

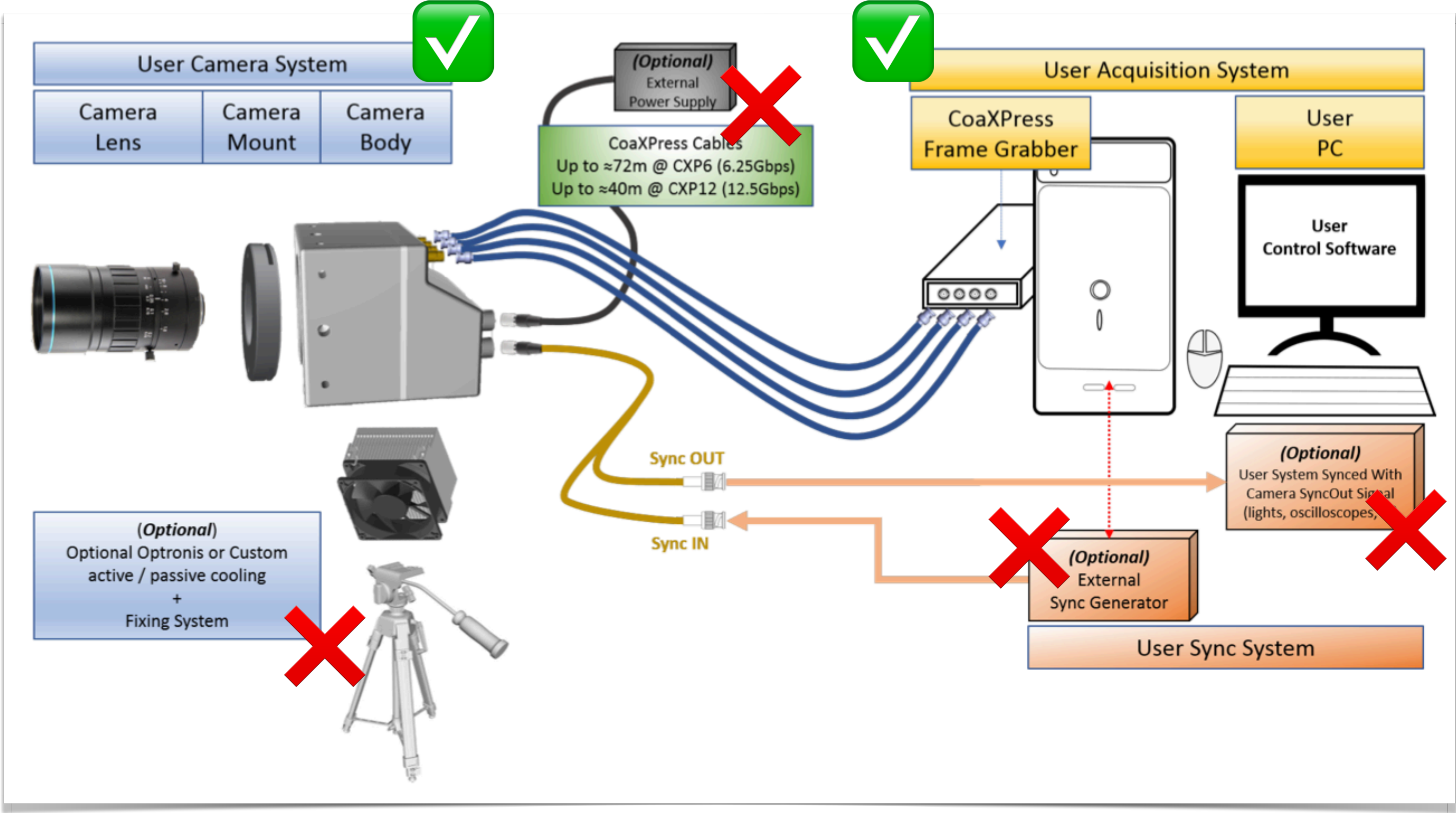
Studio del noise per la camera Cyclone-2-2000M



Samuele Lanzi

12/02/2025

Setup



eGrabber Studio

acA1920-25gc

Buffer 9

System

Interface

Device

Remote Device

Data Stream

Root

DeviceInformation

DeviceID

DeviceVendorName

DeviceModelName

DeviceAccessStatus

DeviceType

DeviceDescription

StreamEnumeration

StreamSelector

StreamID

Device-00-30-53-30-40-d7

Basler

acA1920-25gc

OpenReadWrite

GEV

00:30:53:30:40:d7

0

Stream0

AcquisitionStop

AcquisitionFrameCount

TriggerSelector

TriggerMode

TriggerSoftware

TriggerSource

TriggerActivation

TriggerDelayAbs

ExposureMode

InterlacedIntegrationMode

ExposureAuto

ExposureTimeAbs

ExposureTimeRaw

ReadoutTimeAbs

ExposureOverlapTimeMaxAbs

ExposureOverlapTimeMaxRaw

Execute

1

FrameStart

Off

Execute

Line1

RisingEdge

0,000

Timed

NI

Off

6055,000

6055

38290,000

—

WO

RO

RW

RW

WO

RW

RW

RW

NI

RW

RW

RO

NA

NI

Root

StreamInformation

StreamID

StreamType

StreamDescription

ImageFormatControl

PixelFormat

PixelFormatNamespace

PixelFormatSize

PixelFormatComponentCount

Width

Height

TransportLayerControl

PayloadSize

ControlRemoteDevice

BufferHandlingControl

Stream0

GEV

NA

BayerGB8

PFNC_32BIT

8

1

1920

1080

2073600

False

RO

RO

NA

RO

RO

RO

RO

RO

RO

RO

RO

RO

NA

RW

REC

Gives the device's access status at the moment of the last execution of DeviceUpdateList.

This float value sets the camera's exposure time in microseconds.

Defines whether or not to start and stop the remote device by automatically executing AcquisitionStart and AcquisitionStop commands.

```
#include <iostream>
#include <EGenTL.h>

void listCards() {
    Euresys::EGenTL gentl;
    GenTL::TL_HANDLE tl = gentl.tlOpen();
    uint32_t numCards = gentl.tlGetNumInterfaces(tl);
    for (uint32_t n = 0; n < numCards; ++n) {
        std::string id = gentl.tlGetInterfaceID(tl, n);
        std::cout << "[" << n << "]" << id << std::endl;
    }
}

int main() {
    try {
        listCards();
    } catch (const std::exception &e) {
        std::cout << "error: " << e.what() << std::endl;
    }
}
```


Specifiche

PERFORMANCE (EXAMPLES, 8 BIT)

Resolution (HxV)	Frame rate	Resolution (HxV)	Frame rate
1 920 x 1 080	2 166 fps	1 920 x 960	2 435 fps
1 280 x 960	2 587 fps	1 920 x 480	4 840 fps
640 x 480	5 143 fps	1 920 x 120	18 667 fps
512 x 384	6 409 fps	1 920 x 32	61 842 fps
256 x 8	177 935 fps	1 920 x 8	167 504 fps

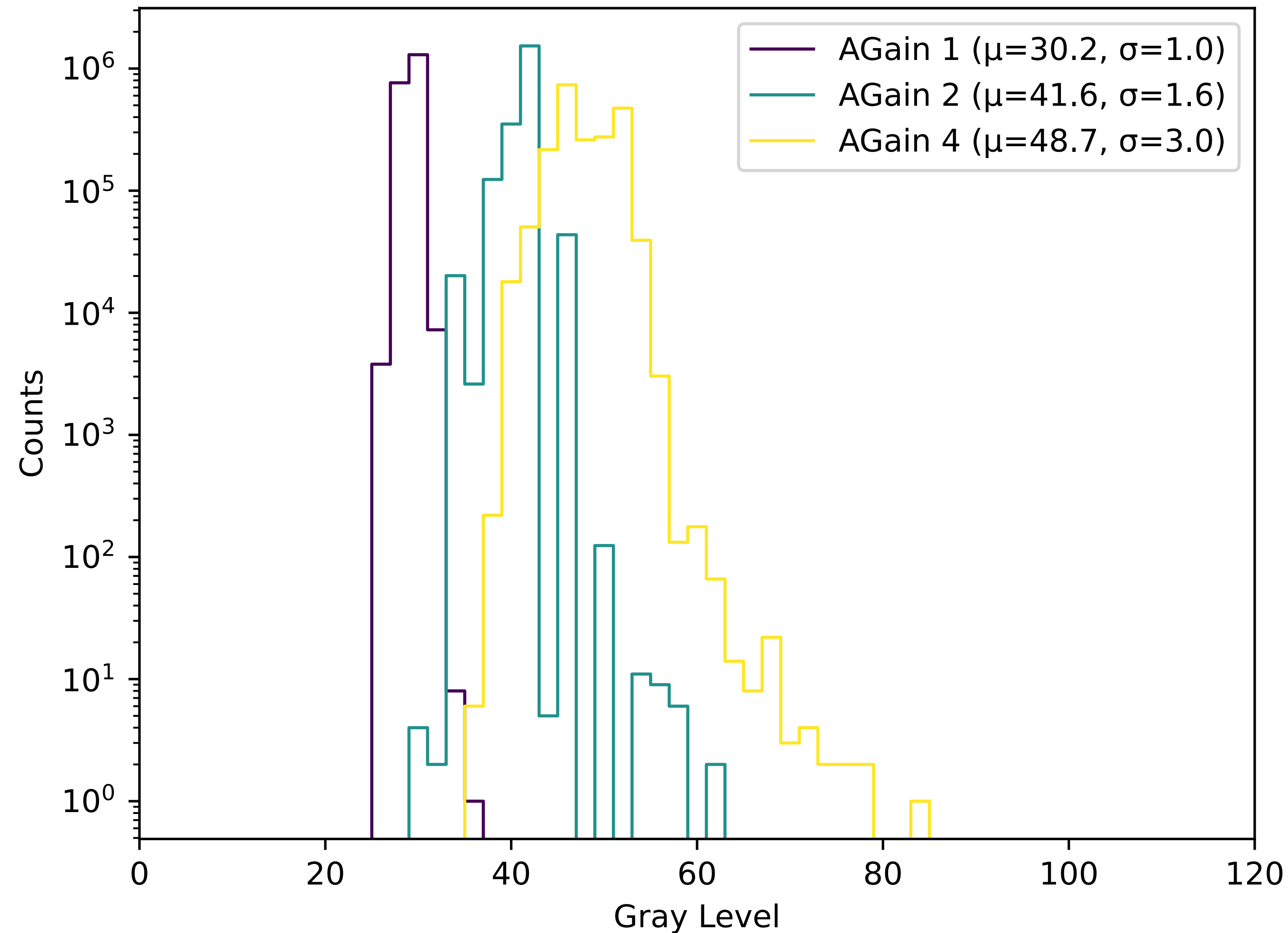
EMVA1288 measurements (v3.1 typ. preliminary)

Dynamic range	TBD dB
Saturation capacity	15 000 e ⁻
Temporal dark noise	10 e ⁻
System gain	15 e ⁻ /DN
Signal-to-Noise Ratio	50 dB

SPECIFICATIONS

Sensor type	LUX19HS, Global Shutter
Resolution	1 920 pixel x 1 080 pixel
Frame rate (full resolution)	20 .. 2 166 fps
Exposure time	4 μs .. 1 / frame rate
Active area / Diagonal	19.20 mm x 10.80 mm / 22.03 mm
Pixel distance	10.0 μm x 10.0 μm
A/D conversion	8/10 Bit
Sensitivity (sensor)	20 V/lux·s @ 550 nm
Quantum efficiency (sensor)	>45 % @ 550 nm
Synchronization	internal, external, CoaXPress
Synchronization signals	Synch IN and Synch OUT, TTL level, electrically isolated
Interface	CoaXPress 2.0 (4 ch., CXP-12) Micro-BNC (HD-BNC) connectors
Power	12 W, PoCXP, ext.
Temperature Ranges	
- operation, amb., no cooling	0 .. +30°C / 32 .. 86°F
- operation, amb., with CY-HIS	0 .. +40°C / 32 .. 104°F
- operation, amb., with CY-FAN	0 .. +55°C / 32 .. 131°F
- operation, case temperature	0 .. +70°C / 32 .. 158°F

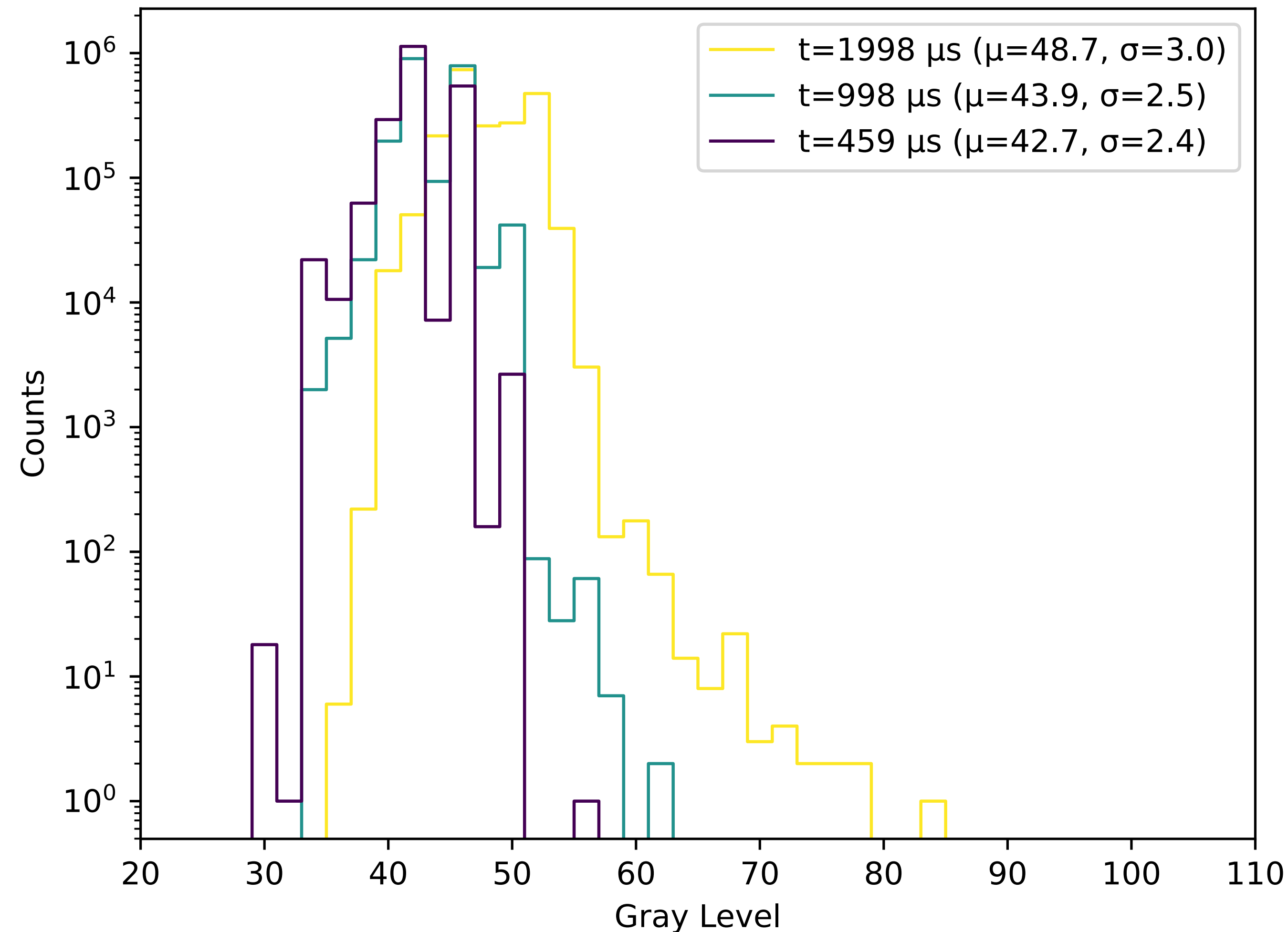
Analisi preliminare - Gray level vs gain



Camera a frame-rate
fisso e variando il gain
nei suoi possibili valori

Media e deviazione
standard crescono
all'aumentare del gain

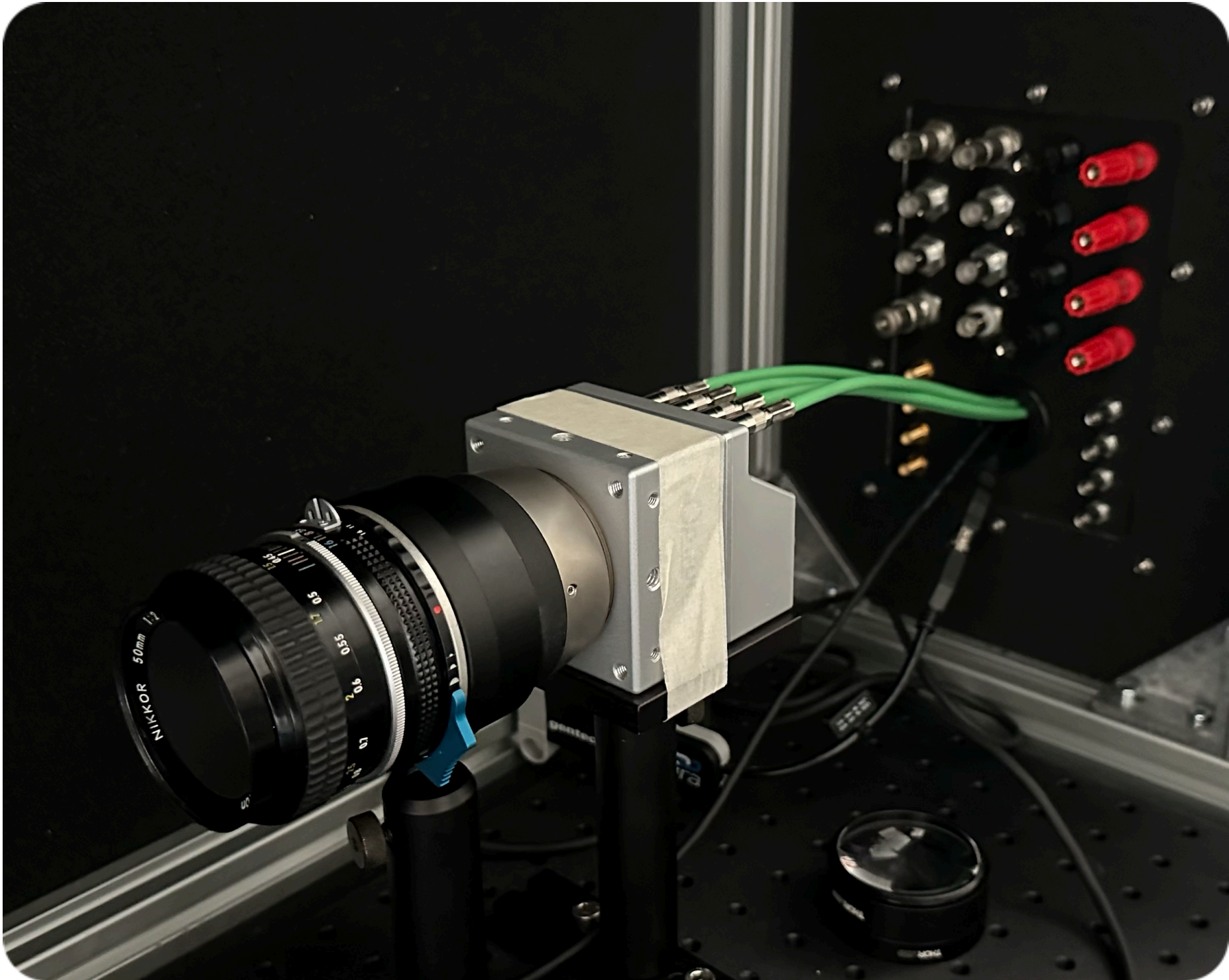
Analisi preliminare - Gray level vs esposizione



Camera a gain
massimo variando il
tempo di esposizione

Media e deviazione
standard crescono
all'aumentare del
tempo di esposizione

Camera bias



- 1. Camera tappata
- 2. Frame rate 2166 Hz
- 3. 100 frame
- 4. Stack delle immagini
- 5. GL medio dello stack



	Mean Bias (GL)	Std (GL)
AGain 1	30,716	0,009
AGain 2	39,824	0,015
AGain 4	42,708	0,017

Camera gain

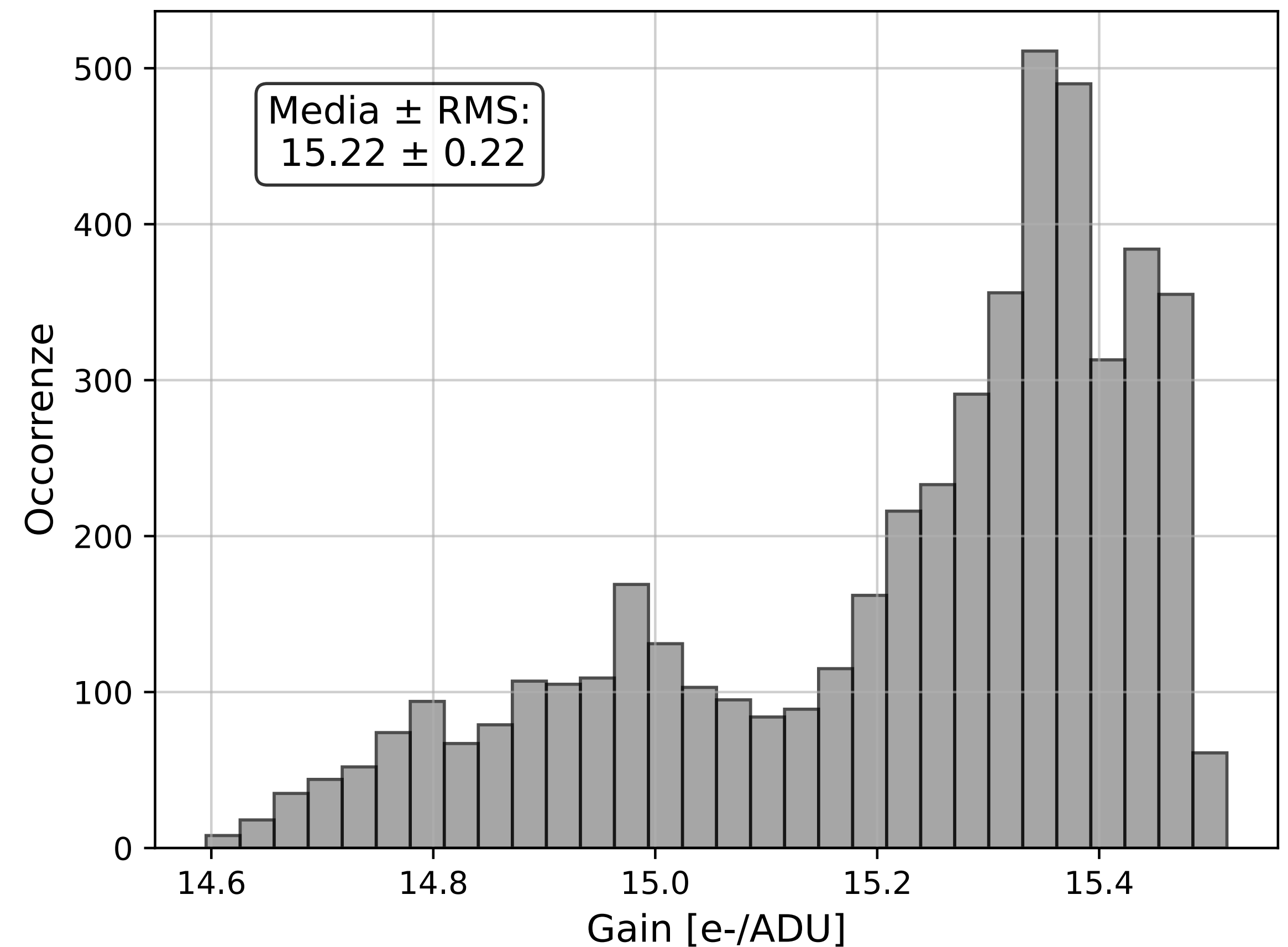
1. 100 frame bias
2. 2 light frame ad esposizione 5 ms
3. Media tra le medie dei light frame $\mu_{1,2}$
4. Differenza tra i light frame
5. Calcolo deviazione standard della differenza σ_{diff}
6. Calcolo della varianza

$$V = \sigma_{diff}^2 / 2$$

7. Calcolo il gain

$$G = (\mu_{1,2} - \text{bias}) / V$$

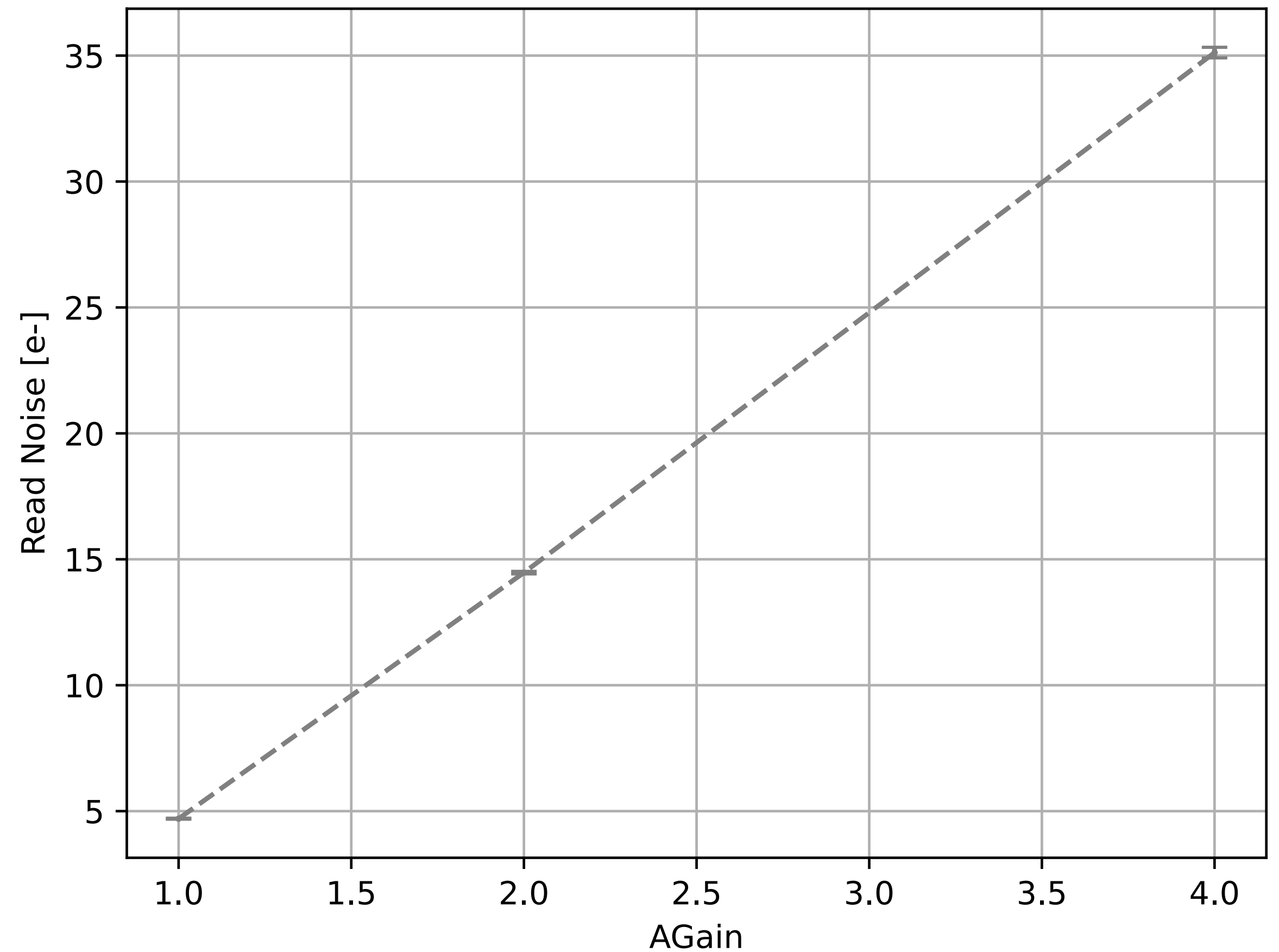
Cambiando ad ogni iterazione le coppie di immagini



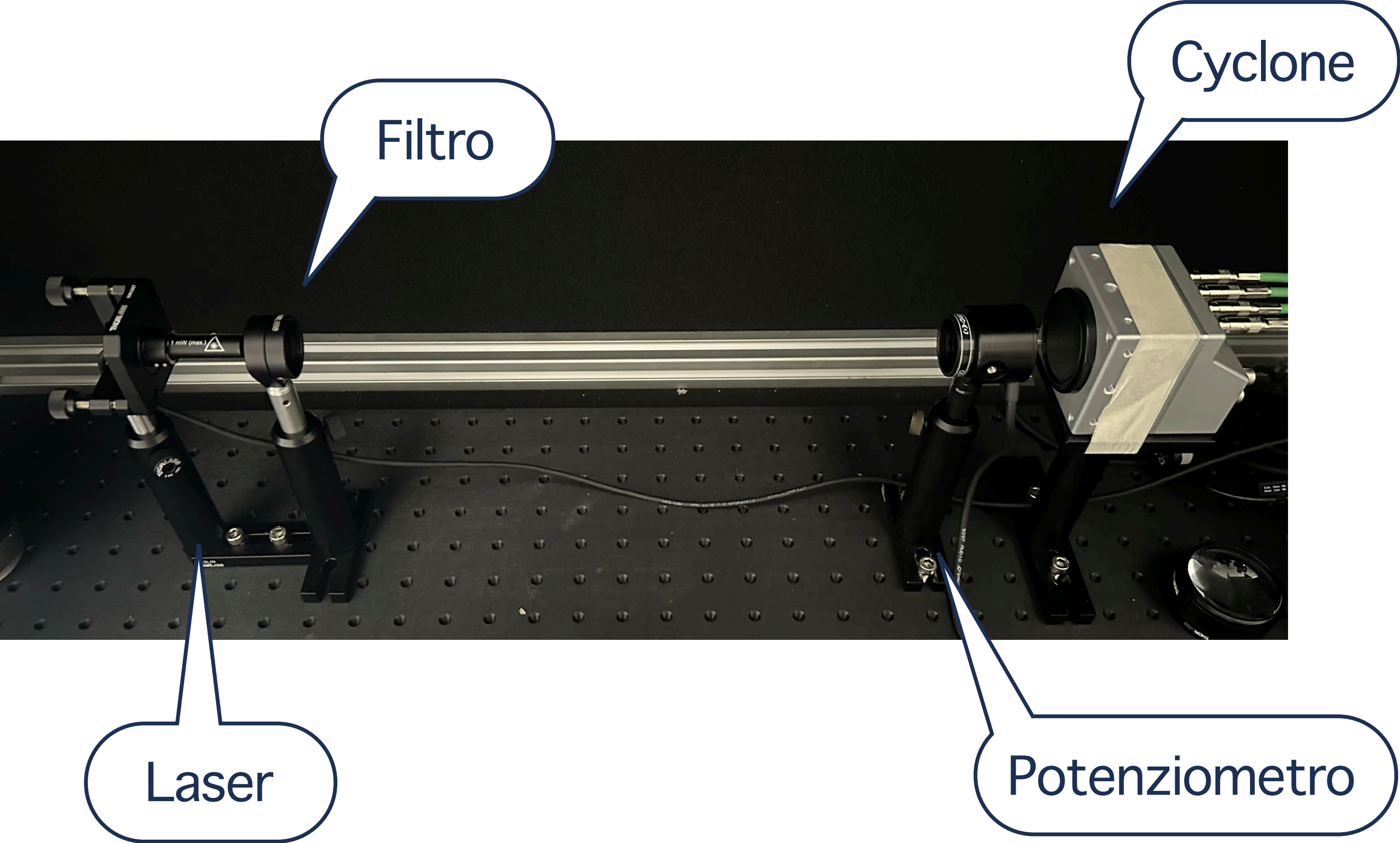
Camera read noise

1. 2 bias
2. Differenza tra i due bias
3. Calcolo deviazione standard della differenza σ_{diff}
4. Calcolo del read noise

$$RN = (\sigma_{diff}G)/\sqrt{2}$$



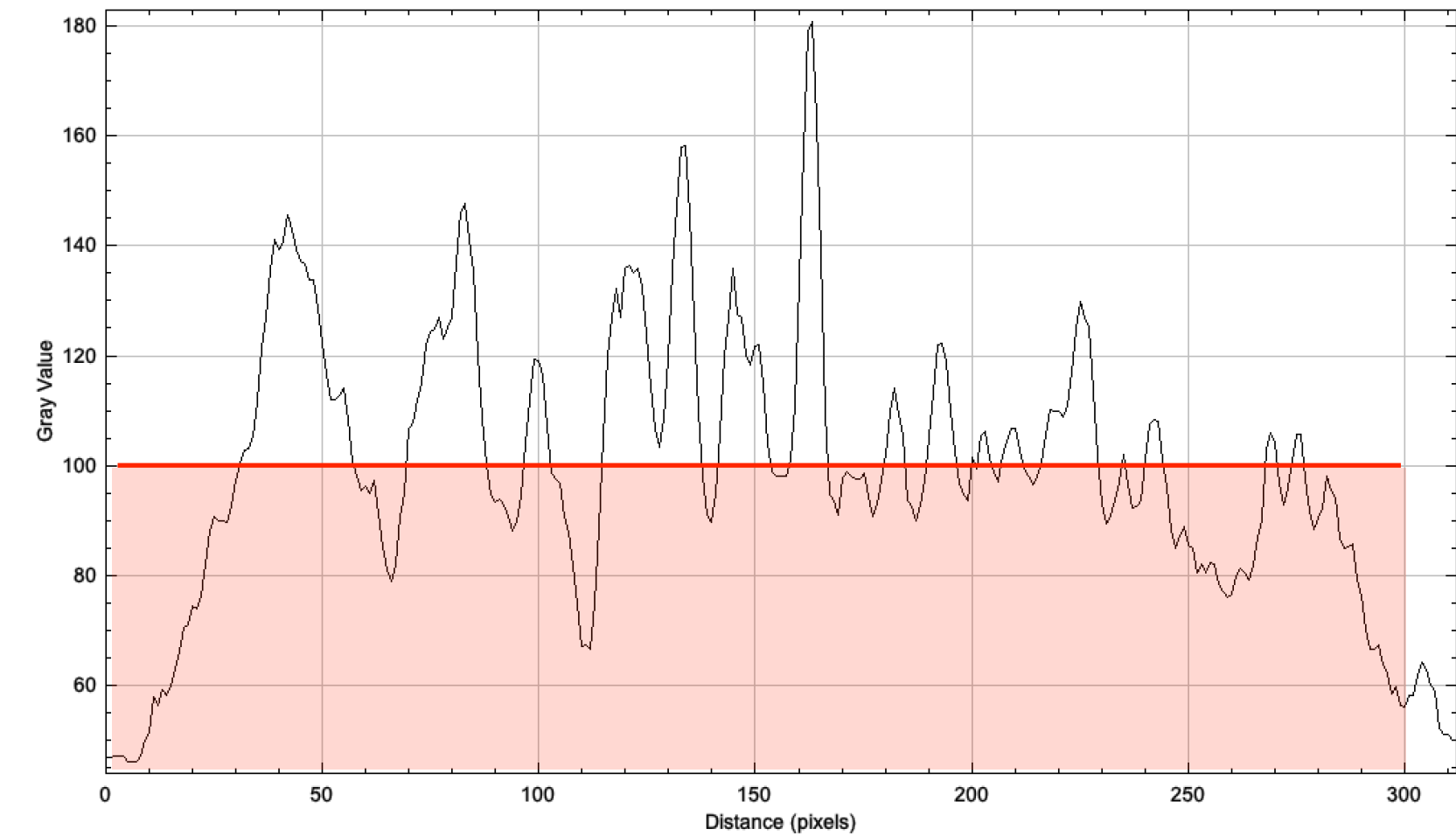
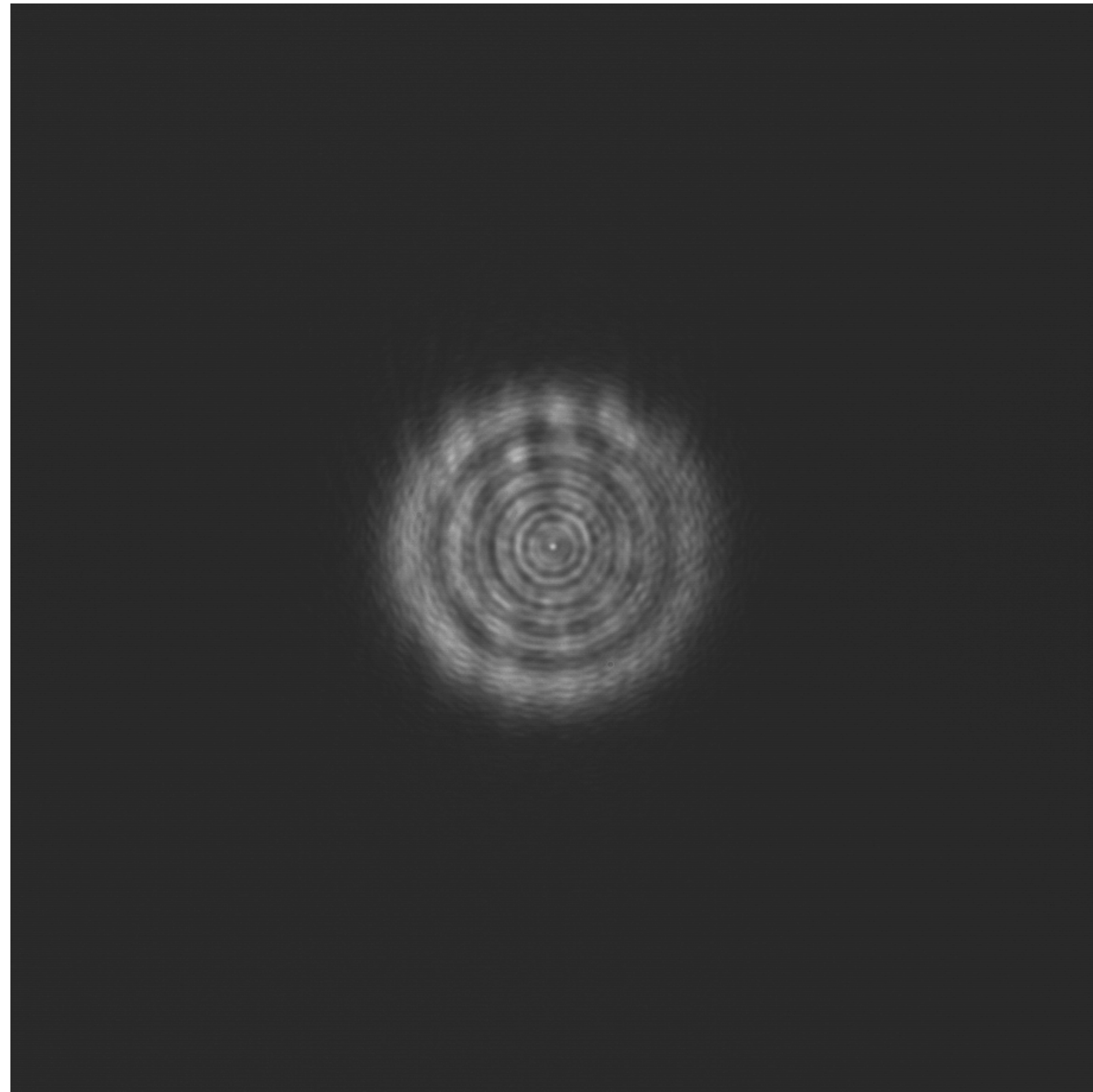
Laser spot



	Potenza
No filtro	106 μ W
NE05A-A	32,7 μ W
NE10A-A	11,3 μ W
NE30A	43,6 nW

0.04 % della
potenza iniziale

Laser spot



$$(\text{Signal}_{GL} - \text{bias})G \sim 2 \text{ Me-}$$

Da conti fatti sulla potenza
del laser dovrebbero arrivare
 $\sim 10 M_\gamma$

Conclusioni

1. La camera **Cyclone** è stata installata con successo.
2. È una camera ad alta velocità, in grado di acquisire immagini fino a **2166 Hz** alla massima risoluzione.
3. Sono state condotte analisi per caratterizzare il sensore, quantificando:
 - **Bias,**
 - **Gain,**
 - **Read noise.**
4. È stata effettuata una prima stima del numero di **elettroni generati sul sensore** dall'illuminazione di uno **spot laser**.
5. Il passo successivo è il calcolo dell'**efficienza quantica (Quantum Efficiency, QE)** del sensore.

