Piccante,
a gentle introduction

dr. Francesco Banterle

francesco.banterle@isti.cnr.it

Piccante: who we are

dr. Francesco Banterle - architect and programmer
francesco.banterle@isti.cnr.it

dr. Luca Benedetti - library packer and programmer
l.benedetti@bath.ac.uk

Piccante: when and how it started

- Piccante started as a debayering tool for PTGrey cameras in early 2010 for acquiring Polynomial Texture Maps…

- More functionalities were soon added such as a noise removal, basic operators, filters… until we got Piccante

Piccante: is

- A C++11 open source library header-only; *no need to compile it*
- A library for processing LDR and HDR images
  - It is for Imaging!
- MPL2 License
- Modular —> *easy to extend!!!*
- Multi-platform —> it compiles on Windows, OS X, and Linux
- Fast —> multi-threading using C++11 threads, and GPU via OpenGL 4.0 Core Profile

Piccante: comparisons

- OpenCV: great library meant for computer vision
- PFS: HDR imaging library similar in scope and aim
- ImageStack: great but abandoned project for imaging
- Image Magick: great library for reading images
- Free Image: great library for reading images
- ImageJ: great library for medical imaging
- Halide: great high-performance language for Imaging

Piccante: what can be done

CRF recovery
HDR assembly

Piccante: what can be done

Tone Mapping

Piccante: what can be done

Color Space Conversion

Piccante: what can be done

Filtering: linear, edge-preserving, non-linear, etc.

Piccante: what can be done

Extracting features, matching them, aligning images

Piccante: depends on

- **OpenGL**: if you want to access the OpenGL module. *You can choose your favorite OpenGL loading library; e.g. GLEW, GL3W, etc.*

- **QT**: for running some examples and for some I/O - It is multi-platform

- **Eigen**: linear algebra library; already included in the folder "*include/externals*"

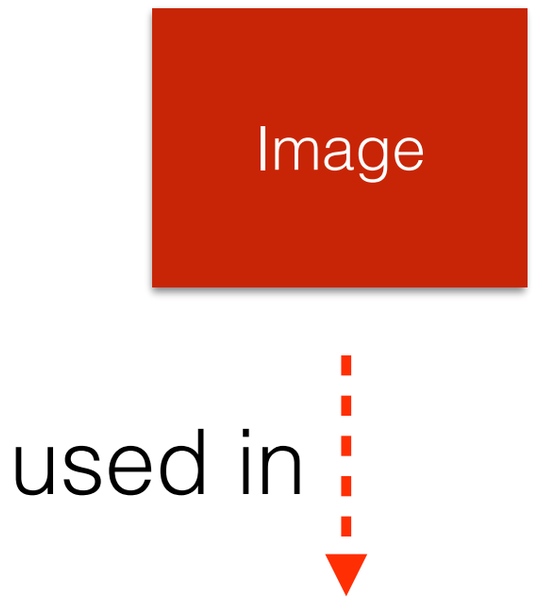
Piccante: how to installation

- Download from:
  - http://piccantelib.net
- Unzip Piccante into your favorite folder
- Add “piccante/include” in you include path
- Add in your code “#include “piccante.hpp”

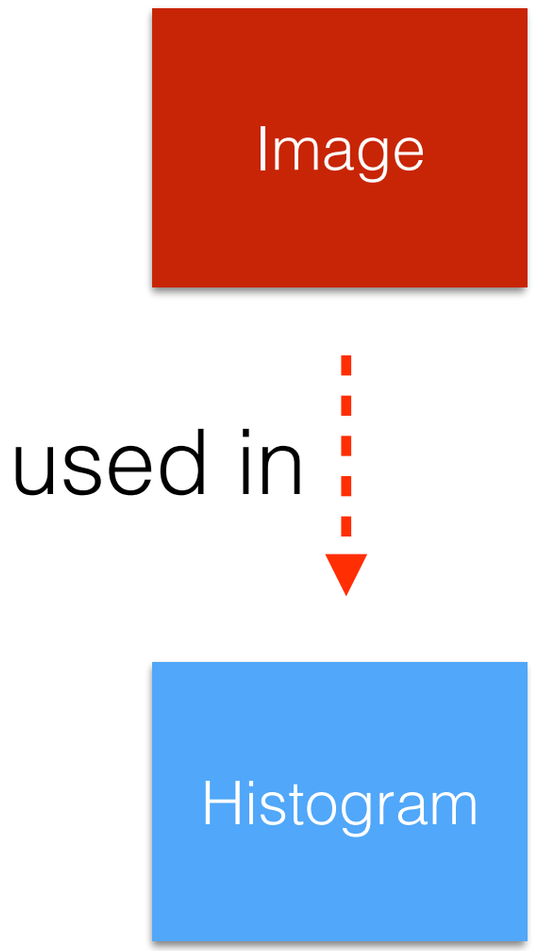
Piccante: main classes

Image

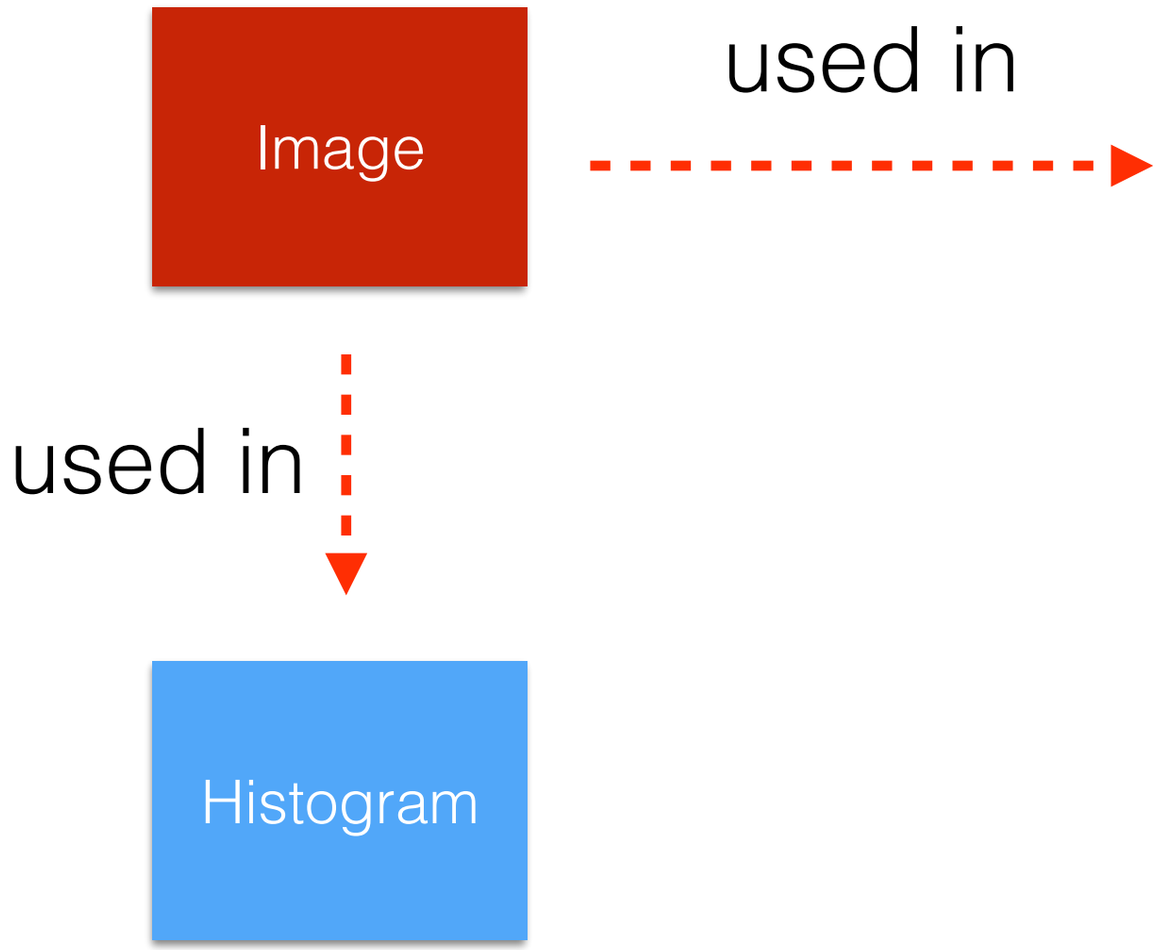
Piccante: main classes



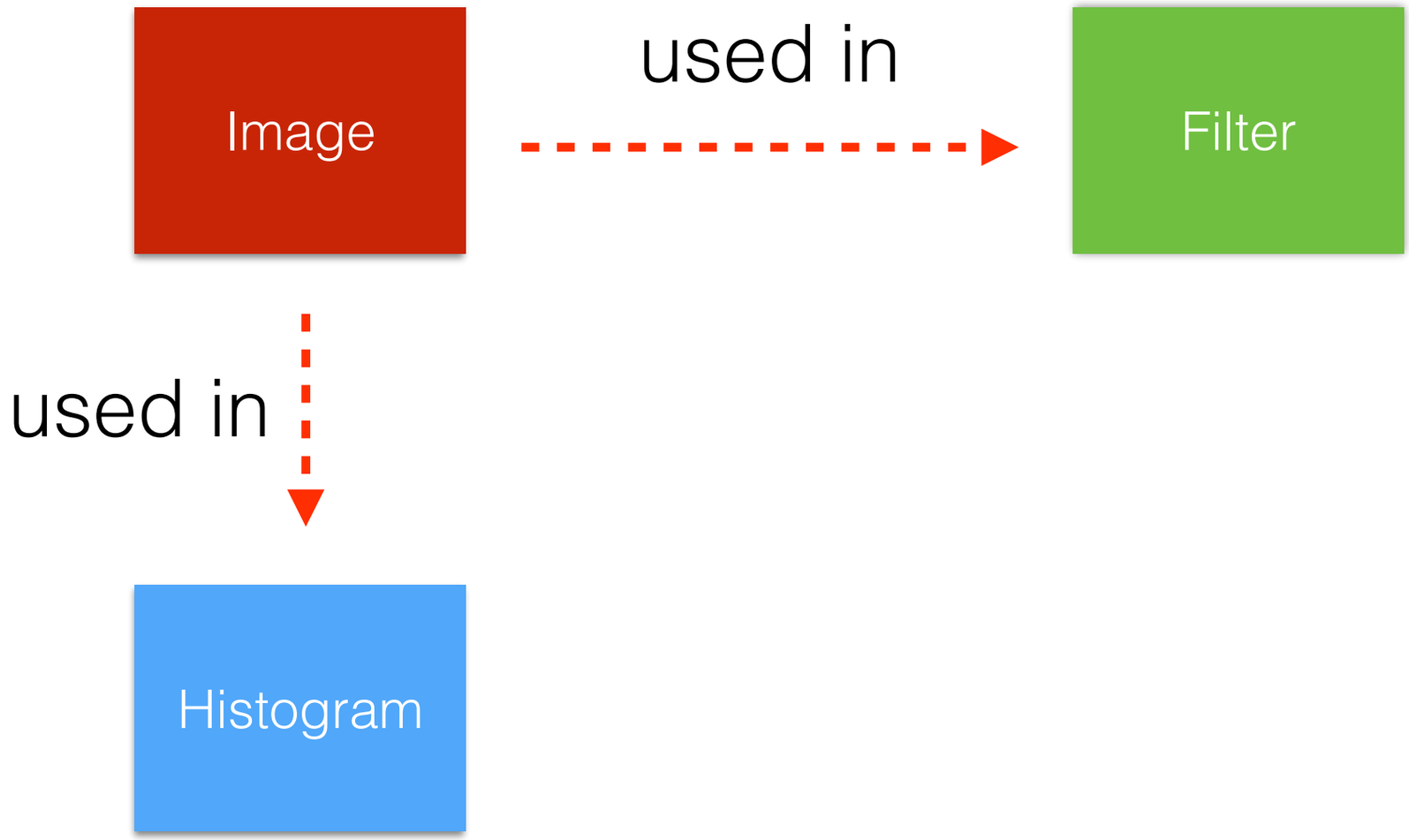
Piccante: main classes



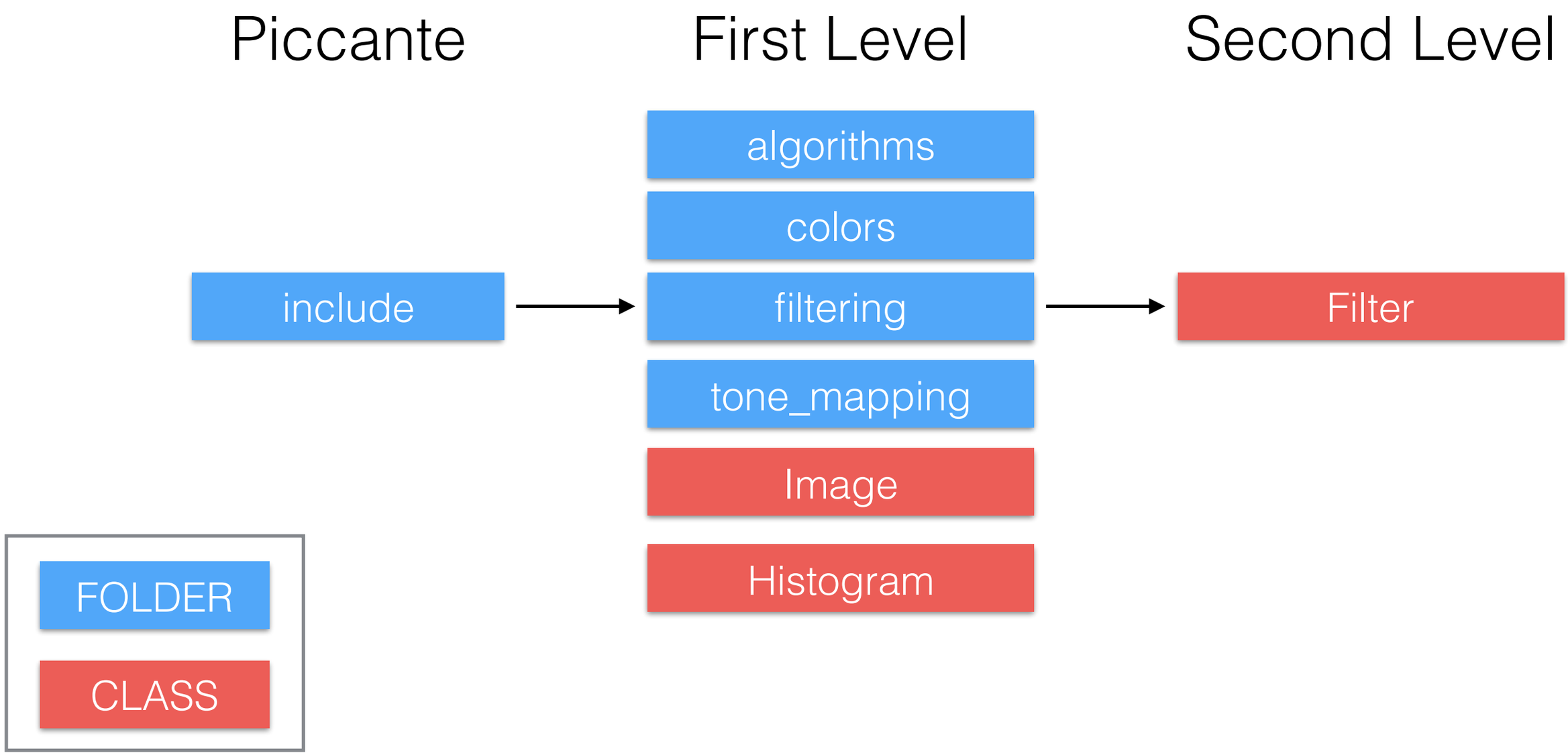
Piccante: main classes



Piccante: main classes

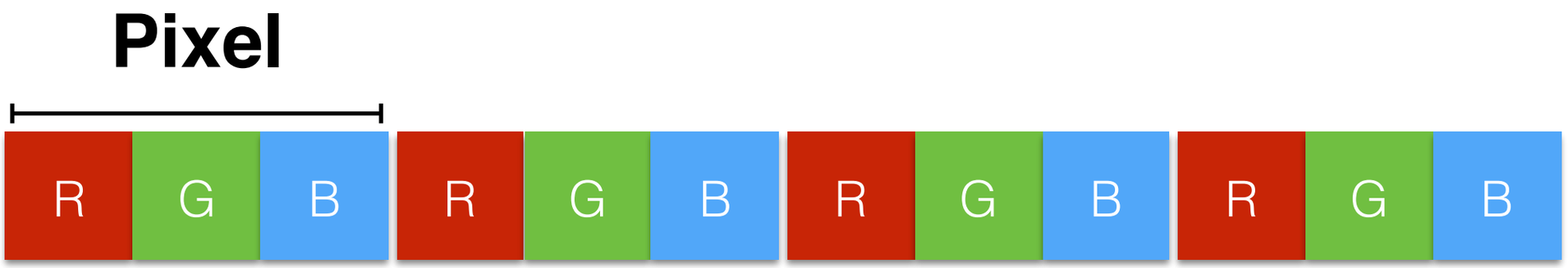


Piccante: mapping the space



Piccante: Image class

- An image in Piccante is a linear array of pixels stored in a *packed* format and *row major*:



- Number of color channels are specified by the user; i.e. images are not limited to have three components

- Images can have multiple frames; images can be videos or 3D images!

Piccante: Image class

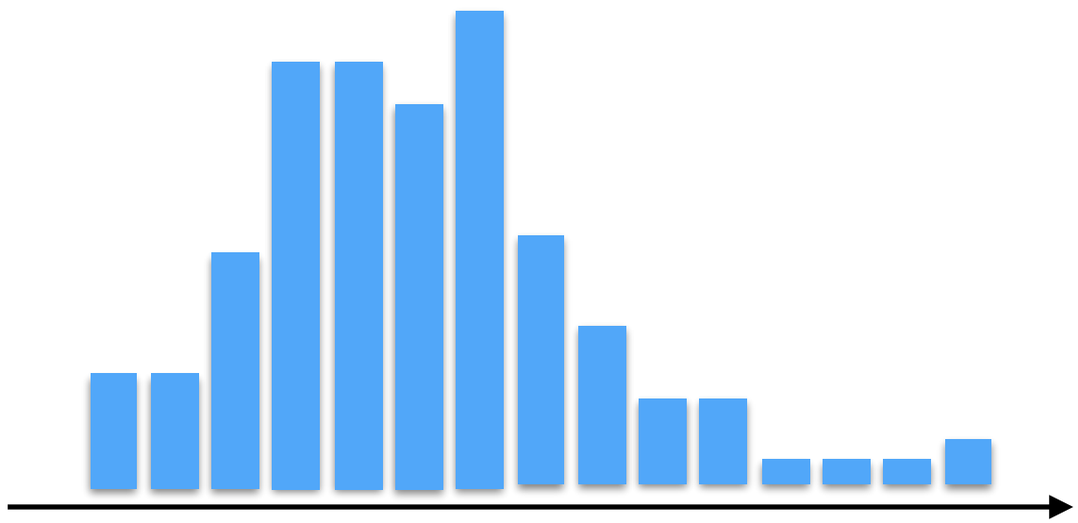
- It provides:
  - functions for memory management
  - accessing pixels (stored as float)
  - basic operators between images and single values: max, min, +, -, *, /
  - functions for gathering statistics for the entire image or a window of it: mean, variance, max, min, etc..

Piccante: Image class

- It provides functions for I/O:
  - **Native**: HDR, PFM EXR (through *optional* OpenEXR library), BMP, TGA, PPM, PGM, PBM
  - **Through QT**: PNG, JPEG, TIFF, etc.
    - QImage are managed as well

Piccante: Histogram class

- Given an image, it creates a classic histogram
- It is general; working for LDR and HDR images
- It is meant for a single color plane
- It can be embedded into an Image for using it in Filter



Piccante: Filter class

- This class defines a Piccante filter:

$$n \text{ Image} \xrightarrow{f} 1 \text{ Image}$$

- It provides:
  - Multi-threading management
  - Mechanisms for allocating memory for outputs

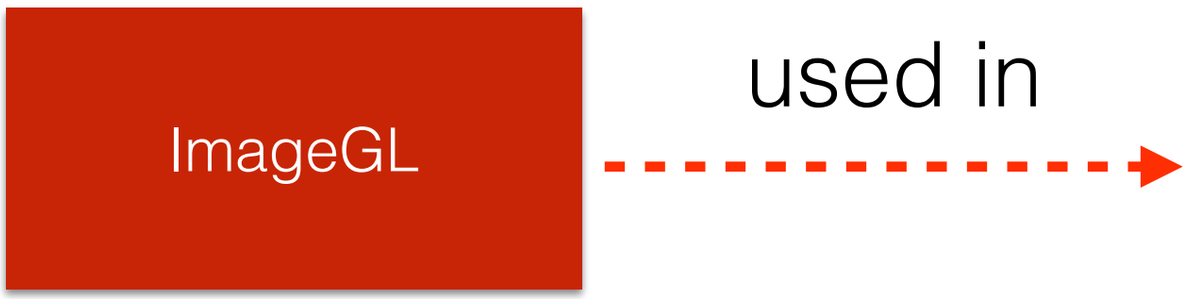
Piccante: the OpenGL module

- This is a subset of Piccante
- It exploits the graphics pipeline for fast computations on the GPU
- It is based mostly on OpenGL 4.0 Core profile
- Future versions of this module may use GL Compute shaders (from the OpenGL 4.3 Core profile)
- Not bound to a particular OpenGL Loading Library

Piccante: main classes

ImageGL

Piccante: main classes



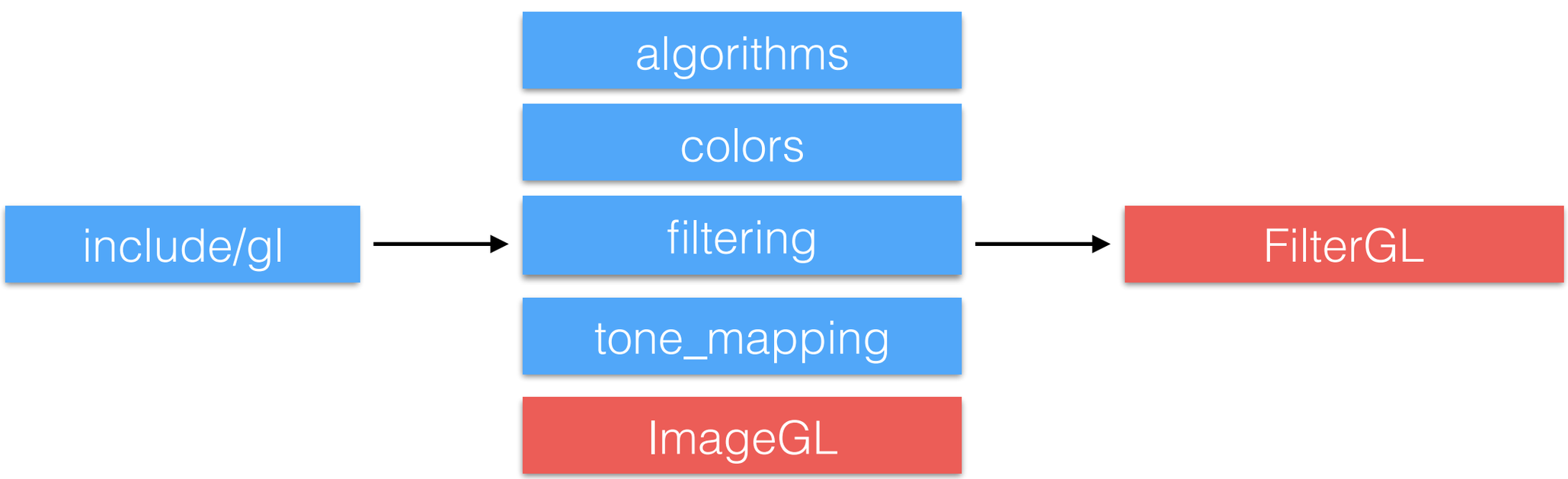
Piccante: main classes

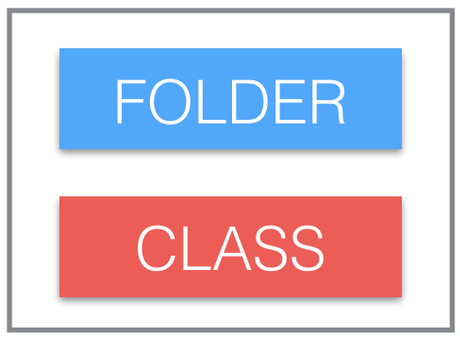
Piccante: mapping the space

Piccante

First Level

Second Level





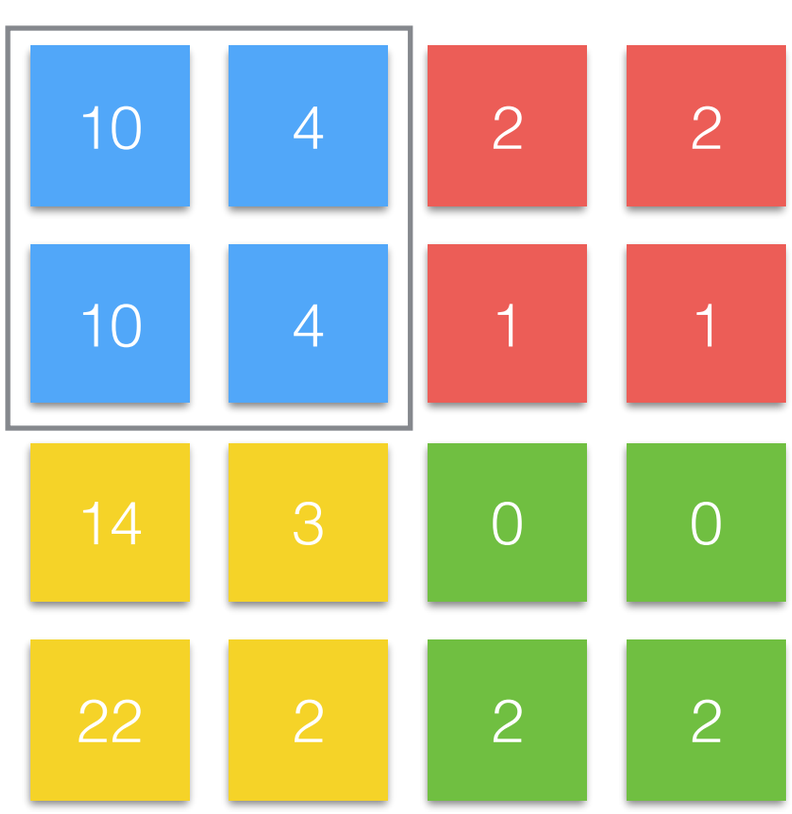
Piccante: ImageGL

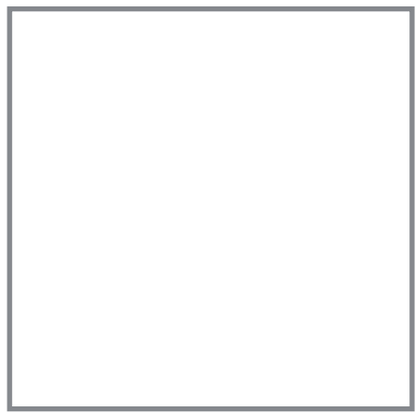
- Inheriting from Image
- Basically an OpenGL version of Image
  - mechanisms for allocating memory on the GPU and copying data
- Operators are implemented as filters…
- Supporting different kind of images:
  - GL_TEXTURE_2D, GL_TEXTURE_3D, GL_TEXTURE_2D_ARRAY, GL_TEXTURE_CUBE_MAP

Piccante: redux paradigm

Computing the minimum:

First Pass

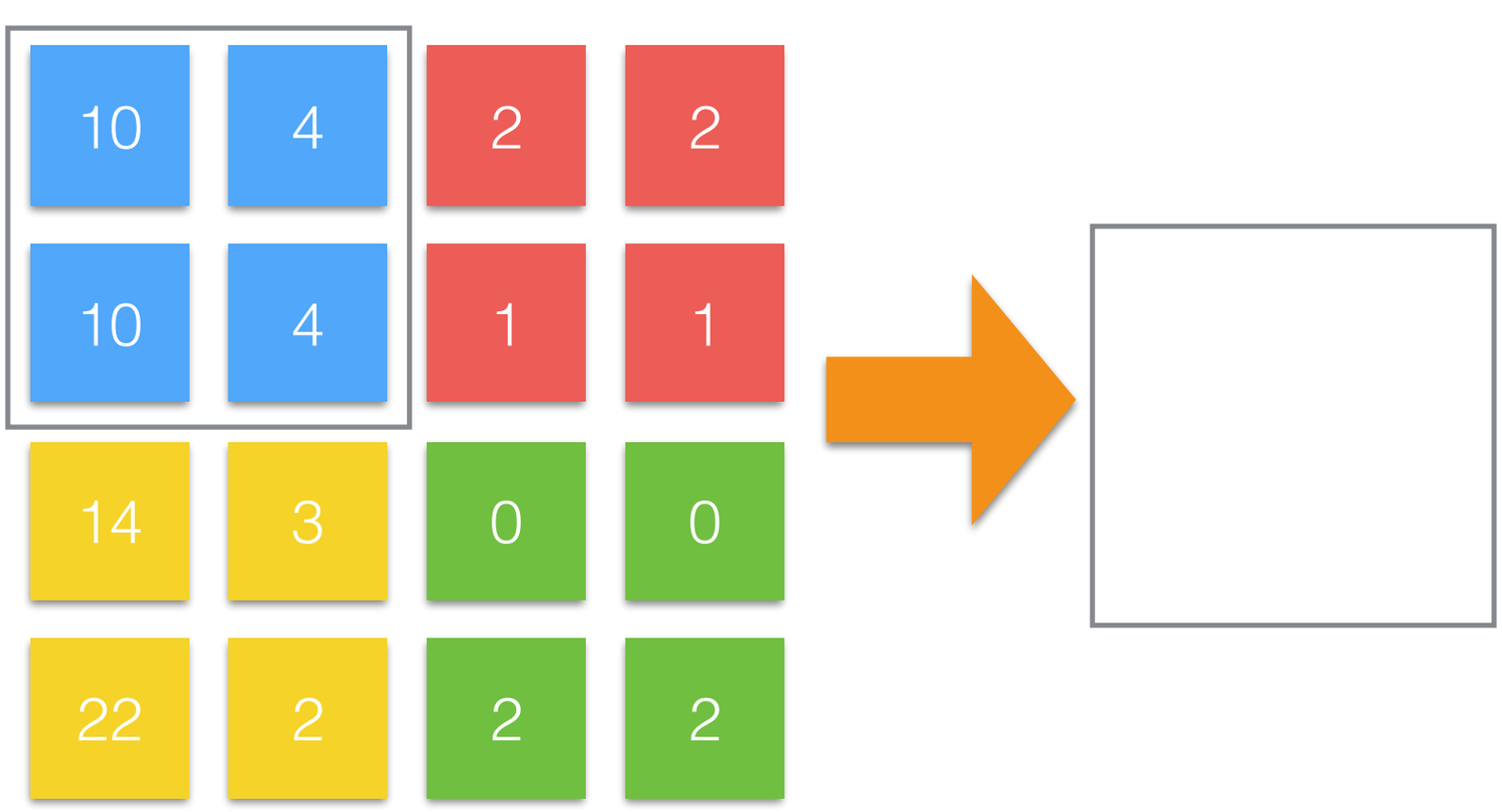




Piccante: redux paradigm

Computing the minimum:

First Pass

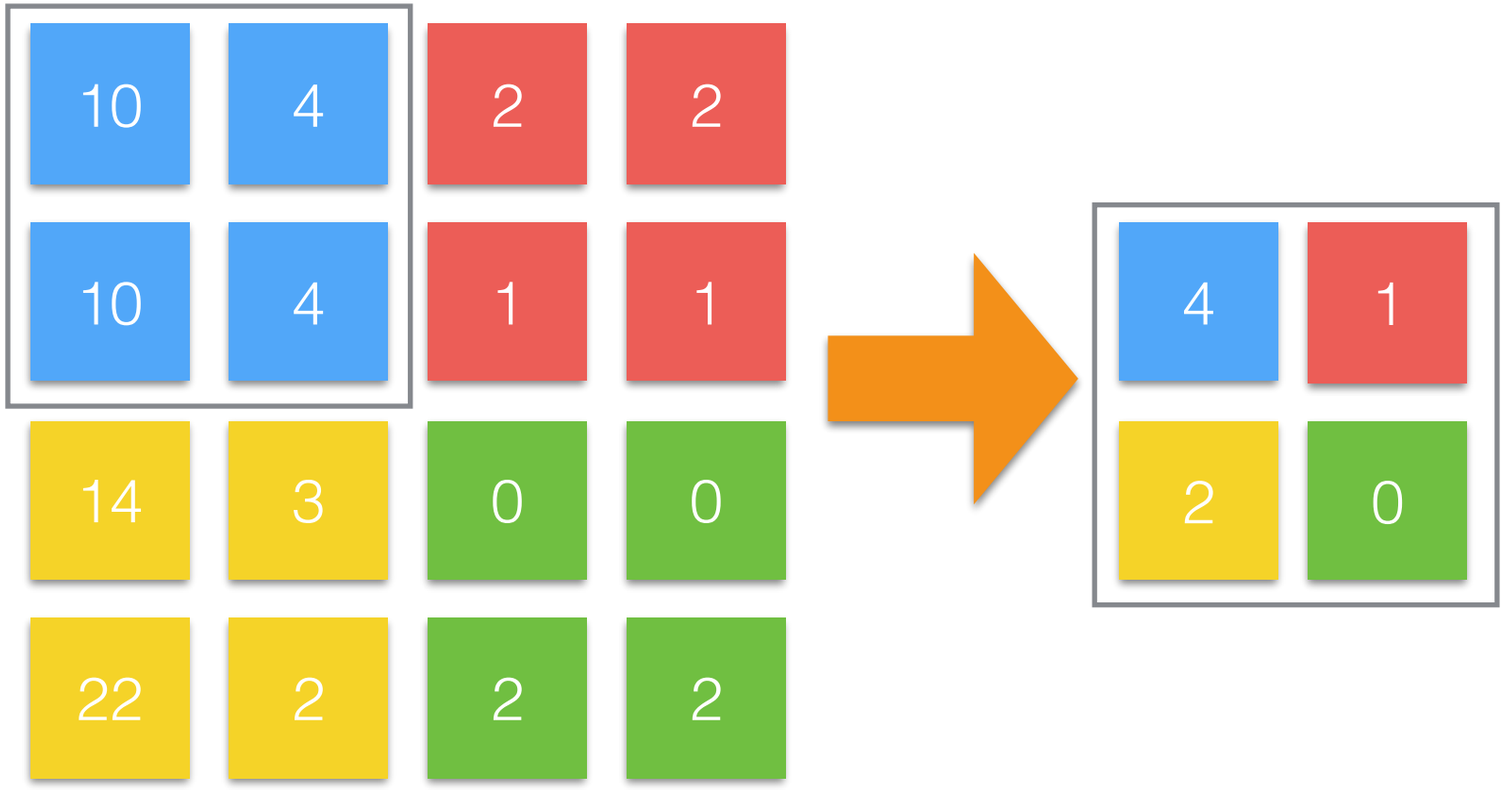


Piccante: redux paradigm

Computing the minimum:

First Pass

Second Pass

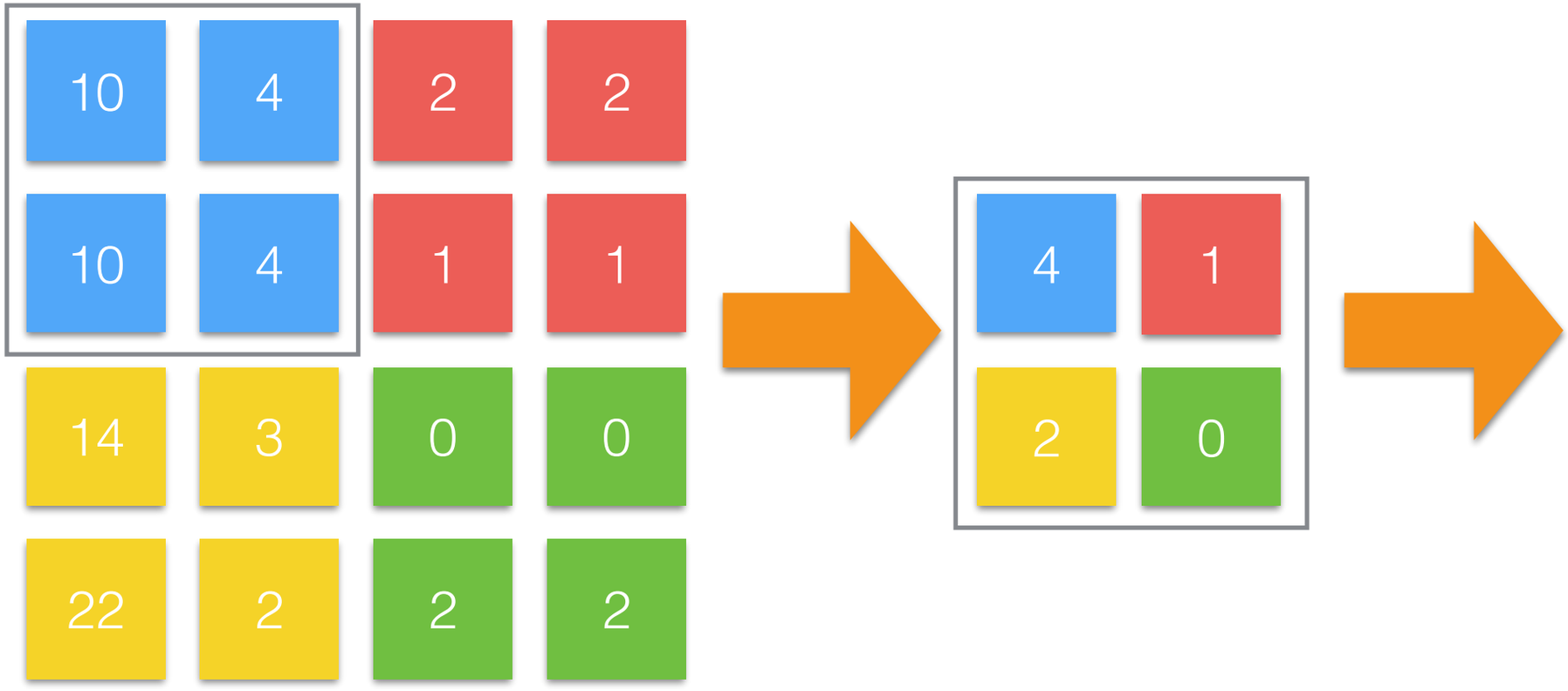


Piccante: redux paradigm

Computing the minimum:

First Pass

Second Pass



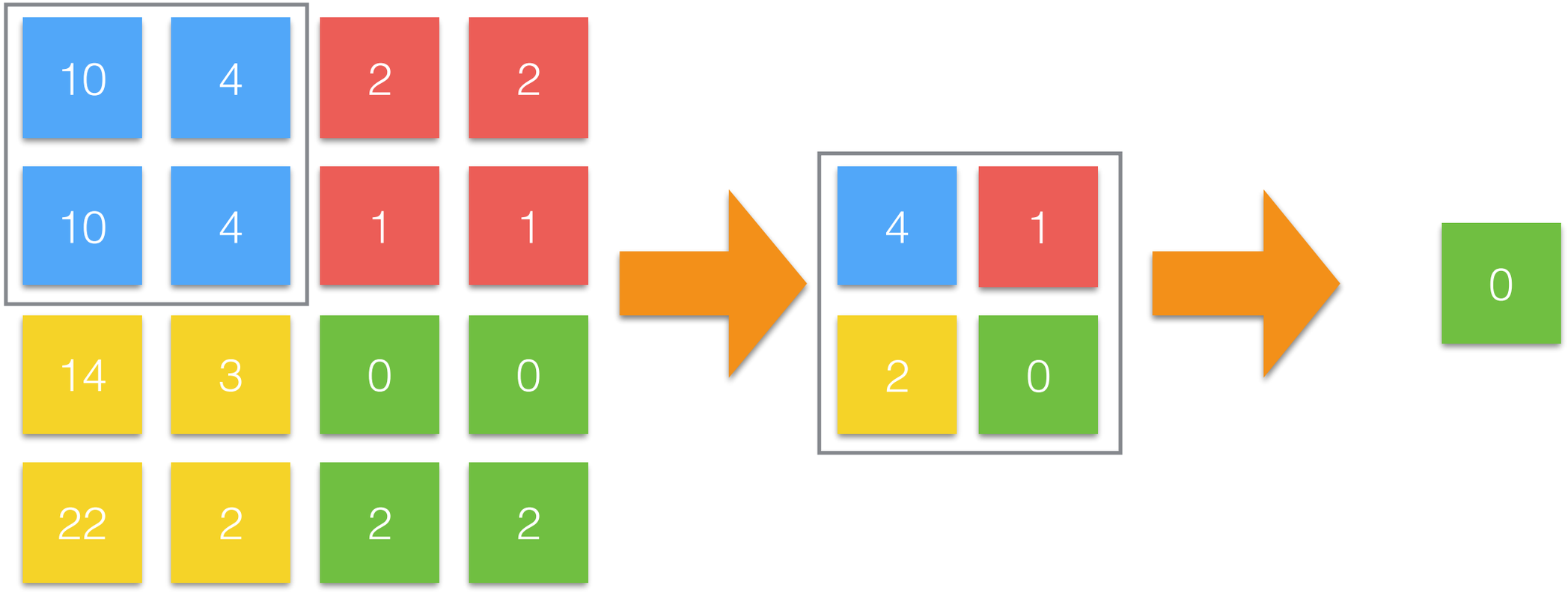
Piccante: redux paradigm

Computing the minimum:

First Pass

Second Pass

Third Pass



Piccante: FilterGL

- Basically an OpenGL version of Filter
- It provides mechanisms for managing OpenGL with ease: setting shaders up, allocating memory for outputs, etc.

and now…

Piccante in action