

# Merge photons split between pixels

- Algorithm
  - Original array
  - Split array for integer and float
  - Find local maximums
  - Find couples of connected pixels
  - Merge coupled pixels
  - Sum arrays
- Implementation
- Presentation in the Detector package



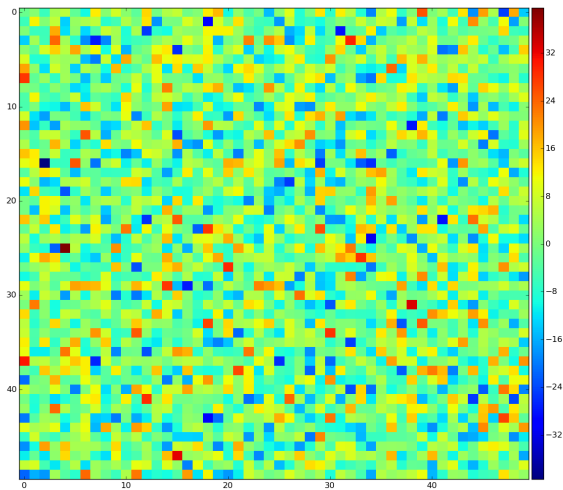
Images are shown for DIFFERENT EVENTS for demonstration purpose only

## Algorithm

Chuck suggested merging algorithm for photons split between pixels as follows.

### Original array

This algorithm consumes (float) calibrated data expressed in terms of number of photons. For test purpose use image shape=(50,50) random standard normal distribution with sigma=10



### Split array for integer and float

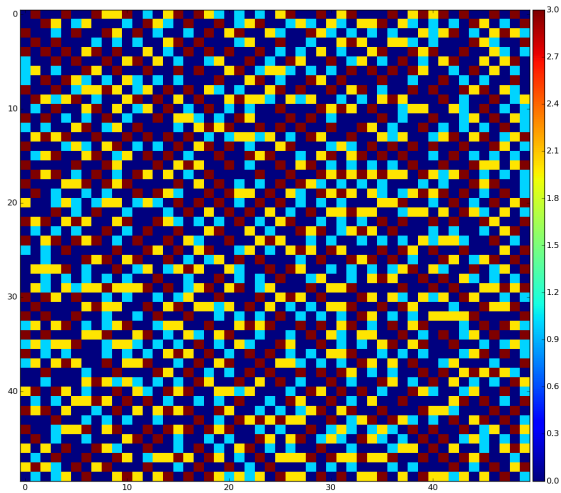
Input array with float number of photons per pixel is split for two

- floor of input (uint) and
- difference between input and previous uint array - (float) fractional array [0,1).

Pixels with intensity below zero and masked are set to 0 (photon) in both arrays.

### Find local maximums

Fractional float array is used to search for local maximums in the rank=1 cross region (among 4 pixels closest to the central):

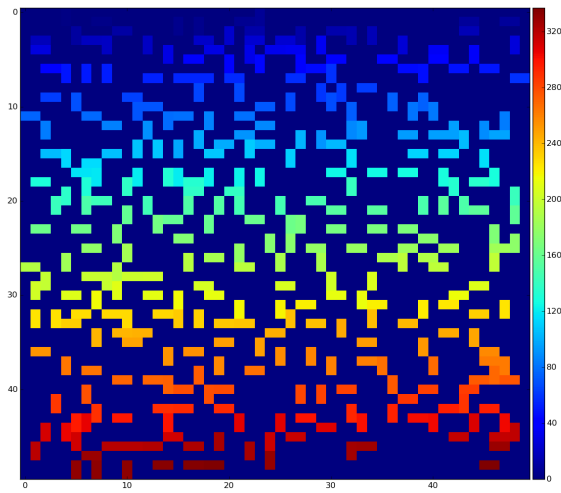


Color coding:

- (0) dark blue - not maximum
- (1) bright blue - local maximum of rank 1 in the row
- (2) yellow - local maximum of rank 1 in the column
- (3) red - local maximum of rank 1 in the cross region

## Find couples of connected pixels

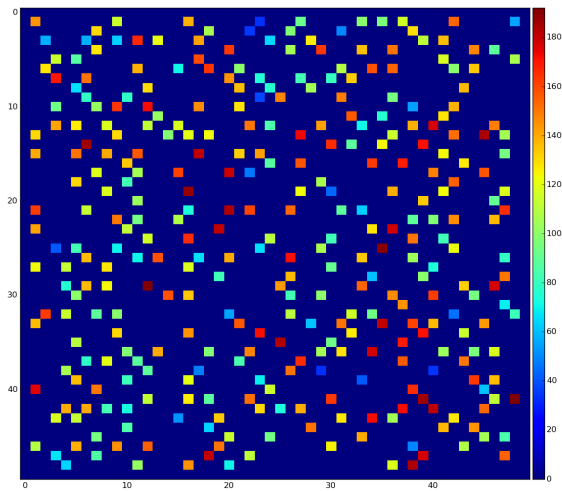
Local maximum (with threshold  $>0.25$ ) is connected to the nearest (not yet used) neighbor with maximal intensity:



Different colors show pairs of connected pixels

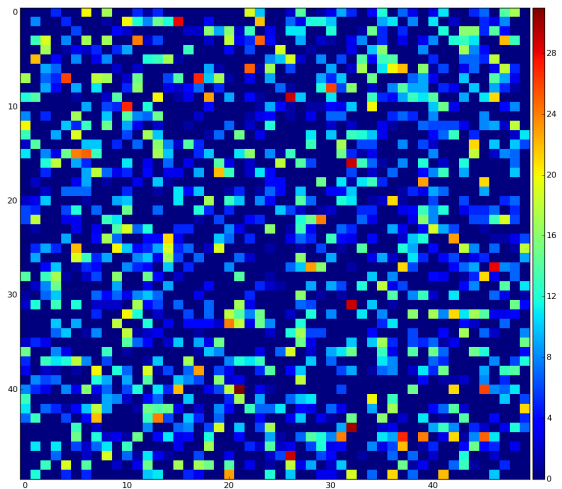
## Merge coupled pixels

Each pair total intensity (with threshold 0.75) is used to count additional integer number of photons.



## Sum arrays

Add array with merged photons with array of integer number of photons:



## Implementation

- C++ algorithm is implemented in `ImgAlgos::AlgImgPros::mapOfPhotonNumbersV1` for 2-d arrays only
- Boost-python wrapper is in `ImgAlgos.pyImgAlgos.cpp`
- Pure python wrapper `ImgAlgos.PyAlgos.py` - supports 2-d and n-d arrays.

## Presentation in the Detector package

- Reference to `ImgAlgos.PyAlgos.photon` is in `AlgoAccess.py`
- `AreaDetector.py` has `photon(...)` method