Emva1288

Camera simulator

Federico Ariza. R&D Manager
Jean-Francois Taillon. SW developer

- HQ in Montreal, offices in SF and Munich
- 30 employees
- Active in academic & industry communities

$10M Series A

May 2018

| Top Deep Learning Startups | Gold & Silver Innovators Awards | Vision Product of the Year for Automotive | Best Object Recognition solution | 10 most Innovative companies in Machine-Learning | Software Innovation Winner | Top Companies to Watch | Most Innovative Autonomous Driving Solution |

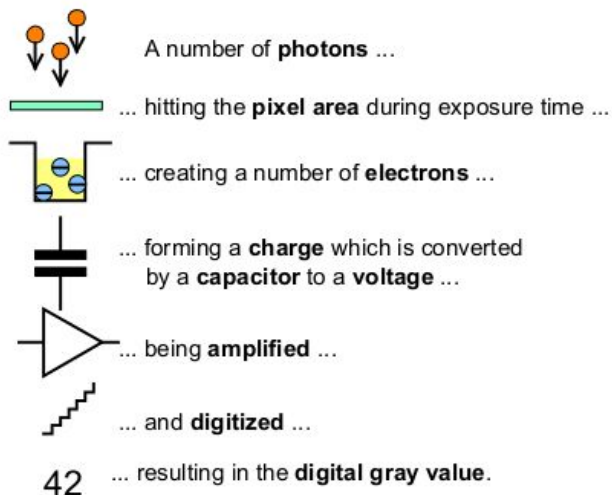The EMVA1288 python module

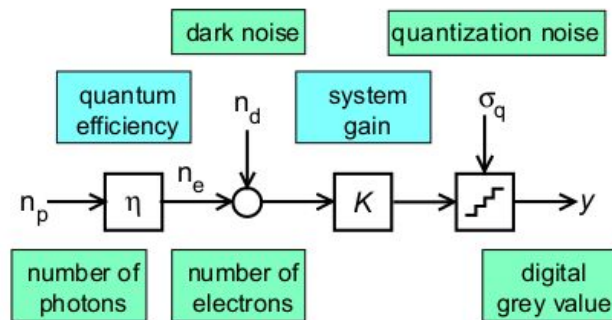https://github.com/EMVA1288/emva1288

The Camera simulator:

```
from emva1288.camera import Camera
```

Process of capturing an image

Figure 1: **a** Physical model of the camera and **b** Mathematical model of a single pixel.

Taken from https://www.emva.org/wp-content/uploads/EMVA1288-3.0.pdf

Simulation workflow

RGB image
[pixels] → Spectrum [Irradiance(λ)] → Radiance(λ) [w sr^{-1} cm^{-2}] → Photons (per pixel)

Dark noise ← Conversion to electrons (per pixel) ← Qe / non-uniformity

Conversion to digital (from system gain) → Final image (Grey value)

algolux

From RGB to photons(λ)
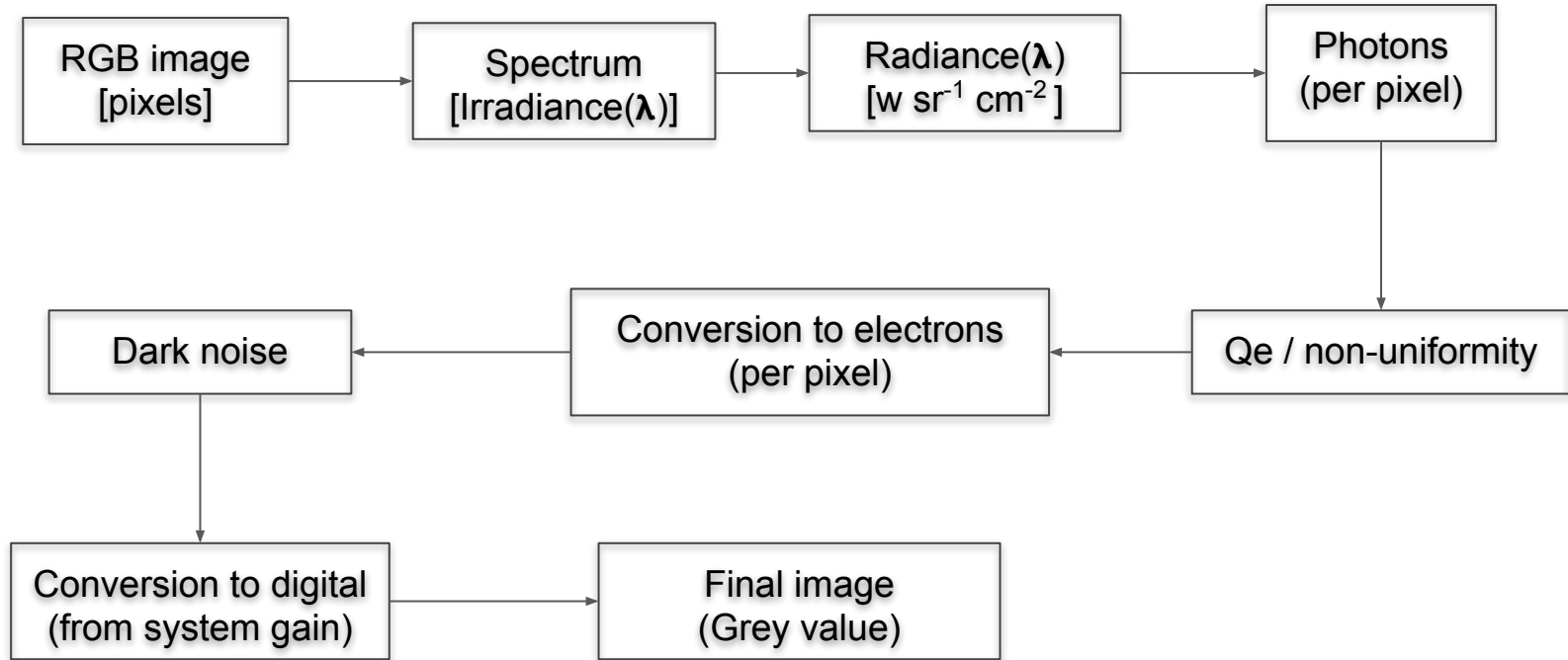
```
┌──────────────┐              ┌──────────────┐
│  RGB image   │─────────────▶│   Spectrum   │
│   [pixels]   │              │              │
└──────────────┘              └──────────────┘
```
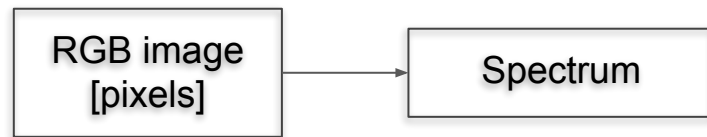
- Not deterministic
- Depends on environment
- Arbitrary results

It is possible to achieve plausible solutions

QE

Radiance(λ)
[w sr⁻¹ cm⁻²] → Photons (per pixel) → Qe / non-uniformity

- QE object
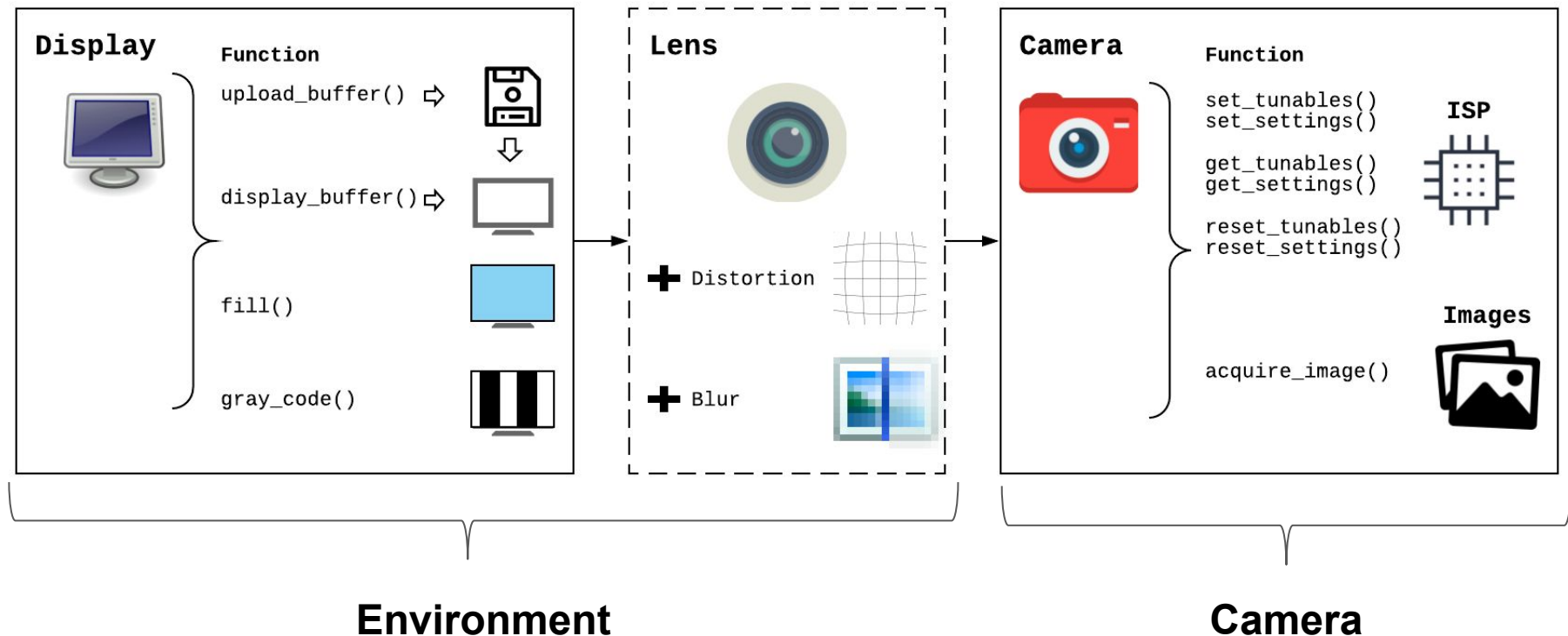- Bayer filter
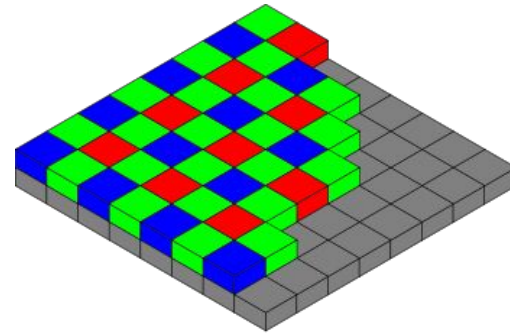
https://en.wikipedia.org/wiki/Bayer_filter

algolux

Grab

- Parameters to change (exposure, qe, etc.)
- Time of computation

Usage

Environment

Camera

We are hiring!

algolux