Image Processing Basics and InterfaceGl

Giovanni Dicanio
giovanni.dicanio@gmail.com

pluralsight
hardcore dev and IT training

Topics

- **Image processing basics**
  - Brightness/contrast adjustment
- **InterfaceGl**
- **Demo**

Introducing Lenna

Standard Test Image

Lena Söderberg

http://en.wikipedia.org/wiki/Lenna

Image Processing

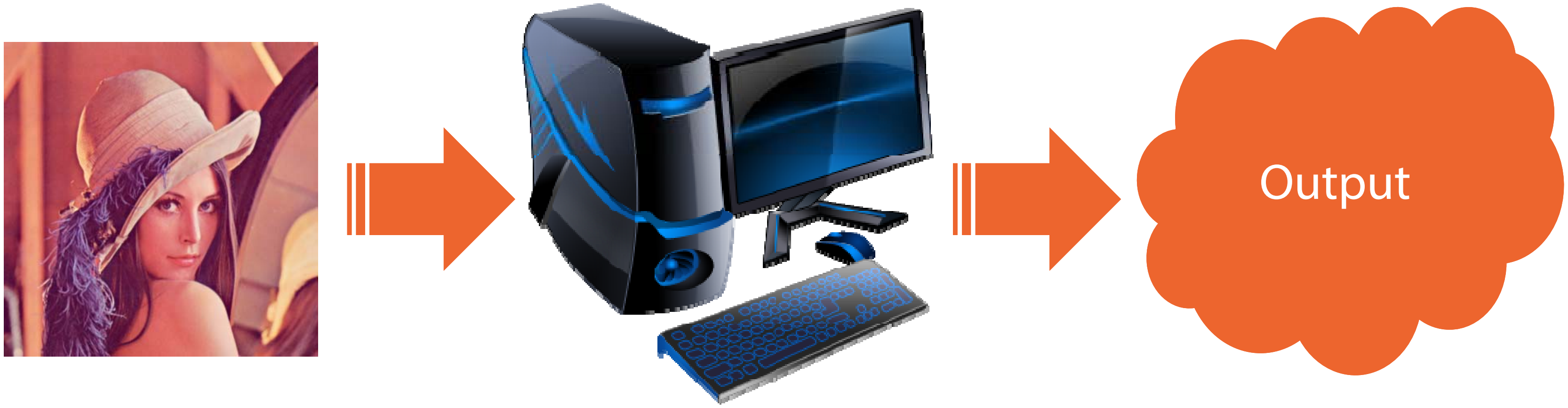


Image In → Image Out

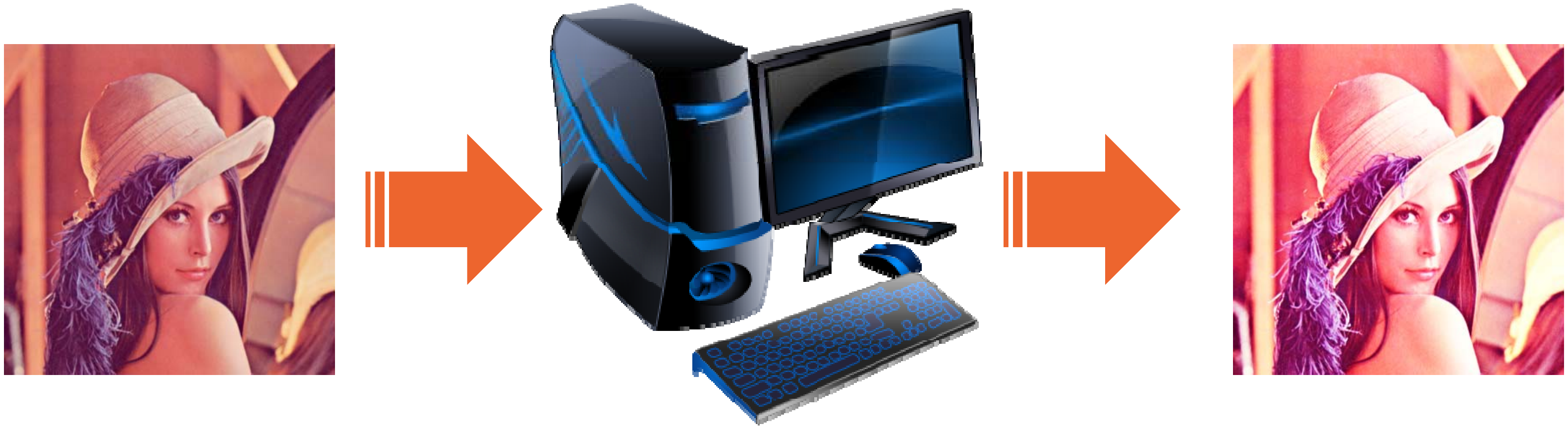
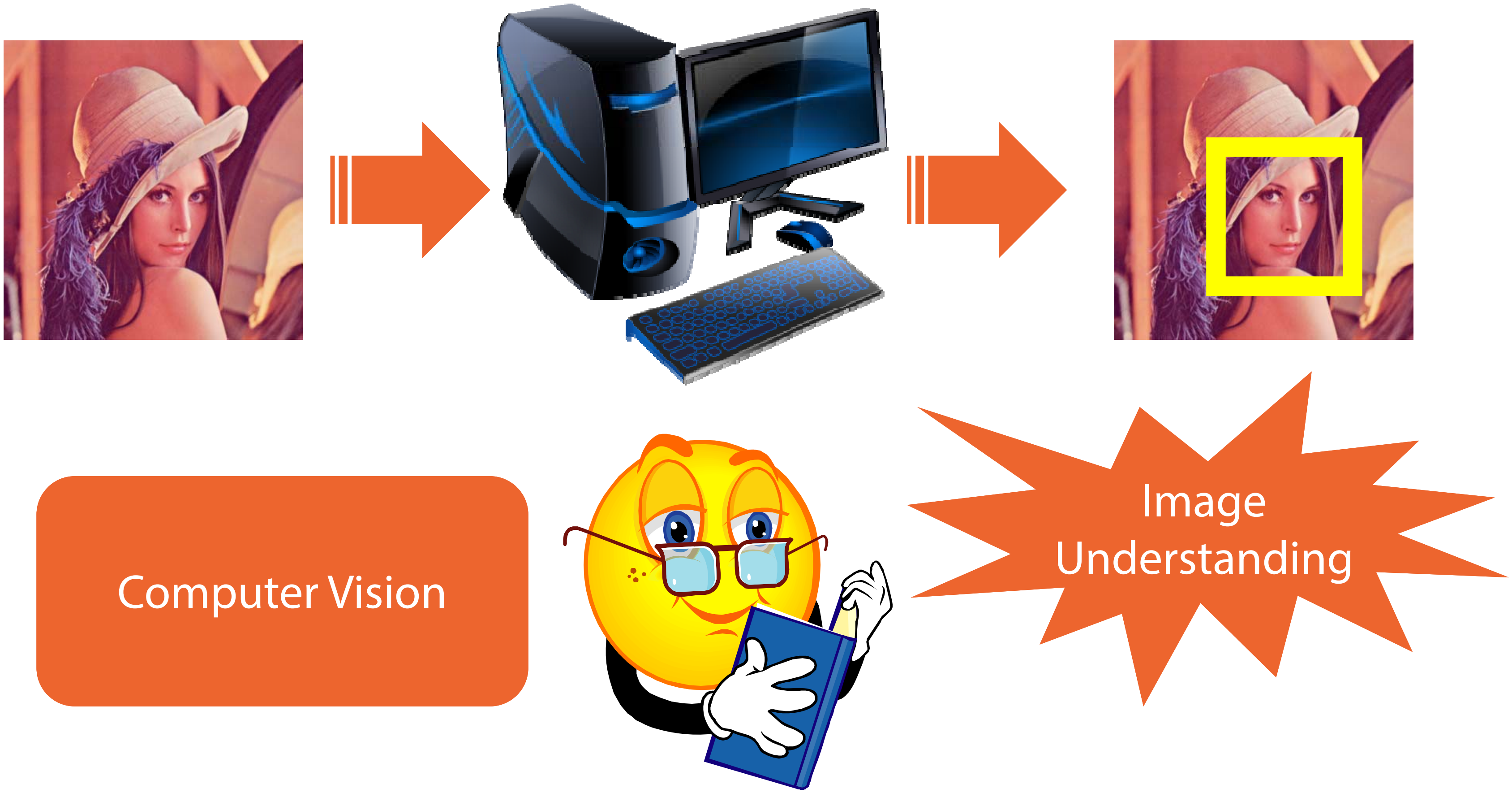


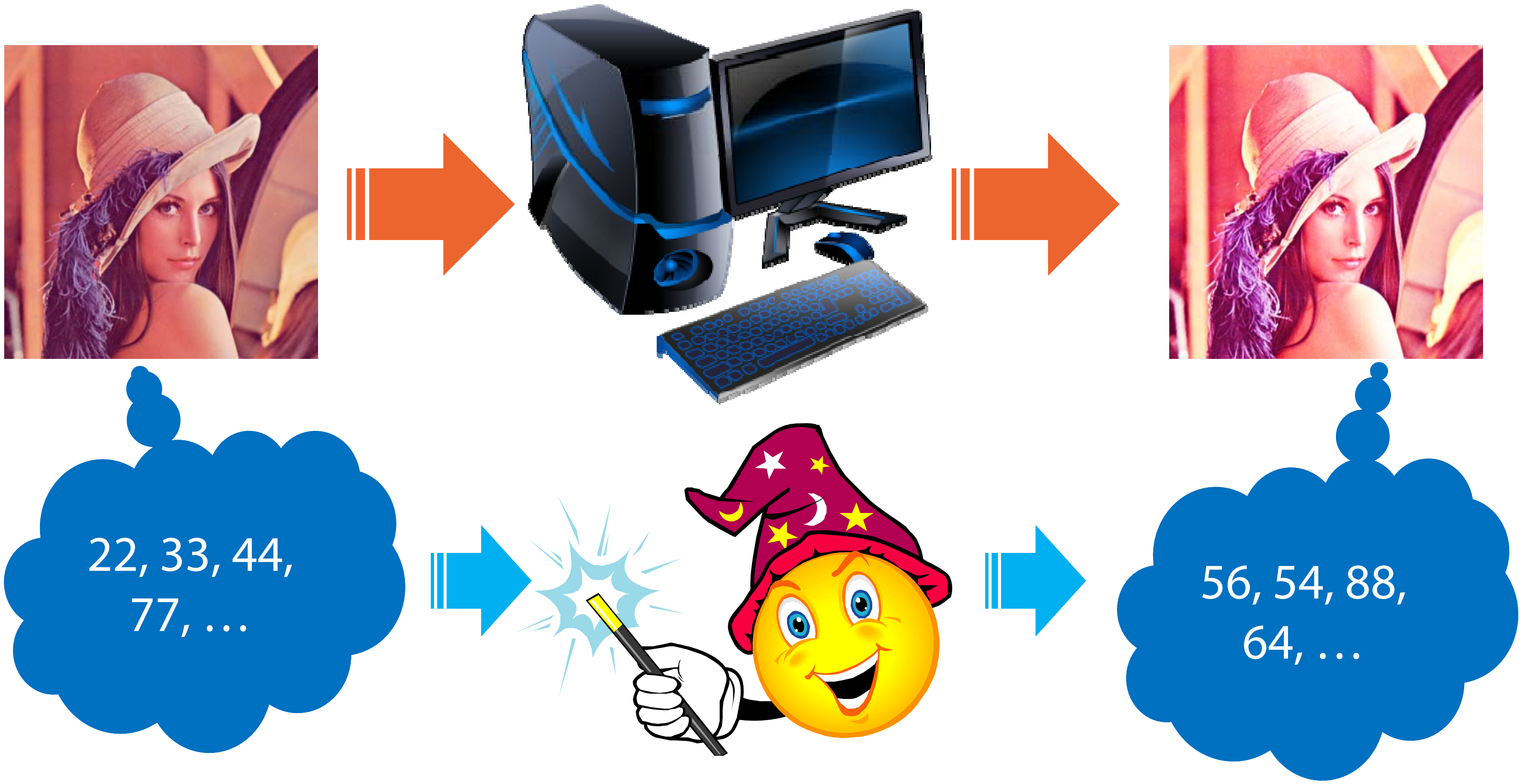
Image In → High-Level Information Out



Computer Vision

Image Understanding

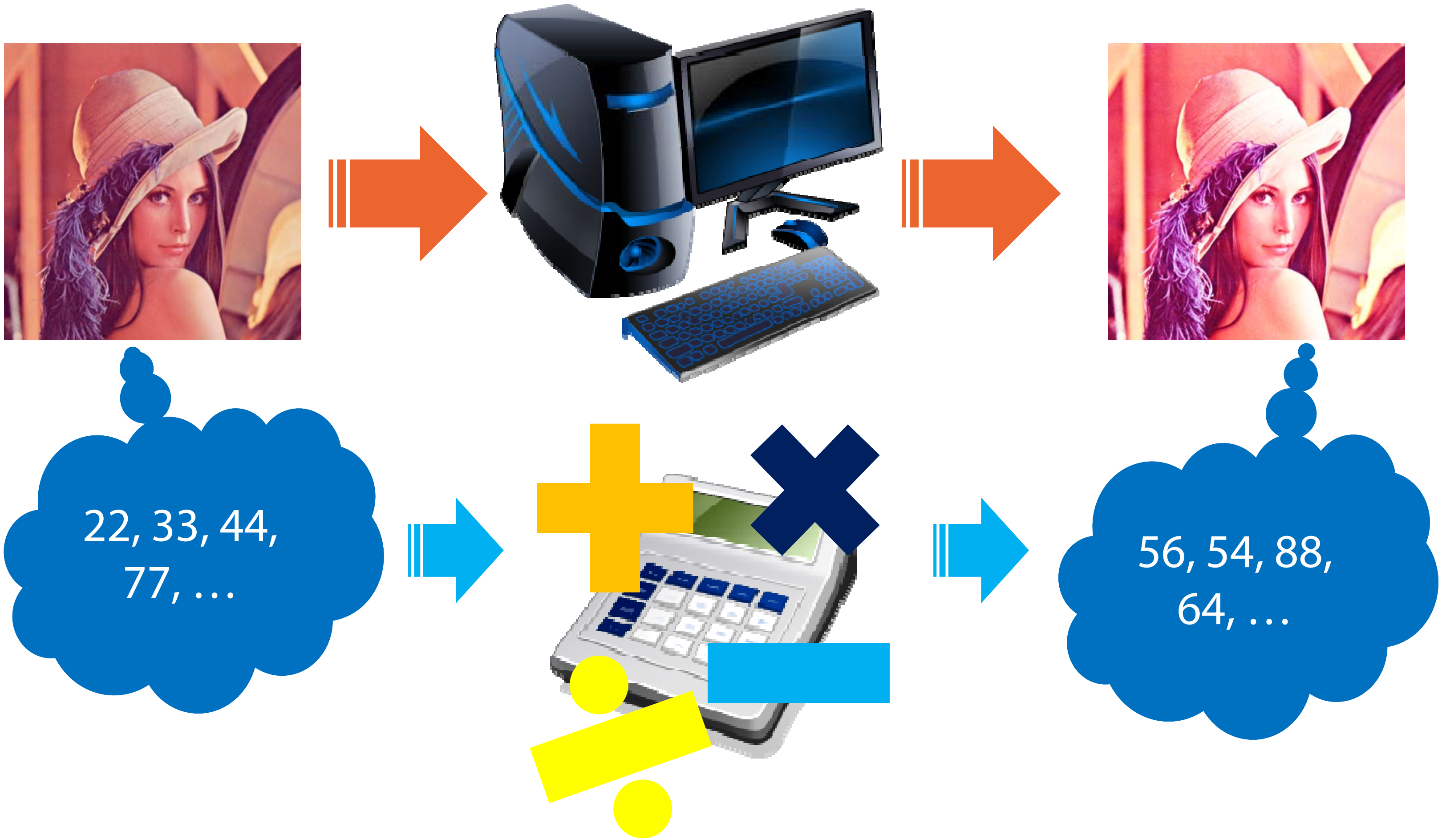
Image Processing



22, 33, 44, 77, …

56, 54, 88, 64, …

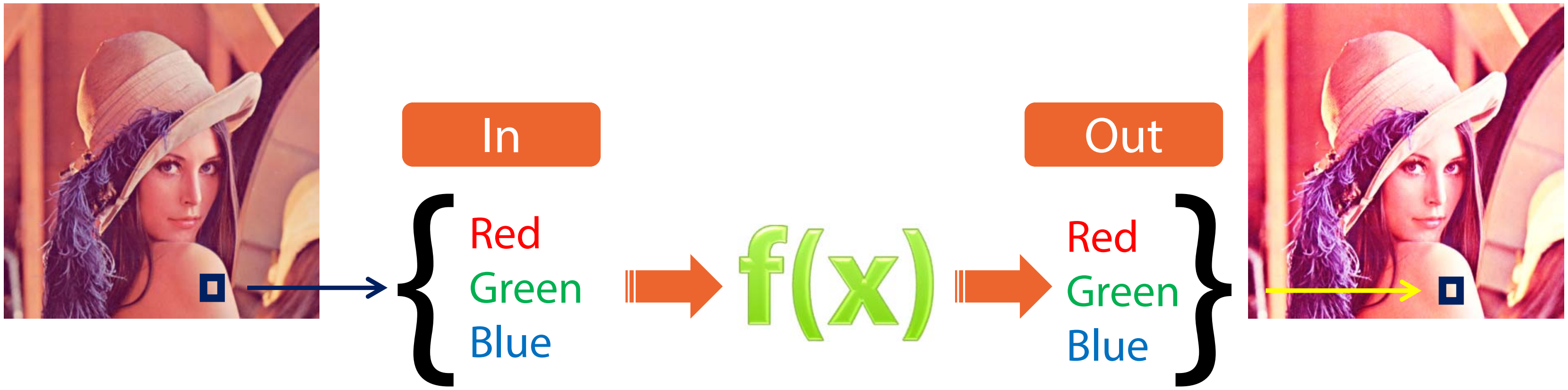
Image Processing



22, 33, 44,
77, …

56, 54, 88,
64, …

Pixel Transforms



In

Red
Green
Blue

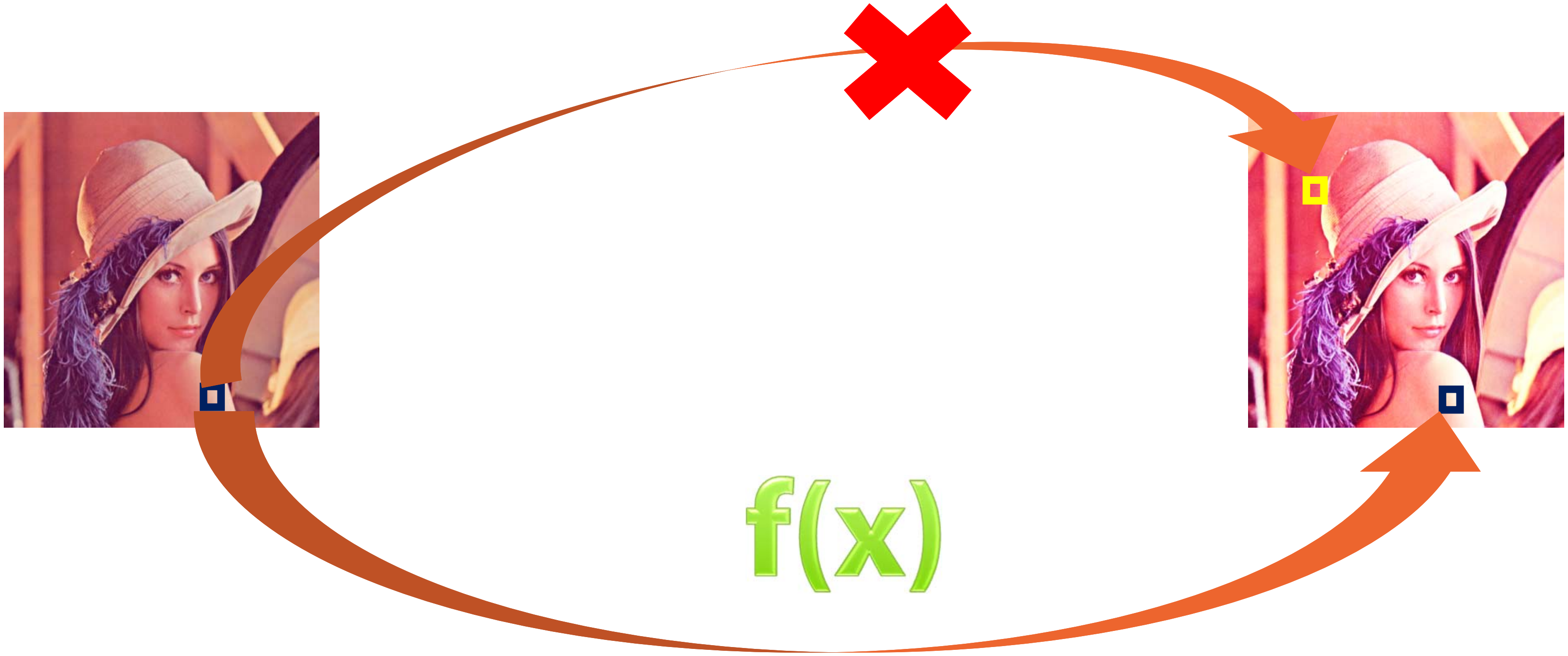
$f(x)$

Out

Red
Green
Blue

For Each Pixel
in Source Image

Pixel Transforms



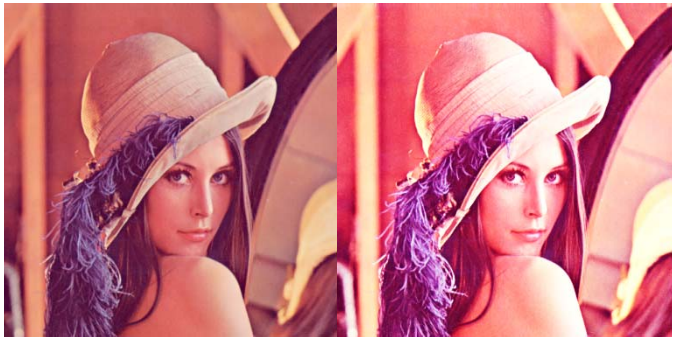
A Cooking Metaphor

In → $f(x)$ → Out

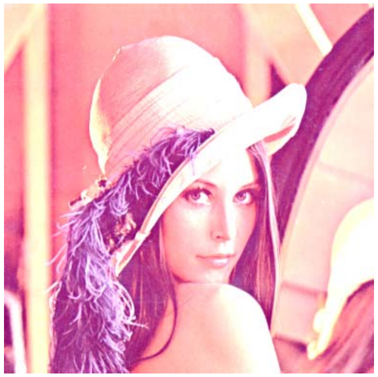
- Red
- Green
- Blue

Brightness and Contrast Adjustment

- Example of pixel transforms
- Operations involved:
  - Addition
  - Multiplication
- Parameters involved:
  - Brightness
  - Contrast



Brightness Formula



=

+

Brightness

New Value = Old Value + Brightness

Brightness R,G,B Formulae

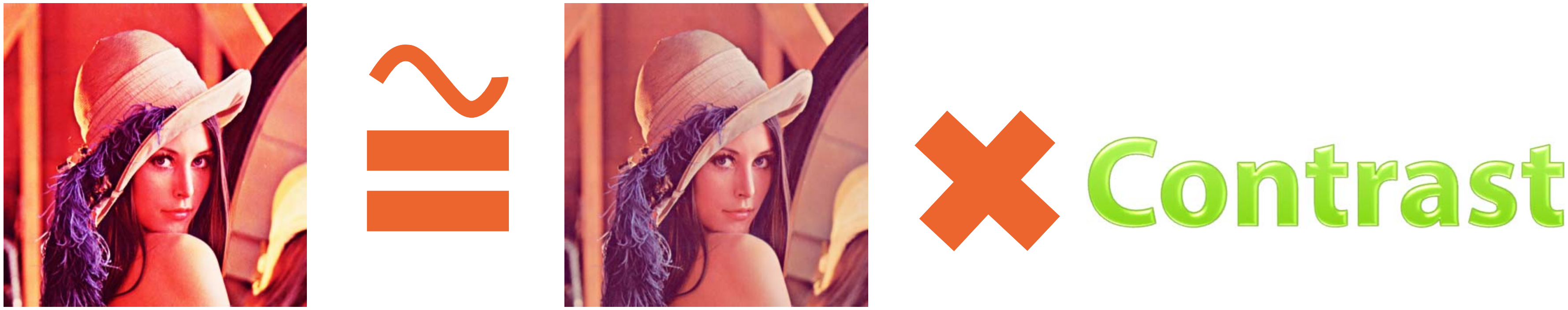
New Red = Old Red + Brightness

New Green = Old Green + Brightness

New Blue = Old Blue + Brightness

For Each
Pixel

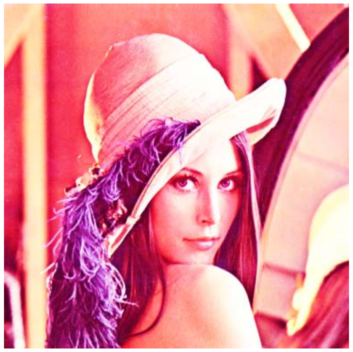
Contrast Formula



New Value = (Old Value - 0.5) * Contrast + 0.5

For each *pixel*
and
for each *R,G,B component*

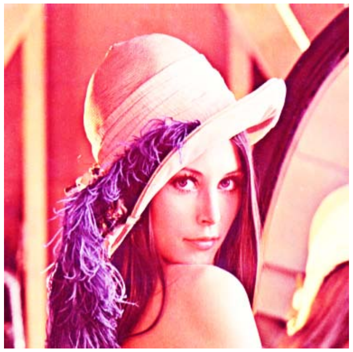
Brightness and Contrast Combined Formula

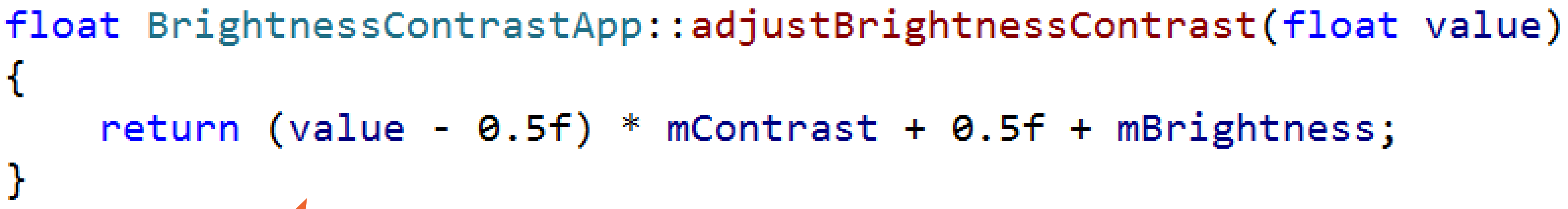


New Value = (Old Value - 0.5) * Contrast + 0.5 + Brightness

For each *pixel*
and
for each *R,G,B component*

Brightness-Contrast Formula in C++



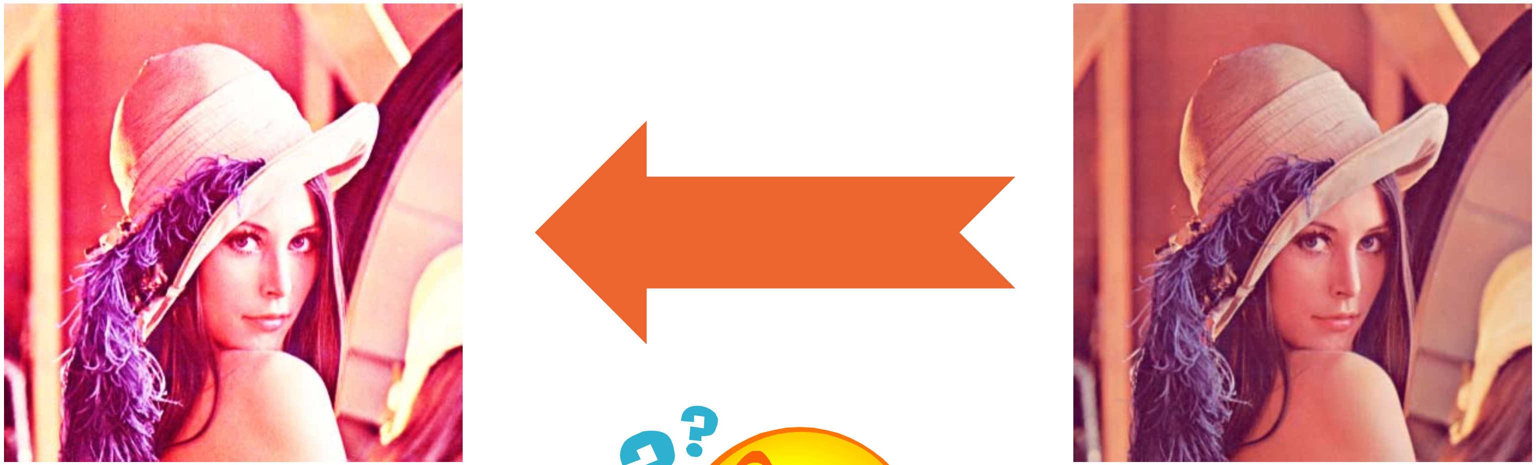


```
float BrightnessContrastApp::adjustBrightnessContrast(float value)
{
    return (value - 0.5f) * mContrast + 0.5f + mBrightness;
}
```

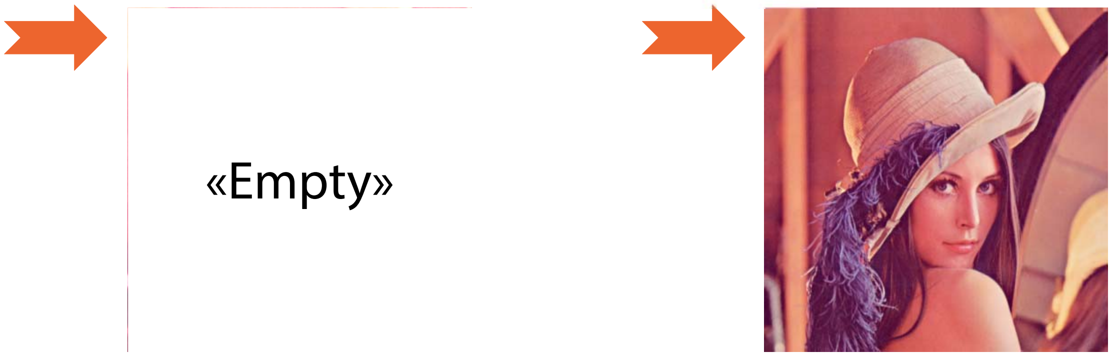
Surface32f

For each *pixel*
and
for each *R,G,B component*

Brightness and Contrast Adjustment in a Surface



Brightness and Contrast Adjustment in a Surface



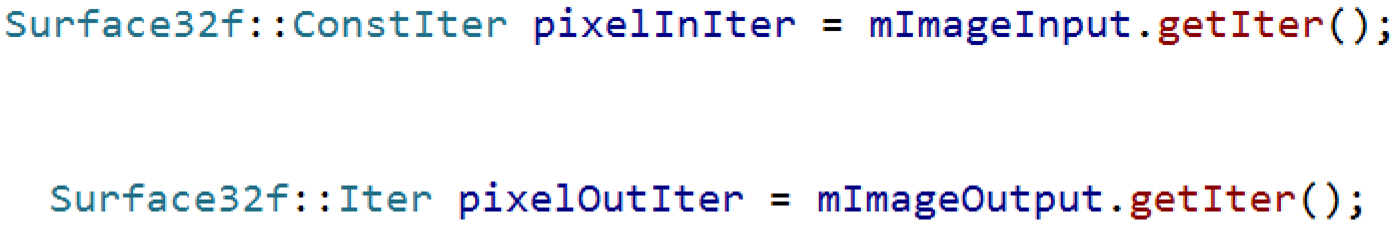
pixelOutIter

pixelInIter

«Empty»

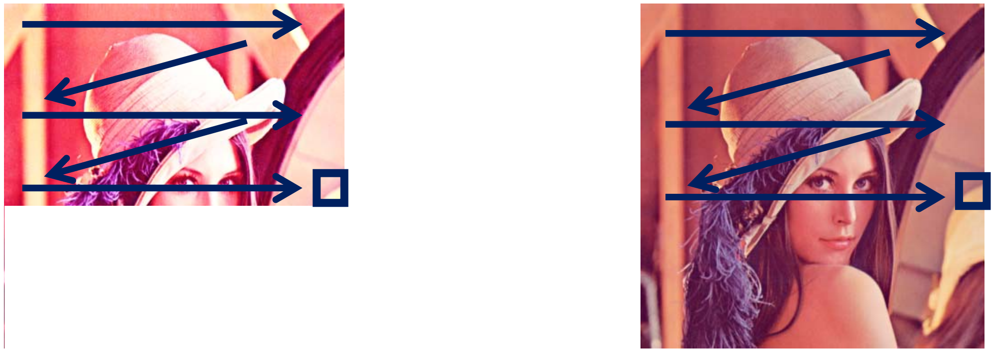
mImageOutput

mImageInput



```
Surface32f::ConstIter pixelInIter = mImageInput.getIter();

  Surface32f::Iter pixelOutIter = mImageOutput.getIter();
```

Brightness and Contrast Adjustment in a Surface

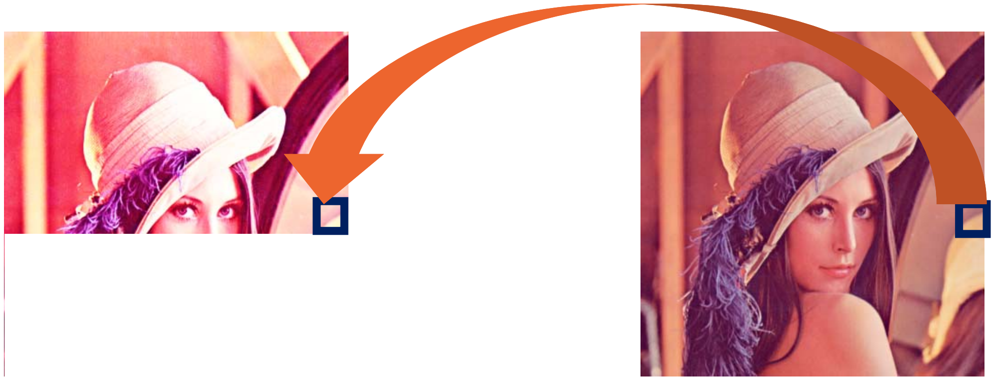


```
while (pixelInIter.line()) {
    pixelOutIter.line();

    while (pixelInIter.pixel()) {
        pixelOutIter.pixel();

            ● ● ●

    }
}
```

For each line…

For each pixel in current line…

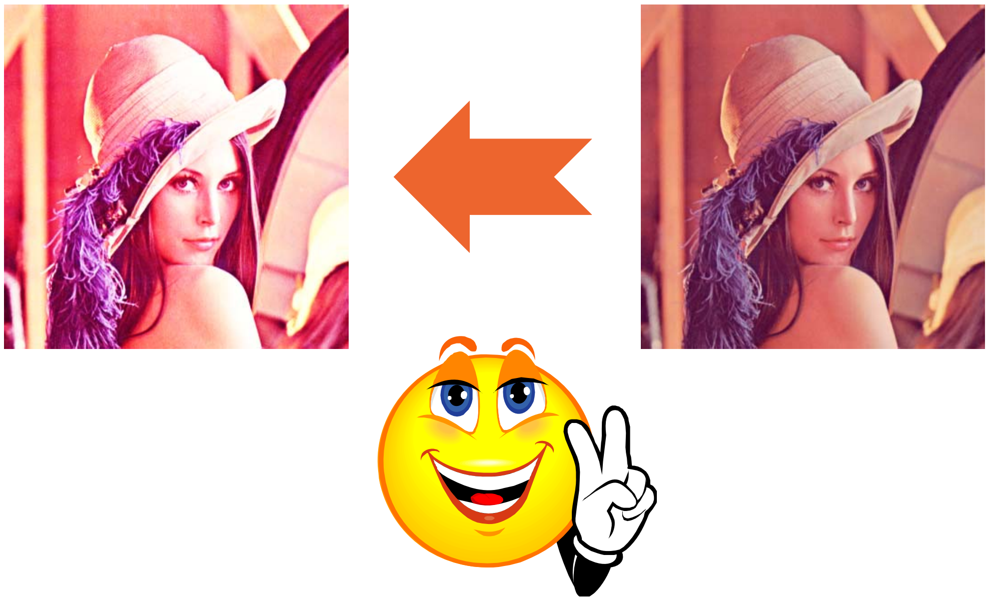
Brightness and Contrast Adjustment in a Surface

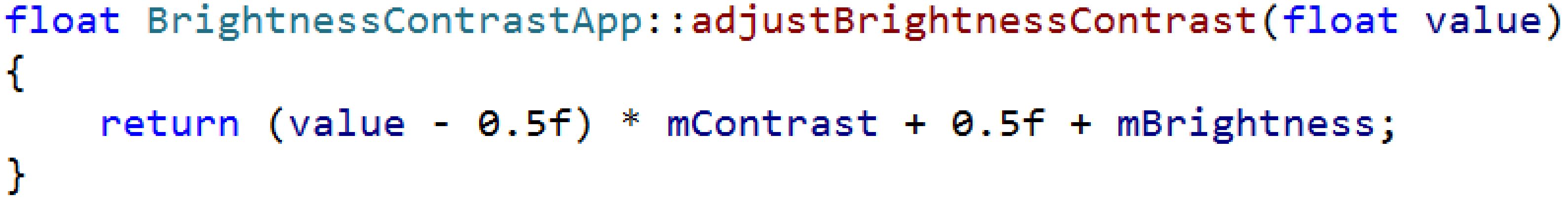


Transform R,G,B components in current pixel

```
while (pixelInIter.line()) {
    pixelOutIter.line();

    while (pixelInIter.pixel()) {
        pixelOutIter.pixel();

        pixelOutIter.r() = adjustBrightnessContrast(pixelInIter.r());
        pixelOutIter.g() = adjustBrightnessContrast(pixelInIter.g());
        pixelOutIter.b() = adjustBrightnessContrast(pixelInIter.b());
    }
}
```

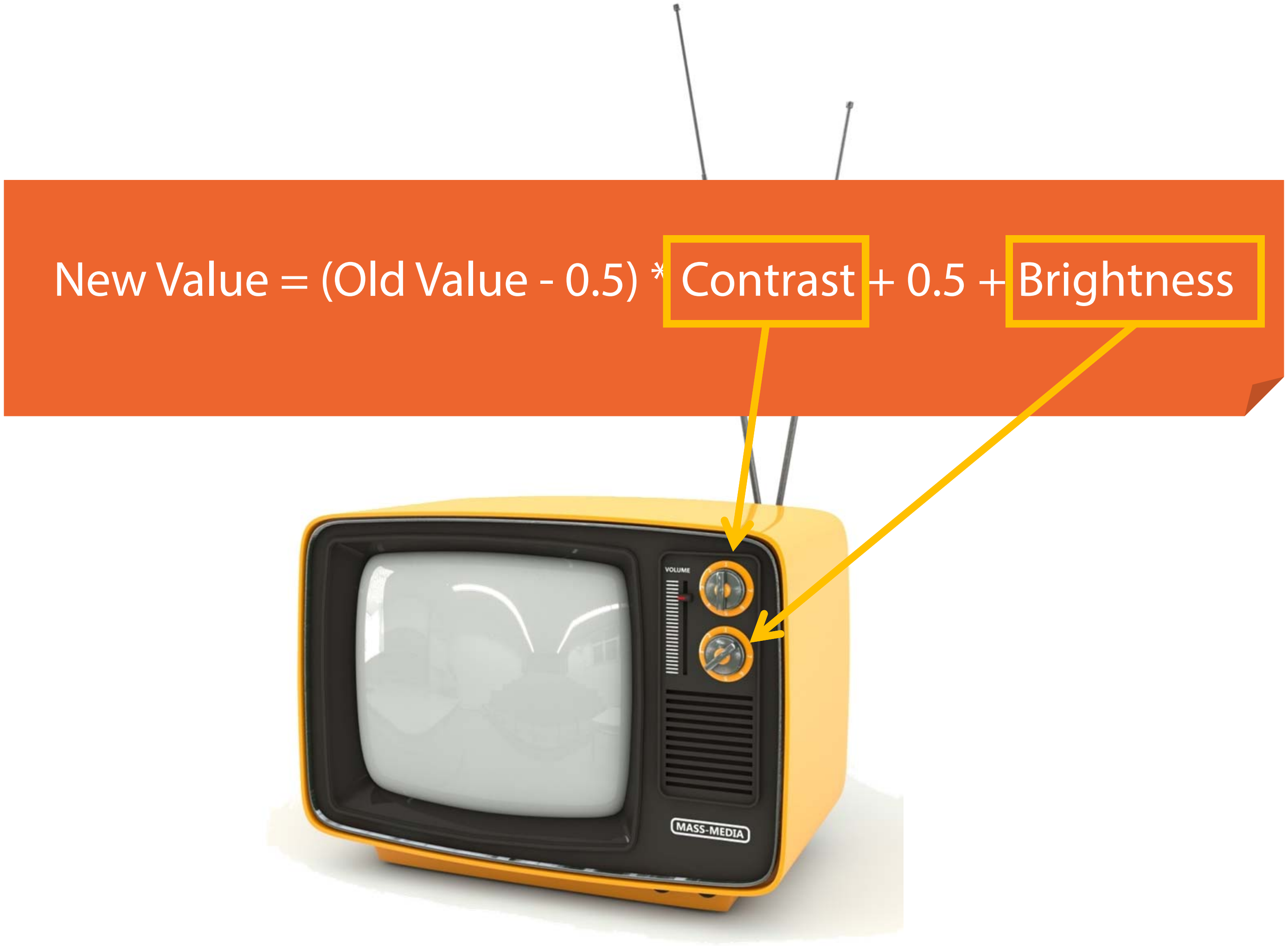
Brightness and Contrast Adjustment in a Surface





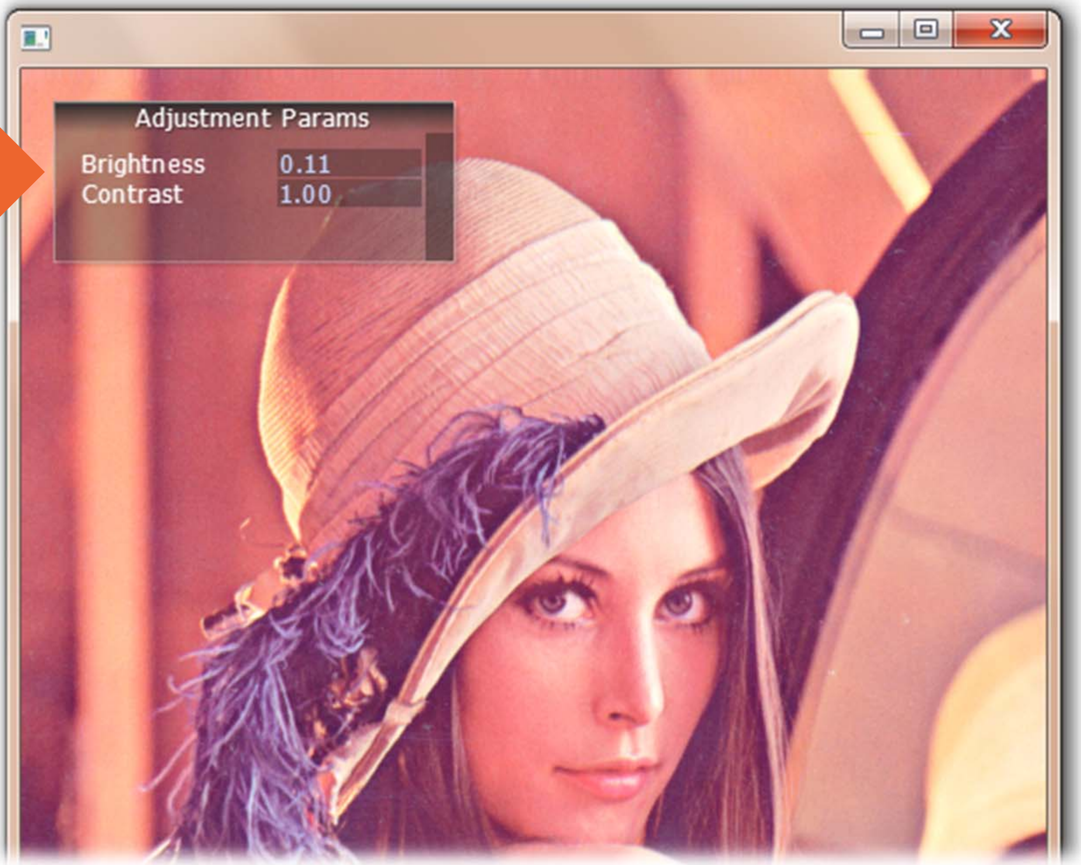
```
float BrightnessContrastApp::adjustBrightnessContrast(float value)
{
    return (value - 0.5f) * mContrast + 0.5f + mBrightness;
}
```

Brightness and Contrast Parameters



New Value = (Old Value - 0.5) * Contrast + 0.5 + Brightness

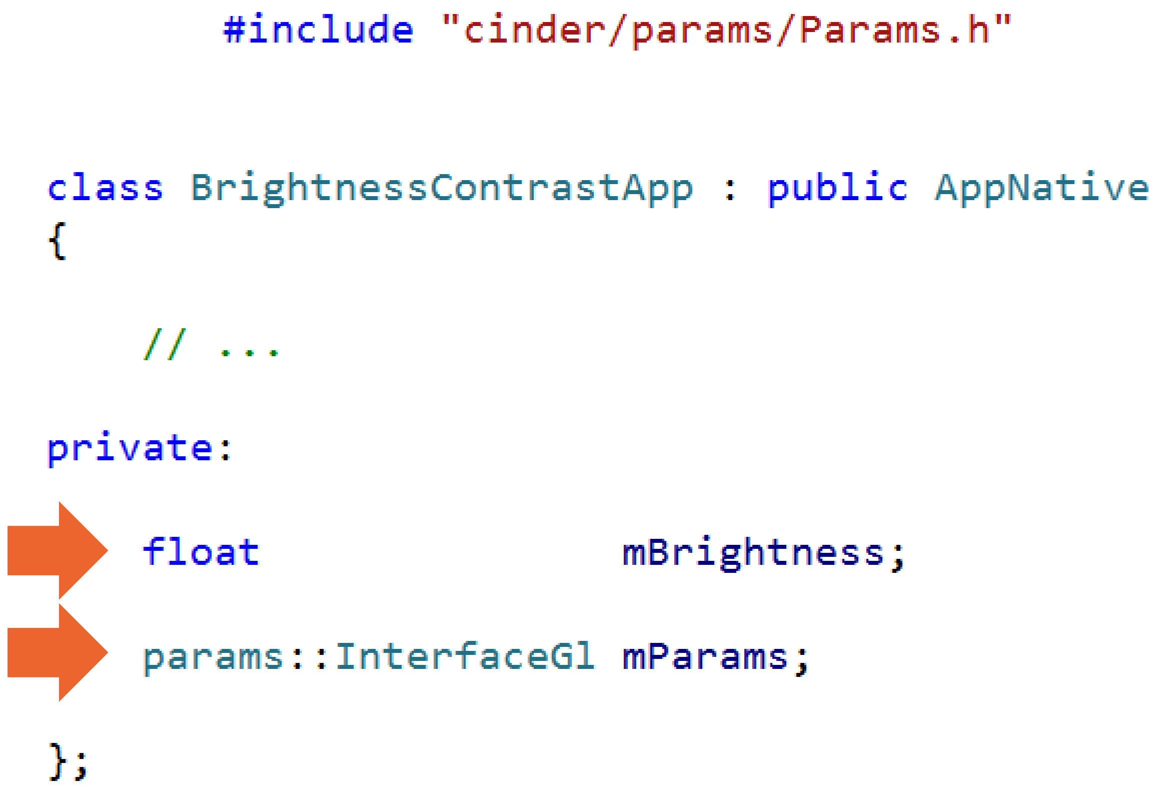
InterfaceGl



Adjustment Params

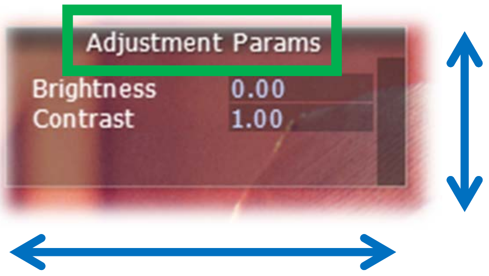
```
Brightness   0.11
Contrast     1.00
```

InterfaceGl: Preparing the Hosting Class



```
       #include "cinder/params/Params.h"

class BrightnessContrastApp : public AppNative
{

    // ...

private:

    float               mBrightness;

    params::InterfaceGl mParams;

};
```

InterfaceGl: Setting Up



Adjustment Params

Brightness 0.00

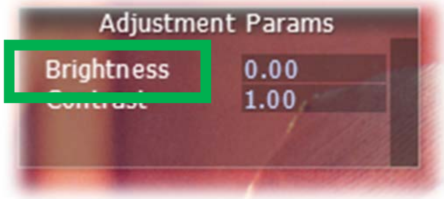
Contrast 1.00

```
void BrightnessContrastApp::setup()
{
    // ...

    mParams = params::InterfaceGl("Adjustment Params", Vec2i(200, 80));



    // ...
}
```

Title

Size

InterfaceGl: Adding Parameters

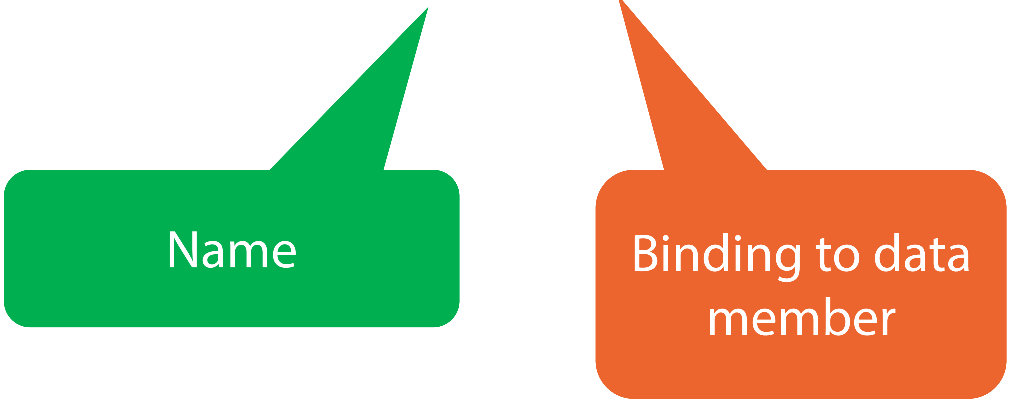


Adjustment Params

Brightness 0.00

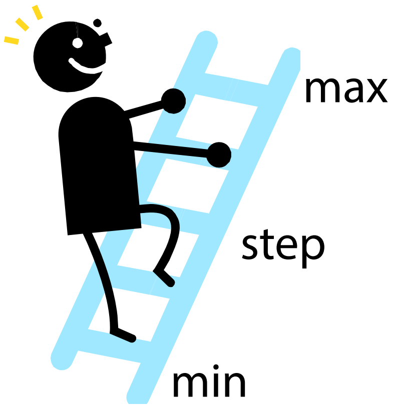
Contrast 1.00

```
mParams.addParam("Brightness",  &mBrightness, "min=-0.5 max=0.5 step=0.01");
```

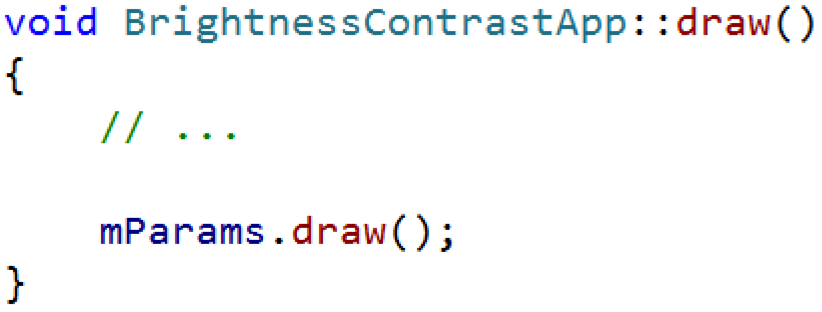


Name

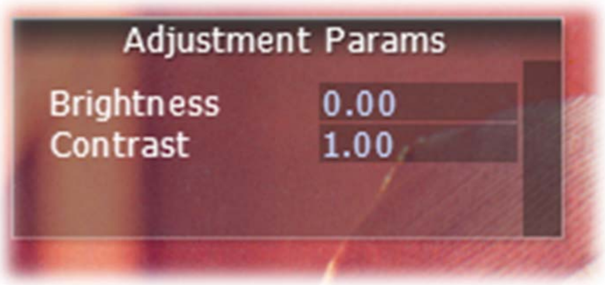
Binding to data member



InterfaceGl: Showing on Screen



```
void BrightnessContrastApp::draw()
{
    // ...

    mParams.draw();
}
```



Adjustment Params

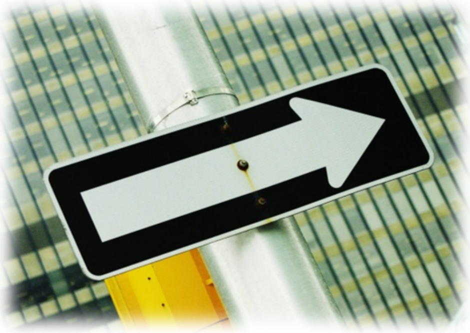
```
Brightness      0.00
Contrast        1.00
```

InterfaceGl: Recap

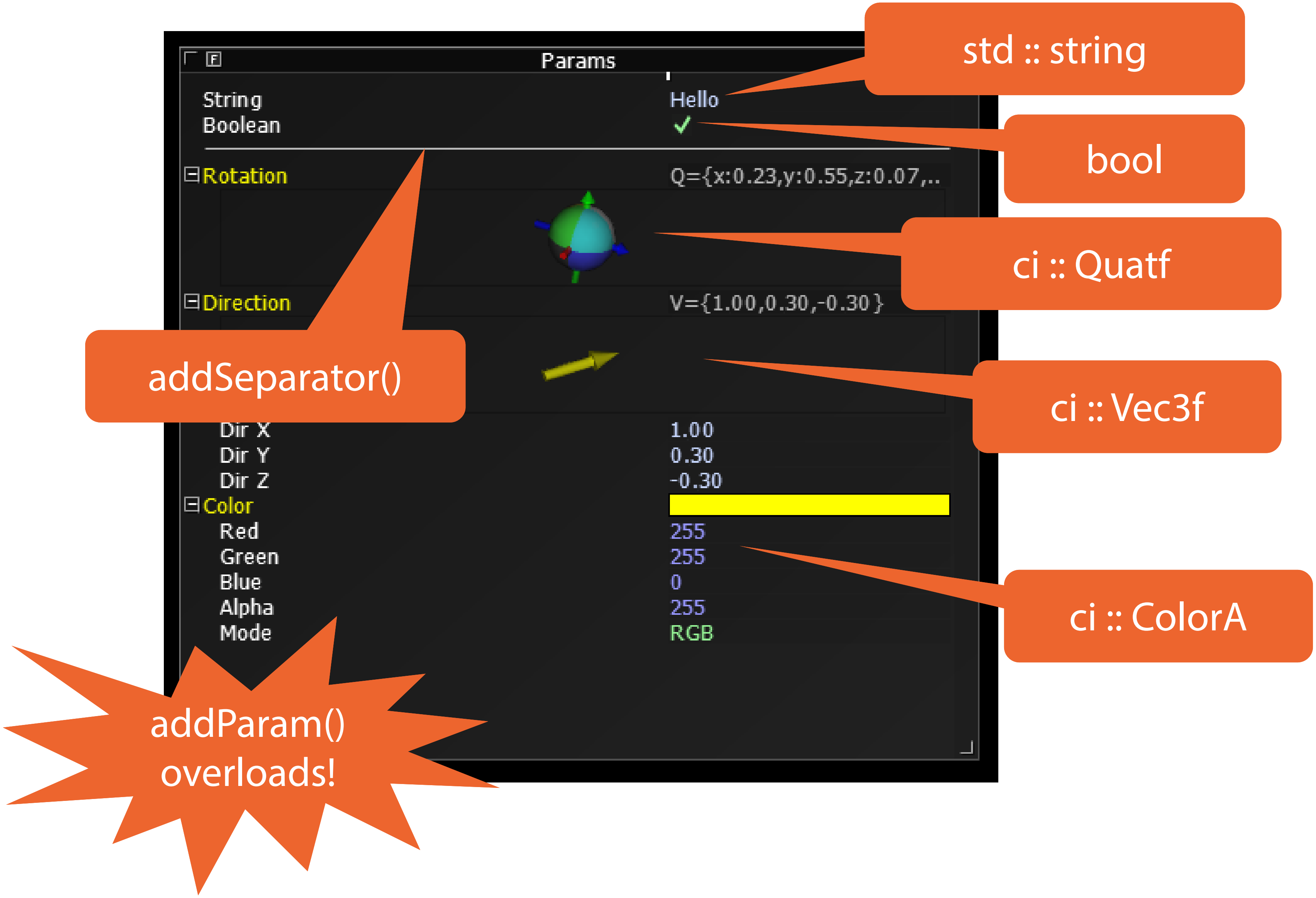
- **Define an InterfaceGl data member**
- **In app’s setup() method:**
  - Create InterfaceGl defining title and size
  - Add parameters using addParam()
- **In app’s draw() method:**
  - Display InterfaceGl GUI using draw()

InterfaceGl: There's More!

- **Specify keyboard shortcuts**
  - keyIncr / keyDecr
- **Manage more complex parameters**
  - Vectors
  - Colors
  - Quaternions (e.g. for rotations)
  - …



More InterfaceGl Types



Params

String Hello
Boolean ✓

Rotation Q={x:0.23,y:0.55,z:0.07,..

Direction V={1.00,0.30,-0.30}
Dir X 1.00
Dir Y 0.30
Dir Z -0.30

Color
Red 255
Green 255
Blue 0
Alpha 255
Mode RGB

std :: string

bool

ci :: Quatf

ci :: Vec3f

ci :: ColorA

addSeparator()

addParam()
overloads!

Demo: Brightness and Contrast Adjustment

Summary

- **Image processing basics**
  - Pixel transforms
- **Brightness and contrast adjustment**
  - Brightness-contrast equation
  - C++/Cinder implementation
- **InterfaceGl**
  - GUI to control parameters (e.g. brightness and contrast)
  - addParam()
  - draw()