Lab3 passive two-view stereo

姓名：	吴奕谦	学号：	3170104915	专业：	计算机科学与技术
姓名	上官越	学号	3170104168	专业	计算机科学与技术
姓名	颜烨	学号	3170104149	专业	数字媒体技术
指导老师	吴鸿智	实验地点	曹光彪二期	实验时间	2019.9.16

Ⅰ、content and its purpose

1. Perform stereo calibration with OpenCV sample to get a better understanding of 3D reconstruction.
2. Compute a dense depth map / 3D point cloud from two calibrated input views via triangulation. Evaluate the impact of different parameters (e.g., patch size) over final quality.
3. Resolve the color discrepancy between two views and produce the final colored 3D point cloud, along with the cameras, in a single .ply file.We may want to perform radiometric calibration first to get a better result.

Ⅱ、environment

Dual basler dart machine vision sub3 color cameras + lens
Visual studio + opencv + pylon
Pylon viewer
matlab

Ⅲ、operation steps and results

1. Perform stereo calibration with OpenCV sample

We use the code from opencv-4.1.1\sources\samples\cpp stereo_calib.cpp to generate an exe to perform stereo calibration.

The input format of the cmd is:

stereo_calib.exe -w=7 -h=5

(where w is the number of grids in the horizontal direction , h is the number of grids in the vertical direction, they all need to reduced by one.)

We just show the most important parts of code here:

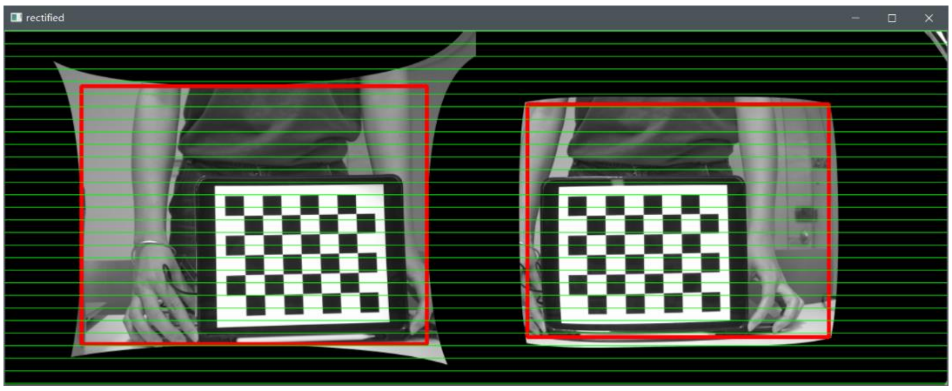
```
for( int scale = 1; scale <= maxScale; scale++ )
                {
                    Mat timg;
                    if( scale == 1 )
                        timg = img;
                    else
                        resize(img, timg, Size(), scale, scale, INTER_LINEAR_EXACT);
                    found = findChessboardCorners(timg, boardSize, corners,
                        CALIB_CB_ADAPTIVE_THRESH | CALIB_CB_FAST_CHECK |
CALIB_CB_NORMALIZE_IMAGE);
                    if( found )
                    {
                        if( scale > 1 )
                        {
                            Mat cornersMat(corners);
                            cornersMat *= 1./scale;
                        }
                        break;
                    }
                }
                if( displayCorners )
                {
                    cout << filename << endl;
                    Mat cimg, cimg1;
                    cvtColor(img, cimg, COLOR_GRAY2BGR);
                    drawChessboardCorners(cimg, boardSize, corners, found);
                    double sf = 640./MAX(img.rows, img.cols);
                    resize(cimg, cimg1, Size(), sf, sf, INTER_LINEAR_EXACT);
                    imshow("corners", cimg1);
                    char c = (char)waitKey(500);
                    if( c == 27 || c == 'q' || c == 'Q' ) //Allow ESC to quit
                        exit(-1);
                }
                else
                    putchar('.');
                if( !found )
                    break;
                cornerSubPix(img, corners, Size(11,11), Size(-1,-1),
                             TermCriteria(TermCriteria::COUNT+TermCriteria::EPS,
                                          30, 0.01));
```

```
Mat cameraMatrix[2], distCoeffs[2];
    cameraMatrix[0] = initCameraMatrix2D(objectPoints,imagePoints[0],imageSize,0);
    cameraMatrix[1] = initCameraMatrix2D(objectPoints,imagePoints[1],imageSize,0);
    Mat R, T, E, F;

    double rms = stereoCalibrate(objectPoints, imagePoints[0], imagePoints[1],
                    cameraMatrix[0], distCoeffs[0],
                    cameraMatrix[1], distCoeffs[1],
                    imageSize, R, T, E, F,
                    CALIB_FIX_ASPECT_RATIO +
                    CALIB_ZERO_TANGENT_DIST +
                    CALIB_USE_INTRINSIC_GUESS +
                    CALIB_SAME_FOCAL_LENGTH +
                    CALIB_RATIONAL_MODEL +
                    CALIB_FIX_K3 + CALIB_FIX_K4 + CALIB_FIX_K5,
                    TermCriteria(TermCriteria::COUNT+TermCriteria::EPS, 100, 1e-5) );
    cout << "done with RMS error=" << rms << endl;

// CALIBRATION QUALITY CHECK
// because the output fundamental matrix implicitly
// includes all the output information,
// we can check the quality of calibration using the
// epipolar geometry constraint: m2^t*F*m1=0
    double err = 0;
    int npoints = 0;
    vector<Vec3f> lines[2];
    for( i = 0; i < nimages; i++ )
    {
        int npt = (int)imagePoints[0][i].size();
        Mat imgpt[2];
        for( k = 0; k < 2; k++ )
        {
            imgpt[k] = Mat(imagePoints[k][i]);
            undistortPoints(imgpt[k], imgpt[k], cameraMatrix[k], distCoeffs[k], Mat(), cameraMatrix[k]);
            computeCorrespondEpilines(imgpt[k], k+1, F, lines[k]);
        }
        for( j = 0; j < npt; j++ )
        {
            double errij = fabs(imagePoints[0][i][j].x*lines[1][j][0] +
                                imagePoints[0][i][j].y*lines[1][j][1] + lines[1][j][2]) +
                           fabs(imagePoints[1][i][j].x*lines[0][j][0] +
                                imagePoints[1][i][j].y*lines[0][j][1] + lines[0][j][2]);
            err += errij;
        }
        npoints += npt;
    }
    cout << "average epipolar err = " <<    err/npoints << endl;
```

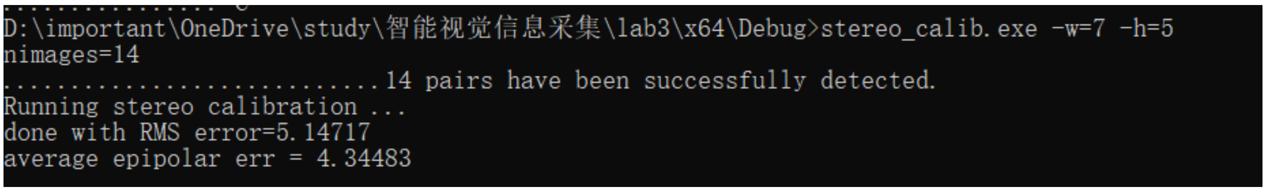
As for the performance of the calibration, firstly we consider patch-size, distance from the camera to the ipad, chessboard-size as factors that influence the result. However, while doing data analysis, we find out in surprise that some factors may also contribute to the result's performance. For example, advanced operations like resizing the photos, grayscaling may cause difference, which we do not take it serious before.

<u>The **result** is: </u> (we just show several most typical results here)

1. The black and white chessboard of 8*6 with scale of 150%:

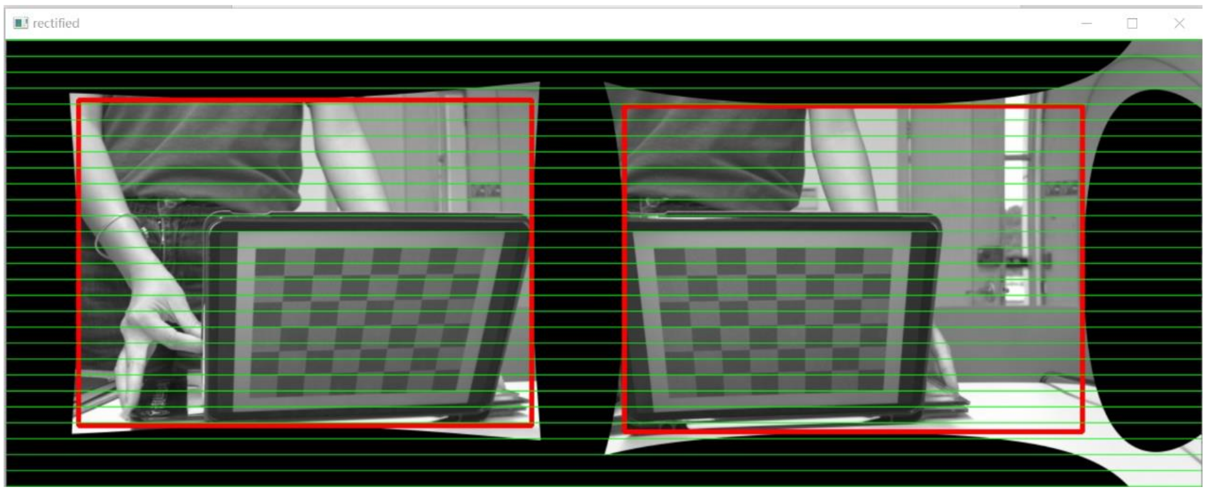


The result error of the calibration :

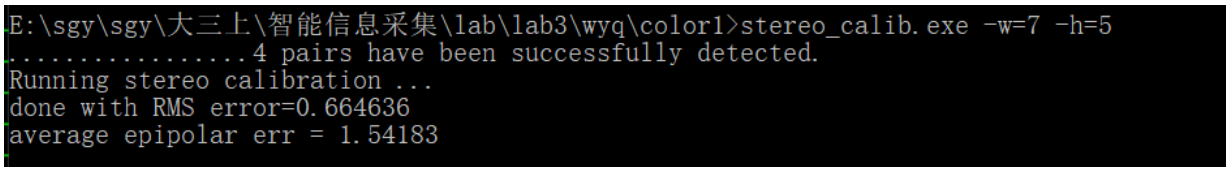


```
D:\important\OneDrive\study\智能视觉信息采集\lab3\x64\Debug>stereo_calib.exe -w=7 -h=5
nimages=14
............................14 pairs have been successfully detected.
Running stereo calibration ...
done with RMS error=5.14717
average epipolar err = 4.34483
```

2. The colorful pink and blue chessboard of 8*6 with no resize and no grayscale:



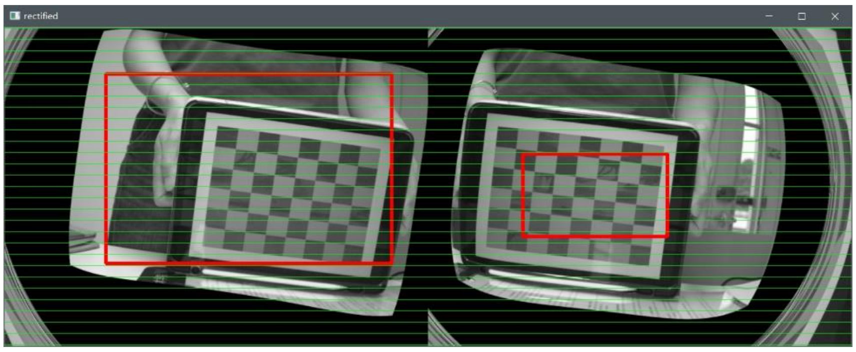
The result error of the calibration :



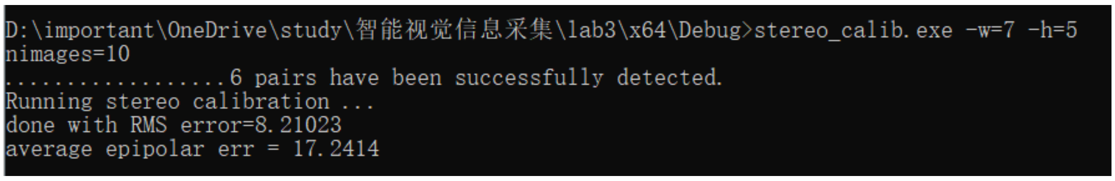
```
E:\sgy\sgy\大三上\智能信息采集\lab\lab3\wyq\color1>stereo_calib.exe -w=7 -h=5
.................4 pairs have been successfully detected.
Running stereo calibration ...
done with RMS error=0.664636
average epipolar err = 1.54183
```

We just use the origin image of colorful chessboard of 8*6, and it has a relatively ideal result

3. The colorful pink and blue chessboard of 8*6 with resize(use the resize code to resize) and grayscale:

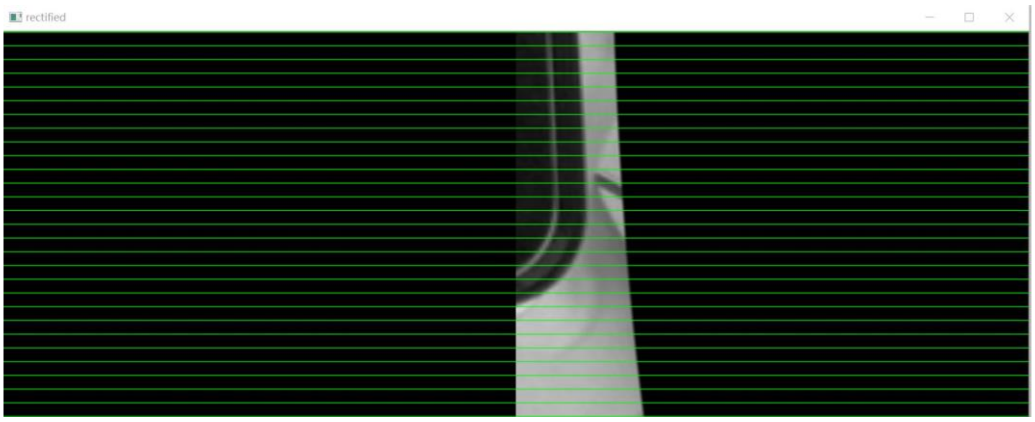


The result error of the calibration :

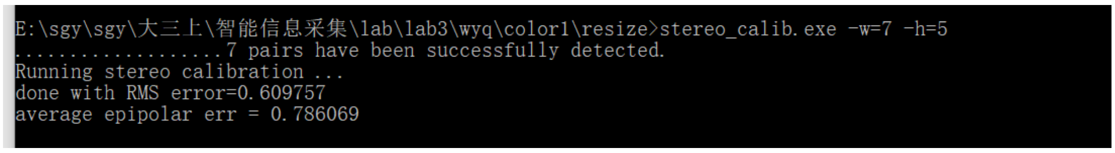


```
D:\important\OneDrive\study\智能视觉信息采集\lab3\x64\Debug>stereo_calib.exe -w=7 -h=5
nimages=10
.................6 pairs have been successfully detected.
Running stereo calibration ...
done with RMS error=8.21023
average epipolar err = 17.2414
```

4. The colorful pink and blue chessboard of 8*6 with resize(use the "draw" application to resize) and no grayscale:



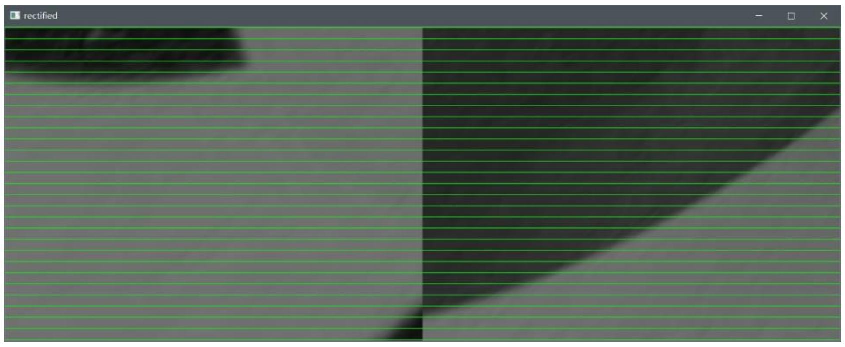
The result error of the calibration :



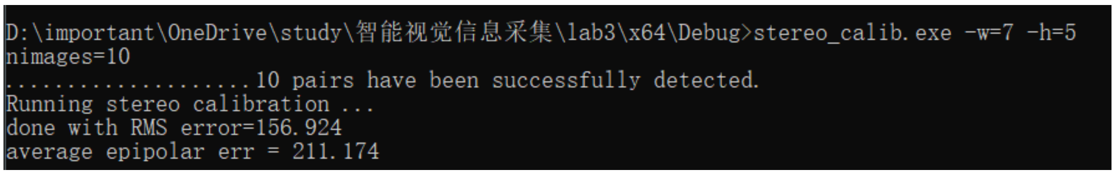
```
E:\sgy\sgy\大三上\智能信息采集\lab\lab3\wyq\color1\resize>stereo_calib.exe -w=7 -h=5
..................7 pairs have been successfully detected.
Running stereo calibration ...
done with RMS error=0.609757
average epipolar err = 0.786069
```

The weird thing is, the error is respectively small, but we get completely wrong result, we think it is caused by the way that the application use to resize images.

5. The colorful green and red chessboard of 8*6 with no resize and no grayscale:

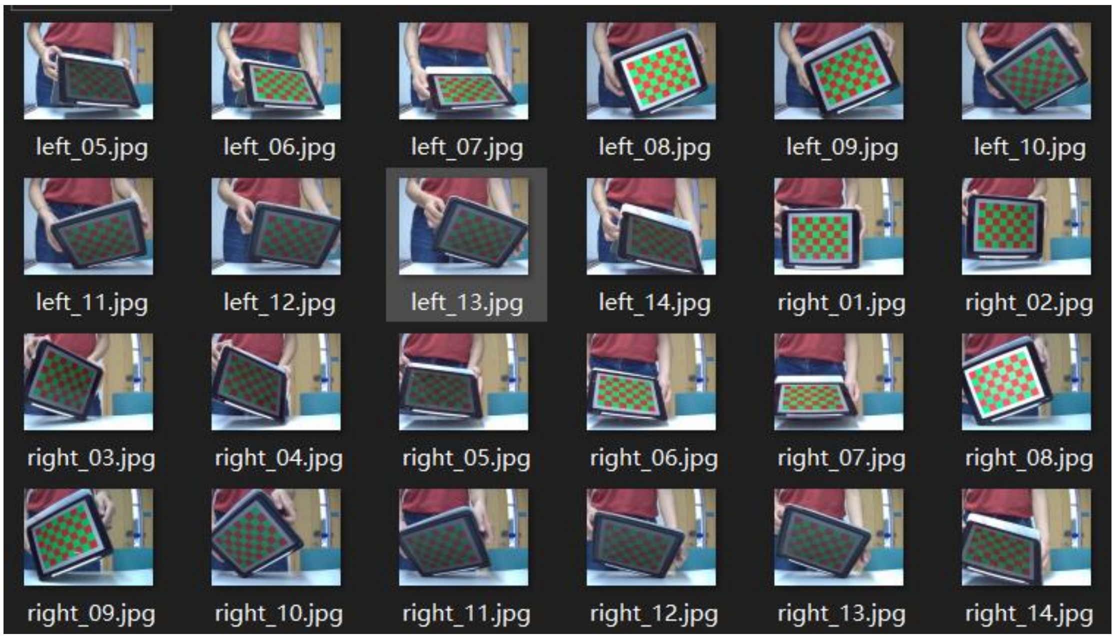


The result error of the calibration :

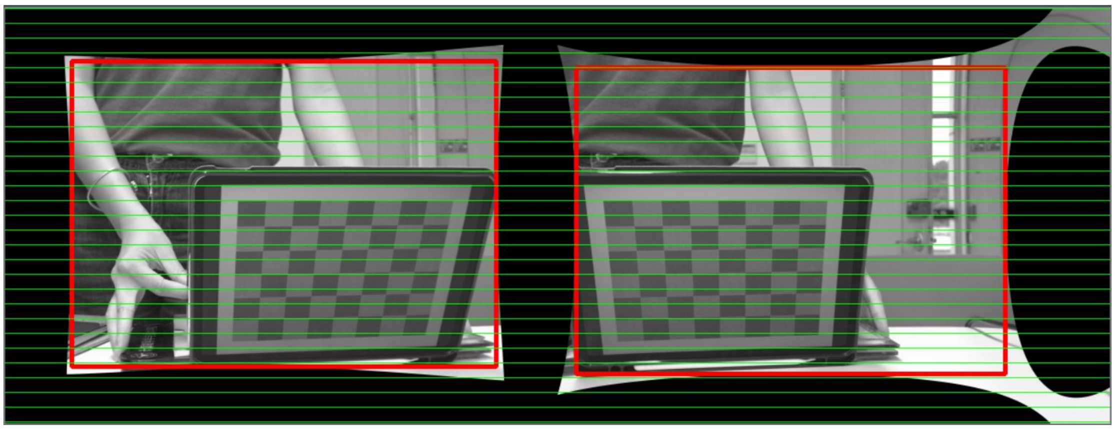


```
D:\important\OneDrive\study\智能视觉信息采集\lab3\x64\Debug>stereo_calib.exe -w=7 -h=5
nimages=10
....................10 pairs have been successfully detected.
Running stereo calibration ...
done with RMS error=156.924
average epipolar err = 211.174
```

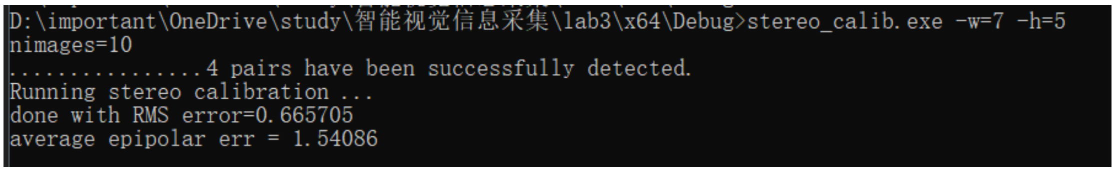
The result is not ideal, the error is too big, we think it is caused by the light and the quality of the images.



6. The colorful pink and blue chessboard of 8*6 with grayscale:



The result error of the calibration :



```
D:\important\OneDrive\study\智能视觉信息采集\lab3\x64\Debug>stereo_calib.exe -w=7 -h=5
nimages=10
...............4 pairs have been successfully detected.
Running stereo calibration ...
done with RMS error=0.665705
average epipolar err = 1.54086
```

7. **We evaluate the impact of different parameters (resize, grayscale) over final quality in the Ⅲ、further analysis section**

2. Compute a dense depth map / 3D point cloud from two calibrated input views via triangulation. Evaluate the impact of different parameters (e.g., patch size) over final quality

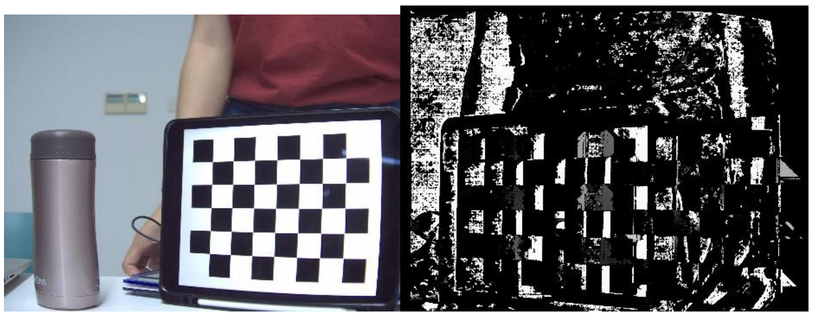
```
if( !intrinsic_filename.empty() )
    {
        // reading intrinsic parameters
        FileStorage fs(intrinsic_filename, FileStorage::READ);
        if(!fs.isOpened())
        {
            printf("Failed to open file %s\n", intrinsic_filename.c_str());
            return -1;
        }

        Mat M1, D1, M2, D2;
        fs["M1"] >> M1;
        fs["D1"] >> D1;
        fs["M2"] >> M2;
        fs["D2"] >> D2;

        M1 *= scale;
        M2 *= scale;

        fs.open(extrinsic_filename, FileStorage::READ);
        if(!fs.isOpened())
        {
            printf("Failed to open file %s\n", extrinsic_filename.c_str());
            return -1;
        }

        Mat R, T, R1, P1, R2, P2;
        fs["R"] >> R;
```

```
            fs["T"] >> T;

            stereoRectify( M1, D1, M2, D2, img_size, R, T, R1, R2, P1, P2, Q, CALIB_ZERO_DISPARITY, -1,
img_size, &roi1, &roi2 );

            Mat map11, map12, map21, map22;
            initUndistortRectifyMap(M1, D1, R1, P1, img_size, CV_16SC2, map11, map12);
            initUndistortRectifyMap(M2, D2, R2, P2, img_size, CV_16SC2, map21, map22);

            Mat img1r, img2r;
            remap(img1, img1r, map11, map12, INTER_LINEAR);
            remap(img2, img2r, map21, map22, INTER_LINEAR);

            img1 = img1r;
            img2 = img2r;
        }
if(!point_cloud_filename.empty())
        {
            printf("storing the point cloud...");
            fflush(stdout);
            Mat xyz;
            reprojectImageTo3D(disp, xyz, Q, true);
            saveXYZ(point_cloud_filename.c_str(), xyz);
            printf("\n");
            }
```

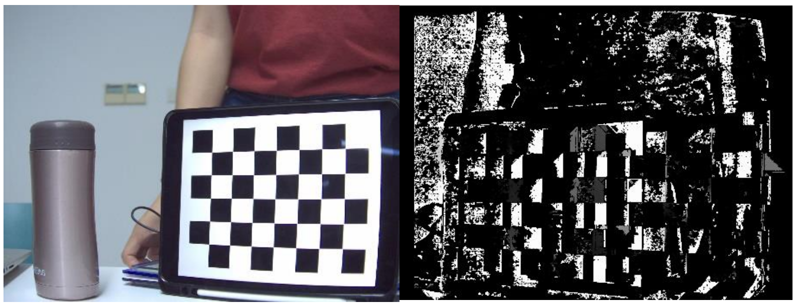
The **result** is: (we just show several results that has respectively small error in 1.here)

1. The black and white chessboard of 8*6 with scale of 150%:

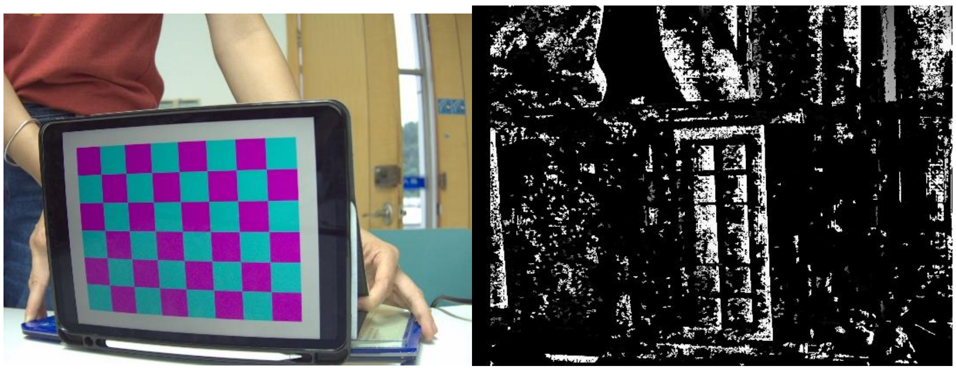
Depth map with max_disparity of 64



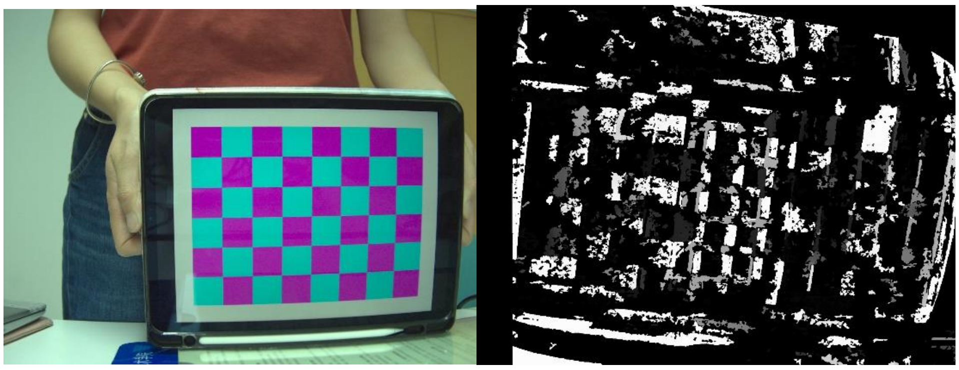
Depth map with max_disparity of 96



2. The colorful pink and blue chessboard of 8*6 with no resize and no grayscale:



3. The colorful pink and blue chessboard of 8*6 with resize(use the resize code to resize) and grayscale

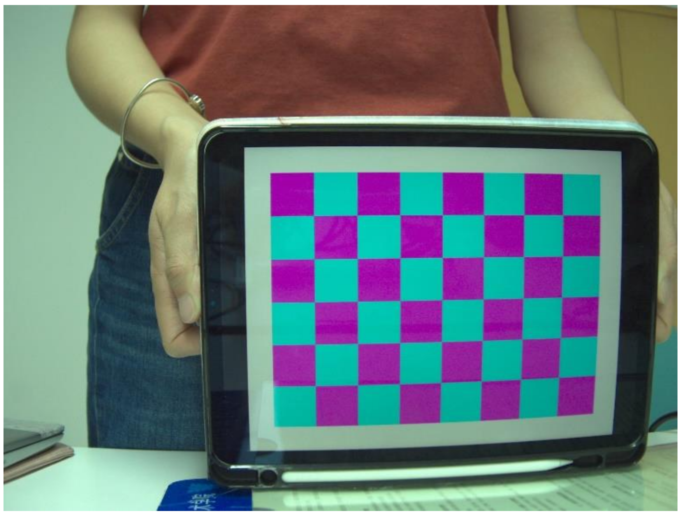


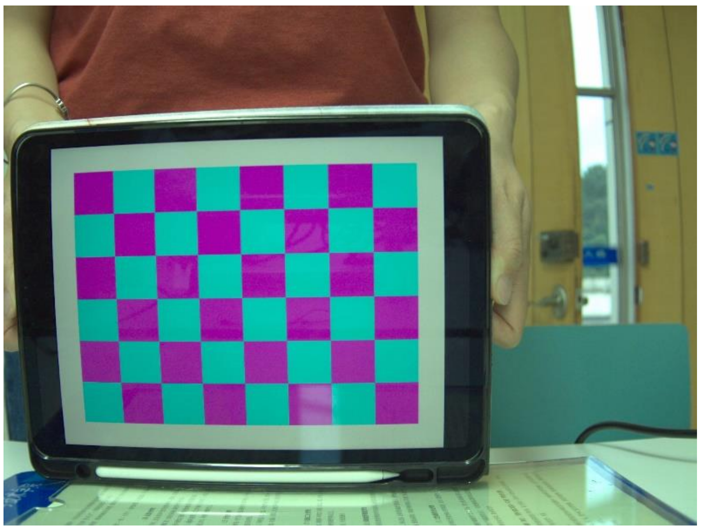
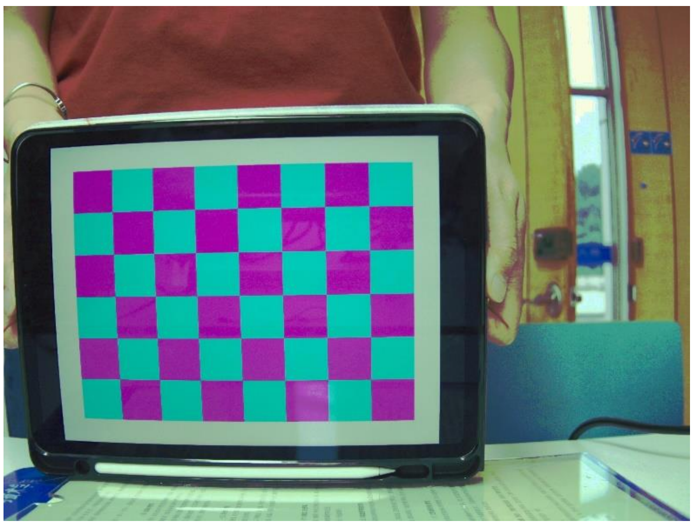
4.

3. Resolve the color discrepancy between two views and produce the final colored 3D point cloud, along with the cameras, in a single .ply file

Firstly, we apply radiometric calibration on the origin colorful images:

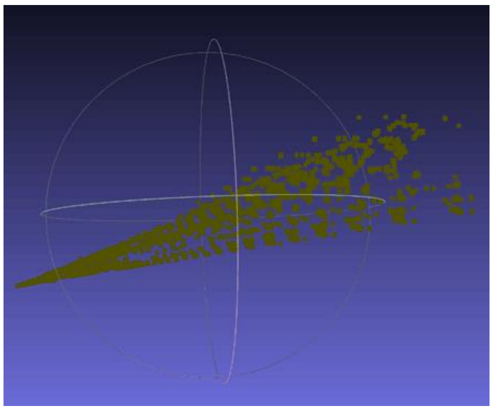
```
void HistogramMatching(Mat srcImg, Mat dstImg)
{
	Mat out(srcImg);
	double srcAddup_hist[256] = { 0.0 };
	statistictHistogram(srcImg, srcAddup_hist);
	double dstAddup_hist[256] = { 0.0 };
	statistictHistogram(dstImg, dstAddup_hist);
	int histMap[256];
	int Tag;
	for (int i = 0; i < 256; i++)
	{
		double minMap = 1.0;
		for (int j = 0; j<256; j++)
		{
			if (minMap > abs(srcAddup_hist[i] - dstAddup_hist[j]))
			{
				minMap = abs(srcAddup_hist[i] - dstAddup_hist[j]);
				Tag = j;
			}
		}
		histMap[i] = Tag;
	}
	for (int nrow = 0; nrow < out.rows; nrow++)
	for (int ncol = 0; ncol < out.cols; ncol++)
	{
		int pixel = out.at<uchar>(nrow, ncol);
		out.at<uchar>(nrow, ncol) = histMap[pixel];
	}
}
```

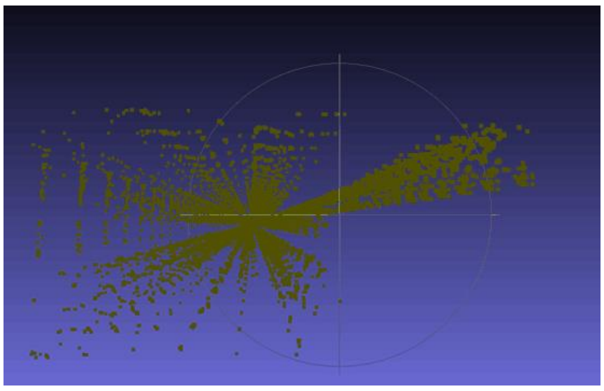
The left image (the base image)	

The origin right image	
The right image after radiometric calibration	

To produce the final colored 3D point cloud, at first , we simply store the points into the .ply, but we get a strange results as follows:

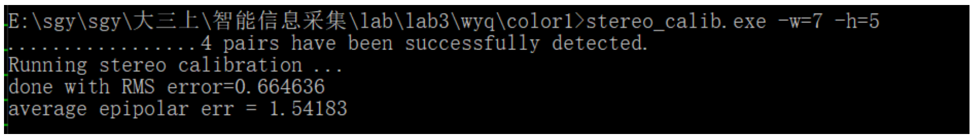
We use the direct output ply to recontruct the model. Maybe we should do sth on it rather than use it directly.



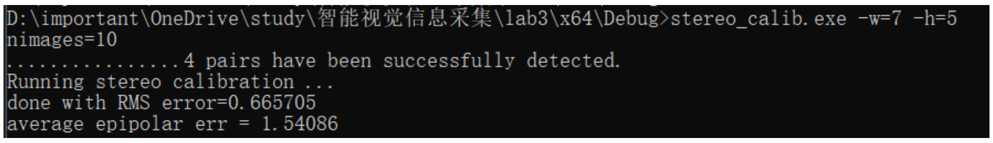


Ⅲ、 further analysis

- We find that when we are doing the stereo calibration on the colorful image, after we grayscale the images, the results have a little difference.

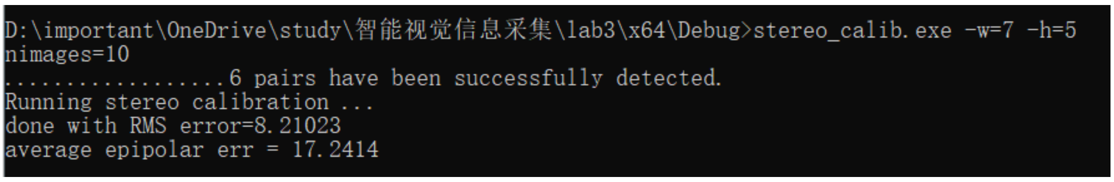


```
E:\sgy\sgy\大三上\智能信息采集\lab\lab3\wyq\color1>stereo_calib.exe -w=7 -h=5
................4 pairs have been successfully detected.
Running stereo calibration ...
done with RMS error=0.664636
average epipolar err = 1.54183
```



```
D:\important\OneDrive\study\智能视觉信息采集\lab3\x64\Debug>stereo_calib.exe -w=7 -h=5
nimages=10
...............4 pairs have been successfully detected.
Running stereo calibration ...
done with RMS error=0.665705
average epipolar err = 1.54086
```

But after we resize and grayscale the origin image, we can detect more pairs of images, but the results are not ideal,

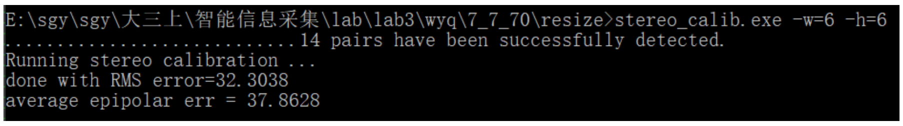


```
D:\important\OneDrive\study\智能视觉信息采集\lab3\x64\Debug>stereo_calib.exe -w=7 -h=5
nimages=10
.................6 pairs have been successfully detected.
Running stereo calibration ...
done with RMS error=8.21023
average epipolar err = 17.2414
```

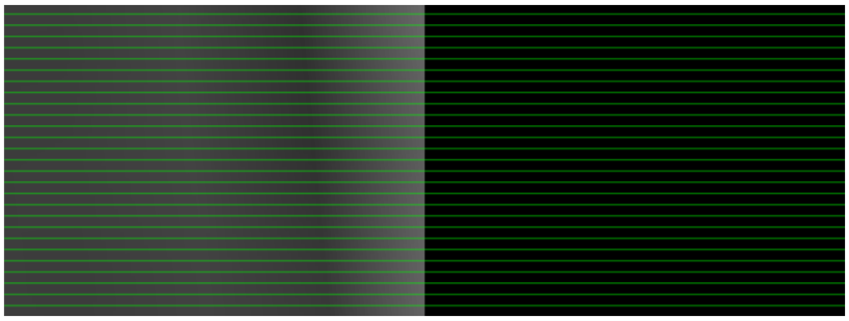
we think that the resized images have less information than the origin images so that it can be detected easier, meanwhile, as for accuracy, we pay a price.

- when we are doing the stereo calibration on the black and white image, there are still some less ideal results as follows:

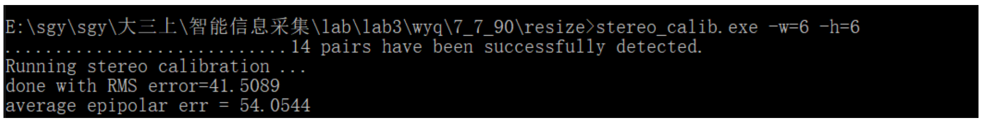
The black and white chessboard of 7*7 with scale of 70%



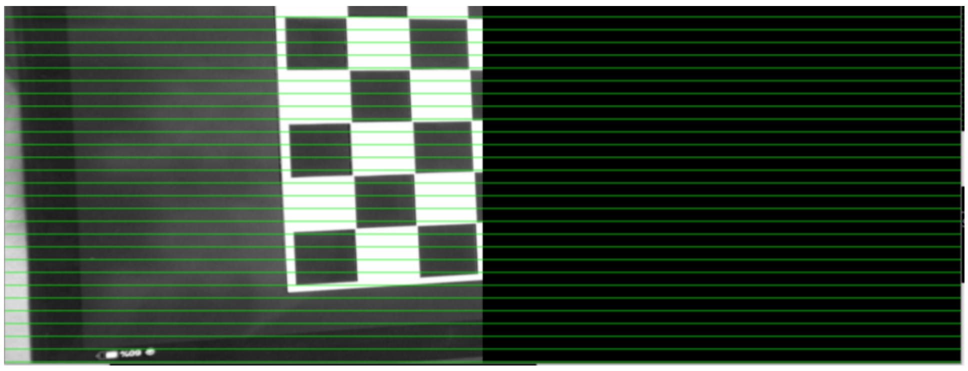
```
E:\sgy\sgy\大三上\智能信息采集\lab\lab3\wyq\7_7_70\resize>stereo_calib.exe -w=6 -h=6
...........................14 pairs have been successfully detected.
Running stereo calibration ...
done with RMS error=32.3038
average epipolar err = 37.8628
```



The black and white chessboard of 7*7 with scale of 90%



```
E:\sgy\sgy\大三上\智能信息采集\lab\lab3\wyq\7_7_90\resize>stereo_calib.exe -w=6 -h=6
...........................14 pairs have been successfully detected.
Running stereo calibration ...
done with RMS error=41.5089
average epipolar err = 54.0544
```



As for some bad performance of our result, we consider its cause lies in our data. Also, we think that if the chessboard is too dense, it also harm the performance.

V、 Discussion and gain

In this assignment, we designed several comparative Experiment, for example, resize and origin, grayscale and origin, and we found really interesting result, which provided us with inspiration and more ideas. We also learn about the importance of designing effective experiment.

.