

## WEB カメラで X 線を簡単に計数できる

### ■WEBカメラでX線を検出・可視化

放射線教育では線源の入手が課題の一つとなっている。そこで、自然線源以外の選択肢として、人工線源の可能性を検討する。

WEB カメラは低エネルギーの放射線を検出できることから、クルックス管から出る微弱 X 線を効率よく検出して可視化できることが知られている。

そこで、画像から X 線の痕跡を検出して計数するプログラムを作成し、各種の放射線実験に適用してみた。

### ■クルックス管と簡易高圧電源

クルックス管は、ケニス製の十字入りクルックス管 HJ 形を使用した。

高圧電源はインダクションコイルでもよいが、持ち運びに便利のように、14kV 高電圧発生ユニットを組み込んだコンパクトなクルックス管用の簡易高圧電源を試作した。

スイッチ付きの単三電池 3 本の電池ボックスを外部駆動電源としたので、要すれば遠隔での ON-OFF も可能である。

### ■WEB カメラ

WEB カメラは、ロジクール製 C270n を使用した。解像度は HD720p/30fps となっている。

WEB カメラ前面にある、動作表示 LED は不要なので、黒ビニールテープで遮光し、その上から、アルミテープを巻き付けて電氣的シールドとした。この電氣的シールドは、WEB カメラをクルックス管に接近して使用する際にクルックス管の高電圧の帯電による電撃で素子が壊れるのを防ぐ目的である。

### ■プログラムの特徴

プログラムでは、毎秒 30 コマの動画から 0.1 秒ごとのシーンを読み込んでモノクロ画像とした後、次々と加算し、10 秒間加算された画像を作る。この画像から各画素の明るさが閾値を超える画素を抽出して計数し、10/3 秒間の計数率として 6 回反復して出力する。

1 コマ 1/30 秒ごとの加算も可能だが、検出状況を見た結果、可視化に支障がないようにサンプリング数を減らした。また、これまでの経験から、画像の周辺にノイズが現れることが分かっていたので、中心部 75%×75%の画素を計数範囲とした。

別添 1 には、使用したプログラムのテキストを載せた。

### ■実験例

別添 2 の図-1 は、遮へいの実験の例で、プラスチック板とアルミ板とを比較した。横軸は  $\rho \times t$  ( $\text{g}/\text{cm}^2$ )、縦軸は CPM となっている。このように、 $\beta$  線や  $\gamma$  線とは異なり、X 線では遮へい材の原子番号依存性が現れてくる特徴がある。

図-2 は、距離の実験の例で、クルックス管の場合は、端面の広い範囲から X 線が出るため、点線源にならない。そこで、0.5mm 厚の鉛板に 6mm の丸穴を開けてスリットとした。その結果、計数率が大幅に低下して、距離による低減が大きく、2 条件しか測定できなかったが、逆二乗測に沿う結果となった。

このように、クルックス管を人工線源として放射線実験に利用できる可能性が示された。

```

import cv2 #画像処理ライブラリをインポート

cap = cv2.VideoCapture(0) #カメラ指定
ret, frame1 = cap.read() #初期画像入力
img1 = cv2.cvtColor(frame1, cv2.COLOR_BGR2GRAY)
#カラー画像をモノクロ画像に変換
img0 = img1 #画像初期値
threshold = input("Threshold(0<255)? ") #計数の閾値
count = 0 #計数
m = 0 #10 秒用コマ数カウンタ
n = 0 #0.1 秒用コマ数カウンタ
k = 1 #繰り返し数

while k <= 6: #6 回繰り返し
    ret, frame2 = cap.read() #画像入力
    img2 = cv2.cvtColor(frame2, cv2.COLOR_BGR2GRAY)
#カラー画像をモノクロ画像に変換
    if m == 300: #10 秒ならば
        m = 0 #10 秒用コマ数カウンタをクリア
        for a in range(60, 419): #画像の縦範囲を制限
            for b in range(80, 559): #画像の横範囲を制限
                px = img1[a, b] #画素(a,b)の輝度
                if px > int(threshold): #輝度が閾値以上
                    count += 1 #計数を 1 増やす
            print('CP10/3S= ' + str(count)) #10 秒率を出力
        k += 1 #繰り返し数を 1 増やす
        count = 0 #計数をリセット
        img1 = img0 #画像をリセット

    if n == 3: #0.1 秒ならば
        n = 0 #0.1 秒用コマ数カウンタカウンタをクリア
        img = img1 + img2 #画像を加算
        img1 = img #加算した画像を次に送る

    m += 1 #10 秒用コマ数カウンタを 1 増やす
    n += 1 #0.1 秒用コマ数カウンタを 1 増やす

cap.release() #カメラをリリース

```

## ■今後の展開

クルックス管と WEB カメラを組み合わせることで、従来にない放射線実験が行えることが分かった。

また、Python によるプログラミングは簡易で、随意に拡張が可能であることから、今後の ICT 教育の一環としても期待できる。