#include <opencv2\core\core.hpp>

#include <opencv2\highgui.hpp>

#include <opencv2\imgproc.hpp>

#include <iostream>

#include <string>

#include <math.h>

using namespace std;

using namespace cv;

int main() {

Mat left, right, result;

float SSD, temp\_SSD;

int k;

String leftName("D:\\Program/OpenCV-3.2.0/opencv/sources/samples/data/left.png");

String rightName("D:\\Program/OpenCV-3.2.0/opencv/sources/samples/data/right.png");

left = imread(leftName.c\_str(), IMREAD\_GRAYSCALE);

right = imread(rightName.c\_str(), IMREAD\_GRAYSCALE);

result = left.clone();

//전체 그림 root 단, 끝 처리 제외(+ - 5)

for (int x = 0 + 5; x < left.rows - 5; x++) {

for (int y = 0 + 5; y < left.cols - 5; y++) {

SSD = 1000000000.0;

k = 0;

//Disparity 0 ~ 16

for (int d = 0; d <= 16; d++) {

temp\_SSD = 0;

//SSD

for (int ssdX = 0; ssdX < 5; ssdX++) {

for (int ssdY = 0; ssdY < 5; ssdY++) {

if (y + d + ssdY > left.cols - 5 - 1)

continue;

temp\_SSD += pow((left.at<uchar>(x + ssdX, y + d + ssdY) - right.at<uchar>(x + ssdX, y + ssdY)),2.0);

}

}

if (temp\_SSD < SSD) {

SSD = temp\_SSD;

k = d;

}

}

result.at<uchar>(x, y) = (uchar)k \* (255/16);

}

}

//normalize(result, result, 0, 255, NORM\_MINMAX);

//Display the result

namedWindow("Display window result", WINDOW\_AUTOSIZE);

imshow("Display window result", result);

waitKey(0);

return 1;

}

Result image

