

# Implementación de una librería para la detección y el análisis de interacciones de partículas con CMOS

DARÍO FEDERICO BALMACEDA

Laboratorio Detección de Partículas y Radiación. Centro Atómico Bariloche

leschatten@gmail.com

6 de mayo de 2018

## Resumen

*Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.*

## I. INTRODUCCIÓN

**L**orem ipsum dolor sit amet, consectetur adipiscing elit.

II. Interacción de la radiación de partículas con la materia

III. Rayos cósmicos

IV. Picos  $K_\alpha$  y  $K_\beta$

## II. CONFIGURACIÓN

### EXPERIMENTAL

### I. Sensores CMOS-APS

Un sensor de píxeles activos (APS por sus siglas en inglés) es un sensor que detecta la radiación basado en tecnología CMOS. Este tipo de sensores son ampliamente utilizados de manera comercial debido a su bajo costo de producción y sus buenos resultados en fotografía. Estos sensores están presentes en teléfonos celulares, ordenadores portátiles, cámaras de acción, y en la mayoría de los dispositivos electrónicos que posee una cámara.

Los sensores consisten en un arreglo matricial de fotodiodos, que producen una corriente eléctrica que varía en función de la intensidad de luz recibida. Por cada fotodiodo, se incorpora un amplificador y un conversor analógico digital (ADC) para la lectura de los datos.

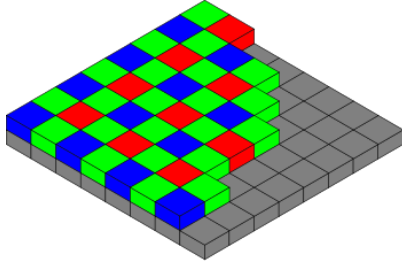
I. Raspberry

II. Sensor CMOS

Se ha utilizado el sensor OmniVision OV5647 de la cámara Raspicam V1.3 cuyo valor ronda los 25 dólares. El sensor posee una resolución de 2592x1944 píxeles, lo que le da una resolución total de 5 MP. El tamaño de un píxel es de  $1.4\mu\text{m} \times 1.4\mu\text{m}$ . El FULL-WELL-CAPACITY es de 4300. El sensor posee un ADC de 10 bits por píxel.

El sensor CMOS posee 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 una longitud de onda, esto permite una composición de la imagen en 3

colores diferentes, de manera análoga al ojo humano.



**Figura 1:** Filtro de Bayer típico de un sensor CMOS-APS. Por cada 4 píxeles 2 son verdes, 1 es rojo y 1 es azul. El color verde es utilizado dos veces debido a la sensibilidad al verde del ojo humano.

### III. librería *raspiraw*

Para la adquisición de datos se utilizó la librería *raspiraw*[1] 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.

## III. RESULTADOS

Para probar el correcto funcionamiento de la librería, se utilizó el poder de procesamiento en dos aplicaciones diferentes. Una de ellas consistió en detectar Rayos Cósmicos, y la otra la de observar los picos  $K_\alpha$  y  $K_\beta$  de diferentes elementos.

### I. Medición de Rayos Cósmicos

### II. Medición de picos $K_\alpha$ y $K_\beta$

1. *Cobre*
2. *Aluminio*
3. *Hierro*
4. *Calcio*

## IV. CONCLUSIONES

## V. REFERENCIAS

### REFERENCIAS

- [1] 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 su flexibilidad a la hora de trabajar

A. ALTERNATIVAS A RASPIRAW

B. DOCUMENTACIÓN DE LA  
LIBRERÍA

- I. rawImages
- II. rawEvent
- III. rawFilters