

#include <opencv2/core/core.hpp>

#include <opencv2/imgproc/imgproc.hpp>

#include <opencv2/highgui/highgui.hpp>

using namespace cv;

void imageAdd(Mat& image1, Mat& image2, Mat& result);

int main()

{

Mat img[20];// , output;

img[0] = imread("C:\\Users\\User\\Desktop\\visual\_stdio\\Project1\\Project1\\lenna\_noise\\lenna\_gray\_00.png", IMREAD\_COLOR);

img[1] = imread("C:\\Users\\User\\Desktop\\visual\_stdio\\Project1\\Project1\\lenna\_noise\\lenna\_gray\_01.png", IMREAD\_COLOR);

img[2] = imread("C:\\Users\\User\\Desktop\\visual\_stdio\\Project1\\Project1\\lenna\_noise\\lenna\_gray\_02.png", IMREAD\_COLOR);

img[3] = imread("C:\\Users\\User\\Desktop\\visual\_stdio\\Project1\\Project1\\lenna\_noise\\lenna\_gray\_03.png", IMREAD\_COLOR);

img[4] = imread("C:\\Users\\User\\Desktop\\visual\_stdio\\Project1\\Project1\\lenna\_noise\\lenna\_gray\_04.png", IMREAD\_COLOR);

img[5] = imread("C:\\Users\\User\\Desktop\\visual\_stdio\\Project1\\Project1\\lenna\_noise\\lenna\_gray\_05.png", IMREAD\_COLOR);

img[6] = imread("C:\\Users\\User\\Desktop\\visual\_stdio\\Project1\\Project1\\lenna\_noise\\lenna\_gray\_06.png", IMREAD\_COLOR);

img[7] = imread("C:\\Users\\User\\Desktop\\visual\_stdio\\Project1\\Project1\\lenna\_noise\\lenna\_gray\_07.png", IMREAD\_COLOR);

img[8] = imread("C:\\Users\\User\\Desktop\\visual\_stdio\\Project1\\Project1\\lenna\_noise\\lenna\_gray\_08.png", IMREAD\_COLOR);

img[9] = imread("C:\\Users\\User\\Desktop\\visual\_stdio\\Project1\\Project1\\lenna\_noise\\lenna\_gray\_09.png", IMREAD\_COLOR);

img[10] = imread("C:\\Users\\User\\Desktop\\visual\_stdio\\Project1\\Project1\\lenna\_noise\\lenna\_gray\_10.png", IMREAD\_COLOR);

img[11] = imread("C:\\Users\\User\\Desktop\\visual\_stdio\\Project1\\Project1\\lenna\_noise\\lenna\_gray\_11.png", IMREAD\_COLOR);

img[12] = imread("C:\\Users\\User\\Desktop\\visual\_stdio\\Project1\\Project1\\lenna\_noise\\lenna\_gray\_12.png", IMREAD\_COLOR);

img[13] = imread("C:\\Users\\User\\Desktop\\visual\_stdio\\Project1\\Project1\\lenna\_noise\\lenna\_gray\_13.png", IMREAD\_COLOR);

img[14] = imread("C:\\Users\\User\\Desktop\\visual\_stdio\\Project1\\Project1\\lenna\_noise\\lenna\_gray\_14.png", IMREAD\_COLOR);

img[15] = imread("C:\\Users\\User\\Desktop\\visual\_stdio\\Project1\\Project1\\lenna\_noise\\lenna\_gray\_15.png", IMREAD\_COLOR);

img[16] = imread("C:\\Users\\User\\Desktop\\visual\_stdio\\Project1\\Project1\\lenna\_noise\\lenna\_gray\_16.png", IMREAD\_COLOR);

img[17] = imread("C:\\Users\\User\\Desktop\\visual\_stdio\\Project1\\Project1\\lenna\_noise\\lenna\_gray\_17.png", IMREAD\_COLOR);

img[18] = imread("C:\\Users\\User\\Desktop\\visual\_stdio\\Project1\\Project1\\lenna\_noise\\lenna\_gray\_18.png", IMREAD\_COLOR);

img[19] = imread("C:\\Users\\User\\Desktop\\visual\_stdio\\Project1\\Project1\\lenna\_noise\\lenna\_gray\_19.png", IMREAD\_COLOR);

int width = img[0].cols;

int height = img[1].rows;

Mat output(height, width, CV\_8UC1);

imageAdd(img[0],img[1], output);

for(int i=2; i<20; i++)

imageAdd(img[i], output, output);

imshow("original", img[0]);

imshow("heart", img[1]);

imshow("add Image", output);

waitKey();

}

void imageAdd(Mat& image1, Mat& image2, Mat& result)

{

int numOfLines = image1.rows; // number of lines in the image

int numOfPixels = image1.cols; // number of pixels per a line

result.create(image1.rows, image1.cols, image1.type());

for (int r = 0; r < numOfLines; r++)

{

const uchar\* data1\_in = image1.ptr<uchar>(r);

const uchar\* data2\_in = image2.ptr<uchar>(r);

uchar\* data\_out = result.ptr<uchar>(r);

for (int c = 0; c < numOfPixels\*3+40; c++)

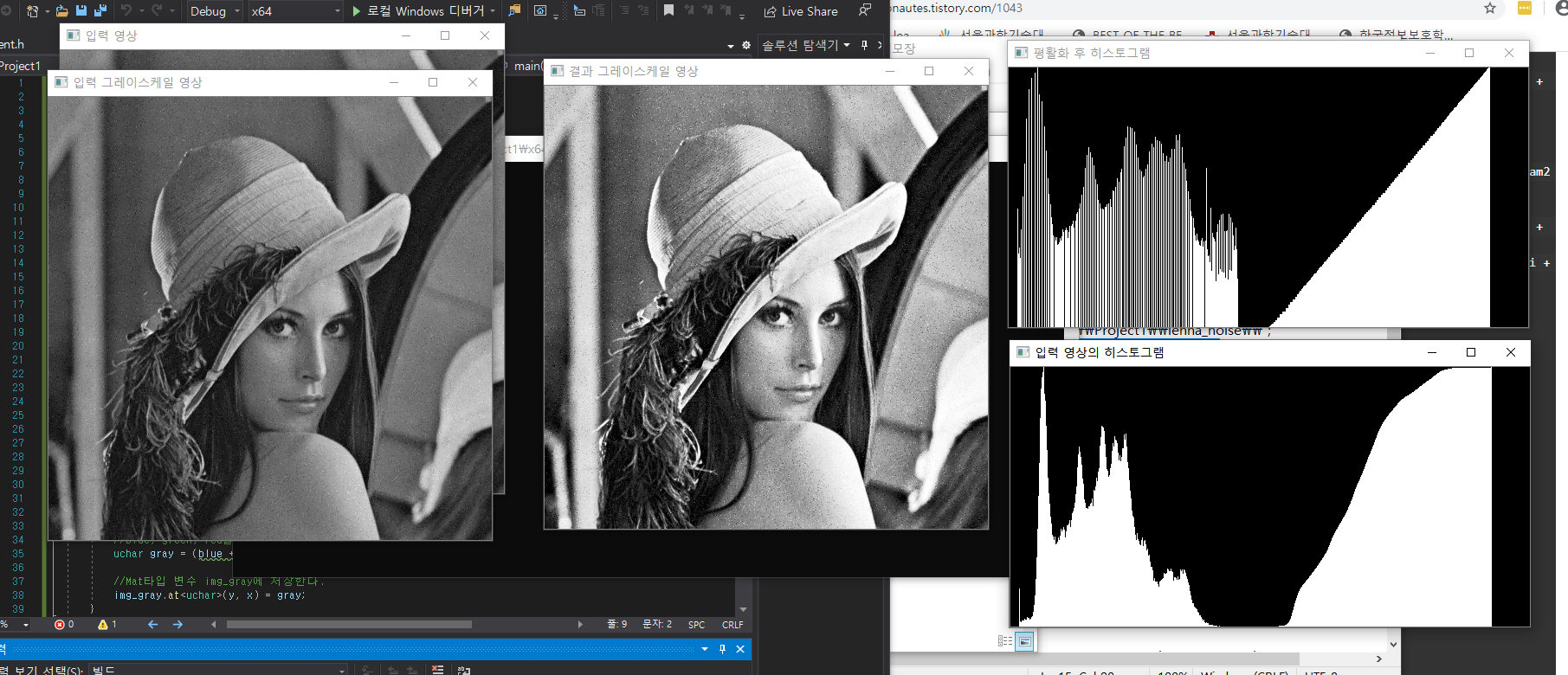
{

data\_out[c] = saturate\_cast<uchar>((data1\_in[c] + data2\_in[c])/2);

}

}

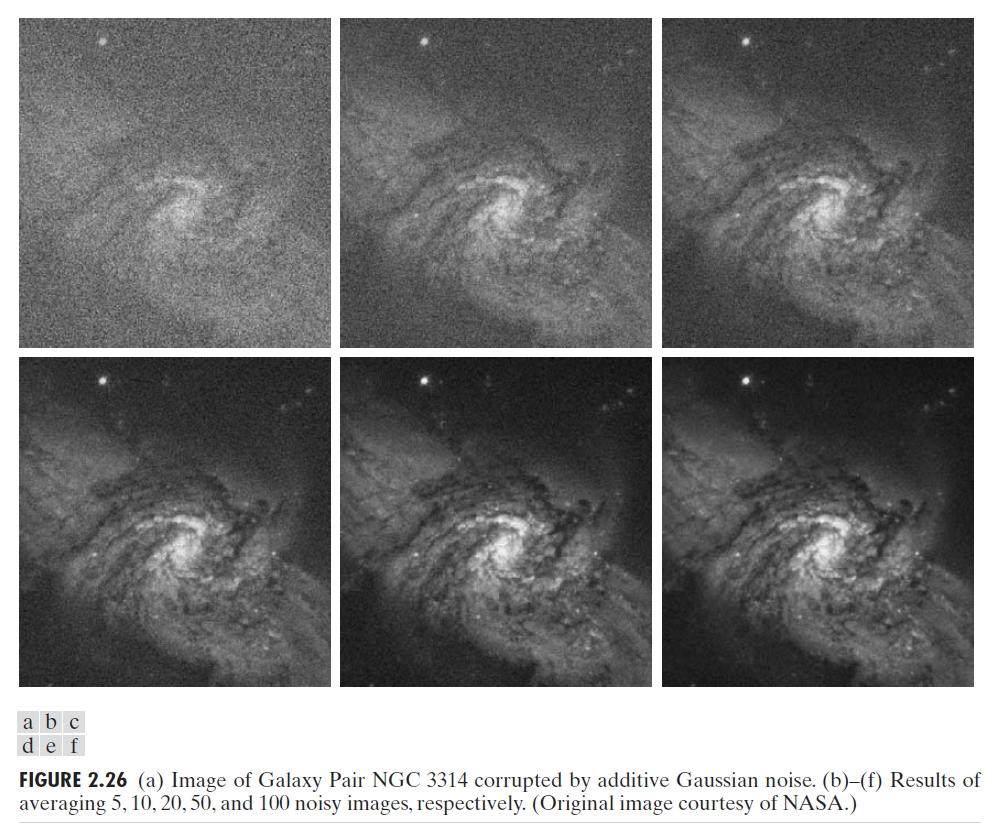
}



<https://webnautes.tistory.com/1043>

**덧셈(Addition)**

위 그림은 노이즈가 첨가된 영상과, 원본 영상, 노이즈간의 관계를 나타낸 식이다. 식의 의미를 살펴보면 이렇다. 노이즈가 낀 영상들의 평균은 원본 영상이라는 것이다. 다만, 이 경우 노이즈가 낀 영상이 1개, 2개 있다고 해서 되는 것이 아니고, 수 많은 노이즈가 낀 영상들이 필요하다. 그렇게 되면, 노이즈끼리의 상충작용이 일어나서, 원본영상에 가까워 진다는 것이다. 그리고 노이즈가 낀 영상들의 평균은 원본영상 이었으므로, 당연히, 노이즈가 낀 영상들은, 원본영상을 중심으로 노이즈만큼 차이가 날테니까, 노이즈가낀 영상들의 분산은 노이즈들의 분산들과 같다는 것을 알 수 있다. 이렇게, 노이즈가낀 영상들 수십, 수백개를 평균내어서, 노이즈가 없는 원본영상을 찾으려고 하는 과정을 **영상 평균화(image averaging)** 이라 한다. 다음 그림은 영상평균화의 예시를 나타낸다.



산술 연산

<https://sunshower76.github.io/cv(computervision)/2019/08/26/Digital-Image-Processing-Digital-image-fundamentals_4-(chapter-2)/>

<https://nauco.tistory.com/5>