

Análisis de interacciones de partículas mediante implementación de una librería para su detección con sensores CMOS

DARÍO FEDERICO BALMACEDA

Laboratorio Detección de Partículas y Radiación. Centro Atómico Bariloche

leschatten@gmail.com

24 de mayo de 2018

Resumen

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.

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 costo inferior que los CCDs utilizados en los detectores de partículas.

En base a esto se propone implementar una librería con filosofía de programación orientada a objetos capaz de hacer uso de estos sensores CMOS para la detección de eventos de partículas.

I. 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 utili-

zan 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 y un conversor analógico digital (ADC) para la lectura de los datos.

1. 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.[2]

Si bien este filtro resulta invisible a los rayos X; el distinguir fotografías por sus colores permitió caracterizar de los distintos parámetros

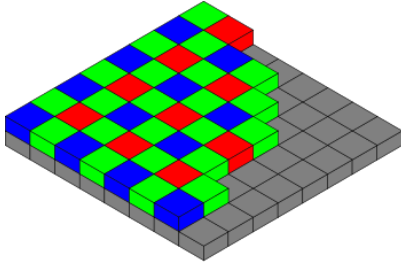


Figura 1: Filtro de Bayer típico de un sensor CMOS-APS. Cada 4 píxeles hay 2 con filtro verde, 1 con rojo y el último con azul. El color verde es utilizado dos veces debido a la sensibilidad al verde del ojo humano.

a la hora del tomar una fotografía, tal como se describe en el apéndice A.

II. Detección de partículas con el sensor CMOS

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

En el caso particular de los rayos X, estos interactúan con el Si principalmente por efecto fotoeléctrico, por lo que el fotón de energía E es absorbido, y un electrón es expulsado con una energía $E - E_b$, siendo E_b la energía de enlace del electrón eyectado (1.78 KeV para el Si)[1].

Si la densidad de eventos en una imagen es baja¹, 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 *deep learning* e inteligencia artificial. [3]

¹En relación con el tamaño de los eventos

III. 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, los valores de energía de los fotones emitidos se encuentran 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.[4].

IV. 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, estas partículas son capaces de producir hadrones cargados, neutrones, fotones (rayos γ), muones, electrones y positrones, [5] cuyos efectos son medibles.

II. CONFIGURACIÓN EXPERIMENTAL

I. 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 es 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. 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.

III. Librería raspiraw

Para la adquisición de datos se utilizó la librería *raspiraw*[6] 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*) eran guardados con extensión raw. Los mismos 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 I). De la misma forma, los últimos 24 bits de cada fila no son utilizados y deben ser descartados.

El valor de carga acumulada por cada píxel se representa mediante un número de 10 bits. En una misma columna, 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.[2] La manera en la que los bits están ordenados en un grupo se muestra en la Fig. 2

Los parámetros que permite variar esta librería son:

		Bits								
		MSB								LSB
		8	7	6	5	4	3	2	1	
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 las filas se indica en número de byte (siendo 5 en total).

En cada celda se encuentra un número que representa el bit de información de un píxel, siendo 1 el bit menos significativo (LSB) y 10 el bit más significativo (MSB).

Tiempo de exposición: Se define como el tiempo en el que el sensor acumula carga. A mayor tiempo de exposición, mayor es la carga leída por el detector.

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.

IV. 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 visible para evitar 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 valores 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).



Figura 3: Configuración experimental utilizada. Se

v. Observación de rayos cósmicos.

Utilizando la librería 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 píxeles que superen un umbral establecido. Debido al ruido de fondo, dicho umbral se ajustó en 75 Unidades de ADC. Como se verá en la sección III, con esta configuración fue posible detectar eventos, por lo que el umbral establecido resultó válido y suficiente. Se tomaron 2 imágenes por segundo, pero estas eran capaces de procesarse a una velocidad del orden de 1 imagen cada 3 segundos, por lo que la frecuencia de eventos debido a rayos cósmicos detectado sea menor a la frecuencia de eventos esperada, que es del orden de $100 \text{ m}^{-2} \text{ s}^{-1}$.

vi. 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 matriz (arreglo modulado). Cada foto presenta un ancho, un largo y un conjunto de datos que corresponden al valor de cada píxel. En esta clase se definen los métodos (funciones) para encontrar eventos, basándose en píxeles adyacentes, para recortar imágenes, encontrar el valor medio, la mediana y hasta para encontrar la desviación estándar de los valores de los píxeles.

A su vez, cada evento es representado por otra clase cuya identidad viene dado por una lista de píxeles (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 píxeles; para obtener el centro de masa del evento, definida como una promedio de los vectores posiciones de cada píxel, ponderado con el valor del píxel; para determinar el radio de un evento, definido como la raíz cuadrada de la varianza de las posiciones de los píxeles al centro de masa, también ponderado por el valor de cada píxel.

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

III. RESULTADOS Y DISCUSIÓN

Se probó el correcto funcionamiento de la librería en tres aplicaciones diferentes. En primer lugar se determinó el ruido de fondo de la cámara, otra de ellas consistió en observar los picos K_α y K_β de diferentes elementos, y por último, en detectar Rayos Cósmicos.

i. Determinación del ruido de fondo

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

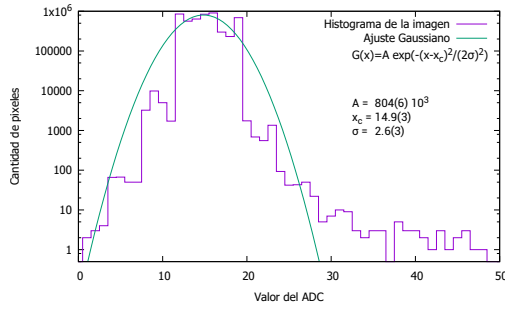


Figura 4: Histograma de los valores de los pixeles con un ajuste gaussiano. El histograma se muestra en escala logarítmica. La gaussiana se encuentra centrada en 14,8(3) unidades de ADC, mientras que su dispersión es 2,5(3) unidades de ADC.

Se tomaron 30 imágenes con tiempo de exposición de 0.5 ms a 2 cuadros por segundos (fps). Se promediaron las 30 imágenes para obtener una estimación sobre el ruido de fondo con la configuración dada. En la Fig. 5 se muestra el promedio por píxel de las fotografías tomadas.

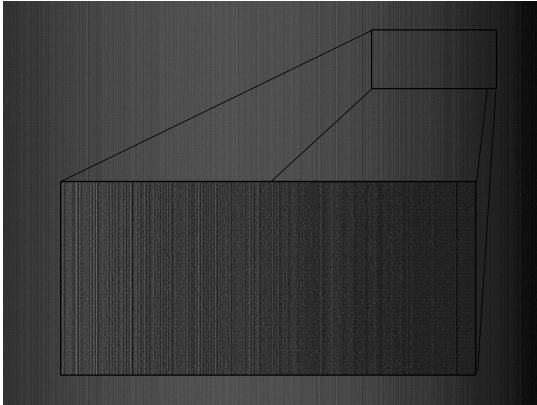


Figura 5: Promedio de las 30 imágenes. En la misma se observa una dependencia del valor promedio como función de la posición. Esta dependencia está fuertemente relacionada con la columna del píxel.

En base a la fuerte dependencia del valor medio de los pixeles como función de la columna, se implementó una función que permite sustraer, a cada columna, la mediana por columna. Esto se hizo con la premisa de que, en una misma columna, los pixeles que poseen

información de eventos son menos que el resto. Esta función será aplicada a cada procesamiento de datos de imágenes. El histograma de la misma imagen que en la Fig. 4, con la respectiva sustracción de la mediana por columna, se muestra en la Fig. 6

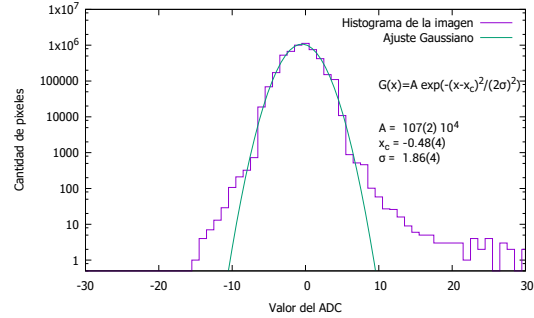


Figura 6: Histograma de los valores de los pixeles, con otro ajuste gaussiano.

II. 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. Para esto se realizó un histograma de la carga de los eventos. 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, esta identificación y asociación nos permitió hacer una calibración de energía como función del canal. tal como se muestra en la Fig. 7.

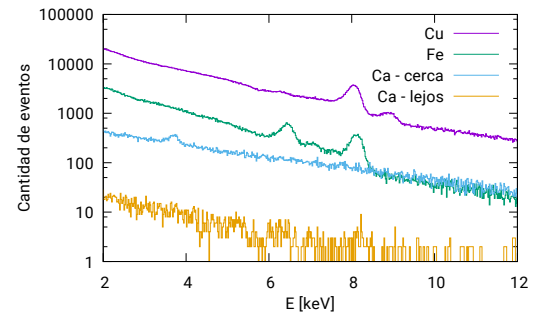


Figura 7: Espectro patronum!

Conociendo la energía característica de cada uno de estos picos [7], y bajo la suposición que los electrones generados pierden todas su

energía, 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

En estos gráficos no fue posible el análisis de la atenuación debido a la presencia de Fe ni de Ca ya que la distancia del sensor a la fuente fueron distintas en todos los casos, y por lo tanto, el ángulo sólido subtendido por el detector fue diferente en todos los casos. Un análisis más cuidadoso sobre las intensidades permitiría encontrar el coeficiente de atenuación para distintos elementos.

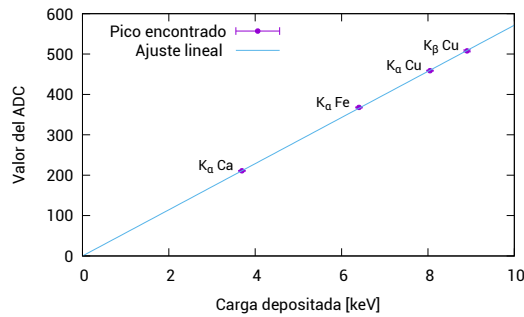


Figura 8: Calibración del canal del sensor como función de la energía de los picos K_{α} y K_{β} encontrados. En el mismo se puede apreciar que la relación es lineal, y la relación carga por unidad de ADC obtenida es de $17.50(3)$ eV

III. Medición de Rayos Cósmicos

Utilizando la librería 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 muestra uno de los eventos detectados.

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.

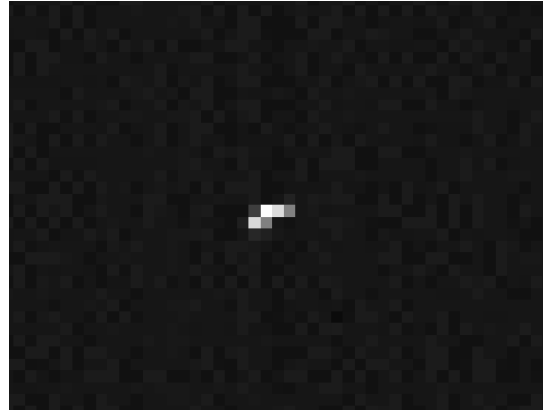


Figura 9: Recorte de una imagen tomada en la que se observa un evento de rayo cósmico. En este caso el color negro representa el 0 unidades de ADC, y el color blanco, 180 unidades de ADC.

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.

REFERENCIAS

- [1] M. Pérez, M. S. Haro, I. Sidelnik, L. Tozzi, D. R. Brito, C. Mora, J. J. Blostein, M. G. Berisso, and J. Lipovetzky. Commercial cmos pixel array for beta and gamma radiation particle counting. In *2015 Argentine School of Micro-Nanoelectronics, Technology and Applications (EAMTA)*, pages 11–16, July 2015.
- [2] Dave Jones. Picamera 1.13 documentation. <https://picamera.readthedocs.io/en/release-1.13/>, Febrero 2017.
- [3] Byron P. Roe, Hai-Jun Yang, Ji Zhu, Yong Liu, Ion Stancu, and Gordon McGregor. Boosted decision trees as an alternative to artificial neural networks for particle identification. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spec-*

trometers, Detectors and Associated Equipment, 543(2):577 – 584, 2005.

- [4] Albert C. Thompson and Douglas Vaughan. X-ray data booklet, 2009.
- [5] Peter K.F. Grieder, editor. *Cosmic Rays at Earth*. Elsevier, Amsterdam, 2001.
- [6] 6by9. raspiraw. <https://github.com/6by9/raspiraw>, 2018.
- [7] A Thompson, D Attwood, E Gullikson, M Howells, KJ Kim, J Kirz, J Kortright, I Lindau, P Pianetta, A Robinson, et al. X-ray data booklet, (lawrence berkeley national laboratory, berkeley, ca, 2009). 2009.

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

A. ALTERNATIVAS A RASPIRAW

I. 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 debe ser utilizado de manera ilustrativa y cuantitativa, pero no cuantitativa.

Para verificar el funcionamiento de la cámara se expuso la cámara ante una fuente de rayos alfa. Utilizando raspivid, se tomó 1 minuto de exposición a 30 cuadros por segundos. Por pixel, se tomó el máximo valor de las 1800 imágenes, generando la imagen que se muestra en la Fig. 10

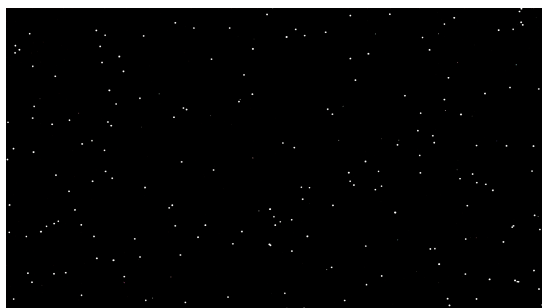


Figura 10: Imagen de los eventos con exposición de 1 minuto ante una fuente de ^{241}Am , que emite radiación alfa de 5.486 MeV. Para la composición de la misma se tomó el valor máximo por pixel de cada fotograma del video.

II. Picamera, librería en Python

La librería Picamera presenta en su documentación una forma para obtener los datos *raw*, sin procesamiento previo. Esta librería, debido a que está dedicada para la toma de fo-

tografías, permite controlar los valores de ISO, tiempo de exposición, balance de blancos.

Como primera aproximación al estudio de los datos *raw*, se analizó la dependencia de los histogramas de los valores de los píxeles. En primer lugar se estudio la dependencia con el ISO, tal como se muestra en la Fig. 11

Luego se analizó la dependencia del histograma para distintos valores de balance de blancos. Se observó que los valores de los datos *raw*, no depende del balance de blancos; lo que indica que el balance de blancos es un post-procesamiento. Tal como se muestra en la Fig. 12

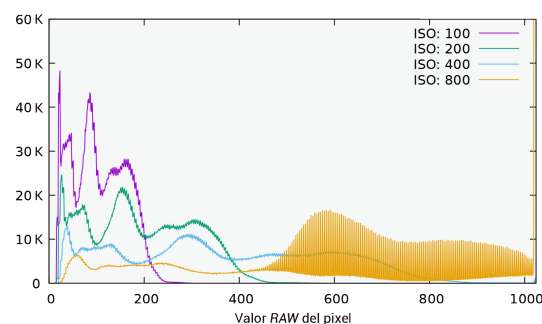


Figura 11: Vale por un caption.

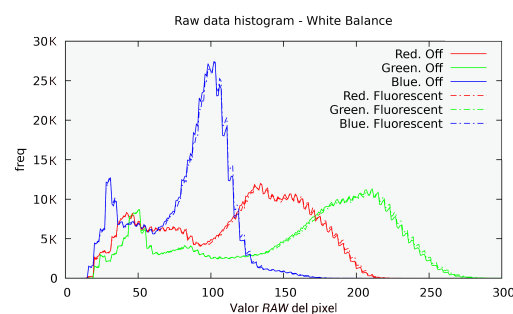


Figura 12: Vale por un caption.