Caracterización de Sensores de Imagen CMOS

Trabajo Final del Grado en Ingeniería de las Tecnologías de Telecomunicación

Manuel Lorente Almán

Sevilla, Junio 2018

Presentación

- Formando parte del equipo de Caracterización de Sensores de Imagen CMOS de Teledyne-AnaFocus desde 2015 (Sevilla, España).
- Integrante del equipo de Desarrollo y Validación de Sistemas de Visión de Teledyne-e2v durante 2017 (Grenoble, Francia).
- Desde 2018 en el grupo de Aplicaciones y Soporte Técnico al Cliente de Teledyne-AnaFocus (Sevilla, España).

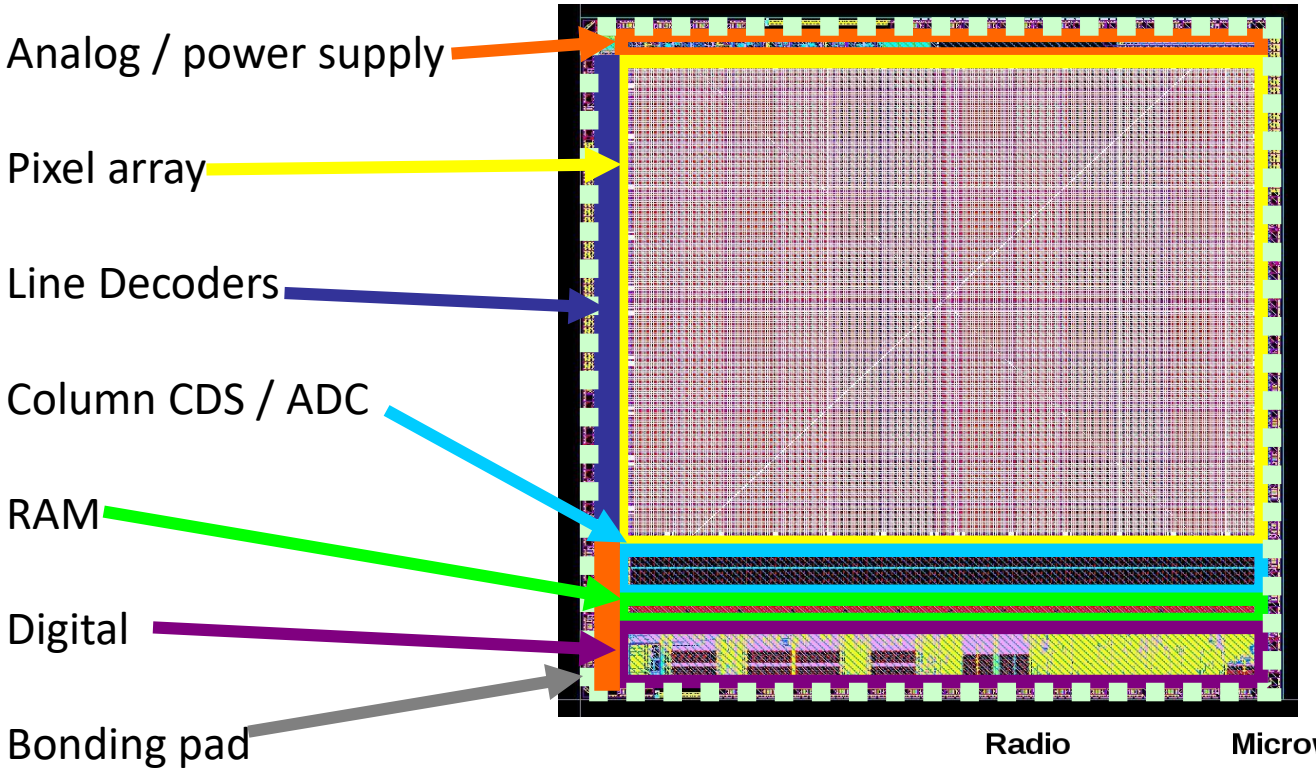
Objetivo y motivación

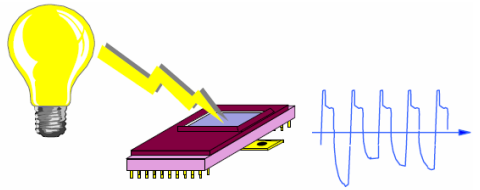
- En los próximos 20 minutos verán introducidos conceptos clave sobre Sensores de Imagen CMOS (CIS) y cómo caracterizarlos.

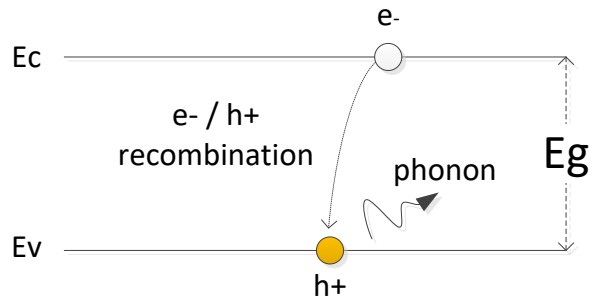
- Tras la presentación serán capaces de entender:
  - La arquitectura fundamental de los CIS.
  - Principales características electro-óptica de los CIS.
  - Caracterización de un CIS sirviéndose de la librería basada en el estándar EMVA1288 desarrollada en Python en detrimento de Matlab, empleado hasta la actualidad.

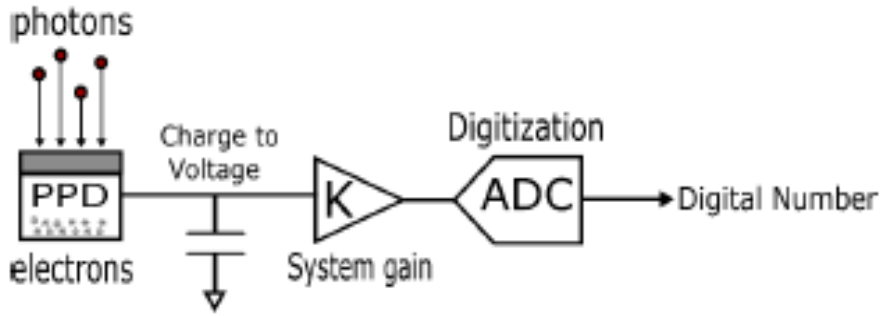
Generalidades sobre CIS

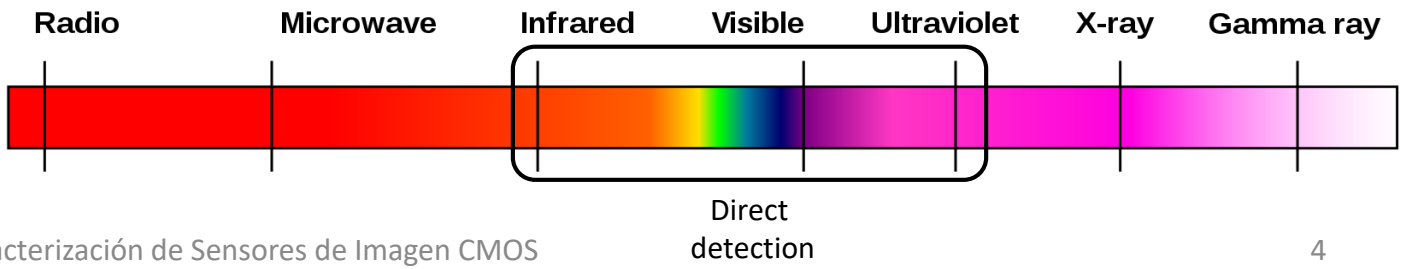
Los CIS son dispositivos semiconductores empleados en la fabricación de cámaras digitales. Detectan la información en forma de luz u otra radiación electromagnética y crea una imagen que representa dicha información basándose en el efecto fotoeléctrico.











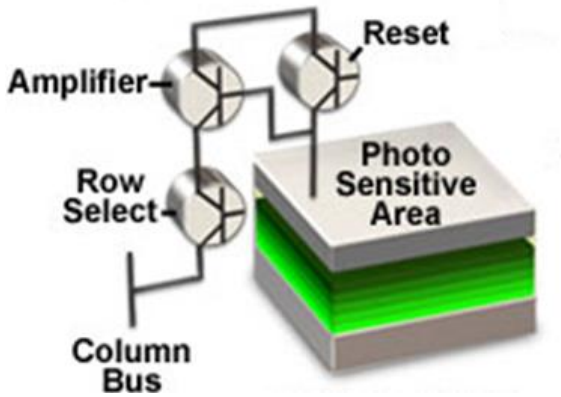
Arquitectura del píxel

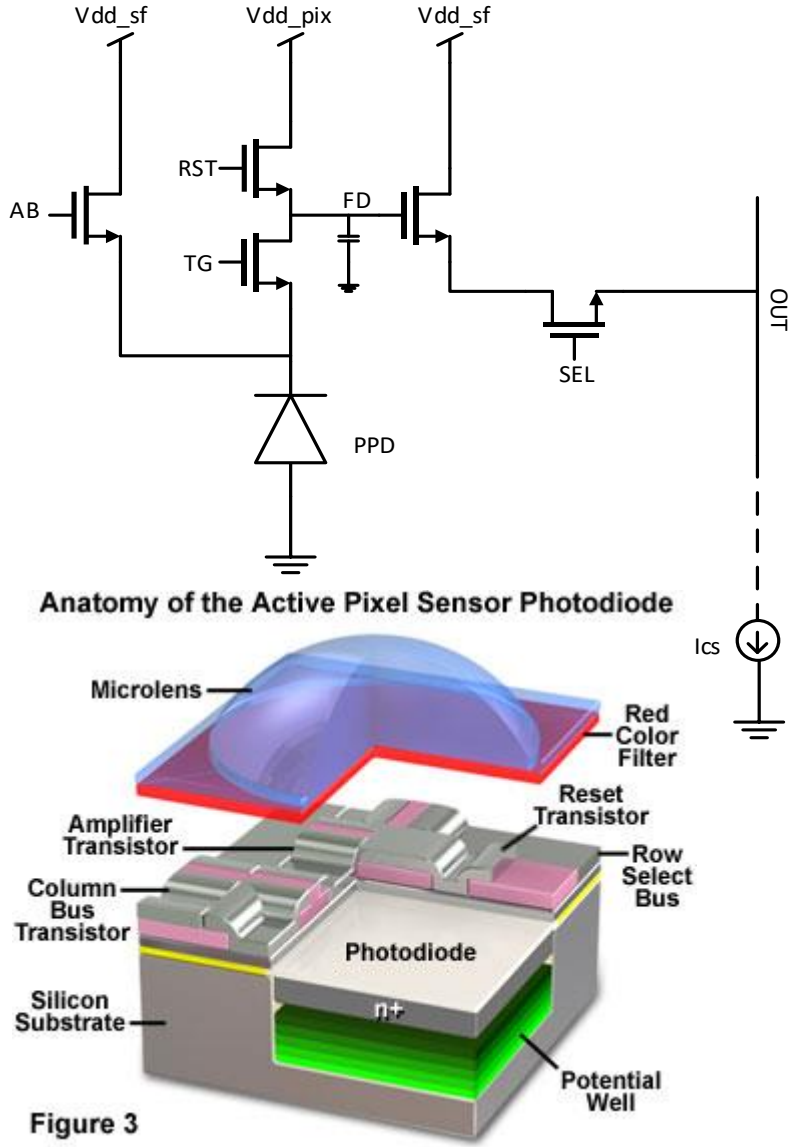
El **píxel** es la parte del CIS donde la luz es absorbida y transformada primero en **electrones** y más tarde **en tensión**.

El píxel activo estándar CMOS consiste en:

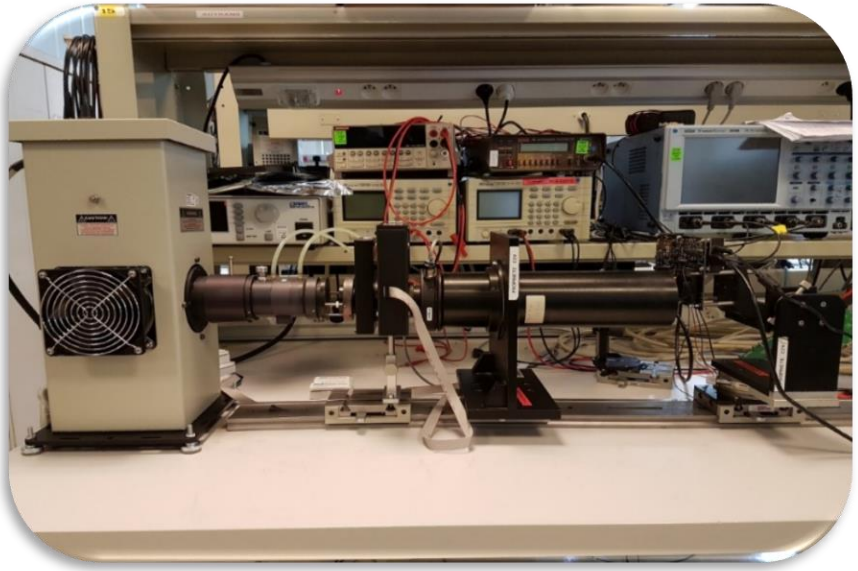
- **Dos "pozos"** donde se colectan y almacenan electrones: El **fotodiodo pinned** (PPD) y el **nodo de memoria** (también llamado *floating diffusion*, FD).
- Cuatro llaves o transistores para mover los electrones: **la *transfer gate*** (TG), el ***reset*** (RST), el *selector* (SEL) y el ***source-follower*** (SF) como transistor de lectura.

Si el píxel precisa operar en ***global-shutter*** (todos los píxeles en el sensor capturan la imagen en el mismo instante de tiempo), se precisa de un **quinto transistor** llamado *anti-blooming* (AB).

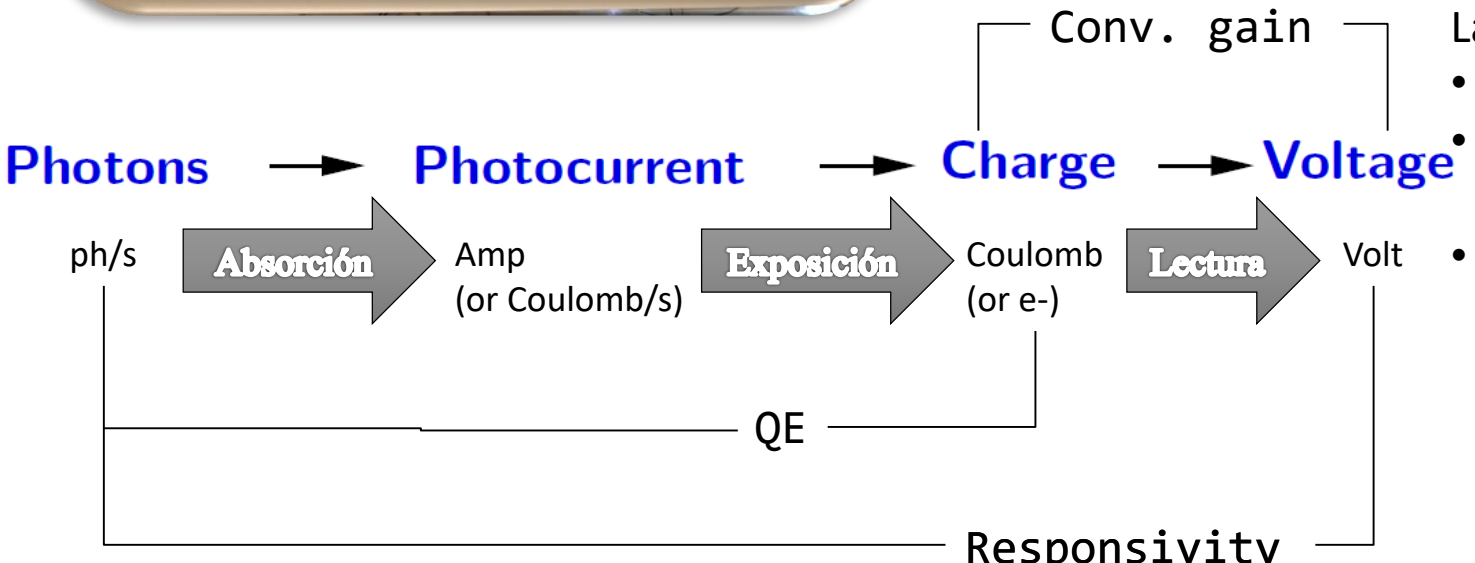




Caracterización de CIS



La $EMVA_{1288}$ es un estándar de medida desarrollado por la Asociación Europea de Visión Industrial (European Machine Vision Association) cuyo propósito es definir métodos de medida y cálculos para la caracterización electro-óptica de sensores de imagen CMOS y cámaras utilizados en aplicaciones industriales y es en el que nos hemos basado en la realización de este proyecto.



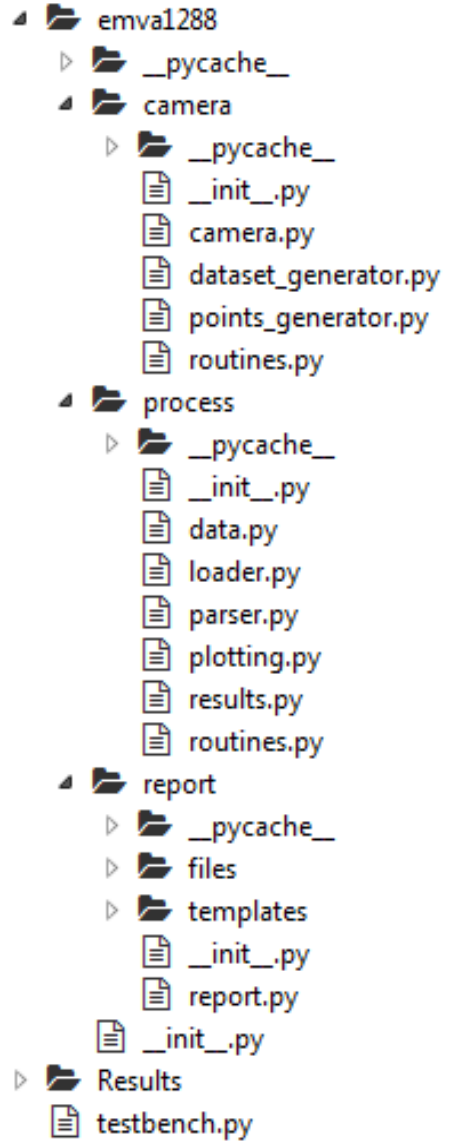
La caracterización de CIS comprende también:

- Búsqueda del punto de operación del CIS.
- Medidas electro-ópticas específicas: MTF, QE, PSRR, eficiencia del shutter, lag, blooming, etc.
- Validación funcional: secuencias internas, bloques digitales, consumo, salida LVDS, etc.

Matriz de caracterización de un CIS

	Test	Tarea	Descripción
Test funcionales (A realizar a la llegada del primer silicio)	Secuencia de inicio	*Bonding*	Comprobar que cada *pad* del anillo de *pads* está correctamente conectado según el diseño
		Power up	Comprobar integridad de las alimentaciones y las tierras
		Consumo de corriente	Medir consume en espera y en funcionamiento del sensor y comparar con valores de simulación
	Comunicaciones y programabilidad	Funcionalidad SPI	Lectura ID del chip
			Lectura STATUS del chip
			Correcta lectura/escritura de registros internos del sensor
			Lanzar instrucción START/STOP de operación del sensor y comprobar su aplicación
		LVDS	Comprobar con un patrón conocido y medida diagrama de ojo
		Reset externo	Escribir registros internos con valores conocidos no por defecto
			Aplicar un reset externo al sensor
			Comprobar que los registros contienen los valores por defecto
	Bloques de señal mixta	Power on Reset	Chequear reset al inicio del sistema
		Referencias internas	Comprobar referencias internas de tensión de los bloques digitales y analógicos
	Circuitería de lectura	Funcionalidad	Leer fila de test con y sin iluminación para medida de ruido
Medidas de caracterización (A realizar para cada nueva versión o configuración)	Performance óptica		DR, SNR, DSNU, PRNU, VFPN, HFPN, Ruido Temporal, NEE, CG, Corriente Oscura, Respuesta, QE
	Consumos	Medida corrientes	Medida de consumo para los modos más depurados y demandados

Librería Python EMVA1288

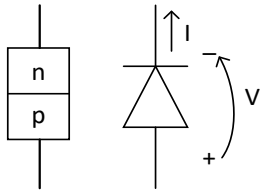


Script	Descripción
camera.camera.py	Contiene las clases y rutinas necesarias para simular el comportamiento de un CIS a partir de unos parámetros típicos propios del sensor.
camera.dataset_generator.py	Permite crear un fichero descriptor y sus correspondientes imágenes siguiendo los parámetros generados por camera.py.
camera.points_generator.py	Contiene las clases y rutinas precisas para poder crear vectores de puntos para parámetros como la exposición o la irradiancia.
camera.routines.py	Rutinas para calcular QE, irradiancia y demás parámetros útiles haciendo uso de las ecuaciones matemáticas descritas en la memoria.
process.data.py	Clase que permite leer del fichero descriptor generado, cargar las imágenes indicadas y crea un objeto de datos para ser procesado según la EMVA1288.
process.loader.py	Clase que obtiene la información relevante de una imagen para el cálculo de performances.
process.parser.py	Clase que traduce la información del fichero descriptor a información útil para la carga de imágenes.
process.plotting.py	Clase que a partir de los resultados del procesamiento dibuja las gráficas necesarias para el reporte final.
process.results.py	Clase que a partir de los datos procesados realiza todos los cálculos siguiendo el estándar EMVA1288 como se ha descrito a lo largo del proyecto.
process.routines.py	Funciones útiles a la hora de calcular ajustes lineales, formas de las imágenes, FFT, etc.
report.report.py	Conjunto de funciones que a partir de los resultados y las gráficas generadas crea un reporte automático en LaTeX.
testbench.py	Script de pruebas utilizado para testar todas las funciones y realización de las simulaciones propuestas.

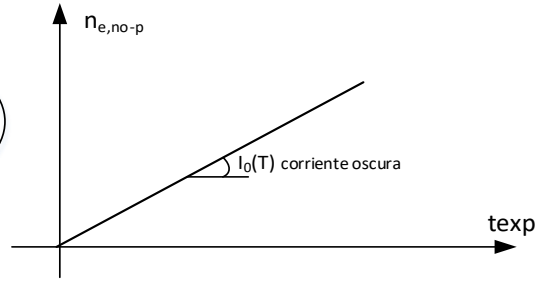
Caracterización Electro-Óptica

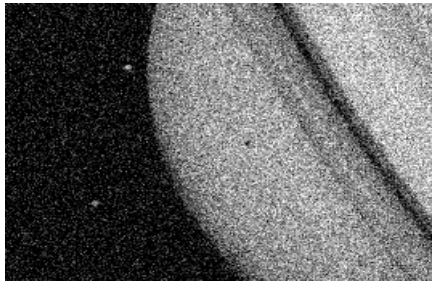
Corriente Oscura

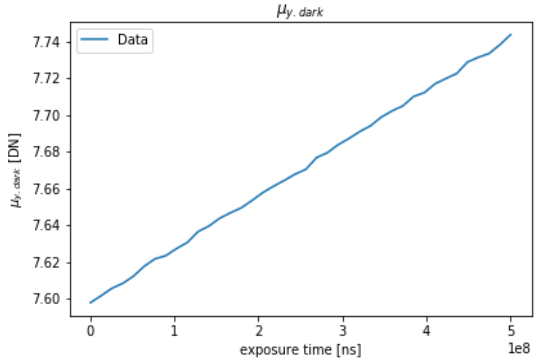
Electrones no-fotogenerados en una unión p-n inversamente polarizada.



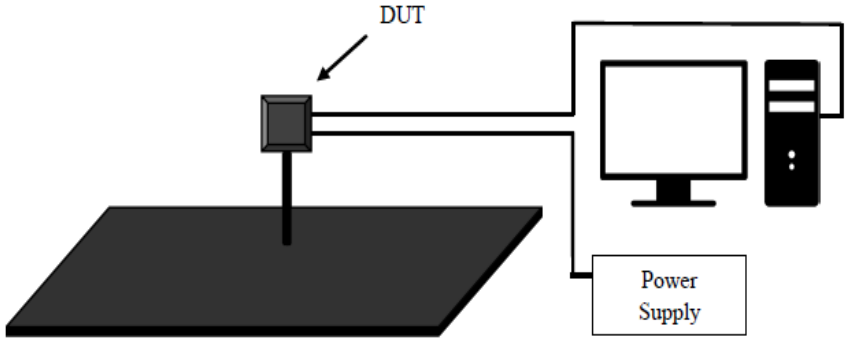
Debido a la temperatura, hay electrones en el fotodiodo que adquieren la suficiente energía para "saltar" de la banda de valencia a la banda de "conducción".







Corriente oscura - Método de medida



1. Tomar una imagen con un **corto** periodo de integración (t_{int1})
2. Tomar una imagen con un **largo** periodo de integración (t_{int2}) al menos 100 o 1000 veces mayor que t_{int1} para que la corriente oscura sea significativa.
3. Capturar una tercera imagen *dummy* con algo más de tiempo de integración que t_{int2} con el fin de comprobar que las imágenes 1 y 2 se encuentran en una **región lineal.**
4. Cálculo de la corriente oscura como la **pendiente**.

Adquisición de imagen vía Sapera CamExpert

'EMVA1288descriptor.txt'
n 12 640 480
d 50.0
i images\image1.tiff
d 50000.0
i images\image2.tiff

Análisis de datos según formato

```
from emva1288.process.parser import ParseEmvaDescriptorFile
from emva1288.process.results import Results1288
from emva1288.process.plotting import Plotting1288
parser = ParseEmvaDescriptorFile('EMVA1288descriptor.txt')
imgs = LoadImageData(parser.images)
data = Data1288(imgs.data)
```

Cálculo y representación de corriente oscura del CIS

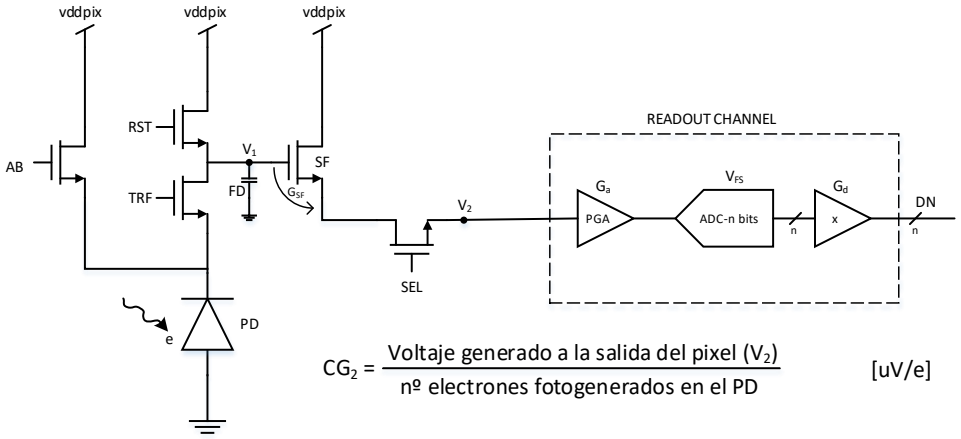
```
EOperf= Results1288(data.data)
DS = Plotting1288(EOperf.u_I_mean_DN)
DS.plot()
```

$$DS\left[\frac{e^-}{s}\right] = \frac{\mu_{image2} - \mu_{image1}}{t_{int2} - t_{int1}}$$

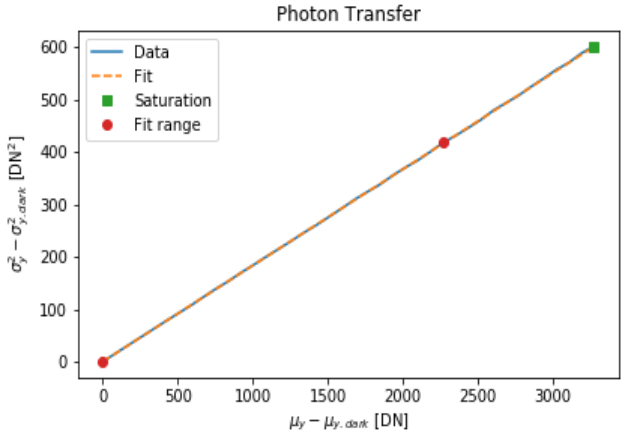
Caracterización Electro-Óptica

Conversion Gain

Hasta 3 definiciones



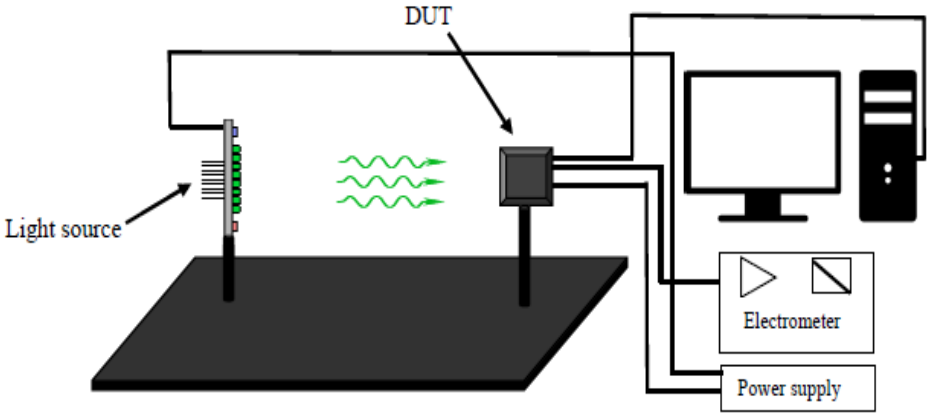
$$CG_2 = \frac{\text{Voltaje generado a la salida del pixel } (V_2)}{\text{nº electrones fotogenerados en el PD}} \quad [uV/e]$$



1. Calibrar y controlar la corriente de la fuente de LED con el fin de **barrer la luz** desde condiciones de oscuridad a saturación.
2. Para cada paso de iluminación **tomar al menos 2 imágenes** y calcular la media en [V] a la vez que se mide la irradiancia E [W/cm^2]
3. Tomar una imagen en oscuridad para obtener el offset de referencia (μ_{dark}).
4. Trazar la curva de varianza entre (σ^2) y (μ - μ_{dark}) y la **pendiente** entre el 0% y 70% de esta curva (PTC) es la ganancia de conversión en [DN/e$^-$].

$$CG\left[\frac{DN}{e^-}\right] = \frac{\sigma^2{}_{70\%} - \sigma^2{}_{dark}}{\mu_{70\%} - \mu_{dark}}$$

Ganancia de Conversión - Método de medida



Adquisición de imagen vía Sapera CamExpert

'EMVA1288descriptor.txt'

n 12 640 480

b 100000.0 7.167

i images\image0.tiff

d 100000.0

i images\image1.tiff

…

…

b 474364102.6 33997.451

i images\image148.tiff

d 474364102.6

i images\image149.tiff

Análisis de datos según formato

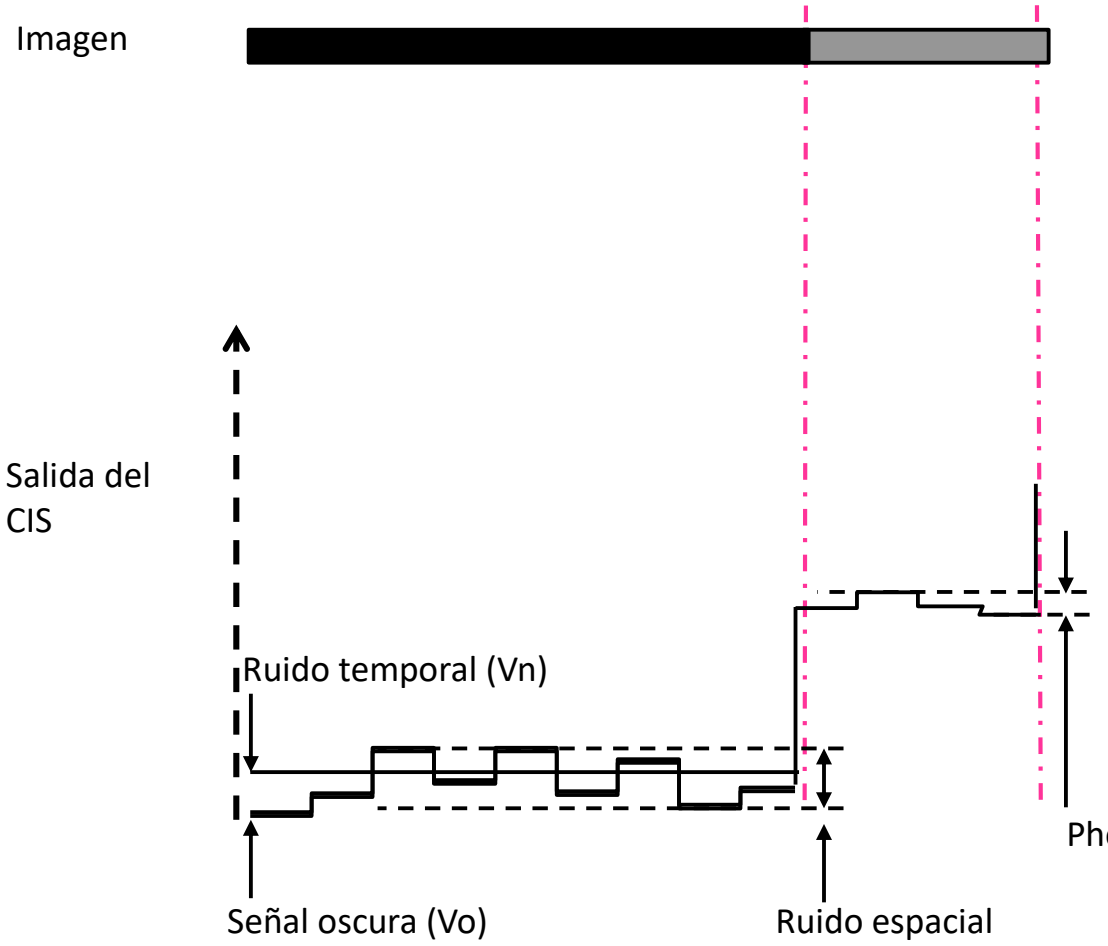
```
from emva1288.process.parser import ParseEmvaDescriptorFile
from emva1288.process.results import Results1288
from emva1288.process.plotting import Plotting1288
parser = ParseEmvaDescriptorFile('EMVA1288descriptor.txt')
imgs = LoadImageData(parser.images)
data = Data1288(imgs.data)
```

Cálculo y representación de ganancia de conversión del sistema

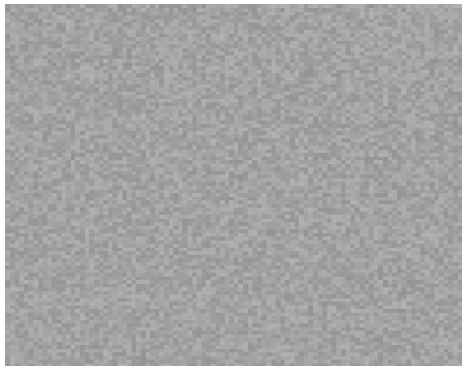
```
EOperf= Results1288(data.data)
CG = Plotting1288(EOperf.K)
CG.plot()
```

Caracterización Electro-Óptica

Ruidos

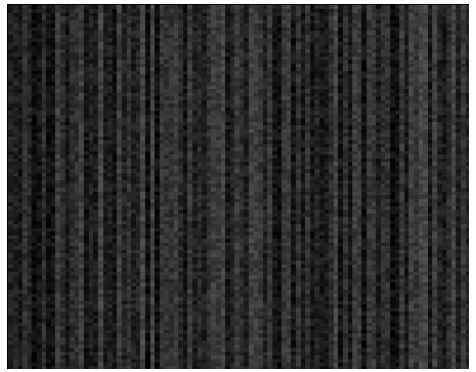


- - Dark Signal non Uniformity (DSNU %)
- - Fixed Pattern Noise (FPN)



Ruido fijo en el píxel (DSNU, PRNU):

- No correlado
- Puede ser corregido sustrayendo imagen en oscuridad (DSNU) y con corrección de ganancia (PRNU)

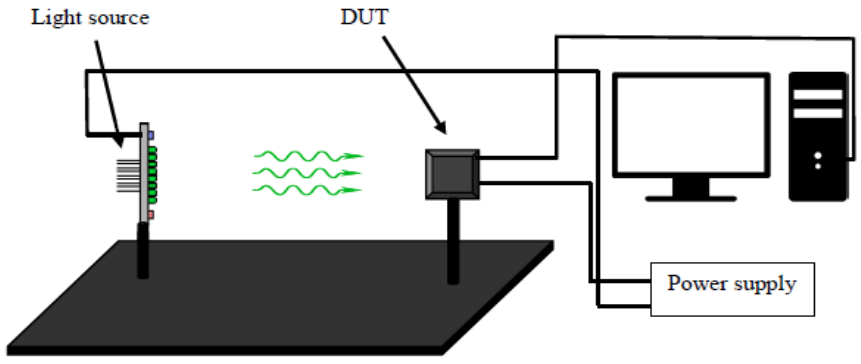


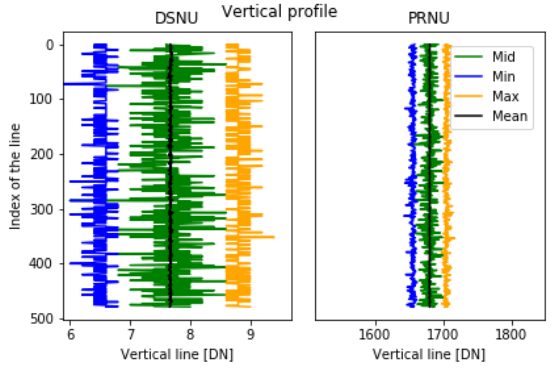
Ruido de patrón fijo(FPN):

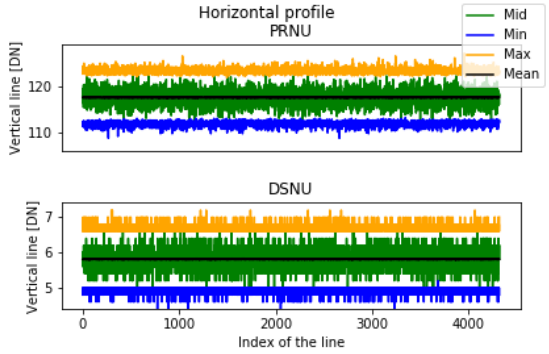
- Correlado o por fila o por columna
- Visible frente al ruido temporal
- Puede ser corregido *on-chip* utilizando filas y columnas del CIS

Caracterización Electro-Óptica

Ruidos - Método de medida







1. Tomar capturas dummy (al menos 20 o 30) para que el sistema sea **estable**.
2. Tomar al menos 10 o 20 imágenes para los cálculos de manera que promediando se eliminen errores de offset o de patrón fijo y con un **corto periodo de integración** para que la influencia del DSNU sea mínima en oscuridad.
3. Encender la fuente de **luz** y esperar unos 10 minutos hasta que sea **térmicamente estable**.
4. Capturar al menos 10 o 20 imágenes al 50% del nivel de saturación y en saturación.

$$Ruido_{temporal}[V\ o\ DN] = \sqrt{\sigma}$$

$$DSNU[\%] = \frac{\sigma_{dark}}{\mu_{dark}} x\ 100$$

$$PRNU[\%] = \frac{\sigma_{(light-dark)}}{\mu_{(light-dark)}} x\ 100$$

Adquisición de imagen vía Sapera CamExpert

'EMVA1288descriptor.txt'

n 12 640 480

b 100000.0 7.167

i images\image0.tiff

d 100000.0

i images\image1.tiff

…

…

b 474364102.6 33997.451

i images\image148.tiff

d 474364102.6

i images\image149.tiff

Análisis de datos según formato

from emva1288.process.parser import ParseEmvaDescriptorFile

from emva1288.process.results import Results1288

from emva1288.process.plotting import Plotting1288

parser = ParseEmvaDescriptorFile('EMVA1288descriptor.txt')

imgs = LoadImageData(parser.images)

data = Data1288(imgs.data)

Cálculo y representación de ruidos

EOperf= Results1288(data.data)

*temp = Plotting1288(EOperf.*sigma_y_dark*)*

*prnu = Plotting1288(EOperf.*PRNU1288*)*

*dsnu = Plotting1288(EOperf.*DSNU1288_DN*)*

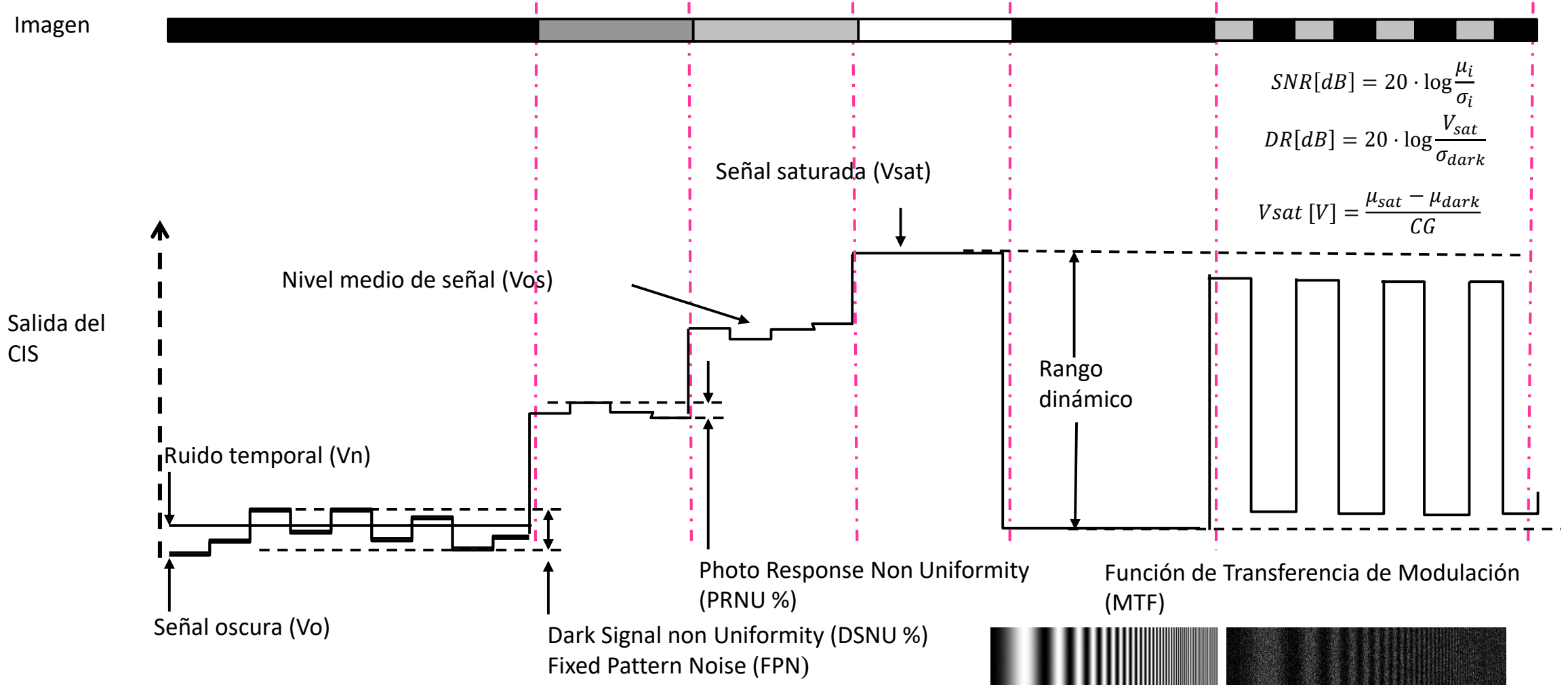
temp.plot()

prnu.plot()

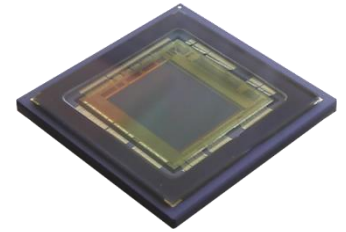
dsnu.plot()

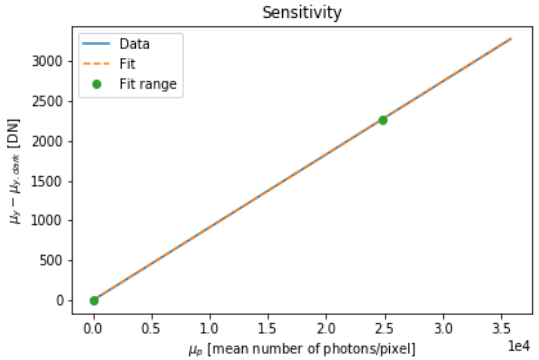
Caracterización Electro-Óptica

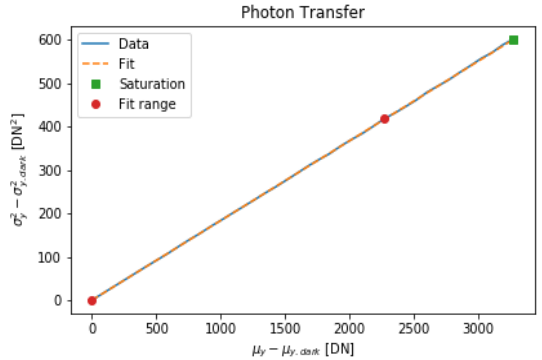
Otros parámetros importantes

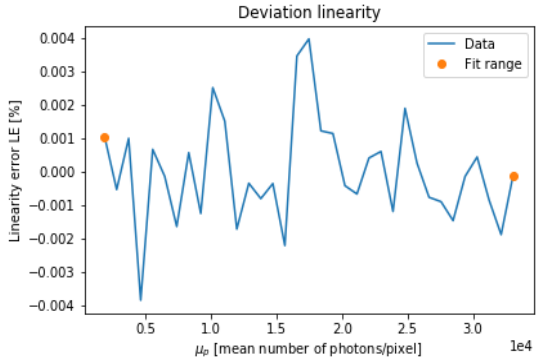


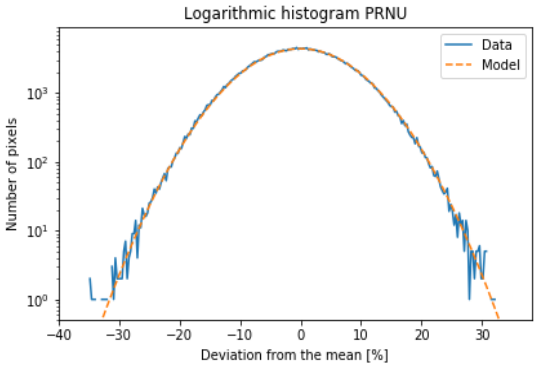
Caracterización Electro-Óptica Lince5M

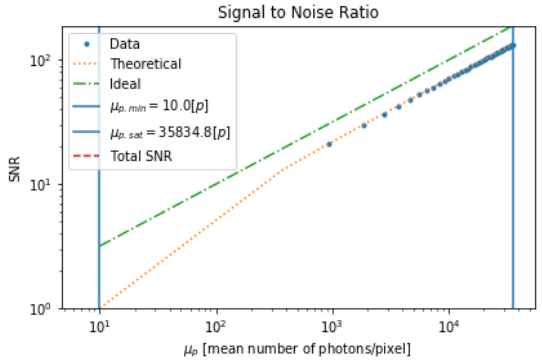


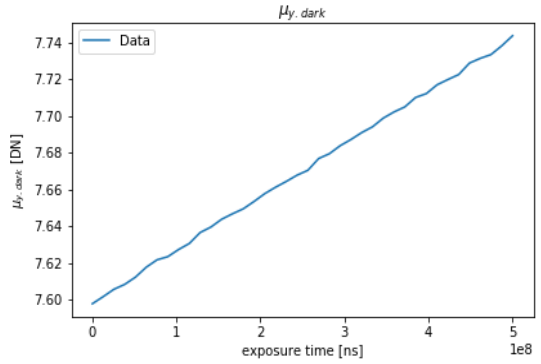


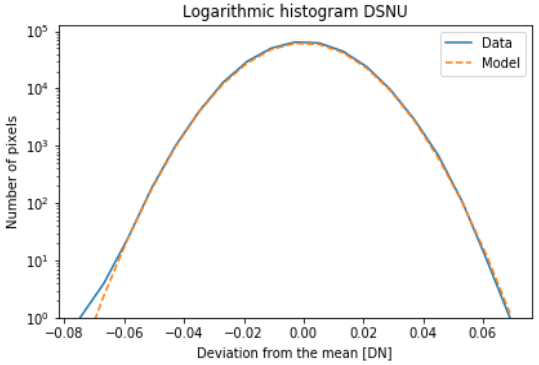


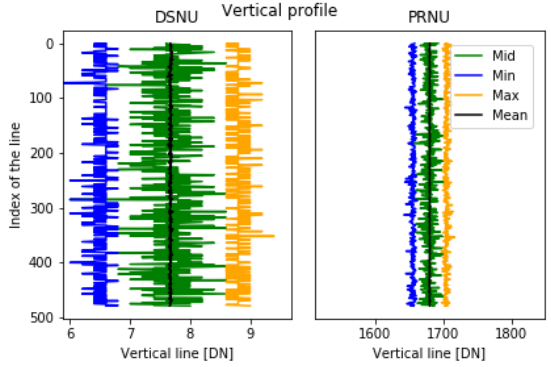


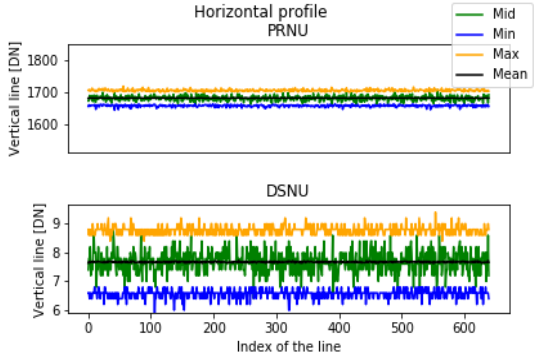












	Lince5M
Sensor type	Area
Resolution [pixels]	640x480
Data output [bits]	12
Pixel size [µm]	5x5
Frame-rate	120fps
Interface type	GigE

	Lince5M
QE [%]	49.86
CG [DN/e$^-$]	0.183
$\sigma_{y,dark}$ [e$^-$]	4.182
SNR_{max} [dB]	42.52
$\mu_{e,min}$ [e$^-$]	4.968
$\mu_{e,sat}$ [e$^-$]	17867
DR [dB]	71.1
DSNU [%]	< 0.1
PRNU [%]	<0.1
Linearity error [%]	0.004
Dark current [e$^-$/s]	1.573

Conclusiones

- Necesidad para el uso de la librería desarrollada, el formateo de las imágenes según se indica en el fichero descriptor EMVA1288.

- Para medir las características de un CIS y obtener información sobre la linealidad, valor de ganancia de conversión, rango dinámico, etc. Idealmente es independiente barrer el tiempo de exposición manteniendo la luz constante que barrer la luz con una fuente de luz lineal manteniendo el tiempo de exposición constante.

- Gracias a la librería desarrollada se ha logrado de migrar desde Matlab a Python para el análisis de imágenes en los grupos de Aplicaciones y Soporte al Cliente tanto en Teledyne-AnaFocus (Sevilla, España) como el equipo de Teledyne-e2v (Grenoble, Francia).

Líneas futuras de trabajo

- Creación de una GUI (Graphic User Interface) que encapsule esta librería.

- Añadir nuevas medidas para CIS fuera del estándar EMVA1288 como podrían ser las orientadas a sensores 3D basados en el concepto *Time Of Flight*.

- Integrar la adquisición de imágenes, el control del sensor y el procesamiento. Implementar la adquisición de imágenes a través de Python de interfaces como GigE, CameraLink o USB.

Referencias

- Arnaud Camboly, "Characterization of CIS", *FillFactory internal*, 2004
- Francis Pang, "CIS characterization Handbook", *e2v internal*, 2007
- Rafael Dominguez, Fernando Medeiro, "Image sensor measurements based in EMVA Standard 1288", *AnaFocus internal*, 2015
- Arnaud Camboly, "EMVA1288 VS AF methodology review", *AnaFocus internal*, 2016
- Rafael Domínguez, Fernando Medeiro, "CMOS pixel architecture", *AnaFocus internal*, 2015
- EMVA Standard 1288, "Standard for Characterization of Image Sensors and Cameras", Release 3.1, November 2012
- Junichi Nakamura, "Image Sensors and Signal Processing for Digital Still Cameras", 2006
- Federico Ariza, “Python library to create LaTeX files”, https://github.com/fariza/PyLaTeX

Gracias por la atención