Table of Contents

FSCR BUSH DIGITAL SPECTROMETER JUPYTER NOTEBOOK

BDS Header Block

In [ ]:
```
# BUSH DIGITAL SPECTROMETER SOFTWARE INTERACTIVE VERSION
# Author - Chandru Narayan
# TEMPLATE FOR FCSR STUDENTS
# CN Version_12i  11/25/2019 cloned from automated version v11
#
#    120219 CN "Added function call to print BDS parameters"
#    120419 CN "Added function call to compute peaks in the spectrum wavelen
#    120619 CN "Added cell for bdscfg parms"
#    120619 CN "Added Try-Except Block for creating Camera Objects"
#    040424 CN "Udated code for deprecated methods"
#    040424 CN "Added Cloudy Sky Spectrum Standard"
```

In [ ]:
```
#!pip install --upgrade pip
#!pip install opencv-python-headless
#!pip install peakutils
```

Importing Libraries and Notebooks

In [ ]:
```
import import_ipynb
from IPython.core.display import Image
from IPython.core.display import display
from IPython.display import IFrame
#import PIL
from PIL import Image as pilimg
from PIL import ImageDraw as pildraw
from PIL import ImageFont as pilfont
```

```
import os, sys
import time
from datetime import datetime
import cv2
import matplotlib.pyplot as plt

class StopExecution(Exception):
    def _render_traceback_(self):
        pass
```

In [ ]:
```
# BUSH LIBRARY FUNCTIONS FOR BUSH DIGITAL SPECTROMETER SOFTWARE INTERACTIVE
# Author - Chandru Narayan
# TEMPLATE FOR FCSR STUDENTS
# CN Version_11i  12/1/2019 cloned from automated version v11
# IMPORT BDSLIB AND BDSCFG HERE
import bdslibv5
```

BDS Configuration Parameters

In [ ]:
```
###
#    BUSH DIGITAL TELESCOPE SOFTWARE CONFIG SECTION
#    TO BE USED IN THE INTERACTIVE VERSION ONLY
#    FOR DETAILED DESCRIPTION OF PARMS SEE BDS CONFIG DOC
###

#
# NAMING
#
source = 'cloudy'
element = 'cloudy sky'
desc = 'cloudy sky spectrum'

#
# CAMERA
#
shutter = 1000000

#
# CALIBRATION
#
wavelength_factor = 0.69
spectrum_angle = 0
slit_topadj = 0
slit_botadj = -0

#
# PLOTS
#
samp_th = 0.05
wlen_th = 35
```

BDS File Output Names

In [ ]:
```
# First let us set the date and time and we may not have internet access
# Uncomment/Edit/RUN statements below if spectroscope is not connected to th
# !date -s '2024-04-08 12:46:30'
# !date
```

In [ ]:
```
#!date -s '2024-04-04 09:38:30'
#!date
```

In [ ]:
```
# STEP 1. SETUP FILE BASENAMES WITH TIMESTAMPS
#        setup the source or basename for files
#        make it indicative of the spectrum you are taking
#        keep it short but meaningful. Do not name "a1" etc!
#source = 'cfls'

# Filenames be appended with date and time
# such that they will not be overwritten
now = datetime.now()
name = source + now.strftime("%m%d%H%M%S")
raw_filename = name + "_raw"
ovl_filename = name + "_ovl"
cht_filename = name + "_cht"
tbl_filename = name + "_tbl"
par_filename = name + "_par"
pks_filename = name + "_pks"
```

In [ ]:
```
################## STUDENT TO ADD EDITS BELOW ################################
## WRITE A STATEMENT TO PRINT THE 4 OUTPUT NAMES FROM THE BDS SOFTWARE TO FA

print(raw_filename)
print(ovl_filename)
print(cht_filename)
print(tbl_filename)
print(par_filename)
print(pks_filename)
```

In [ ]:
```
################## STOP HERE STUDENT/INSTRUCTOR TO VALIDATE STEP 1 #########
## VALIDATE THE NAMES OF FILES TO BE CREATED - DO THEY LOOK RIGHT ??? ##

# DO NOT GO FORWARD UNTIL INSTRUCTOR VALIDATES
```

Obtaining Image of Spectrum

In [ ]:
```
# STEP 2. CREATE THE CAMERA OBJECT
#          CAPTURE THE RAW SPECTRUM IMAGE
#          THIS WILL BE EXAMINED FOR ANY ADJUSTMENTS NEEDED
#          FOR EXAMPLE IMAGE BRIGHTNESS LIGHT LEAKAGE ETC
#          DISPLAY CAPTURED IMAGE
```

In [ ]:
```
raw_jpg_filename=bdslibv5.take_image(shutter)
bdslibv5.show_image(raw_jpg_filename)
```

Displaying Processed Image of Spectrum with parameters

In [ ]:
```
#       view image and apply putty or tape inside spectroscope to prevent li
#       remember - image is flipped laterally from left right!
display(Image(raw_jpg_filename))
bdslibv5.display_bds_params(name,desc,shutter,slit_topadj,slit_botadj,spectr
```

Draw Visual Aperture and Measure Emission Spectral Peaks

In [ ]:
```
# STEP 3. PROCESS THE IMAGE AND LOCATE THE SLIT (APERTURE)
#         READ RAW JPG FILE OBTAINED IN A PIXEL ARRAY
#         RECORD THE PIXEL WIDTH AND HEIGHT
#         NARROW THE PIXEL WINDOW FOR SLIT TOP AND BOTTOM
#         FOR EXAMPLE IMAGE BRIGHTNESS LIGHT LEAKAGE ETC
#         DISPLAY CAPTURED IMAGE
```

In [ ]:
```
#         READ RAW JPG FILE OBTAINED IN A PIXEL ARRAY
im = pilimg.open(raw_jpg_filename)
pic_pixels = im.load()
#         record the pixel width and height
width = im.size[0]
height = im.size[1]
print("width is %d, height is %d" % (width, height))
#        The slit needs to be shortened in height at times due to light leak
#        inside spectrometer. This small adjustment can be made here.
#        bigger negative numbers for smaller for bottom slit
#        bigger positive numbers for smaller top slit
#        for daylight or bright spectrum we need to narrow the slit greatly.
#        default values are set above
#        Adjust and uncomment below if you need
#slit_topadj = 0
#slit_botadj = -0

#        call library function to find the aperture in the raw image (pixel
aperture = bdslibv5.find_aperture(pic_pixels, width, height, slit_topadj, sl
#        draw the aperture
draw = pildraw.Draw(im)
bdslibv5.draw_aperture(aperture, draw)
```

In [ ]:
```
#        Draw scan line using the Spectrum angle
#        This is the angle that the camera and diffration grating makes with
#        The Spectrum Angle trignometric tangent of the angle the camera and
#        with the line of sight to the entry slit. This usually does not nee
#        as it manipulates where in the observation area the spectrum falls.
#        approximate such that pixel counter can find it
#        default values are set above
#        Adjust and uncomment below if you need
#spectrum_angle = 0.01
```

```
#        draw the scan lline
bdslibv5.draw_scan_line(aperture, draw, spectrum_angle)
```

In [ ]:
```
#        The wavelength_factor is the variable used for calibrating the spec
#        the calibration spectral line matches the known standard for that e
#        The wavelength_factor is close to 0.90 for the 1000 lines/mm diffra
#        The wavelength_factor is close to 0.60 for the 500 lines/mm diffrat
#        default values are set above
#        Adjust and uncomment below if you need
#wavelength_factor = 0.9
try:
    results, max_result = bdslibv5.draw_graph(draw, pic_pixels, aperture, sp
except:
    #camera.close()
    print("Exception while creating an aperture")
    print("This run **** TERMINATED PREMATURELY **** ...")
    print("Maybe the result of misaligned light path a very dim spectrum")
    print("Adjust Light Path Alignment OR Increase Shutter parameter and try
    raise StopExecution
else:
    print("Producing graphical result");
```

In [ ]:
```
#        Display actual and ideal targets for camera exposure corrections
bdslibv5.inform_user_of_exposure(max_result)
```

In [ ]:
```
#       Create the spectrum image overlaid with aperture and scan line
ovl_jpg_filename = ovl_filename + ".jpg"
bdslibv5.save_image_with_overlay(im, ovl_jpg_filename)
```

In [ ]:
```
#       View the Overlaid image fix parameters and rerun STEP 3 ONLY from th
display(Image(ovl_jpg_filename))
bdslibv5.display_bds_params(name,desc,shutter,slit_topadj,slit_botadj,spectr
```

In [ ]:
```
################## STOP HERE STUDENT/INSTRUCTOR TO VALIDATE STEP 3 #########
## IS THE ACTUAL EXPOSURE WITHIN THE TARGET LIMITS ??
## DID A RECTANGULAR WINDOW APPEAR OVERLAID ON THE IMAGE  ENCLOSING THE SPEC
## IS THE SCAN LINE VISIBLE ??
## IS THE SCAN LINE ALIGNED WITH THE SLIT ??
## IF NOT WE HAVE TO MAKE ADJUSTMENTS BEFORE PROCEEDING
## READ INSTRUCTIONS IN VARIOUS CELLS ON THIS STEP
## MAKE CHANGES AND ASK FOR ME TO VALIDATE BEFORE PROCEEDING

# DO NOT GO FORWARD UNTIL INSTRUCTOR VALIDATES
```

Display Emission Spectrum and Compare with NIST Standard values

In [ ]:
```
cht_png_filename = cht_filename + ".png"
print(cht_png_filename)
```

In [ ]:
```
# STEP 4 FINAL STEP! NORMALIZE AND CREATE/DISPLAY SPECTRUM CHART
# MAKE ADJUSTMENTS AND RERUN FROM THE BEGINNING IF NEEDED
```

```
normalized_results = bdslibv5.normalize_results(results, max_result)
```

In [ ]:
```
#       Create the spectrum chart overlaid with the proper wavelengths
#       and color map according to frequency
cht_png_filename = cht_filename + ".png"
bdslibv5.export_diagram(cht_png_filename, normalized_results)
```

In [ ]:
```
#display(Image(cht_png_filename))
#bdslibv5.display_bds_params(name,desc,shutter,slit_topadj,slit_botadj,spect
```

In [ ]:
```
#       Print the Spectral Peaks table of wavelengths
#       for current spectral image obtained
csv_tbl_filename = tbl_filename + ".csv"
bdslibv5.export_csv(tbl_filename, normalized_results)

#       Uncomment and change these thresholds if necessary if
#       you would like to increase or decrease the number
#       of Spectral peaks found

#samp_th = 0.2
#wlen_th = 10
#       Call function to draw the Spectral Peaks which will
#       Plot the peaks and return a list of Peak Wavelengths
pks_png_filename = pks_filename + ".png"
peak_wl, t1, t2 = bdslibv5.draw_spectral_line_peaks(element,csv_tbl_filename
bdslibv5.display_bds_params(name,desc,shutter,slit_topadj,slit_botadj,spectr
par_txt_filename = par_filename + ".txt"
bdslibv5.write_bds_params(par_txt_filename,name,desc,shutter,slit_topadj,sli
```

In [ ]:
```
pattern = pilimg.open(cht_png_filename).convert('RGBA')
#txt = pilimg.new('RGBA', pattern.size, (255,255,255,0))
size = width, height = pattern.size
draw = pildraw.Draw(pattern,'RGBA')
font = pilfont.truetype('Lato-Regular.ttf', 12)
#print(size)
draw.text((0,0), desc.upper(), font=font, fill='#000')
draw.text((0,20), t1, font=font, fill='#000')
draw.text((0,40), t2, font=font, fill='#000')
#draw.text((0,100), "Hello World", (0, 0, 0, 0),font=font)
pattern.save(cht_png_filename)
```

In [ ]:
```
bdslibv5.display(Image(cht_png_filename))
#bdslibv5.display_bds_params(name,desc,shutter,slit_topadj,slit_botadj,spect
```

In [ ]:
```
# Show sky spectrum
# ref: https://en.wikipedia.org/wiki/Diffuse_sky_radiation

# cfls spectrum
# ref: https://www.bealecorner.org/best/measure/cf-spectrum/
# ref: https://commons.wikimedia.org/wiki/File:Fluorescent_lighting_spectrum

if 'sky' in desc.lower() or 'cloud' in desc.lower() :
    bss = "blue_sky_spectrum.png"
    display(Image(bss))
```

```python
    bss = "cloudysky_wiki.png"
    display(Image(bss))
elif 'cfls' in desc.lower() or 'flourescent' in desc.lower() :
    cfls1 = "cfls_standard.png"
    display(Image(cfls1))
    cfls2 = "cfls_plot.png"
    display(Image(cfls2))
    cfls3 = "cfls_table.png"
    display(Image(cfls3))
```

In [ ]:

```python
#camera.close()
```

In [ ]:

```python
################## STOP HERE STUDENT/INSTRUCTOR TO VALIDATE STEP 4 FINAL STE
## CONGRATULATIONS - YOU MADE A FANCY DIGITAL SPECTROSCOPE AND MADE YOUR FIR
##
## DID THE SPECTRAL CHART APPEAR ??
## DOES THE CHART LOOK CORRECT ??
## DOES IT MATCH WITH THE STANDARD FOR ELEMENTS FOUND IN THE STANDARD SPECTR
## IF NOT WE WILL MAKE ADJUSTMENTS TO PARAMETERS ABOVE AS DOCUMENTED
## MAKE CHANGES AND ASK FOR ME TO VALIDATE BEFORE PROCEEDING

# DO NOT GO FORWARD UNTIL INSTRUCTOR VALIDATES
# WHEN YOU HAVE GOOD RESULTS PRINT FROM THE "FILE->PRINT PREVIEW" FROM
# THE JUPYTER NOTEBOOK AND GET THIS NOTEBOOK PRINTED FOR VALIDATION!
```

In [ ]:

```python

```

In [ ]:

```python

```