(1) ファイル一覧

以下、python Spyder で実行することを念頭に説明します。



ダウンロードした pyE5A_Gaf フォルダーを、 Spyder で プロジェクト: プロジェクトを開く

- 1 e5_Gaf.py : 解析 __MAIN__
- 解析結果出力 パワポ (.ppt) と Excel (.xlsx)を出力します。
- ③ Gafスキャン用紙.pptx

照射した Gaf を貼り付けるシート ④ Gaf 解析 subroutines ⑤ Gaf スキャン済データ 定義

⑥を解析する際のパラメータ

6 Gaf スキャン済データ

ビットマップ (.bmp) で保存が必要

【実行】

- (1) ①を実行すると、
- (2) ⑤に従って、
- (3) ⑥が読み込まれ、
- (4) ②が生成されます。

注) Spyder IDE 5.2.2 上で開発しました。 標準モジュール以外で必要なモノは下記です。

import: python-pptx package (used v0.6.18)
https://python-pptx.readthedocs.io/en/latest/

> pip install python-pptx

このモジュールは、conda install ではダメみたいです。

(2) 準備:スキャナ読取り

- (1)読取りシート③に、感光させたGaf Filmを貼る。
 (#1~#6の順で、6枚まで貼れます。6枚以下でもOK)
 (2) PC用スキャナで、75dpi ()で読取り。
- (3) ビットマップ形式(.bmp) で保存6。(ファイル名は任意)



(3) 解析結果の出力(概要)

②a _scan.ppt	2b _scanShift.ppt	②c _anaOD.ppt	②d _anaOD.xlsx
1	1		A B C D E F G H I J K L M N O P
			$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$
2 manufacture to the second se	2 Determinant for the INP left		18 10 37.4 0<
			IZ AA AB AC AD AE AF AG AH AI AJ AK AL AM 85 0 0 0 0 0225 0225 0225 0225 0226 0228 0228 0228 0228 0228 0228 0228 0228 0228 0228 0228 0228 0225 0228 0228 0225 0228 0225
4		0.0. 迴舟桁柘朱	■ D222 0225 0226 0226 0226 0226 0226 0226 0
スキャナ画像の読込み結果 そのまま	同 ②a で、 画像分割具合の調整用		

そのまま

(4) e5_Gaf.py : __MAIN_ の説明



①【修正箇所】

rDB_Gaf[](後述)の何番目を解析するかの指定

- ② rDB_Gaf[]から、解析パラメータを取得
- ③a,b スキャン済イメージファイルのフォルダー指定
 左図の場合、同 pyE5A_Gaf フォルダー中の
 ./_GafData/ にある .bmp ファイル
- ④ 解析実行、レポートを出力 _scan.ppt, _scanShift.ppt, _anaOD.pptを出力
- ⑤【必要なら】スキャンイメージを出力
 _scanShift.ppt 中の各種イメージを、.png.csv で出力
- ⑥ O.D. 解析結果を Excelで出力 _anaOD.xlsx を出力



(5) e5DB_Gaf.py の説明



pyE5A:e5_Gaf.py



スキャン済.bmp ファイルを読み込み、6枚の画像を切り出し、 RGB分離後、O.D.値に変換します。



[pg.1] .bmp を読み込んだ 1頁全体の画像

[0,1,…,-1]], 6枚の解析type の指定方法

上図の場合、Gaf Film の感光条件は、 #1:Wobbler磁石=OFF → 0指定 #2~#5:Wobbler=ON → 1指定 #6:Film無し → -1指定 でしたので、 rDB_Gaf []のパラメータ指定は、 [0,1,1,1,1,-1]], となります。





- ← 白丸枠 が crop 中心を示します。 用紙の貼り方や、スキャナの癖で、中心がズレます
- ← crop 画像を R,G,B 分離した画像
 - R,G,B 成分ごとに 256階調で表示。 黒濃さが階調です。 背面色 R,G,B は、識別のため。意味ありません。

[pg.3] R,G,B画像 に「円形マスク」をした画像

白丸枠の大きさでマスクをかけて、 以降の解析を行っています。 この枠の外には、サインペンなどでメモ書き してあっても結構です。

(参考文献) Optical Density: O.D.値について
 上松、花屋、小嶋(JAEA 高崎研)
 「GAFフイルム線量計とイメージスキャナを
 組み合わせた、簡便なイオンビームの
 2次元線量相対分布計測システム」
 RADIOISOTOPES, 57 87-98 (2008)
 吸光度 O.D.=Log10 (1/T)
 =Log10 (255/RGB 値)

[[]pg.4] マスク画像 を O.D.値に変換した画像



[pg.1~4] の構成は、_scan.ppt と同じです。
 6枚画像の切り出し具合の調整用です。

[pg.2] page画像を 6枚に crop した結果 を見て下さい。

[(dx,dy),(dx,dy),…], 6枚の Crop Shift指定方法

Separated images & those RGB-band



 ← 6枚全てにおいて (dx=0, dy=0) にした場合。
 [(0,0), (0,0), (0,0), (0,0), (0,0), (0,0)],

白丸枠 に合うように crop 位置を調整する必要があります。

上図に、「dx dy値の方向」を示しました。 白丸枠内の目盛と dx,dy の整数値は、必ずしも一致しませんので、 [(dx,dy),(dx,dy),…],の値を何度か変化させながら "e5_Gaf.py"を実行して、 最適値を捜して下さい。



← [(2,3), (-2,4), (2,6), (-2,6), (1,10), (0,0)],
 にした場合。
 値は整数値で指定です。

今回のスキャンは、この Crop Shift 値 で、ほぼ正しく白丸枠に収まりました。 これで [pg.3~4] の画像中心も正しく設定されました。

(8) _anaOD.ppt の説明



odG[0]_0000Test1_repGaf3_ana0D.ppt 0000Test 0000Test1 40 - ban-v 40 BG 30 30 20 20 10 10 -(mm) 0 -10 -10 -20 -20 R2) -30 -30 -40 -40 -40 -30 -20 -10 Ó 10 20 30 0.6 0.4 ban-x 0.6 (m,sg,a,R2)= Opt.Dens. 2.19,8.29,0.46,0.99 BG =0.23 max=0.69 0.5 center(x,y)=(-2.2,+1.3)mm avr(x,y)=(0.53,0.53)od 0.4 GausFit ban-x, y: m, sg, a, R2= 0.3 (-2. 19, 8. 29, 0. 46, 0. 996) (+1.35, 8.01, 0.46, 0.996) -40 -30 -20 -10 ò 10 20 30 40 X[mm]

[pg.2] Film#1 解析結果(Gaussian Fit plot)

Wobbler=OFF 測定の場合、

O.D. 値分布の Gaussian Fit 解析も行います。

[左図] band-x, band-y plot では、 2次元プロット中に示した Y軸側:ban-x X軸側:ban-y の"色帯" 位置で切り出した(※1)1次元プロットを表示します。

[右図] Gaussian Fit plot では、 2次元プロット中に示した 黒十字: Gaussian の重心位置 ban-x ban-y の "色帯" 位置で切り出して、1次元Gaussian で fit した 結果をプロットを表示します。

[pg.9] Film#5 解析結果 (band plot)



Wobbler=ON 測定の場合は、 band-plot のみで、

Gaussian Fit解析は 行いません。 (※1) 色帯 band-x, -y の位置指定を変更する場合は、 ./gaf/gaf_anaOD.py の下記の場所を適宜変更して下さい。

odC: ndarray xyw: tuple(int,int)	Optical density of one color band [mm] odC x,y-size in mm
<pre>selfmyclear() self.odC = odC</pre>	
self.nx.self.ny = ode	C. shape
self.xw,self.vw = xv	N
<pre>self.xpix = np.arange</pre>	e(0,self.nx) # pixcel index list
<pre>self.ypix = np.arang</pre>	e(0,self.ny)
<pre>self.xp2m = selfpix</pre>	<pre>x2mm(self.xpix,'x') # pixcel index -> mm</pre>
<pre>self.yp2m = selfpix</pre>	x2mm(self.ypix,'y')
# Wob=ON/OFF 共通の解	祈 までやっておく
<pre>self.set_ticker()</pre>	
<pre>self.set_band('x')</pre>	<pre>def set_band(self, ax, banprm=(-15,15,5,3)):</pre>
self.set_band('y')	""" set band plot of odC[]
<pre>self.set_odBG()</pre>	ax: char 'x' or 'y' axis name
<pre>self.set_odMax()</pre>	<pre>banprm: int*4 [mm] (ban_start,_end,_step,_wid)</pre>

pyE5A解説 : Gaf Film 解析

(9) _anaOD.xlsx の説明

_anaOD.ppt に記した解析結果を、

別途 Excel で解析したい場合に便利でしょう。

Sheet = od(0)_Atrb class anaOD()の Attribution 一覧

	A	В	с	D	E	F (2	G	н	Ι	J	З) _к	L
1		Atr.nm	Atr.x	Atr.y	Xmm	Ymm=+1.3	Ycm=+1.3	Ycm=fit	Ymm	Xmm=-2.2	Xcm=-2.2	Xcm=fit
2	0	nx,ny [pix]	250	250	-42.5	0	0	0	42.5	0	0	0
3	1	xw,yw [mm]	85	85	-42.16	0	0	0	42.16	0	0	0
4	2	od BG [od]	0.23		-41.82	0	0	0	41.82	0	0	0
5	3	od Max [od]	0.69		-41.48	0	0	0	41.48	0	0	0
6	4	od center(x,y) [pix]	118	121	-41.14	0	0	0	41.14	0	0	0
7	5	od center(x,y) [mm]	-2.2	1.28	-40.8	0	0	0	40.8	0	0	0
8	6	od Avr(x,y) [od]	0.53	0.53	-40.46	0	0	0	40.46	0	0	0
9	7	G.fit m(x,y) [mm]	-2.191	1.349	-40.12	0	0	0	40.12	0	0	0
10	8	G.fit sg(x,y) [mm]	8.294	8.008	-39.78	0	0	0	39.78	0	0	0
11	9	G.fit a(x,y) [od]	0.46	0.459	-39.44	0	0	0	39.44	0	0	0
12	10	G.fit R2(x,y)	0.996	0.996	-39.1	0	0	0	39.1	0	0	0
13	11				-38.76	0	0	0	38.76	0	0	0
14	12				-38.42	0	0	0	38.42	0	0	0
15	13	\cup			-38.08	0	0	0	38.08	0	0	0
16	14				-37.74	0	0	0	37.74	0	0	0
17	15				-37.4	0	0	0	37.4	0	0	0
18	16				-37.06	0	0	0	37.06	0	0	0
19	17				-36.72	0	0	0	36.72	0	0	0
20	18				-36.38	0	0	0	36.38	0	0	0
21	19				-36.04	0	0	0	36.04	0	0	0
22	20				-35.7	0.225	0.226	0.226	35.7	0.201	0.229	0.229
23	21				-35.36	0.225	0.226	0.226	35.36	0.229	0.229	0.229
24	22				-35.02	0.226	0.226	0.226	35.02	0.226	0.226	0.226
74	12				-10.0ZI	0.209	0.29	0.29	10.021	0.273	0.272	0,272
-	•	od(0)_Atrb	od(0)_	odC	od(1)_/	Atrb od	(1)_odC	od(2)_/	Atrb	od(2)_od(C od(3)	_Atrb



Sheet = od(0)_odC class anaOD() \mathcal{O} odC [] dump



(1) class anaOD() \mathcal{O} Attribution

フィルムサイズ、OD値、画像中心位置 Gaussian fit の結果など。

詳しくは ./gaf/gaf_anaOD.py class anaOD(): def _myclear(): を参照。

②、③ 色帯 ban-x, -y の数値

Wob=OFF の場合(左図)は、 Gauss. fit した時の Beam 中心位置の ban のみ。 Wob=ON の場合は、 色帯 5本それぞれの ban 値を表示。

詳しくは ./gaf/gaf_anaOD.py class anaOD(): def mkdf_Atrib(): を参照。

(4) class anaOD() 𝔊 odC 𝔊 dump

各フィルムの OD値 配列

pandas の index = Y [mm], columns= X [mm] で dump してあります。

⑤ 同④に、Excel 条件付き書式 を指定して表示させた場合

条件付き書式ルールの管理			
書式ルールの表示(5): 現在の選択範囲	~		
Ⅲ新規ルール(N)	× ルールの削除(D)	■重複ルール(<u>C</u>) へ ∨	
ルール (表示順で適用)	走 書	適用先	
グラデーション カラー スケール		=\$B\$2:\$IQ\$251	t

Excel で読み込んでから、 自分で 条件付き書式 を設定して下さい。

(10)./gaf/gaf_scan.pyの説明

OD値に変換する段階で、 別途 Excel で解析したい場合に便利でしょう。

前述ページ3で記した部分について

⑤【必要なら】スキャンイメージを出力 _scanShift.ppt 中の各種イメージを、.png.csv で出力

if(False): # scanイメージ保存要なら = True # modes 指定は、gaf_scan.py:rdGafScan6.save_all()参照 scn.save_all(fnmH,modes=[2,12,15,42])

modes = [Job番号] の指定値は、 ./gaf/gaf_scan.py class rdGafScan6 ():

def save_all (): を参照。

def	<pre>save_all(self</pre>	,fnH:str='_', modes:list=[2,12,	42]):				
	""" Save images & O.D. array						
	fnH: str	file name header					
	<pre>modes: [int,</pre>] Job modes -> see below					
	mode						
	= 1 :	im[] save as	'.png'				
	= 2 :	imCur[]	'.png'				
	= 11,12,13 :	imR,G,B[]	'.png'				
	= 14,15,16 :	imR,G,B[]	'.csv'				
	= 21,22,23 :	imRm,Gm,Bm[] Masked image	'.png'				
	= 31,32,33 :	<pre>imRod,God,Bod[] O.D.image</pre>	'.csv'				
	= 41,42,43 :	odR,G,B[] O.D. 2Darray	'.csv'				

(11) プログラムをいじりたい場合



gaf_anaOD.py や gaf_scan.py の最後に、 それぞれ debag 用の if _name_ == '_main_': を用意してありますので、 イジッテみてください。

以上です。 お問い合わせは、理研あ吉田まで。 Rev. 23.02/26