, rgb24, 1920x1080 [SAR 1:1 DAR 16:9], q=2-31, 200 kb/s, 29.97 fps, 29.97 tbn, 29.97 tbc
    Metadata:
      encoder         : Lavc57.107.100 ppm
frame=    1 fps=0.0 q=-0.0 Lsize=N/A time=00:00:00.03 bitrate=N/A speed=0.326x
video:6075kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: unknown
amreeta@desktop:~/Downloads/EX3/Q2$
```

Image before application of Median Filter

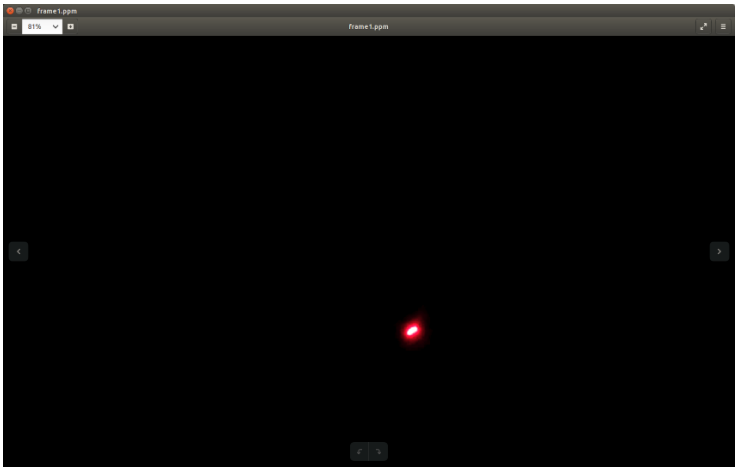
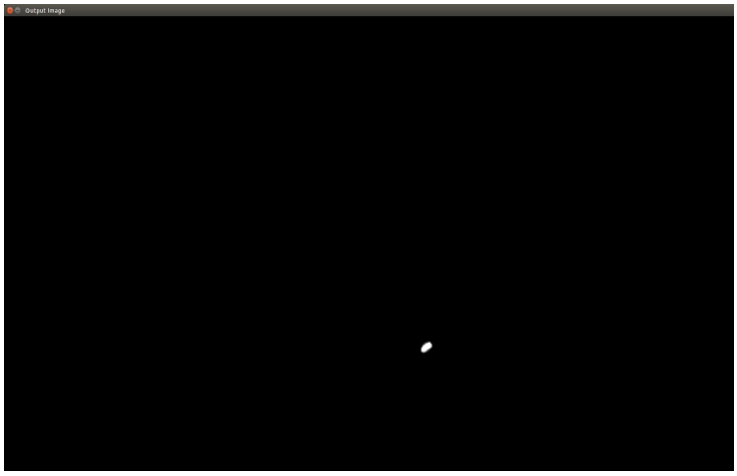


Image after application of Median Filter



Code Explanation

- The frame extracted using ffmpeg is read by using imread function.
- This frame is converted into gray map by extracting only the Green band using the extractChannel function.
- The medianBlur function is used to apply median filter on the frame using a 3X3 kernel.
- The resultant image is displayed using imshow.
- ESC key can be pressed by the user to terminate the program.

Result Description

- While the median filter helps in removing the noise in an image, it also suppresses the edge of the object as it acts as a low pass filter.
- In this case, we have a laser dot as our object of interest. However, this laser dot is not a precise dot, but instead it spread laser spot.
- The median filter helps in removing the noise and preserves the edges to some extent, but it is not very useful in edge enhancement.

QUESTION 3

Build and Run

```
amreeta@desktop:~/Downloads/EX3/Q3$ make
g++ -O0 -g   -c q3.cpp
g++  -O0 -g   -o q3 q3.o `pkg-config --libs opencv` -L/usr/lib -lopencv_core -l
opencv_flann -lopencv_video
amreeta@desktop:~/Downloads/EX3/Q3$ ./q3
Gtk-Message: 21:20:40.656: Failed to load module "canberra-gtk-module"
amreeta@desktop:~/Downloads/EX3/Q3$
```

Code Explanation

- Each frame of the “Dark-Room-Laser-Spot-With-Clutter” is taken.
- For background elimination in order to preserve the region of interest i.e. the laser spot, frame differentiation is used.
- Each present frame is subtracted from the previous frame by using the absdiff function and the resulting frame is displayed using imshow function.
- The frames are saved in the file and ffmpeg is used to convert these frames into a video.
- ESC key can be pressed by the user to terminate the program.

Result Description

- Frame differentiation is quite helpful is removing the clutter and eliminates the background to a great extent.
- In this case though we can see the laser spot, it increases the noise around the moving laser spot.
- Hence, we see a laser spot that is quite spread while moving but the background is mostly eliminated.

Ffmpeg command to re-encode the video

```
amreeta@desktop:~/Downloads/EX3/Q3$ ffmpeg -r 30 -f image2 -i Frame_%d.pgm q3.mpeg
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-a
vresample --enable-avisynth --enable-gnutls --enable-ladspa --enable-libass --enable-libbluray --enable-libbs2b --enable-libcaca --enable-libcdio --enable-libflite --enable-libfontconfig --enable-libfreet
ype --enable-libfribidi --enable-libgme --enable-libgsm --enable-libmp3lame --enable-libmysofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus --enable-libpulse --enable-librubberband --enable-l
ibrsvg --enable-libshine --enable-libsnappy --enable-libsoxr --enable-libspeex --enable-libssh --enable-libtheora --enable-libtwolame --enable-libvorbis --enable-libvpx --enable-libwavpack --enable-libweb
p --enable-libx265 --enable-libxml2 --enable-libxvid --enable-libzmq --enable-libzvbi --enable-omx --enable-openal --enable-opengl --enable-sdl2 --enable-libdc1394 --enable-libdrm --enable-libiec61883 --e
nable-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, image2, from 'Frame_%d.pgm':
  Duration: 00:00:57.97, start: 0.000000, bitrate: N/A
    Stream #0:0: Video: pgm, rgb24, 1920x1080, 30 tbr, 30 tbn, 30 tbc
```

QUESTION 4

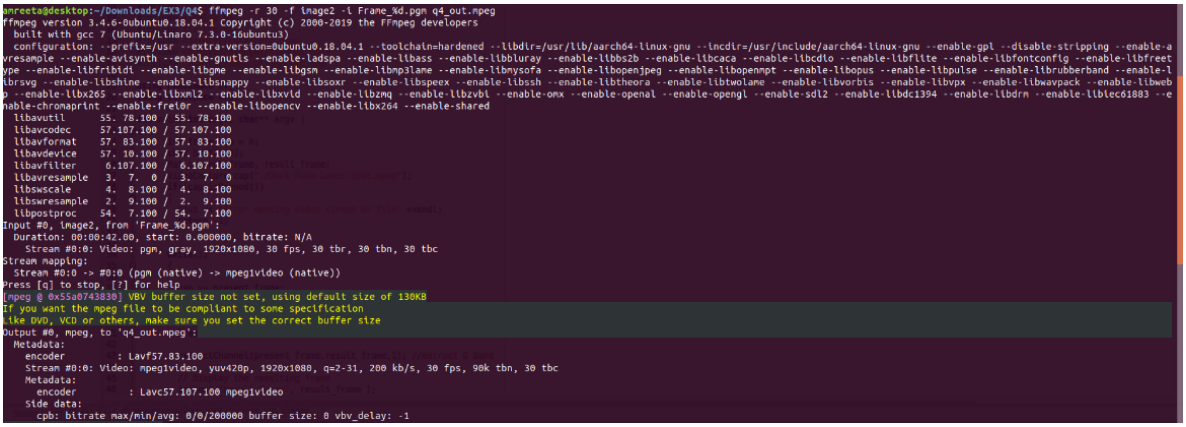
Build and Run

```
amreeta@desktop:~/Downloads/EX3/Q4$ make
g++ -O0 -g   -c q4.cpp
g++  -O0 -g   -o q4 q4.o `pkg-config --libs opencv` -L/usr/lib -lopencv_core -lopencv_flann -lopencv_video
amreeta@desktop:~/Downloads/EX3/Q4$ ./q4
Gtk-Message: 21:06:23.238: Failed to load module "canberra-gtk-module"
amreeta@desktop:~/Downloads/EX3/Q4$
```

Code Explanation

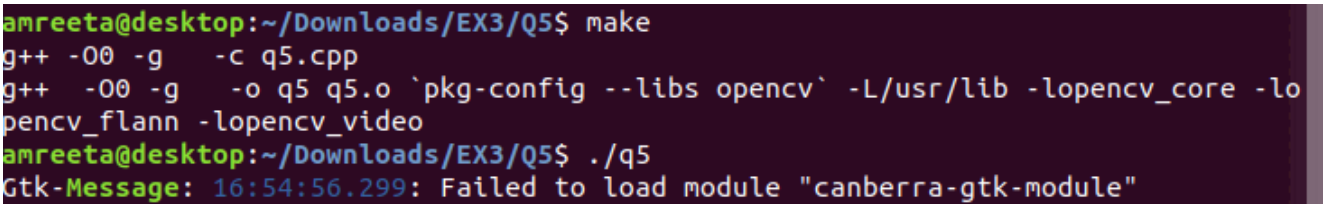
- Each frame of the “Dark-Room-Laser-Spot” is taken.
- These frames are converted into gray map by extracting only the Green band using the extractChannel function.
- The resultant image is displayed using imshow.
- The frames are saved in the file and ffmpeg is used to convert these frames into a video.
- ESC key can be pressed by the user to terminate the program.

Ffmpeg command to re-encode the video



QUESTION 5

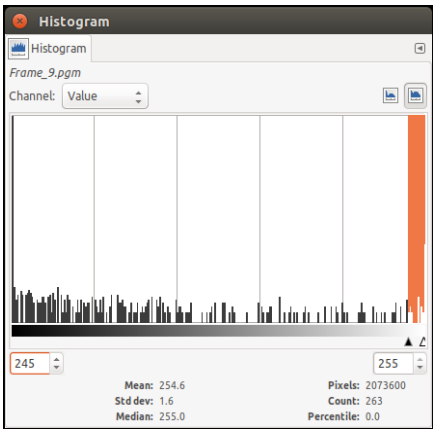
Build and Run



Code Explanation

- Each frame of the “Dark-Room-Laser-Spot.mpeg” is taken and converted into gray map frames by extracting only the Green band.
- For accurate laser spot detection, a threshold function is used to extract the foreground.
- In order to track the laser spot, first the contour of the foreground is determined, and a rectangle is drawn around the contour. Post this approximation to polygons is applied and a rectangle is bound around it. The contour and the rectangle are drawn on the image.
- The center of the contour is determined using the moments function and this point is given as an argument to the drawMarker function to draw a crosshair at the center point.
- Each frame is displayed using the imshow() function and the images are saved and converted to a video.
- ESC key can be pressed by the user to terminate the program.

Threshold value analysis

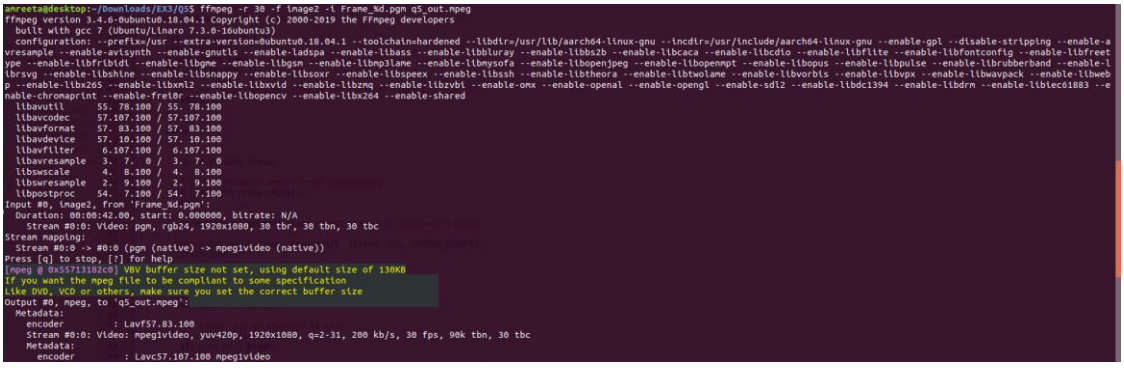


For an accurate detection of the laser beam after gray scaling the image, we need to find the correct value of white color. The white color usually ranges from 170 to 255 in the RGB range. Hence, we perform histogram analysis in this range to check the count of pixel values in this range. Using GIMP, I found the following reading:

Bin Range	Pixel Count
170-255	356
200-255	326
220-255	296
230-255	287
240-255	272
245-255	263
250-255	255
255	243

From the above table, we want to extract the pixels for the most accurate extraction of the laser beam while not taking the blurry pixels surrounding it into account. Hence, I have chosen to take 245-255 bin range as it should cover the region of interest without taking into account the additional pixels.

Ffmpeg command to re-encode the video



QUESTION 6

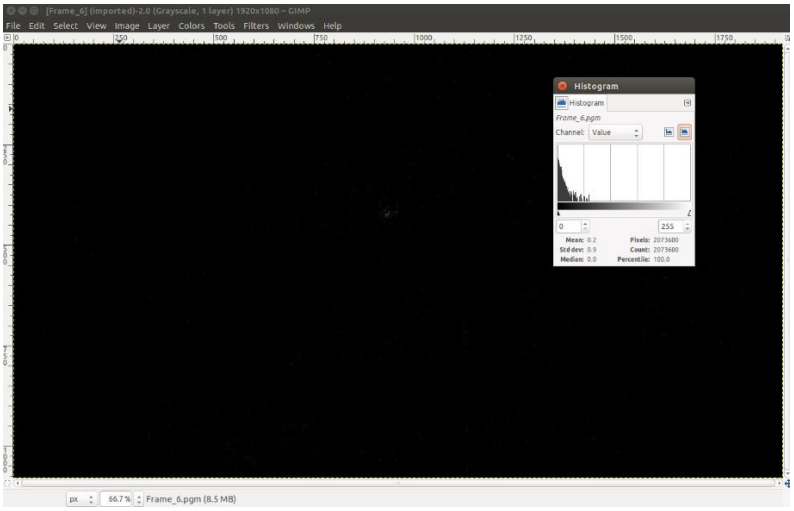
Build and Run

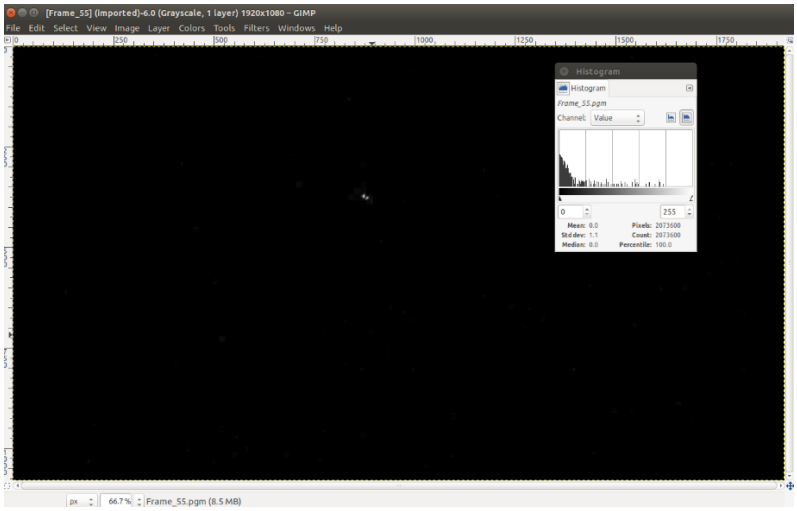
```
amreeta@desktop:~/Downloads/EX3/Q6$ make
g++ -O0 -g   -c q6.cpp
g++  -O0 -g   -o q6 q6.o `pkg-config --libs opencv` -L/usr/lib -lopencv_core -lo
pencv_flann -lopencv_video
amreeta@desktop:~/Downloads/EX3/Q6$ ./q6
Gtk-Message: 22:43:59.916: Failed to load module "canberra-gtk-module"
```

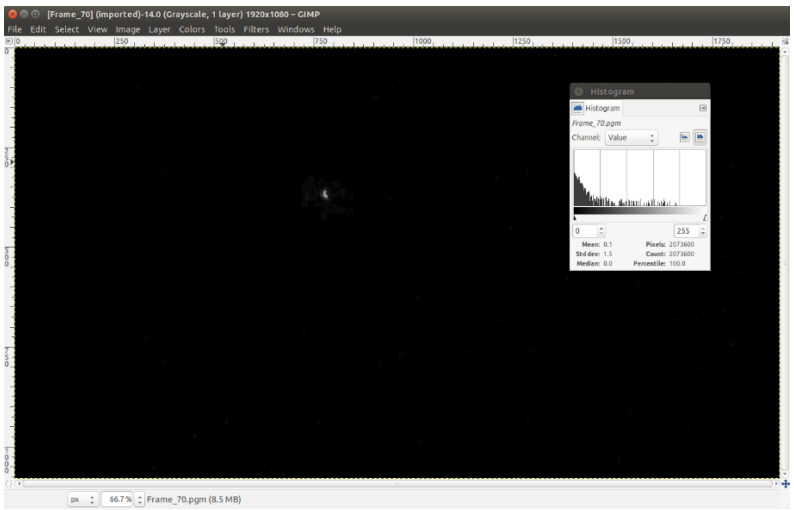
Code Explanation

- Each frame of the "Light-Room-Laser-Spot-with-Clutter.mpeg" is taken and converted into gray map frames by extracting only the Green band.
- For background elimination in order to preserve the region of interest i.e. the laser spot, frame differentiation is used.
- Each present frame is subtracted from the previous frame by using the absdiff function after the frames are converted to their grey scale value.
- To further remove noise, median filter is used.
- For accurate laser spot detection, a threshold function is used to extract the foreground.
- The countNonZero function is used to count the black pixels in the frame and if the entire frame has black pixels, the threshold value is lowered and computed again in order to detect the laser spot.
- In order to track the laser spot, first the contour of the foreground is determined, and a rectangle is drawn around the contour. Post this approximation to polygons is applied and a rectangle is bound around it. The contour and the rectangle are drawn on the image.
- The center of the contour is determined using the moments function and this point is given as an argument to the drawMarker function to draw a crosshair at the center point.
- Since the laser spot is changing rapidly and it is blurry, multiple contours get detected and hence the contour area is calculated, and the maximum value is drawn on the image.
- Each frame is displayed using the imshow() function and the images are saved and converted to a video.
- ESC key can be pressed by the user to terminate the program.

Threshold Value Analysis







- The frames sown above are few of the frames of the video after background elimination and grey scaling. As seen above the, the laser spot (which has spread on the background wall) has a huge variation for different frames and hence a generalized thresholding function will not be capable of tracking the laser spot accurately.
- And hence, in the code, adaptive thresholding is applied which sets different threshold value for different frames based on the requirement of the frame.
- This is done by checking whether the entire image only has black pixels. The countNonZero function is used to count the black pixels in the frame and if the entire frame has black pixels, the threshold value is lowered and computed again in order to detect the laser spot catering to the different frames.
- Thus, this will help in a much better tracking of the laser spot.

Ffmpeg command to re-encode the video

```
amreeta@desktop:~/Downloads/EX3/Q6$ ffmpeg -r 30 -f image2 -i Frame_%d.pgm q6_out.mpeg
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/aa
rch64-linux-gnu --enable-gpl --disable-stripping --enable-avresample --enable-avisynth --enable-gnutls --enable-ladspa --enable-libass --enable-l
ibbluray --enable-libbs2b --enable-libcaca --enable-libcdio --enable-libflite --enable-libfontconfig --enable-libfreetype --enable-libfribidi --e
nable-libgme --enable-libgsm --enable-libmp3lame --enable-libmysofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus --enable-libpulse -
-enable-librubberband --enable-librsvg --enable-libshine --enable-libsnappy --enable-libsoxr --enable-libspeex --enable-libssh --enable-libtheora
 --enable-libtwolame --enable-libvorbis --enable-libvpx --enable-libwavpack --enable-libwebp --enable-libx265 --enable-libxml2 --enable-libxvid -
-enable-libzmq --enable-libzvbi --enable-omx --enable-openal --enable-opengl --enable-sdl2 --enable-libdc1394 --enable-libdrm --enable-libiec6188
3 --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, image2, from 'Frame_%d.pgm':
  Duration: 00:00:57.97, start: 0.000000, bitrate: N/A
    Stream #0:0: Video: pgm, rgb24, 1920x1080, 30 tbr, 30 tbn, 30 tbc
Stream mapping:
  Stream #0:0 -> #0:0 (pgm (native) -> mpeg1video (native))
Press [q] to stop, [?] for help
[mpeg @ 0x556db702c0] VBV buffer size not set, using default size of 130KB
If you want the mpeg file to be compliant to some specification
Like DVD, VCD or others, make sure you set the correct buffer size
Output #0, mpeg, to 'q6_out.mpeg':
  Metadata:
    encoder         : Lavf57.83.100
    Stream #0:0: Video: mpeg1video, yuv420p, 1920x1080, q=2-31, 200 kb/s, 30 fps, 90k tbn, 30 tbc
    Metadata:
      encoder         : Lavc57.107.100 mpeg1video
    Side data:
      cpb: bitrate max/min/avg: 0/0/200000 buffer size: 0 vbv_delay: -1
```

REFERENCES

- https://www.ibm.com/developerworks/library/bd-interactive/index.html
- https://en.wikipedia.org/wiki/Ray_tracing_(graphics)
- http://www.cse.uaa.alaska.edu/~ssiewert/a485_code/
- https://docs.opencv.org