

Detección y análisis de interacciones de partículas con sensores APS-CMOS mediante la implementación de una librería en C++

DARÍO FEDERICO BALMACEDA

Laboratorio Detección de Partículas y Radiación. Centro Atómico Bariloche
leschatten@gmail.com

24 de mayo de 2018

Resumen

Los sensores CMOS constituyen una alternativa de bajo precio para la detección de interacciones de partículas. En este trabajo se implementó una librería para utilizar el sensor OmniVision OV5647 de la cámara Raspicam V1.3, con una Raspberry Pi. Utilizando esta librería se observaron los picos de emisión de rayos X K_α y K_β del Cu y los picos K_α del Fe y del Ca. La determinación de estos picos permitió hacer la correcta calibración de carga depositada en función del valor de los píxeles. Finalmente se utilizó la librería para la detección de interacciones con partículas secundarias producidas por rayos cósmicos con el sensor.

I. INTRODUCCIÓN

Los sensores de imágenes CMOS son ampliamente utilizados en los dispositivos electrónicos comerciales debido a su bajo costo de producción. Estos están presentes en celulares, consolas de videojuegos, cámaras de acción, sistemas de seguridad y un sinnúmero de aplicaciones más. En el ámbito científico, estos sensores están comenzando a tener relevancia [1], ya que permiten realizar estudios de dosimetría, calcular actividad de muestras radioactivas y control de reactores nucleares a un coste muy inferior que el de los CCDs utilizados para determinar la carga depositada en las interacciones sensor-partícula. Como en experimentos como DAMIC (búsqueda de materia oscura)[2] o CONNIE (medición de neutrinos)[3].

En base a esto se decidió implementar una librería con filosofía de programación orientada a objetos capaz de hacer uso de estos sensores CMOS para la detección de interacciones de partículas con el sensor, y poder estimar parámetros como el número de eventos en una imagen (fotograma) y la carga de estos eventos.

I.1. Sensores CMOS-APS

Un sensor de píxeles activos (APS por sus siglas en inglés) es un sensor que detecta la radiación utilizando tecnología CMOS. La tecnología CMOS hace referencia a un conjunto de familias lógicas basadas en semiconductores complementarios de óxido metálico, estos utilizan transistores de tipo pMOS y de tipo nMOS, de esta forma el único consumo en reposo se debe a las corrientes parasitarias.

Estos sensores consisten en un arreglo matricial de fotodiodos, que producen una corriente de electrones que varía en función de la intensidad de luz recibida, basándose en el efecto fotoeléctrico para la generación de pares electrón-hueco. Por cada fotodiodo se incorpora un amplificador. Según el diseño, se incorpora un conversor analógico digital (ADC) para la lectura de los datos, a cada píxel o por columnas.

Filtro de Bayer

Los sensores CMOS destinados para fotografías a color poseen un filtro de Bayer, el mismo consiste en un arreglo de filtros rojos, verdes y azules dispuestos como se muestra en la Fig. 1. De esta forma, cada píxel posee la información de un único rango de longitudes de onda, esto permite una composición de

la imagen en 3 colores diferentes, de manera análoga al ojo humano. Para formar la imagen final en una fotografía, se utiliza un algoritmo de des-Bayerización.[4]

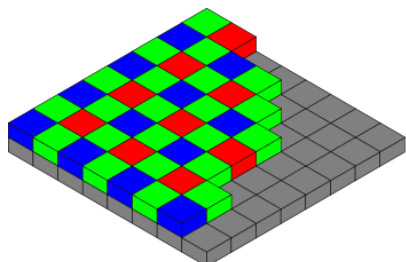


Figura 1: Filtro de Bayer típico de un sensor CMOS-APS. Cada 4 píxeles hay 2 con filtro verde, 1 con rojo y el último con azul. El color verde es utilizado dos veces debido a la sensibilidad al verde del ojo humano.

Si bien este filtro resulta invisible a los rayos X, y no es relevante en el caso de un sensor CMOS para detectar partículas; el distinguir fotografías por sus colores permitió caracterizar de los distintos parámetros a la hora de tomar una fotografía, tal como se describe en el apéndice A.

I.2. Detección de partículas con el sensor CMOS

Una partícula que deposite una energía E en el sensor, es capaz de generar E/a pares de electrón-hueco, siendo a la energía promedio para generar un par electrón-hueco en el material¹. El valor máximo de electrones hasta llegar a saturación es conocido como *Full Well Capacity*.

Si la densidad de eventos en una imagen es baja², es posible identificar eventos a partir de los píxeles de una vecindad, debido a que se desprecia la probabilidad de encontrar un píxel que haya acumulado carga de dos eventos distintos. En caso de haber una densidad de eventos mayor, pueden utilizarse otros métodos, como técnicas basadas en algoritmos de

aprendizaje profundo (*deep learning*) e inteligencia artificial. [6]

I.3. Fluorescencia de rayos X

La emisión de rayos X característicos corresponde a la ionización de un átomo en la que un electrón de las primeras capas es excitado a un estado no ligado. En este caso un electrón de una capa superior ocupa la vacante dejada por el electrón. Durante este proceso, dado que la energía se conserva, se emite un fotón cuya energía es igual a la diferencia entre los niveles de la transición.

Los picos K_α y K_β corresponden a las transiciones de un electrón de un estado $|2p\rangle$ al estado $|1s\rangle$ y a la transición $|3p\rangle$ a $|1s\rangle$, respectivamente.

En el caso particular del Cu, Fe y Ca, 3 elementos que usaremos a continuación, los valores de energía de los fotones emitidos se pueden encontrar en la Tabla 1.

Z	Elem.	K_{α_1}	K_{α_2}	K_β
20	Ca	3692	3688	4012
26	Fe	6404	6391	7058
29	Cu	8048	8028	8905

Tabla 1: Valores de energía, en eV, de los fotones emitidos para los picos de emisión K_α y K_β del Ca, Fe y del Cu.[7].

I.4. Rayos cósmicos

Los rayos cósmicos son partículas que llegan desde el espacio y bombardean constantemente la Tierra desde todas direcciones. La mayoría de estas partículas son protones o núcleos de átomos. Al interactuar con la atmósfera terrestre, los rayos cósmicos de alta energía son capaces de producir hadrones cargados, neutrones, fotones (rayos γ), muones (de 1 a 5 GeV), electrones y positrones (de 1 a 20 MeV), [8] cuyos efectos son medibles.

¹3.7 eV en el caso del silicio[5]

²En relación con el tamaño de los eventos

II. CONFIGURACIÓN EXPERIMENTAL

II.1. Sensor CMOS

Se ha utilizado el sensor OmniVision OV5647 de la cámara Raspicam V1.3 cuyo precio ronda los 23 dólares. El sensor posee 2592x1944 píxeles, lo que le da una resolución total de 5 MP. El tamaño de un píxel de $1.4 \mu\text{m} \times 1.4 \mu\text{m}$. La capacidad de carga máxima hasta llegar a saturación es de 4300 electrones (*Full Well Capacity*). El sensor posee un ADC de 10 bits por píxel. El tamaño total por imagen raw es de unos 6.4 MB.

II.2. Raspberry

Se utilizó una Raspberry Pi modelo B, con una memoria micro SD de 16 GB, para la obtención de imágenes. Debido a la resolución de las imágenes, la profundidad de bits de las imágenes y del tamaño del sistema operativo *Raspbian*, la memoria disponible permitió almacenar del orden de 1000 fotografías en simultáneo.

II.3. Librería raspiraw

Para la adquisición de datos se utilizó la librería *raspiraw*[9] debido a la rapidez con la que se toman los datos. En el apéndice A se muestran otras alternativas que son más lentas para la toma de datos, pero que pueden resultar más simples y sencillas de implementar.

Los datos sin procesamiento (datos *raw*) se almacenaron con extensión *.raw*. El valor de carga acumulada por cada píxel se representa mediante un número de 10 bits. En una misma fila, los píxeles son agrupados de a 4, formando una estructura de 5 bytes. En los primeros 4 bytes se encuentran los bits más significativos de los 4 píxeles que conforman la estructura. El quinto byte contiene los últimos 2 bits menos significativos de cada píxel.[4] La manera en la que los bits están ordenados en un grupo se muestra en la Fig. 2

	Bits								
	MSB	8	7	6	5	4	3	2	LSB
Bytes	1	10	9	8	7	6	5	4	3
	2	10	9	8	7	6	5	4	3
	3	10	9	8	7	6	5	4	3
	4	10	9	8	7	6	5	4	3
	5	2	1	2	1	2	1	2	1

Figura 2: Representación de un grupo 4 píxeles. Los 40 bits de los 4 píxeles están distribuidos como ilustra la figura. Colores distintos corresponden a píxeles distintos. En cada celda se representa el bit de información de un píxel con un número, siendo 1 el bit menos significativo (LSB) y 10 el bit más significativo (MSB).

Estos datos consistían en 1952 filas de 3264 bytes. Las últimas 8 filas de datos no contienen información y no son utilizadas, ya que sólo existen debido a que 1952 es menor múltiplo de 16 mayor a las 1944 filas que posee el sensor (ver sección II.1). De la misma forma, los últimos 24 bits de cada fila no son utilizados y deben ser descartados.

Esta librería permite cambiar varios parámetros de la toma de datos. En particular, los parámetros que se modificaron son:

Tiempo de exposición: Se define como el tiempo en el que sensor acumula carga. A mayor tiempo de exposición, mayor es la carga leída por el detector, cuando se encuentra expuesto a una fuente de luz.

En lo que sigue, se utilizó un tiempo de exposición de 500 ms, a no ser que se aclare lo contrario.

Cuadros por segundos: También conocido como *fps* por sus siglas en inglés. Se refiere a la frecuencia con la que se toman las imágenes y generalmente viene expresado en fotografías por segundo.

En lo que sigue, se capturó a 2 cuadros por segundo, a no ser que se aclare lo contrario.

II.4. Librería propia

Para el procesamiento de las imágenes de raspaw se implementó una librería en C++. En esta librería las imágenes pasan a ser objetos representados por una matriz (arreglo modulado). Cada foto presenta un ancho, un largo y un conjunto de datos que corresponde al valor de cada pixel. En esta clase se definen los métodos (funciones) para encontrar eventos, basándose en pixeles adyacentes, para recortar imágenes, encontrar el valor medio, la mediana y hasta para encontrar la desviación estándar de los valores de los pixeles.

A su vez, cada evento es representado por otra clase cuya identidad viene dado por una lista de pixeles (posición horizontal, posición vertical y valor). Para esta clase se definen métodos que permiten obtener la carga total de un evento, definida como la suma de los valores de los pixeles; para obtener el centro de masa del evento, definida como el promedio de los vectores posiciones de cada pixel, ponderado con el valor del pixel; para determinar el radio de un evento, definido como la raíz cuadrada de la varianza de las posiciones de los pixeles al centro de masa, también ponderado por el valor de cada pixel.

El código fuente puede encontrarse en la plataforma GitHub mediante el siguiente enlace github.com/DBFritz/ParticleDetection

II.5. Emisor de rayos X

En la Fig.3 se muestra la configuración experimental utilizada. La cámara de la Raspberry Pi se colocó en un recipiente de plástico opaco a la luz visible, para evitar detectar los fotones en ese rango de longitudes de onda. El equipo utilizado posee dos ventanas, se utilizó la menos coherente para facilitar el proceso de alineado del haz con la cámara.

El generador de rayos X utilizado corresponde al grupo de materiales del Centro Atómico Bariloche, el mismo utiliza la diferencia de energía entre los niveles del Cu para generar los rayos X. Por este motivo, la emisión de los rayos X no corresponde a un espectro uniforme, sino que presenta máximos locales en los valo-

res que corresponden a diferencias de niveles energéticos.

Para la medición de los picos de Fe y Ca se intercaló una lámina de Fe y una concha marina (rica en Ca). Debido a que los rayos X se generan en base al Cu, los picos de emisión también estarán presentes en el espectro de los demás elementos.

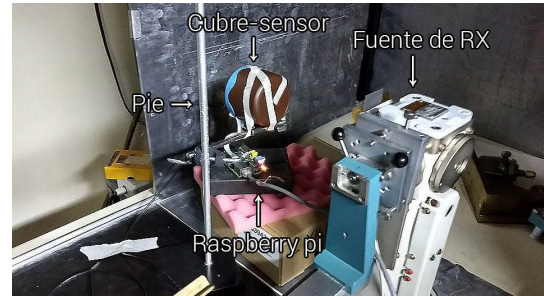


Figura 3: Configuración experimental utilizada para la detección de rayos X. El sensor se colocó dentro de un recipiente plástico para protegerla de la luz visible. El pi permitía alinear correctamente el haz de fotones con la cámara.

II.6. Observación de rayos cósmicos

Utilizando la librería propia desarrollada en el marco de este proyecto (ver sección II.4), es posible identificar eventos en paralelo con la toma de imágenes, por lo que se implementó un sistema que guarde solamente las imágenes que poseen pixeles que superen un umbral establecido, permitiendo así medir durante días sin saturar la memoria, eliminando las fotos vacías de eventos. Debido al ruido de lectura, dicho umbral se ajustó en 75 unidades de ADC. Como se verá en la sección III.3, con esta configuración fue posible detectar eventos, por lo que el umbral establecido resultó válido y suficiente. Se tomaron 2 imágenes por segundo, pero éstas eran procesadas a una velocidad del orden de 1 imagen cada 3 segundos, por lo que se espera que la frecuencia de eventos detectados debido a rayos cósmicos sea notablemente menor a la frecuencia de eventos por unidad de área esperada, que es del orden de $100 \text{ m}^{-2}\text{s}^{-1}$ (para una sección de 10 mm^2 , casi 4 eventos por horas).

III. RESULTADOS Y DISCUSIÓN

Se probó el correcto funcionamiento de la librería, descrita en la sección II.4, en tres aplicaciones diferentes. En primer lugar se determinó el ruido de lectura de la cámara, otra de ellas consistió en observar los picos K_α y K_β de diferentes elementos para calibrar la respuesta del sensor, y por último, en detectar rayos cósmicos.

III.1. Determinación del ruido de lectura

Se realizó un histograma de los valores de los pixeles de una única imagen, para verificar la distribución y la dispersión del ruido. Los resultados obtenidos se muestran en la Fig. 4.

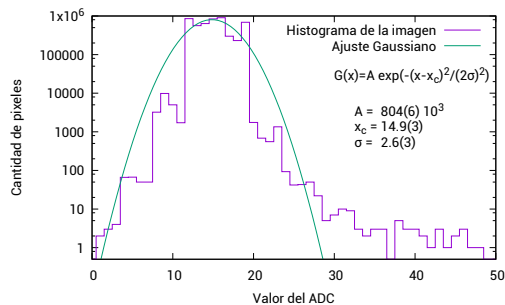


Figura 4: Histograma de los valores de los pixeles con un ajuste gaussiano. El histograma se muestra en escala logarítmica. La gaussiana se encuentra centrada en 14,9(3) unidades de ADC, mientras que su dispersión es 2,6(3) unidades de ADC.

Se observa que el ruido está muy alejado de un comportamiento de ruido blanco (gausiano), indicando la presencia de un FPN (ruido de patrón fijo).

Se tomaron 30 imágenes con tiempo de exposición de 500 ms a 2 cuadros por segundos (fps). Se promediaron las 30 imágenes para obtener una estimación sobre el ruido de lectura con la configuración dada. En la Fig. 5 se muestra el promedio por píxel de las fotografías tomadas.

En base a la fuerte dependencia del valor medio de los pixeles como función de la columna, se implementó una función que per-

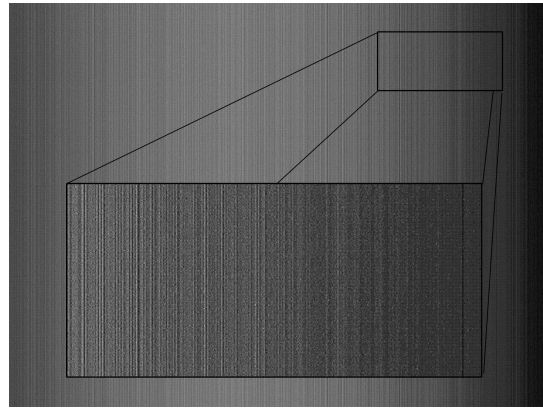


Figura 5: Promedio de las 30 imágenes del fondo. Se observa una dependencia del valor promedio como función de la posición. Esta dependencia está fuertemente relacionada con la columna del píxel.

mite sustraer, a cada columna, el valor medio por columna. Esta función será aplicada a cada procesamiento de datos de imágenes, salvo que se indique lo contrario. Se recalculó el histograma de la misma imagen que en la Fig. 4, con la respectiva sustracción de la mediana por columna. Dicho histograma se muestra en la Fig. 6.

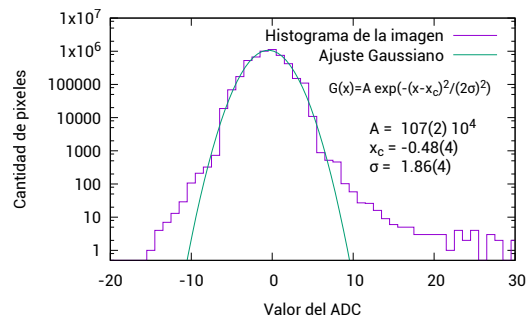


Figura 6: Histograma de los valores de los pixeles, después de corregir por columna, con un ajuste gaussiano. Los valores obtenidos del ajuste se muestran en el gráfico.

En este caso se observa un comportamiento notablemente más gaussiano, con una dispersión un 28 % menor.

III.2. Mediciones de picos K_α y K_β y calibración de energía

Se analizó el espectro de Cu, Fe y Ca utilizando la librería propia, descrita en la sección II.4. Para esto, se realizó un histograma de las cargas de los eventos del orden de 300 imágenes. En estos espectros fue posible identificar los picos K_α y K_β de Cu y los picos K_α del Fe y del Ca, tal como se muestra en la Fig. 7.

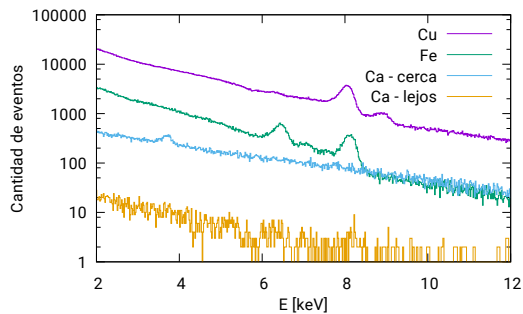


Figura 7: Espectros detectados por el sensor. El espectro del Cu corresponde al espectro de emisión de la fuente utilizada. Los demás espectros se obtuvieron colocando una placa de Fe y de Ca, entre la fuente y el sensor. En el gráfico se ha utilizado la calibración entre energía y unidades de ADC mostrada en la Fig.8.

En el caso del espectro del Fe, puede observarse la presencia del pico K_α del Cu, debido a que la fuente es de Cu. De la misma forma, este pico puede verse en el caso del espectro del Ca cuando éste fue ubicado lejos del sensor.

Conociendo la energía característica de cada uno de estos picos [7], se realizó una calibración del valor raw de un pixel como función de la carga (energía) depositada. Esta calibración se muestra en la Fig. 8

III.3. Medición de Rayos Cósmicos

Utilizando la librería propia, se pudo encontrar 2 eventos en 2 días seguidos, con la cámara en oscuridad y sin la presencia de fuentes radioactivas cerca, por lo que se asocia estos eventos a los rayos cósmicos. En la Fig. 9 se muestran ambos eventos detectados.

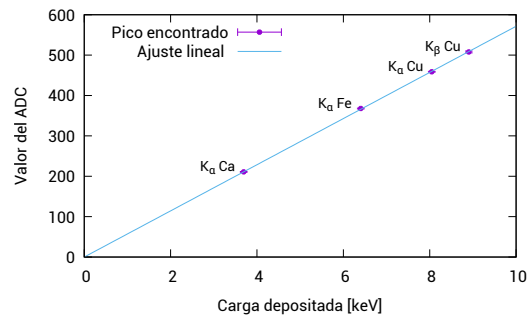


Figura 8: Calibración del canal del sensor como función de la energía de los picos K_α y K_β encontrados. En el mismo se puede apreciar que la relación es lineal, y la relación carga por unidad de ADC obtenida es de 17.50(3) eV / ADC

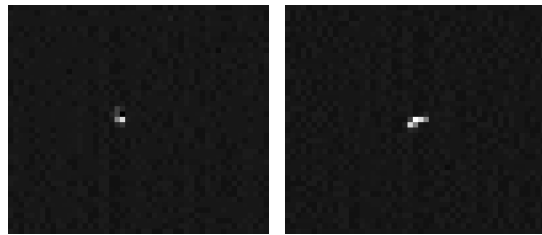


Figura 9: Recortes de las imágenes tomadas en la que se observaron un evento de rayos cósmicos. En este caso el color negro representa el 0 unidades de ADC, y el color blanco, 180 unidades de ADC (3150 eV).

IV. CONCLUSIONES

Se utilizaron sensores CMOS para la detección de eventos con partículas para una descripción cualitativa y esquemática. Se analizó la dependencia con los distintos parámetros de la cámara y en base a esto se decidió una configuración de trabajo.

Gracias a esto se pudo caracterizar y calibrar el sensor para la configuración utilizada mediante la implementación de la librería en C++ con filosofía de Programación Orientada a Objetos. Utilizando esta librería se observó la presencia de eventos producidos por rayos cósmicos y se observaron los picos K_α y K_β del Cu y los picos K_α del Fe y Ca.

El alcance de esta librería es amplio. Uno de los posible usos es la detección de plomo, en

el agua (vía observación del ^{210}Pb). Además podría utilizarse como un contador de Geiger de bajo costo.

Alcalde por proveernos e informarnos sobre el equipo de rayos X.

REFERENCIAS

- [1] M. Pérez and M. S. Haro Et Al. Commercial cmos pixel array for beta and gamma radiation particle counting. In *EAMTA 2015*, pages 11–16, Julio 2015.
- [2] DAMIC collaboration. Damic: a novel dark matter experiment. 2013.
- [3] CONNIE collaboration. The connie experiment. *Journal of Physics: Conference Series*, 761(1):012057, 2016.
- [4] Dave Jones. Picamera 1.13 documentation. <https://picamera.readthedocs.io/en/release-1.13/>, Febrero 2017.
- [5] Don Et. Al Groom. Temperature dependence of mean number of eh pairs per ev of x-ray energy deposit. 2009.
- [6] Byron P. Roe and Hai-Jun Yang Et Al. Boosted decision trees as an alternative to artificial neural networks for particle identification. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 543(2):577 – 584, 2005.
- [7] Albert C. Thompson and Douglas Vaughan. X-ray data booklet, 2009.
- [8] Peter K.F. Grieder, editor. *Cosmic Rays at Earth*. Elsevier, Amsterdam, 2001.
- [9] 6by9. raspiraw. <https://github.com/6by9/raspiraw>, 2018.

AGRADECIMIENTOS

Se agradece la colaboración de Xavier Bertou por la experiencia brindada y por acompañar el trabajo en forma constante. A Miguel Sofo por la disponibilidad y por la ayuda a la hora de hacer mediciones. Al equipo de Fabricio

A. ALTERNATIVAS A RASPIRAW

Antes de desarrollar la librería propia, se usaron alternativas disponibles en internet para entender el funcionamiento de la Picamera.

Raspistill y Raspivid

Ambos vienen por defecto instalados en el sistema operativo *Raspbian*. Raspistill permite capturar fotografías en formato *Joint Photographic Experts Group* (jpeg/jpg). Si bien las imágenes tomadas presentan un post-procesamiento, Raspistill es capaz de añadir a la imagen los datos sin procesar (*raw*).

Por el otro lado, raspivid permite grabar videos en formato *H.264*. Este formato también presenta compresión de datos, al igual que *.jpg*, pero no existe parámetro que permita obtener las imágenes sin procesamiento. Por lo cual solamente puede ser utilizado de manera ilustrativa y cuantitativa, pero no cuantitativa.

Para verificar el funcionamiento del sensor, se expuso la cámara ante una fuente de rayos alfa (^{241}Am), removiendo previamente el sistema óptico de lentes que poseía la cámara *picamera*. Utilizando raspivid, se tomó 1 minuto de exposición a 30 cuadros por segundos, dando un total de 1800 imágenes. Por pixel, se tomó el máximo valor de todas las imágenes, generando la imagen que se muestra en la Fig. 10

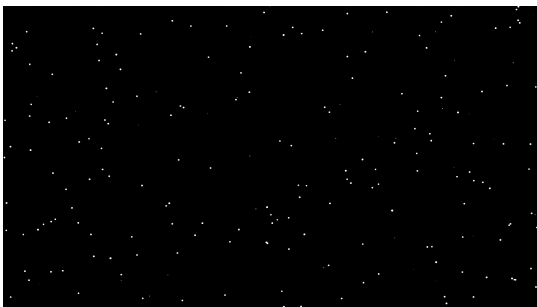


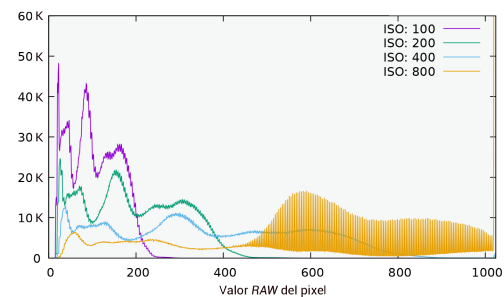
Figura 10: Imagen de los eventos con exposición de 1 minuto ante una fuente de ^{241}Am , que emite radiación alfa de 5.486 MeV. Para la composición de la misma se tomó el valor máximo por pixel de cada fotograma del video.

Picamera, librería en Python

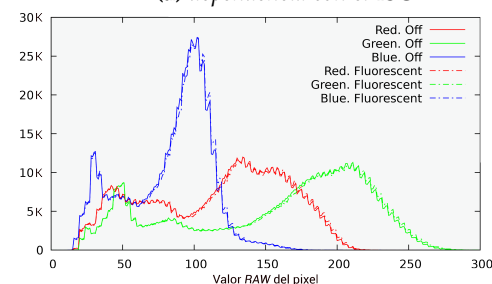
La librería Picamera presenta en su documentación una forma para obtener los datos *raw*[4]. Esta librería, permite controlar los valores de valores de ISO, tiempo de exposición, balance de blancos, resolución, entre otros parámetros para la toma de fotografías.

Como primera aproximación se estudió la dependencia de los histogramas de los valores *raw* de los pixeles como función del ISO, tal como se muestra en la Fig. 11.(a). En base a esto se concluye que el valor de ISO no es un post-procesamiento, ya que no influye en los datos *raw*. La curva de ISO= 800 presenta una preferencia impar-par debido a la electrónica interna del sensor, de la cual no se hizo estudios posteriores.

Luego se analizó la dependencia del histograma para los balances de blancos apagado y "fluorescente". No se observó dependencia del histograma como función del balance de blancos, lo que indica que el balance de blancos es un post-procesamiento. Los histogramas obtenidos, por color, se muestran en la Fig. 11.(b).



(a) dependencia con el ISO



(b) dependencia con el balance de blancos

Figura 11: Histogramas de los valores de los píxeles

Estos estudios confirman el potencial de los datos raw para los estudios de física considerados. Sin embargo la toma de datos era muy lenta, lo que llevó a desarrollar de la librería descrita en la sección II.4.