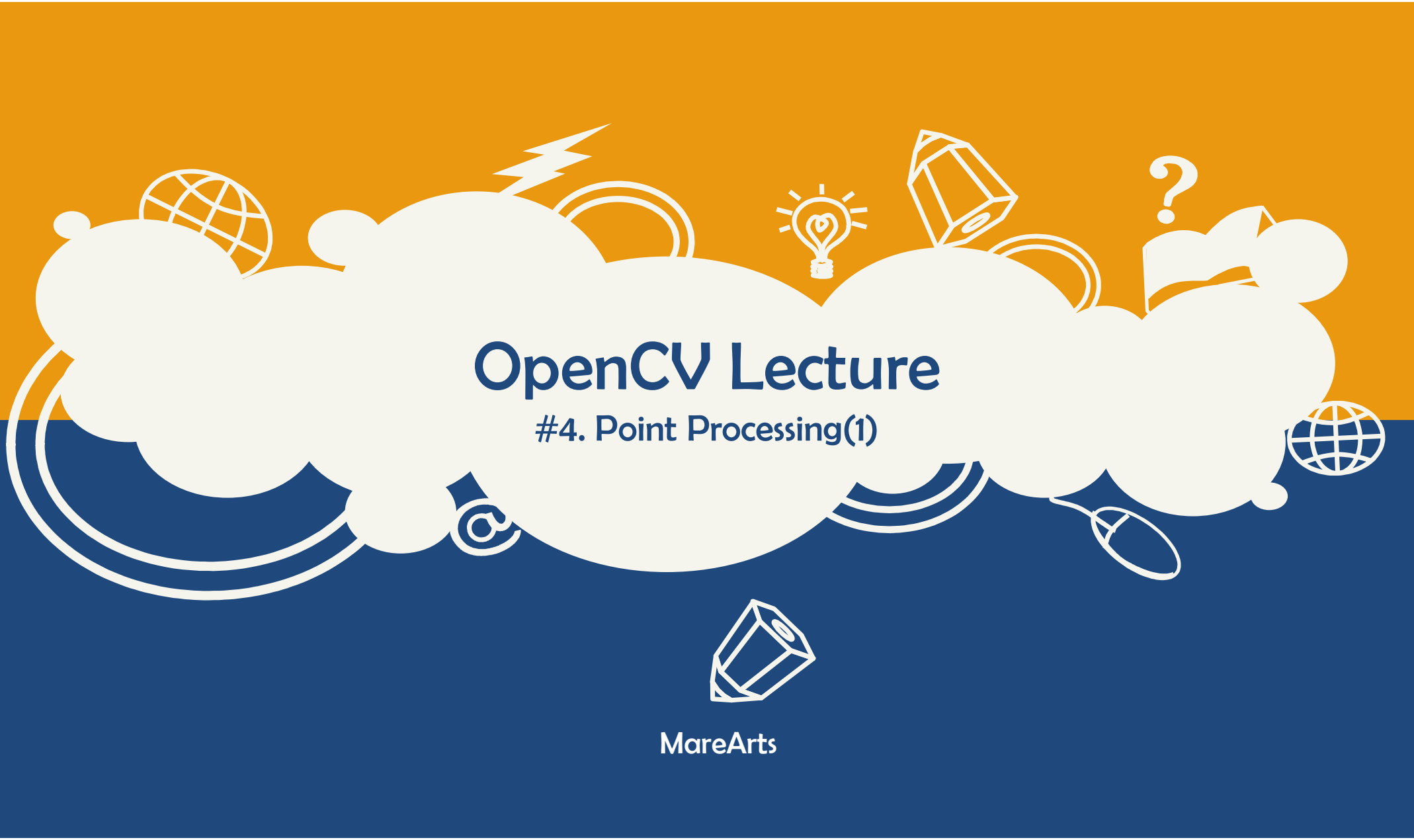


OpenCV Lecture

#4. Point Processing(1)

MareArts

Contents

- Constant(scalar) operation
- Operation between frame(images)
- Other operation
  - and, or, xor, not
  - Noise generation

Constant operation

- +, -, /, * operation by constant
  - add
    - void add(InputArray src1, InputArray src2, OutputArray dst, InputArray mask = noArray(), int dtype = -1);
    - cuda::add...

```
add(img, Scalar(200, 200, 200), img_add); //Value is between 0 and 255
//img_add = img + Scalar(200, 200, 200);
```

```
cuda::add(gimg, Scalar(-200, -200, -200), gout2);
```

  - Subtract
    - void subtract(InputArray src1, InputArray src2, OutputArray dst, InputArray mask = noArray(), int dtype = -1);
    - cuda::subtract

```
subtract(img, Scalar(200, 200, 200), img_subtract); //Value is between 0 and 255
//img_subtract = img - Scalar(200, 200, 200)
```

Refer to : http://study.marearts.com/2017/01/opencv-add-subtract-multiply-divide.html

Constant operation

- +, -, /, * operation by constant
  - absdiff
    - void absdiff(InputArray src1, InputArray src2, OutputArray dst);
    - Cuda::absdiff

```
absdiff(img, Scalar(200, 200, 200), img_absdiff); //Value is between 0 and 255
```

```
cuda::absdiff(gimg, Scalar(10, 2, 100), gout1);
```

  - *, /
    - multiply, divide functions
    - A = A * Scalar, B = B / Scalar

```
//cpu *, /
multiply(img, 20, img_mul);
divide(img, 20, img_div);
//img_mul = img * 20;
//img_div = img / 20;
```

Refer to : http://study.marearts.com/2017/01/opencv-add-subtract-multiply-divide.html

Parallel processing

parallel_for_

- void parallel_for_(const Range& range, const ParallelLoopBody& body, double nstripes=-1.)
  - Create a body by inheriting the ParallelLoopBody class.
  - Parallel processing is made by the range.

```
img_parallel = Mat(img.size(), img.type());
cv::parallel_for_(cv::Range(0, img.rows), Parallel_process(img, img_parallel, Scalar(-200, -200, -200)));
```

```
//TBB need for using this
class Parallel_process : public cv::ParallelLoopBody
{
```

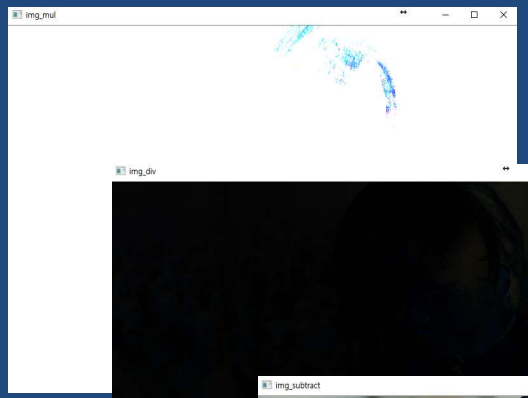
…

Refer to : http://study.marearts.com/2017/01/opencv-add-subtract-multiply-divide.html

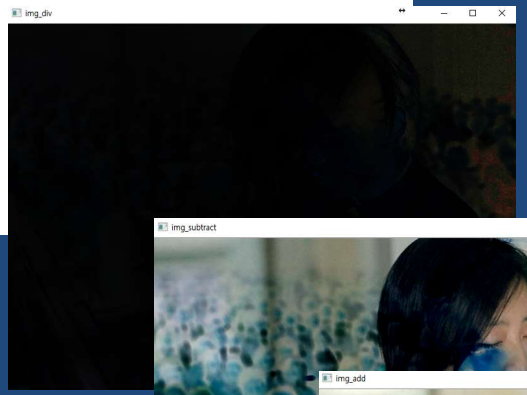
Operation between images

- +, -, /, * operation between images
  - same with scalar operation therefore
    - add, subtraction, multiply, divide
    - also use operation symbols, that is +, -, *, /
    - cuda version is also same.
    - see the example source

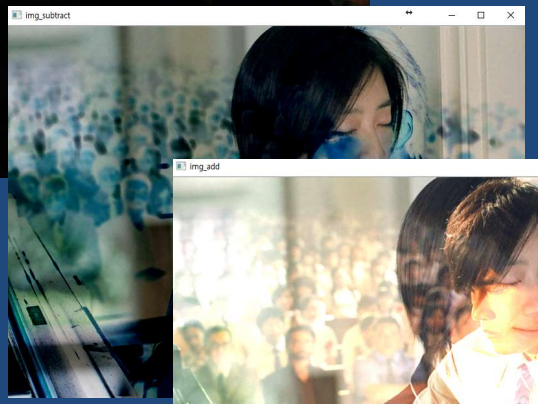
```
//If the sizes of the two images are different, an error occurs.
resize(img2, img2, Size(img.size().width, img.size().height));

add(img, img2, img_add);
//img_add = img + img2;
//cuda::add(img_cuda, img2_cuda, img_add_cuda);

subtract(img, img2, img_subtract);
//img_subtract = img - img2;
//cuda::subtract(img_cuda, img2_cuda, img_subtract_cuda);

multiply(img, img2, img_mul);
//img_mul = img * img2;
//cuda::multiply(img_cuda, img2_cuda, img_mul_cuda);

divide(img, img2, img_div);
//img_div = img / img2;
//cuda::divide(img_cuda, img2_cuda, img_div_cuda);
```



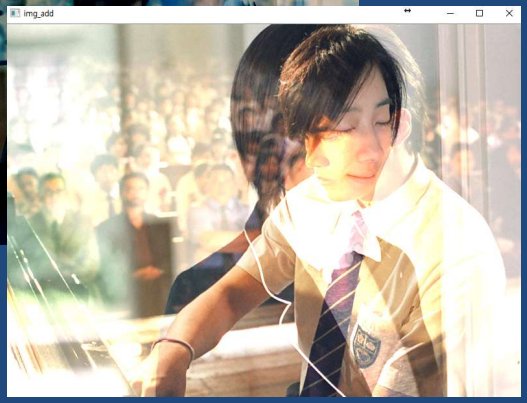
multiply



divide



subtract



add

Refer to : http://study.marearts.com/2017/02/cvlecture-example-code-operation.html

Operation between images

A little more interesting example

- Add with weighted values
  - void addWeighted(InputArray src1, double alpha, InputArray src2, double beta, double gamma, OutputArray dst, int dtype=-1)

$$\mathtt{dst}(I) = \mathtt{saturate}(\mathtt{src1}(I) * \mathtt{alpha} + \mathtt{src2}(I) * \mathtt{beta} + \mathtt{gamma})$$

  - However, it is better that the sum of alpha and beta weights should be 1.

$$g(x) = (1 - \alpha) f_0(x) + \alpha f_1(x)$$

  - see the example source code.

```
beta = (1.0 - alpha);
addWeighted(img, alpha, img2, beta, 0.0, img_wadd);
```

Refer to : http://study.marearts.com/2017/02/cvlecture-example-code-operation.html

Operation between images

And another interesting experiment

- video frame subtraction
  - As you can see in the right code
  - To subtract the old_frame and the current frame.
  - then changes the old_frame to the current frame.
  - That's all.

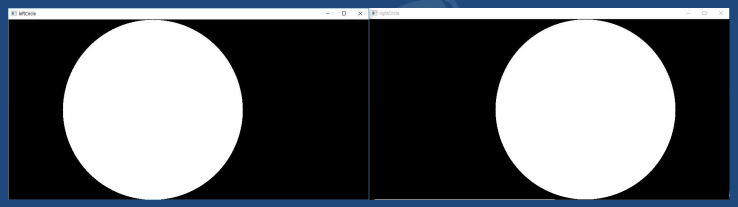
```
while (1)
{

        if (!(stream1.read(frame))) //get one frame form video
                break;

        if (old_frame.empty())
        {
                old_frame = frame.clone();
                continue;
        }


        subtract(old_frame, frame, sub_frame);
        //absdiff(old_frame, frame, absdiff_frame);

        imshow("frame", frame);
        imshow("sub_frame", sub_frame);
        //imshow("absdiff_frame", absdiff_frame);

        old_frame = frame.clone();

        if (waitKey(5) >= 0)
                break;
}
```

Refer to : http://study.marearts.com/2017/02/cvlecture-example-code-video-subtraction.html

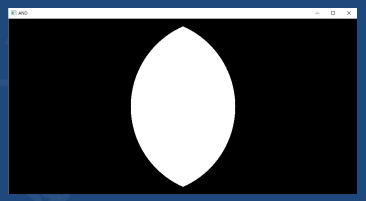
and, or, xor, not operation

- Easily implement bit operation with
  - bitwise_and (&)
  - bitwise_or (|)
  - bitwise_xor (^)
  - bitwise_not (~)

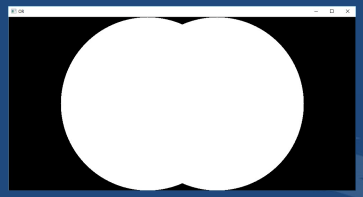
A	1	1	0	0
B	1	0	1	0
And	1	0	0	0
Or	1	1	1	0
Xor	0	1	1	0
A	1	1	1	0
not	0	0	0	1



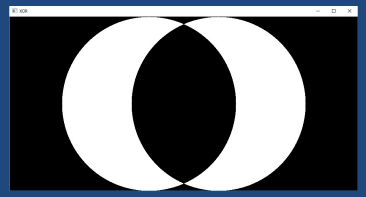
Must be size and channel same!!



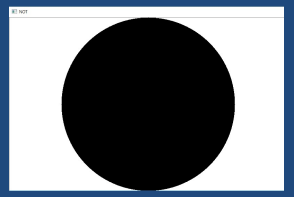
and



or



xor



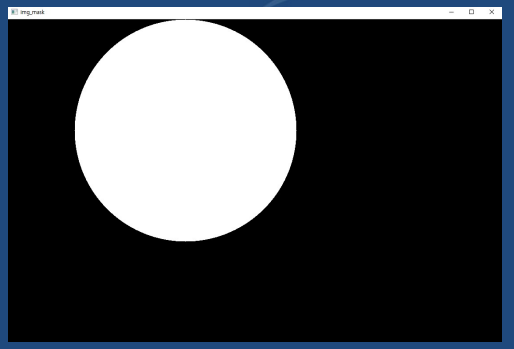
not

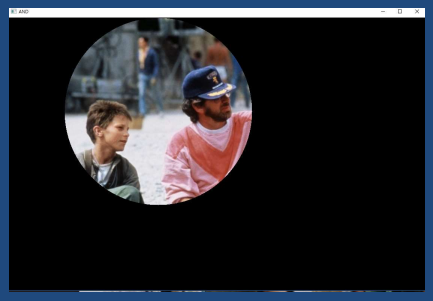
http://study.marearts.com/2017/02/opencv-lecture-and-or-xor-not-example.html

and, or, xor, or operation

- apply to bit operation to image
  - bitwise_and
  - bitwise_or
  - bitwise_xor
  - bitwise_not

Must be size and channel same!!

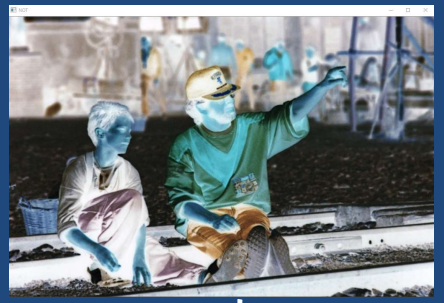




and

or

xor



not

http://study.marearts.com/2017/02/opencv-lecture-and-or-xor-not-example_14.html

other masking method

- Mat setTo and copyTo
  - setTo

```
yy =
    2 2 2
    2 2 2
    2 2 2

xx =
    0 0 0
    0 1 0
    0 0 0

yy.setTo(0, xx) = >

yy =
    2 2 2
    2 0 2
    2 2 2
```

  - copyTo

```
yy =
    2 2 2
    2 2 2
    2 2 2

xx =
    0 0 0
    0 1 0
    0 0 0

yy.copyTo(zz, xx) =>

yy =
    0 0 0
    0 2 0
    0 0 0
```

noise generation

- introduce random function in opencv
  - randn
    - Gaussian distribute rand value generation.
    - this function are input mean and standard deviation.
  - randu
    - It creates a random values between low and max value.
  - The value is generated according to the type of mat

http://study.marearts.com/2017/03/opencv-lecture-4-7-source-code-noise.html

noise generation

- see example
  - randn

```
//gaussian noise
Mat Gaussian_noise = Mat(5, 5, CV_8UC1);
double mean = 0;
double std = 10;
randn(Gaussian_noise, mean, std); //mean, std
cout << Gaussian_noise << endl;
```

```
[  0,   2,   0,   0,  12;
   0,   0,   3,   5,   0;
   8,   0,  13,   0,   0;
   0,   0,  19,   3,   0;
  15,   0,   0,   0,   0]
계속하려면 아무 키나 누르십시오 . . .
```

```
Mat Gaussian_noise = Mat(5, 5, CV_8SC1);
double mean = 0;
double std = 10;
randn(Gaussian_noise, mean, std); //mean, std
cout << Gaussian_noise << endl;
```

```
[  0,   2,  -7,  -4,  12;
  -3,  -6,   3,   5, -11;
   8,   0,  13, -20, -19;
  -5,  -7,  19,   3, -14;
  15,  -8, -25, -15, -14]
계속하려면 아무 키나 누르십시오 .
```

```
//gaussian noise
Mat Gaussian_noise = Mat(5, 5, CV_32F);
double mean = 0;
double std = 10;
randn(Gaussian_noise, mean, std); //mean, std
cout << Gaussian_noise << endl;
```

```
[-1.6030947e-008, 1.5813506, -7.0092797, -4.2859597, 12.387421;
 -2.7800822, -6.4839878, 3.0516329, 5.150146, -11.450438;
 7.6984553, -0.46431211, 13.059061, -19.59684, -18.992142;
 -5.3723822, -7.4828959, 18.964931, 2.7405043, -13.802051;
 15.36549, -7.8635716, -24.748602, -15.479244, -13.660193]
계속하려면 아무 키나 누르십시오 . . .
```

http://study.marearts.com/2017/03/opencv-lecture-4-7-source-code-noise.html

noise generation

- see example
  - randu

```
//gaussian noise
Mat Gaussian_noise = Mat(5, 5, CV_8UC1);
randu(Gaussian_noise, 5, 10); //low, high
cout << Gaussian_noise << endl;
```

```
[  6,   7,   9,   9,   7;
   5,   6,   8,   6,   9;
   6,   8,   7,   8,   5;
   8,   8,   6,   7,   8;
   8,   8,   5,   9,   8]
계속하려면 아무 키나 누르십시오 . . .
```

```
Mat Gaussian_noise = Mat(5, 5, CV_32F);
randu(Gaussian_noise, 5, 10); //low, high
cout << Gaussian_noise << endl;

/*
```

```
[7.6514139, 5.9962959, 7.0052972, 9.0719252, 7.1856651;
 6.2439485, 8.8655252, 8.8104687, 6.5389724, 8.5121584;
 7.3922362, 8.9609499, 5.4292154, 5.3753014, 5.8171167;
 6.4988961, 9.5282698, 8.5484295, 5.7485628, 8.8271999;
 5.621407, 5.0186429, 7.7575679, 9.9909964, 5.794961]
계속하려면 아무 키나 누르십시오 . . .
```

```
//gaussian noise
Mat Gaussian_noise = Mat(5, 5, CV_8SC1);
randu(Gaussian_noise, 5, 10); //low, high
cout << Gaussian_noise << endl;

/*
```

선택 C:₩WINDOWS₩system32₩cmd.exe

```
[  6,   7,   9,   9,   7;
   5,   6,   8,   6,   9;
   6,   8,   7,   8,   5;
   8,   8,   6,   7,   8;
   8,   8,   5,   9,   8]
계속하려면 아무 키나 누르십시오 . . .
```

http://study.marearts.com/2017/03/opencv-lecture-4-7-source-code-noise.html

noise generation

- Let's make noise on Origin image
  - make noise mat as the same size of origin image.
  - and, noise is reflected by 'and operation'

```
//noise adapt
Mat Gaussian_noise = Mat(img.size(), img.type());
double mean = 0;
double std = 10;
randn(Gaussian_noise, mean, std); //mean, std
Mat colorNoise = img + Gaussian_noise;
```

origin

noise add

http://study.marearts.com/2017/03/opencv-lecture-4-7-source-code-noise.html

noise generation

- Other example using iteration
  - this method is using rand function in C++
    - but this is not good for performance..

```
//another noise
int rsize = 1000;
//initialize random seed:
srand( time(NULL) ); //#include <time.h>
Mat imgD = img.clone();
for (int i = 0; i < rsize; ++i)
{
    //but x,y value will be duplicate
    int x = rand()%img.cols; //0 ~ img.cols-1
    int y = rand()%img.rows; //0 ~ img.rows-1

    cout << x << " " << y << " / " << img.cols << " " << img.rows << endl;

    imgD.at< Vec3b >(y, x)[0] = 255;
    imgD.at< Vec3b >(y, x)[1] = 255;
    imgD.at< Vec3b >(y, x)[2] = 255;
}
```

noise add

noise generation

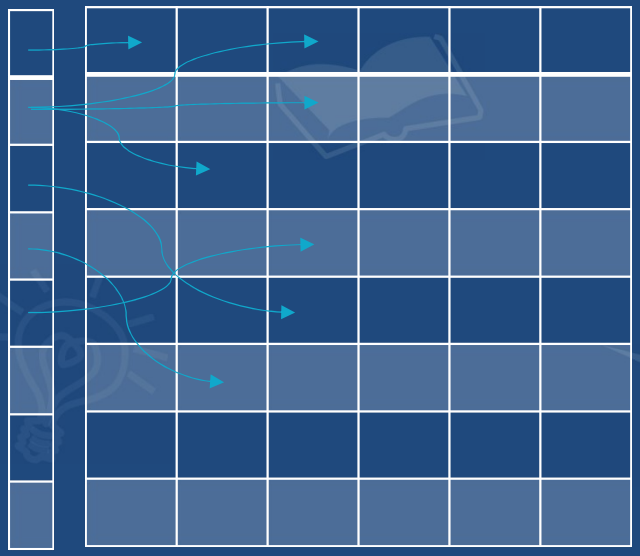
- So apply randn
  - this example will use another functions
    - minMaxIdx : this identify the maximum and minimum values
    - threshold : this is binary to two values

```
Mat noiseGray = img.clone();
cvtColor(noiseGray, noiseGray, CV_RGB2GRAY);
Mat Gaussian_noise2 = Mat(noiseGray.rows, noiseGray.cols, CV_8UC1);
double mean2 = 0;
double std2 = 10;
randn(Gaussian_noise2, mean2, std2); //mean, std
double minV, maxV;
minMaxIdx(Gaussian_noise2, &minV, &maxV);
cout << "min : " << minV << " max :" << maxV << endl;
threshold(Gaussian_noise2, Gaussian_noise2, (minV + maxV) / 2, 255, CV_THRESH_BINARY);


noiseGray = noiseGray + Gaussian_noise2;
```

noise add

Look up Table

- What is Look up table?
  - pre calculated table for repeated operation results
  - ex)



```
unsigned char O[10000][10000] = { 1, 2, };

//very slow;
for (int i = 0; i < 10000; ++i)
{
    for (int j = 0; j < 10000; ++j)
    {
        int t = O[i][j] * 1.14;
        t = (t > 255) ? 255 : t;
        O[i][j] = t;
    }
}
```

```
//fast
unsigned char LUT[256];

for (int i = 0; i < 256; ++i)
{
    int t = i * 1.14;
    t = (t > 255) ? 255 : t;
    LUT[i] = t;
}

for (int i = 0; i < 10000; ++i)
{
    for (int j = 0; j < 10000; ++j)
    {
        O[i][j] = LUT[ O[i][j] ];
    }
}
```

http://study.marearts.com/2018/05/loop-lut-processing-time-compare.html

Look up Table

- Look up table application
  - applyColorMap
    - http://docs.opencv.org/2.4/modules/contrib/doc/facerec/colormaps.html

Class	Scale
COLORMAP_AUTUMN	
COLORMAP_BONE	
COLORMAP_COOL	
COLORMAP_HOT	
COLORMAP_HSV	
COLORMAP_JET	
COLORMAP_OCEAN	
COLORMAP_PINK	
COLORMAP_RAINBOW	
COLORMAP_SPRING	
COLORMAP_SUMMER	
COLORMAP_WINTER	

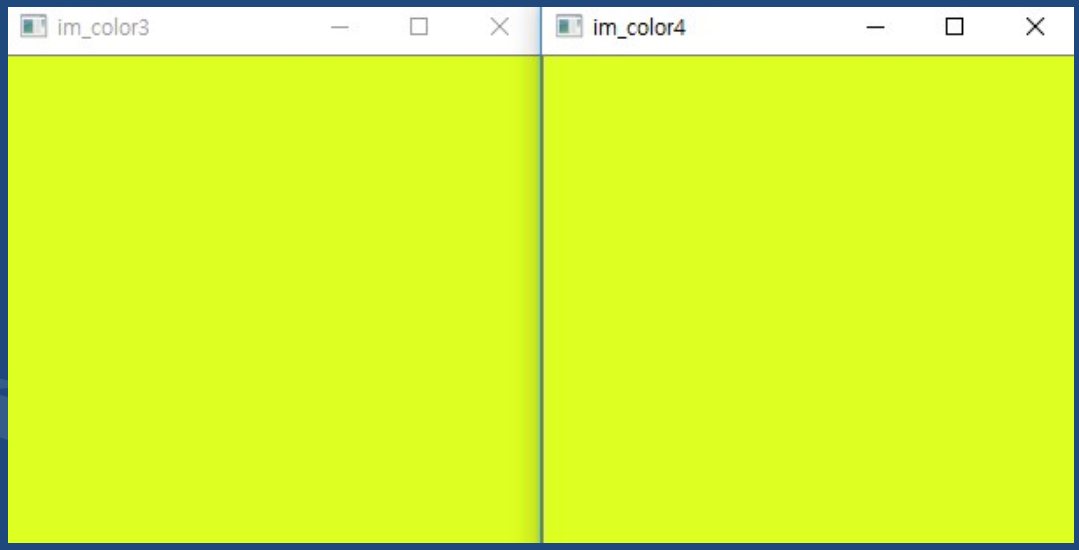

```
//image show.
Mat img = imread("AbyssCGI.jpg");

namedWindow("img", 0);
imshow("img", img);

//////////////////applyColorMap
//basic usage
Mat im_color;
applyColorMap(img, im_color, COLORMAP_AUTUMN);
namedWindow("im_color", 0);
imshow("im_color", im_color);
```

http://study.marearts.com/2018/05/example-for-applycolormap-usage.html

Look up Table

- Look up table application
  - applyColorMap
    - different result when image color and gray?
      - no! same result.

```
//////////////////// gray & color test!!
Mat img2(100, 100, CV_8UC3);
img2.setTo(Scalar(0, 128, 255));
Mat im_color3;
applyColorMap(img2, im_color3, COLORMAP_JET);

namedWindow("im_color3", 0);
imshow("im_color3", im_color3);

//same with resutl of color
Mat img2_gray;
cvtColor(img2, img2_gray, CV_RGB2GRAY);
Mat im_color4;
applyColorMap(img2, im_color4, COLORMAP_JET);
namedWindow("im_color4", 0);
imshow("im_color4", im_color4);
////////////////////////
```



http://study.marearts.com/2018/05/example-for-applycolormap-usage.html

Our LUT(Look Up Table)

- Let's make our LUT
  - 1. make LUT
  - 2. adapt Our LUT to image using LUT function
    - different result when image color and gray?
      - yes! different result.

```
Mat img = imread("AbyssCG12.jpg");

Mat lookUpTable(1, 256, CV_8U);
uchar* p = lookUpTable.data;
int factor = 256 / 10;
for (int i = 0; i < 256; ++i)
{
    p[i] = factor * (i / factor);
    //printf("[%d] = %d ₩n", i, p[i]);
}

Mat reduced;
LUT(img, lookUpTable, reduced);
```

http://study.marearts.com/2018/05/opencv-lutlook-up-table-example-code.html

Thank you.

- See you later
  - Do not forget your assignment!!
  - I will miss you very much!!

Jinhae Marine Park