#include "stdafx.h"

#include <opencv2\opencv.hpp>

#include "opencv2\highgui.hpp"

#include "opencv2\imgcodecs.hpp"

#include "opencv2\imgproc.hpp"

#include <iostream>

#include <iomanip>

using namespace std;

using namespace cv;

int main()

{

//initialize and allocate memory to load the video stream from camera

VideoCapture camera0(1);

VideoCapture camera1(2);

if( !camera0.isOpened() ) return 1;

if( !camera1.isOpened() ) return 1;

Size frameSize(480, 360);

camera0.set(CV\_CAP\_PROP\_FRAME\_WIDTH, frameSize.width);

camera0.set(CV\_CAP\_PROP\_FRAME\_HEIGHT, frameSize.height);

camera1.set(CV\_CAP\_PROP\_FRAME\_WIDTH, frameSize.width);

camera1.set(CV\_CAP\_PROP\_FRAME\_HEIGHT, frameSize.height);

Mat frame0,frame1;

Mat frame0gray,frame1gray;

Mat dispbm,dispsgbm;

Mat dispnorm\_bm,dispnorm\_sgbm;

Mat falseColorsMap, sfalseColorsMap;

int ndisparities = 16\*5; /\*\*< Range of disparity \*/

int SADWindowSize = 21; /\*\*< Size of the block window. Must be odd \*/

Ptr<StereoBM> sbm = StereoBM::create( ndisparities, SADWindowSize );

Ptr<StereoSGBM> sgbm = StereoSGBM::create(0, //int minDisparity

96, //int numDisparities

5, //int SADWindowSize

600, //int P1 = 0

2400, //int P2 = 0

10, //int disp12MaxDiff = 0

16, //int preFilterCap = 0

2, //int uniquenessRatio = 0

20, //int speckleWindowSize = 0

30, //int speckleRange = 0

true); //bool fullDP = false

//-- Check its extreme values

double minVal; double maxVal;

while(true)

{

//grab and retrieve each frames of the video sequentially

camera0 >> frame0;

camera1 >> frame1;

imshow("Video0", frame0);

imshow("Video1", frame1);

cvtColor(frame0,frame0gray,CV\_BGR2GRAY);

cvtColor(frame1,frame1gray,CV\_BGR2GRAY);

sbm->compute( frame0gray, frame1gray, dispbm );

minMaxLoc( dispbm, &minVal, &maxVal );

dispbm.convertTo( dispnorm\_bm, CV\_8UC1, 255/(maxVal - minVal));

sgbm->compute(frame0gray, frame1gray, dispsgbm);

minMaxLoc( dispsgbm, &minVal, &maxVal );

dispsgbm.convertTo( dispnorm\_sgbm, CV\_8UC1, 255/(maxVal - minVal));

imshow( "BM", dispnorm\_bm);

imshow( "SGBM",dispnorm\_sgbm);

//wait for 40 milliseconds

int c = cvWaitKey(40);

//exit the loop if user press "Esc" key (ASCII value of "Esc" is 27)

if(27 == char(c)) break;

}

return 0;

}