Peter Szabo
2020.05.17

VMMC Final report: Section 1

Details:

Resolution of image: 4608 x 2592 pixels

Images are included in the folders: images (for bigger checkboard), small_images (for smaller checkboard).

Size of squares in case of bigger checkboard: 36mm

Size of squares in case of bigger checkboard: 29mm

Bigger checkerboard (internal parameters):

Internal parameters $\mathbf{K} = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$

During the work, multiple cameras were plotted, but only one column is internal parameters, but as we can see, there is not much difference between using multiple or less cameras.

No pattern images	3	4	5	6	7
Scale factor f_x	3656.49	3634.24	3609.075	3606.06	3610.90
Scale factor f_y	3662.83	3635.44	3609.72	3605.95	3611.09
Principal point coordinates (c_x ,c_y)	(2414.70; 1290.27)	2382.93; 1283.25	2366.50; 1291.99	2363.04; 1292.88	2364.18; 1291.73
Angles: degrees between image axes	89.91	89.94	89.92	89.91	89.88
s	-5.74	-3.46	-4.77	-5.1	-7.39

Are the pixels of my camera square?

Based on the camera parameters, we can see, that the f_x and f_y are almost equal (less then 1% difference), which means, that our pixels are rectangular. But looking at s, knowing s = f_x * tan(α). We can calculate that, we have approximately -0.0828° (using the average of the upper values) difference from a square, which is very small. This means the we can state, that we have square pixels.

Which are the degrees of the principal point and the center of the image plane?

The center of the image plane: 2304;1296, the principal point on the other hand 2364;1291. The degree of error: ~2% and less than 1%.

Are the axes of the image plane orthogonal?

Yes, as we can see they the angle is very close to the orthogonal (although there is a bit of noise). (we calculated these axes at the first question). Even though the s value does not seem to be close to zeros, but that is due to the fact that it depends of f_x (s = f_x * tan(α)), which in our case is relatively big.

Peter Szabo
2020.05.17

Smaller images (internal parameters):

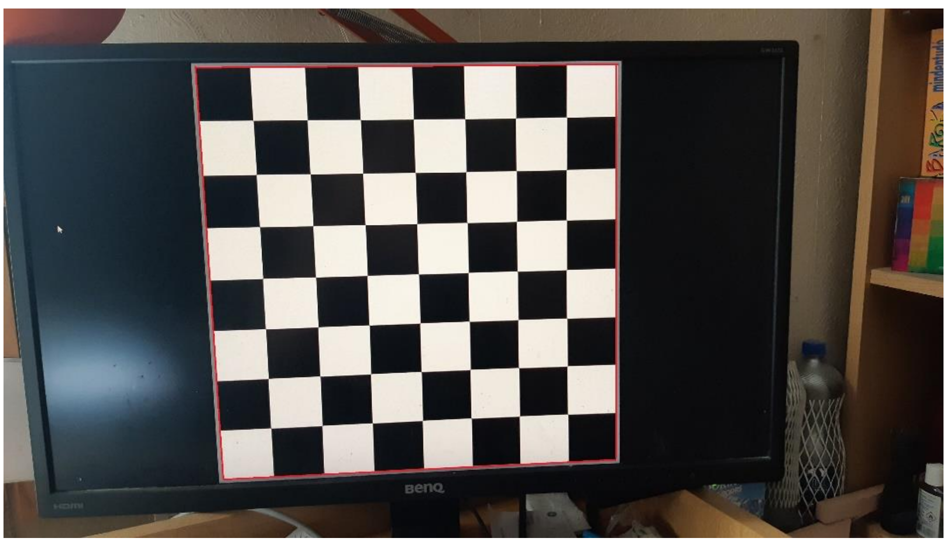
No pattern images	3	4	5	6
Scale factor U	3649.76	3657.16	3688.08	3687.54
Scale factor V	3636.92	3655.33	3673.20	3674.24
Principal point coordinates	2332.48; 1308.79	2358.48; 1319.63	2350.43; 1318.82	2347.41; 1313.39
Angles: degrees between image axes	89.89	89.81	89.91	90.02
s	-6.87	-11.95	-5.29	1.91

Theoretical and practical relationship:

Based on the theory, the intrinsic parameters should be the same, because the same camera was used in both cases. On the other hand, in practice its usually difficult to get exactly the same results, but looking at our results we can see, that approximately it was managed to get them similar.

On the smaller image we could possibly get higher error, due to the human factor. Since that image is smaller it is more difficult to find the matching point of the checkerboards (and the same mistake is proportionally bigger compared to the bigger checkerboard).

Compute radial distortion:

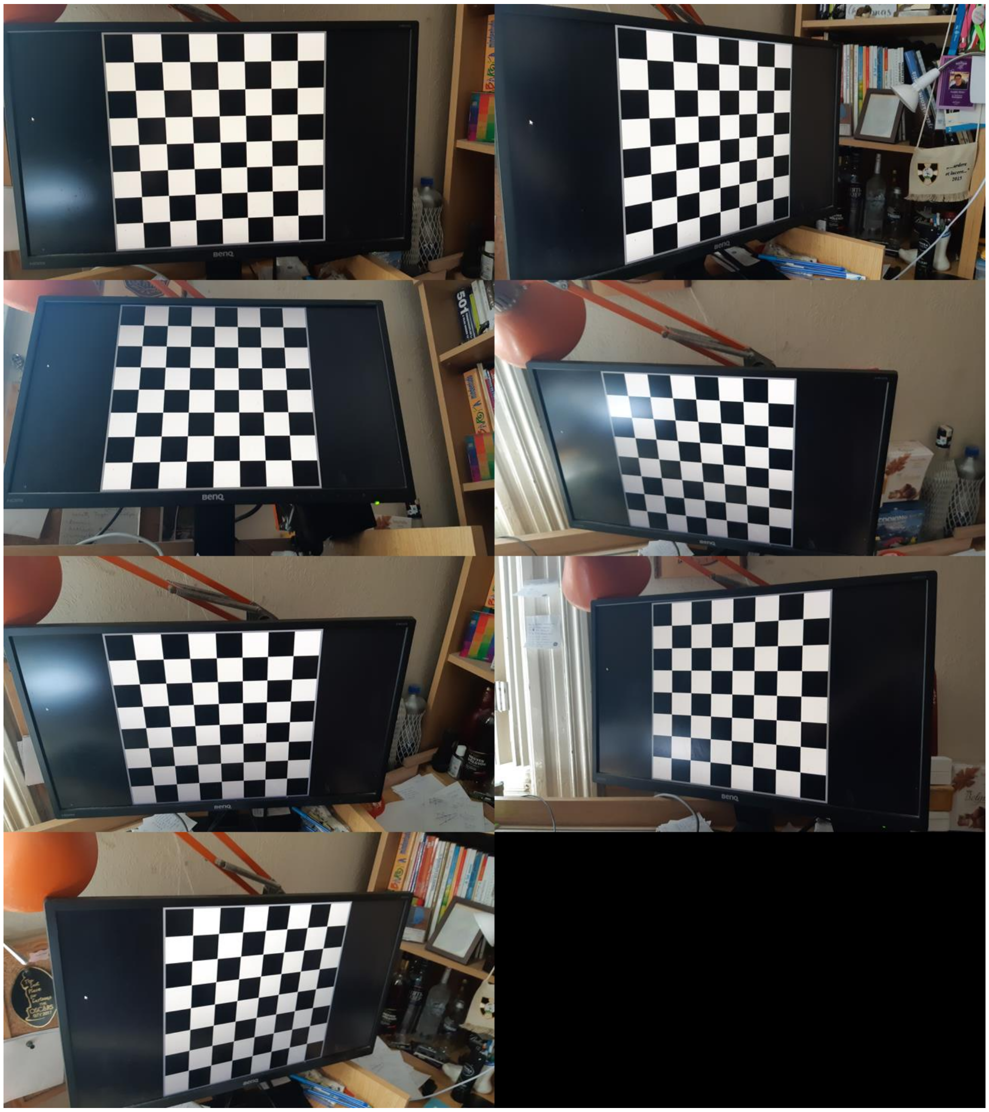


We can also see, that the radial distortion is negligible.

For the following task, I am going to use the intrinsic parameters calculated for the big checkerboard via 5 cameras.

Note: the images are not included, since they have a large resolution, and the data would exceed the limit on moodle! (the image of the bigger checkerboard can be seen of the following montage)

Peter Szabo
2020.05.17



Peter Szabo
2020.05.17

VMMC Final report: Section 2

1. **Object/ Scene capture:**

After multiple experimentations, I constructed the following scene. As we can see 2 standing box were placed, with some object lying on the table, providing it more various texture. Initially I tried to utilize the checkboard scene, but after multiple experiences none of the detectors were able to detect it (only from far and small angle). Moreover, I had to be careful not to put similar objects (2 Nesquik) because that could confuse the matcher. During the search for the best scene I experimented with more the 40 scene, some of them are included in the appendix.

Challenges:

During the experiment, we had to set a large amount of parameters to tune, and multiple tradeoffs to consider. This made this task highly complex and challenging, taking a lot of time, experiment, and creativity to solve.

One of the most important tradeoff was the tradeoff between the angle the number of matching points

Also the distance of the objects was playing a crucial role, the farther we get the bigger angle we could take.

In addition, the images were downscaled in order to decrease the computation cost. This also had an effect on the number of points detected. The more we down-scale, the less points we are able to detect (and describe) but the faster we execute. This can be balanced by the setting of octaves and scales, but that again increases the computational costs. Finally, the downscaling ration was set to be 0.2.

On more interesting thing worth to note was, that the detectors and descriptors preferred the objects parallel to the image plane.

Finally, during setting the parameters the main goal was to maximize the number of detected points, that are also good matches. The final choice of parameters: number of scales: 10. Number of octaves: 8, number of points: 500 (but it doesn't have impact to all detectors).

Results:

	DoH	SURF	Kaze	DSP
1. 7.	0.5052 (97->49)	0.5029 (173-> 87)	0.5 (358->179)	0.5 (134->67)
2. 6.	0.5042 (119->60)	0.5 (188->94)	0.5 (606->303)	0.5 (194->97)
3. 5.	0.5 (258->129)	0.5018 (273 ->138)	0.5 (1238->619)	0.5013 (379->190)
1. 2.	0.5 (412->206)	0.5012 (423->212)	0.5 (2380->1190)	0.5009 (587 ->294)
2. 3.	0.5014 (363->182)	0.5 (388->194)	0.5 (1971->986)	0.5020 (749->376)
3. 4.	0.5 (346->173)	0.5 (342->171)	0.5 (1577->789)	0.5009 (533->277)

The results via RANSAC:

	DoH	SURF	Kaze	DSP
1. 7.	0.2165 (97->21)	0.1965 (173-34)	0.2039 (358->73)	0.3358 (134->45)
2. 6.	0.2269 (119->27)	0.2074 (188->39)	0.2525 (606->153)	0.3144 (194->61)
3. 5.	0.2093 (258->54)	0.2821 (273->77)	0.3 (1238->390)	0.3562 (379->135)
1. 2.	0.2864 (412->118)	0.2340 (423->99)	0.2 (2380->552)	0.2147 (587->126)
2. 3.	0.2314 (363->84)	0.4124 (388->160)	0.4 (1970-818)	0.3778 (749->283)
3. 4.	0.3324 (346->115)	0.3363 (342->115)	0.4 (1577->582)	0.3689 (553->204)

Figure 1. visualizes the results as well.

During the experiment, the effect of closer pairs was examined. As we can see, that as we choose closer part, every descriptor was able to find more matching points. In addition, I also compared the neighboring frames in order to see how much matches were gained by decreasing the angle.

In case of DoH and SIFT, we observe, that it finds the least amount of pairs, and among those pairs we always get around 20% useful one (based on RANSAC). But still finding a decent amount of matches, among a few is not enough point for reconstruction. Also it is worth to point out, that its performance in RANSAC doesn't tend to improve. One more disadvantage of DoH + SIFT combination is that it proved to be the slowest among all the other combinations.

Examining SURF, we see that it finds both object in the middle, with less point. They are also evenly distributed. It is also very fast. Regarding the RANSAC scores it has a mediocre performance, but it still finds enough point for us to be able to perform the reconstruction.

Taking a closer look at Kaze we can we can clearly observe, that it find a lot more matching compared to the rest. Regardless of being lower proportion of those points a good match, it is still more than the others. As we can see on the images, that it detected 2 surfaces, and the points detected on each of them seem to be well distributed. As the qualitative estimation shows, we are able to localize points on the other surface as well, compared to DoH SIFT, where we cannot.

On contrary, DSP SIFT finds less matches, but with higher robustness (according to RANSAC) and better speed (it was designed for this). It also localizes two surfaces, with evenly distributed points.

Comparing DSP-SIFT and Kaze, we can see the difference between them on the Figure 3**.** Meanwhile DSP SIFT localizes points more evenly distributed, Kaze finds the objects in the middle better (due to nonlinearity, that scales are not evenly distributed).

To sum up, our goal was to find the most possible points with the possible greatest angle. Although DSP proved to be faster than Kaze, **my final decision was Kaze detector and Kaze descriptor combination**. The reason behind that that it found the most matches on the frames with the widest angle. And this is crucial for our reconstruction step. On the other hand, we should keep in mind, that our mean-projection error will be a bit higher due to the higher number of point. SIFT-DSP-SIFT would provide less error, and it would still be able to localize enough point, to have an accurate reconstruction, but with more point our object will be visually better with more point (more confidence for the prize of little extra error, which will be hopefully mostly corrugated in bundle adjustment).

An interesting observation is the fact, that the first and second frame (last row of each column) were neighboring frames, they still result less matches compared to 3.-5. pair. This means that there is a higher angle difference between the first and second pair.

In addition, some extra experiments were carried out, in order to measure the relation between two consecutive scenes, and based on that we could infer the complexity of transformation between 2 consecutive frames. In order to do this evaluation, RANSAC results were used. As we can see from the correspondences

Final Parameters:

Scale: 0.2 (downscaling of the original image)
Nscales: 10
Noctaves: 8
Npoints: 500
MaxRatio: 0.6

The chosen pair of images is **image 1. and 7.**

Even though we achieved far more point correspondences with smaller angle, with Kaze we found enough, good quality matches, well distributed on the surfaces. There conditions should be enough to reconstruct.

The following figure presents the correspondences (in the most important ones) form:

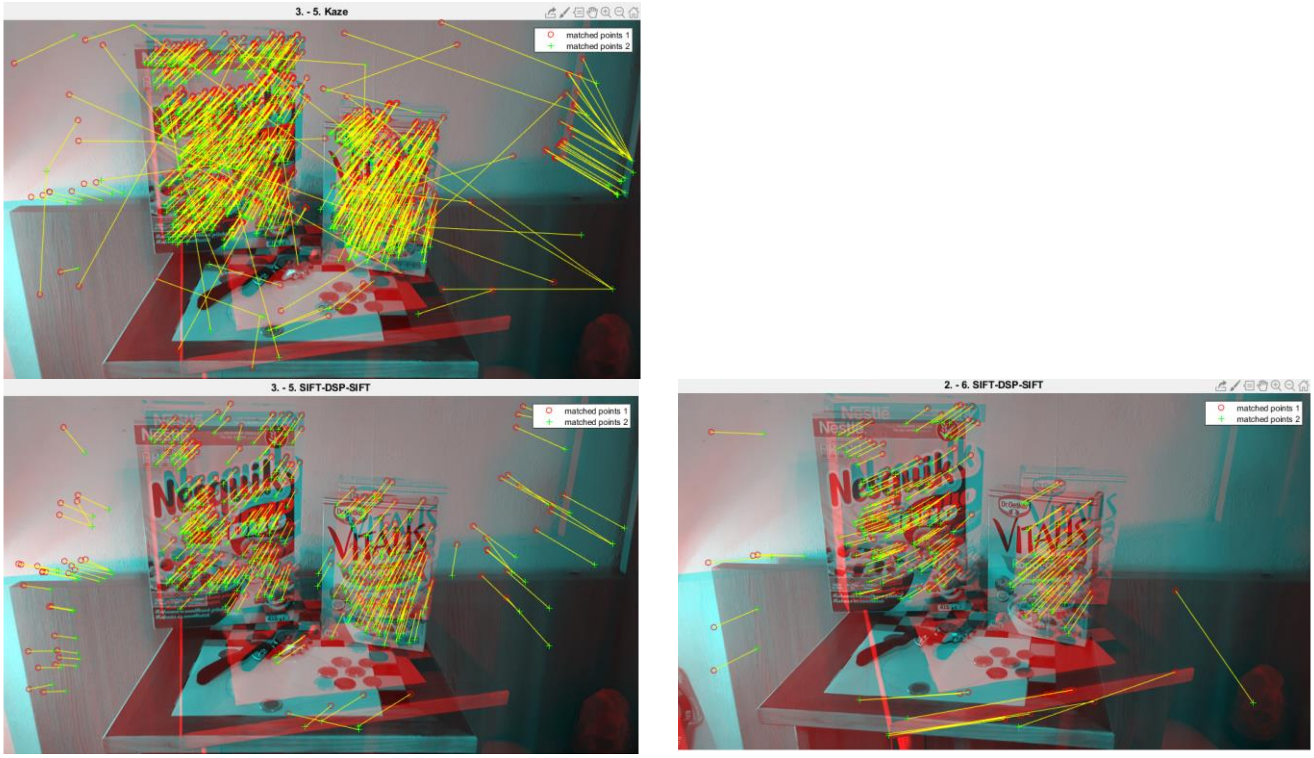


Figure 1. Matching between 2 frames (the title specifies which descripotr – detector was used, and which images were compared)

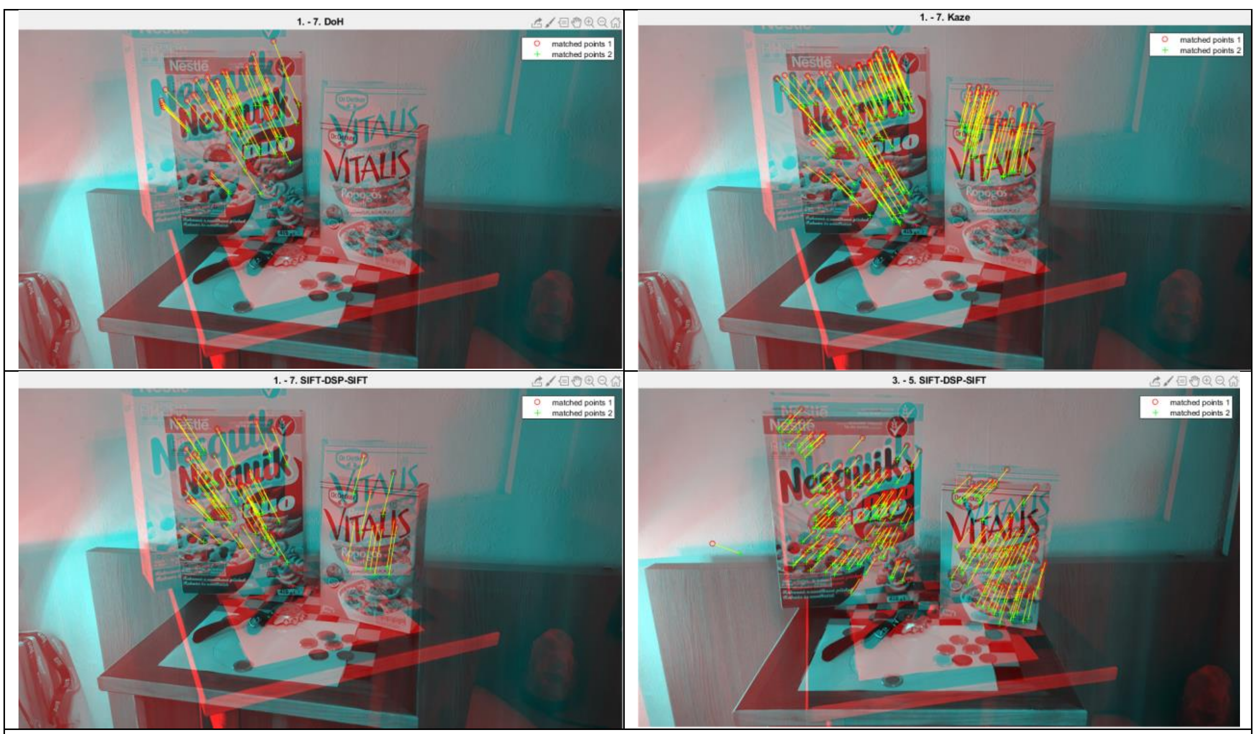


Figure 2. Visualization of matching used to create the fundamental matrix via LMeds method

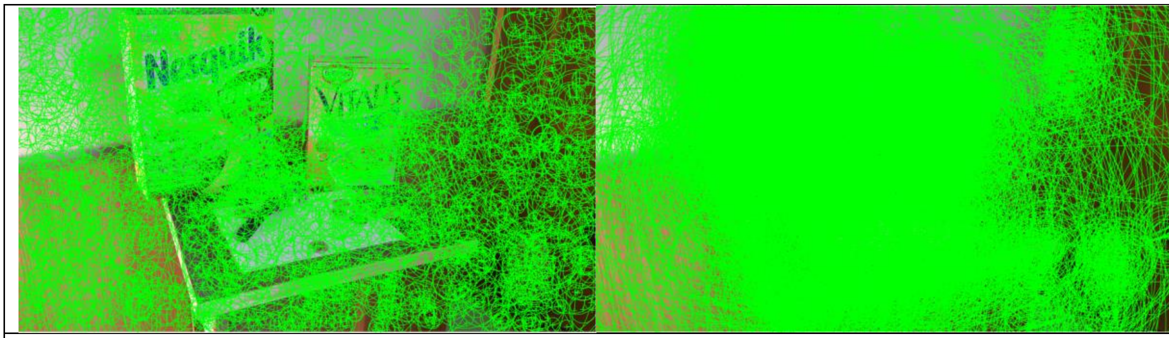
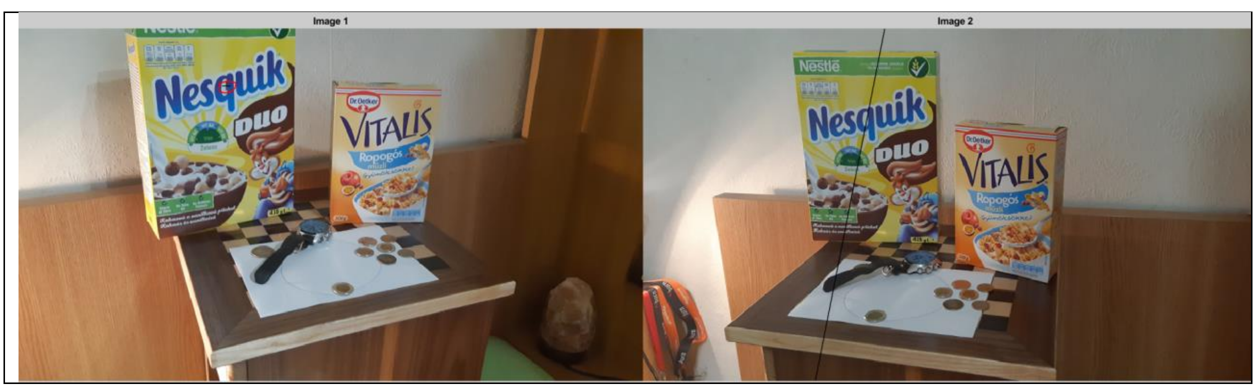
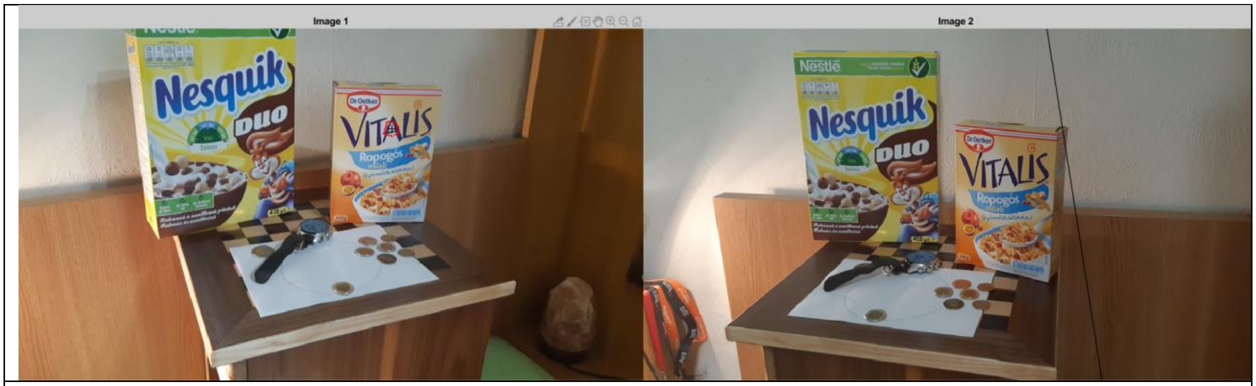


Figure 3. The comparison of Kaze and DSP on the same image. We can see the difference of the distribution of the described points.

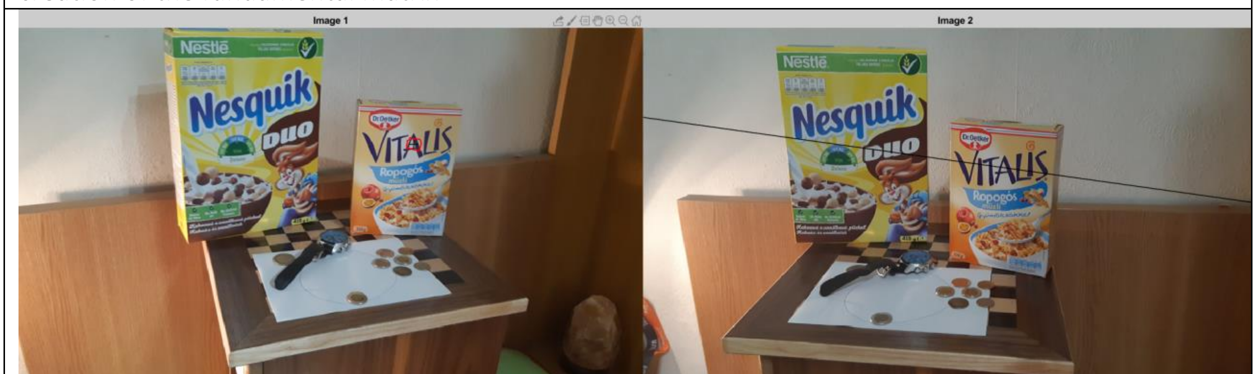
FUNDAMENTAL MATRICIES:

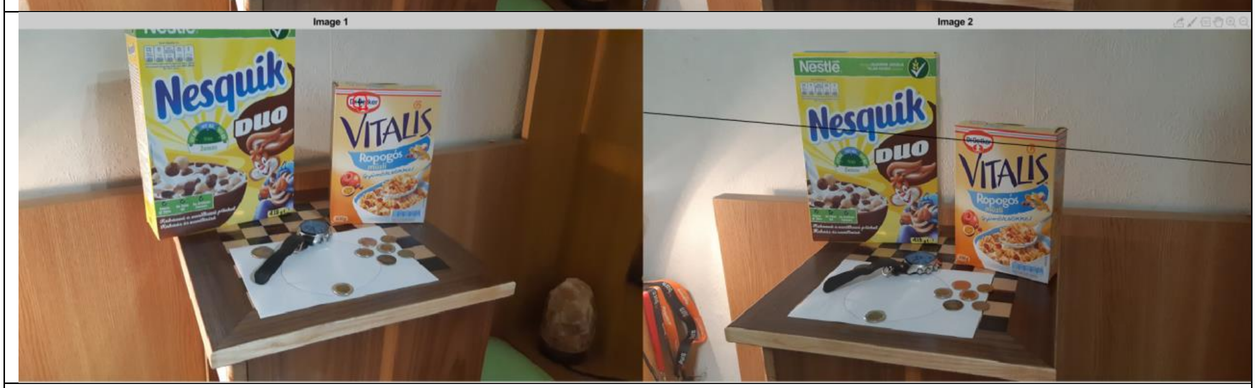
	DoH			SURF			Kaze			SIFT - DSP		
1. 7.	0.0000	-0.0000	-0.0066	-0.0000	-0.0000	0.0020	-0.0000	0.0000	0.0008	0.0000	0.0000	-0.0060
	0.0000	0.0000	-0.0033	0.0000	0.0000	-0.0215	0.0000	0.0000	-0.0184	-0.0000	0.0000	-0.0064
	0.0023	0.0029	1.0000	-0.0017	0.0150	0.9997	-0.0008	0.0116	0.9998	0.0043	0.0014	1.0000
2. 6.	-0.0000	-0.0000	0.0082	0.0000	0.0000	-0.0004	-0.0000	0.0000	0.0053	-0.0000	-0.0000	0.0067
	0.0000	0.0000	-0.0398	-0.0000	0.0000	-0.0411	0.0000	0.0000	-0.0477	0.0000	0.0000	-0.0478
	-0.0070	0.0307	0.9987	0.0006	0.0316	0.9987	-0.0038	0.0374	0.9981	-0.0051	0.0378	0.9981
3. 5.	-0.0000	-0.0001	0.0440	-0.0000	0.0000	0.0048	-0.0000	-0.0000	0.0092	-0.0000	-0.0000	0.0113
	0.0002	0.0000	-0.0541	0.0000	0.0000	-0.0405	0.0000	0.0000	-0.0471	0.0000	0.0000	-0.0485
	-0.0404	0.0488	0.9956	-0.0043	0.0350	0.9985	-0.0083	0.0420	0.9979	-0.0101	0.0432	0.9978
1. 2.	0.0000	0.0000	-0.0128	0.0000	0.0000	-0.0109	0.0000	0.0000	-0.0145	0.0000	0.0000	-0.0140
	-0.0000	0.0000	0.0004	-0.0000	0.0000	-0.0040	-0.0000	0.0000	0.0046	-0.0000	0.0000	0.0020
	0.0117	-0.0032	0.9998	0.0098	0.0015	0.9999	0.0134	-0.0071	0.9998	0.0129	-0.0042	0.9998
2. 3.	-0.0000	-0.0002	0.0465	0.0000	-0.0000	-0.0106	-0.0000	-0.0000	0.0051	-0.0000	-0.0000	0.0011
	0.0002	0.0000	-0.0145	0.0000	0.0000	-0.1552	0.0000	0.0000	-0.1608	0.0000	0.0000	-0.1551
	-0.0467	0.0104	0.9977	0.0102	0.1502	0.9766	-0.0043	0.1557	0.9749	-0.0008	0.1502	0.9767
3 4	0.0000	0.0009	-0.0984	0.0000	0.0000	-0.0492	0.0000	0.0001	-0.0472	0.0000	0.0000	-0.0443
	-0.0009	0.0000	0.2001	-0.0001	-0.0000	0.1544	-0.0001	-0.0000	0.1470	-0.0000	-0.0000	0.1538
	0.0829	-0.1884	0.9540	0.0423	-0.1463	0.9753	0.0404	-0.1399	0.9775	0.0379	-0.1456	0.9759



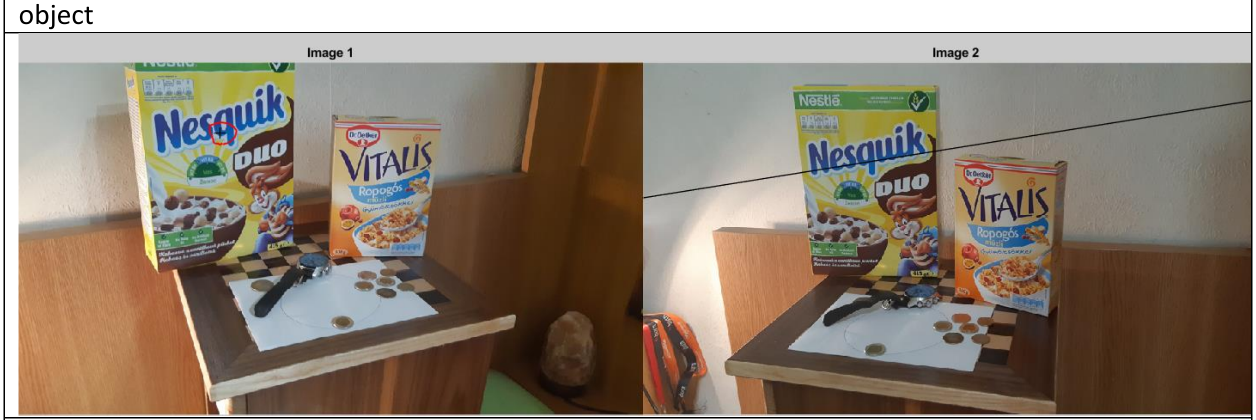


DoH + SIFT (image 1-7) as we can see the epipolar line only crosses points, that contributed the creation of the fundamental matrix

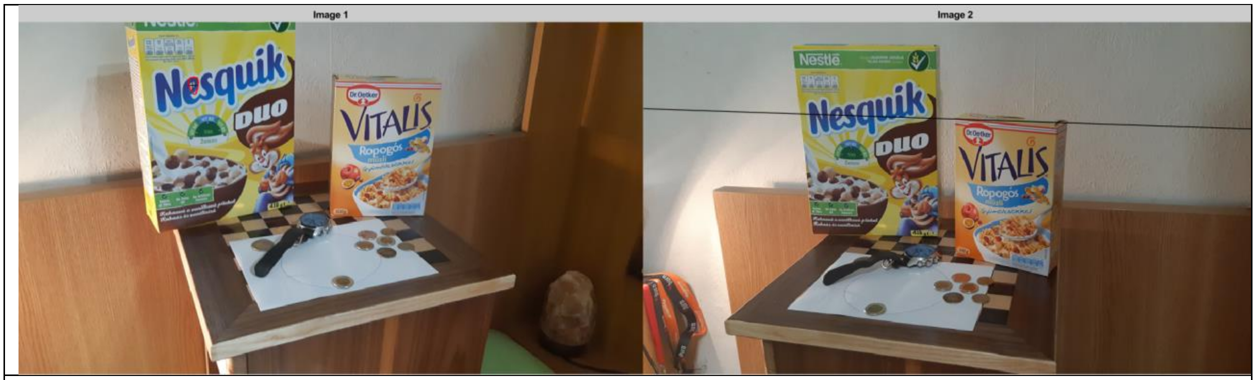




Kaze recognized points on the second object, so the epipolar line fits in the case of Kaze on both object



DSP: also able to fit the epipolar line



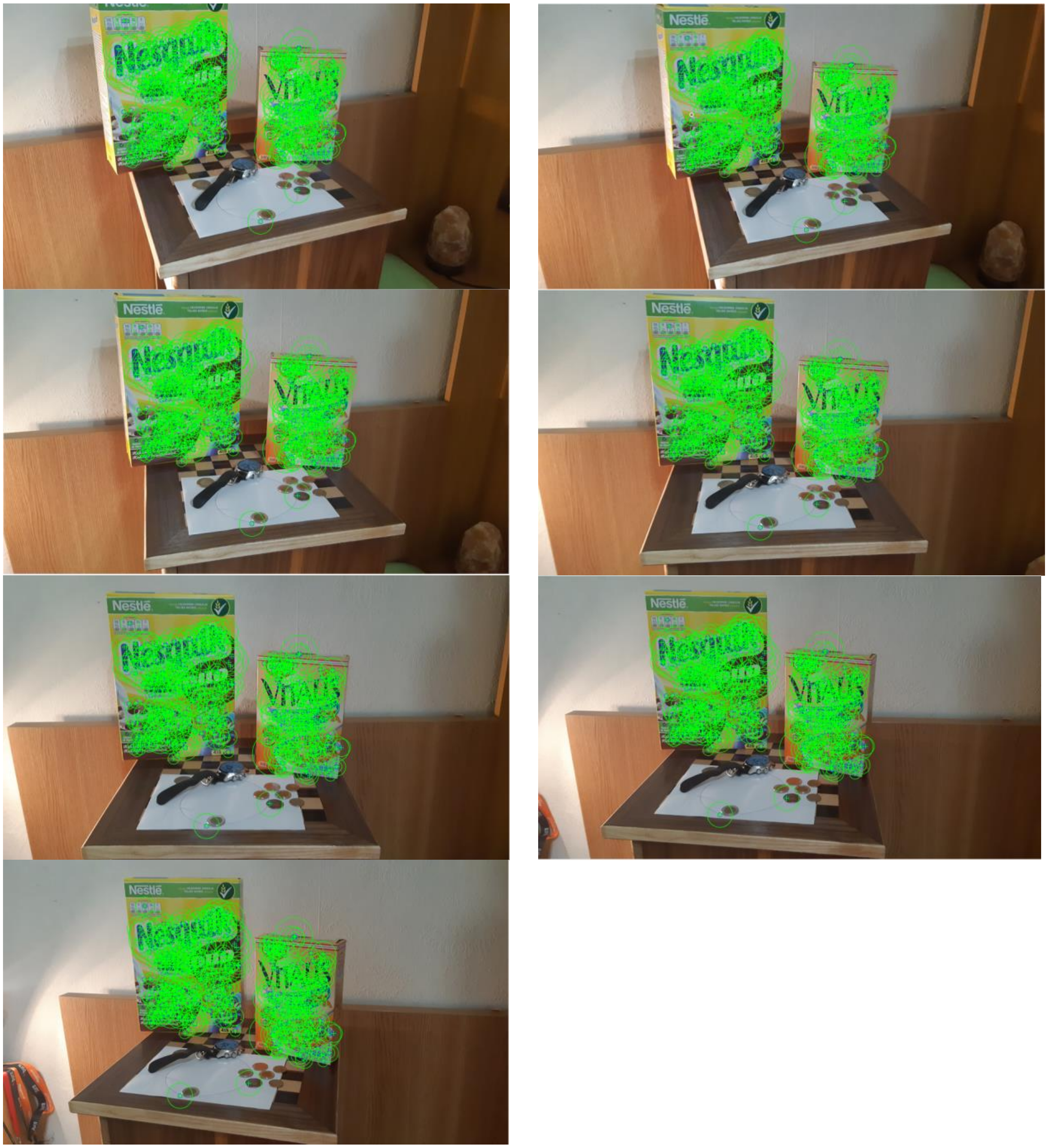
SURF (on image 1 and 7):

Figure 4. Qualitative measurement with the epipolar line

Peter Szabo
2020.05.17

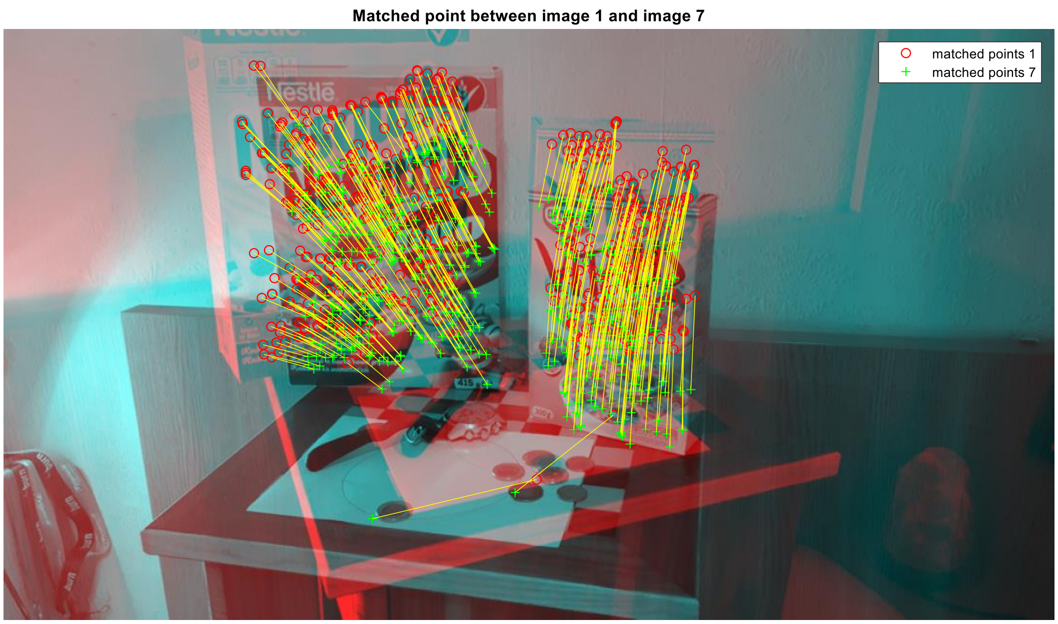
VMMC Final report: Section 3

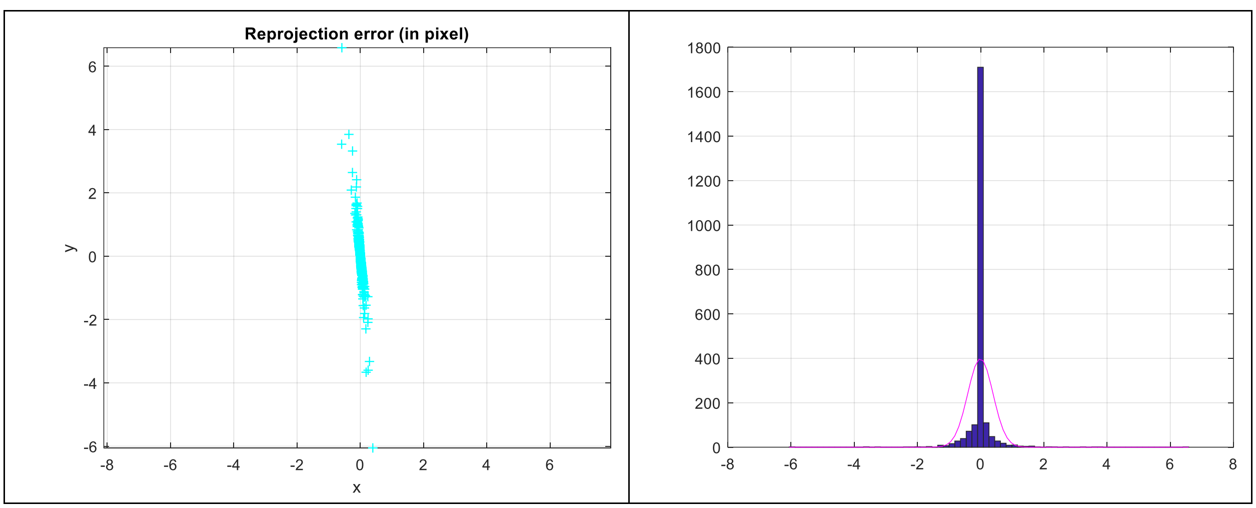
1. **Provide the images, with the detected interest points:**



Peter Szabo
2020.05.17

2. The first and the last (7^{th}) image were used for the reconstruction:





Peter Szabo
2020.05.17

Mean reprojection error = 0.15945

Pixel error: mean = [0.00188 -0.02008]

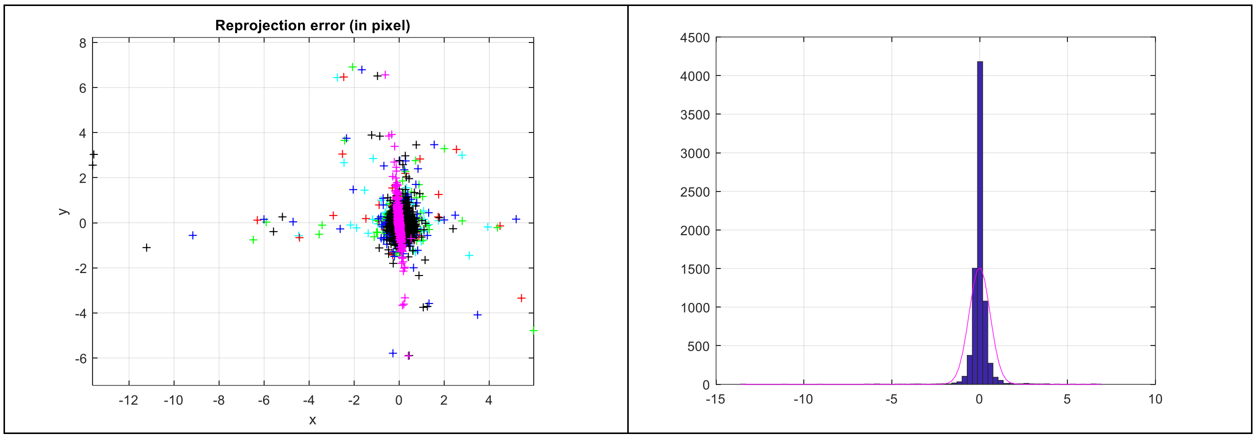
Pixel error: std = [0.05088 0.56231]

3. Projective Bundle adjustment: every image was used during this step

A) Resectioning: mean reprojection error = 0.35419

Pixel error: mean = [0.00211 -0.01247]

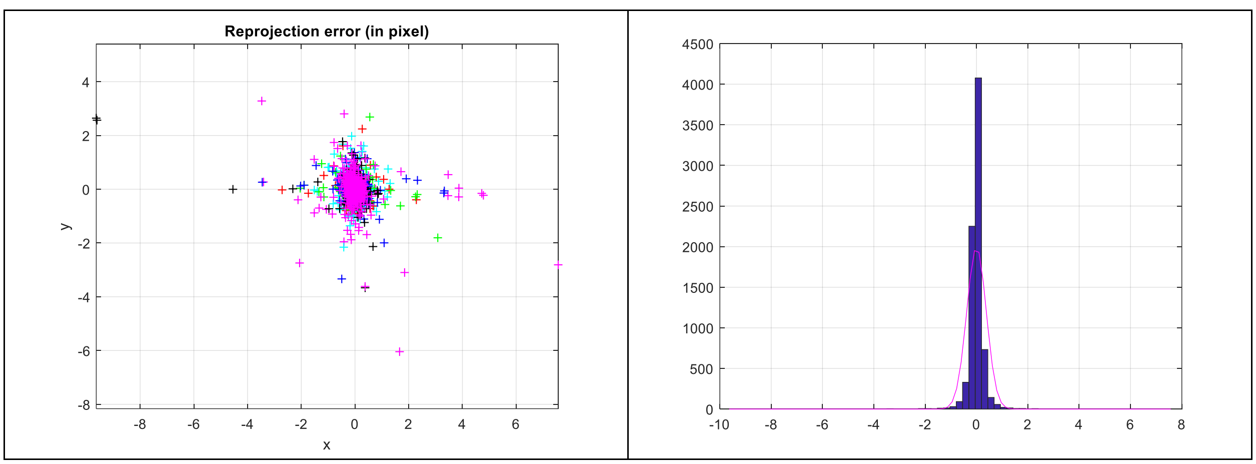
Pixel error: std = [0.59963 0.59063]



B) Projective Bundle adjustment: mean reprojection error = 0.14468

Pixel error: mean = [-0.00000 -0.00000]

Pixel error: std = [0.41218 0.34574]

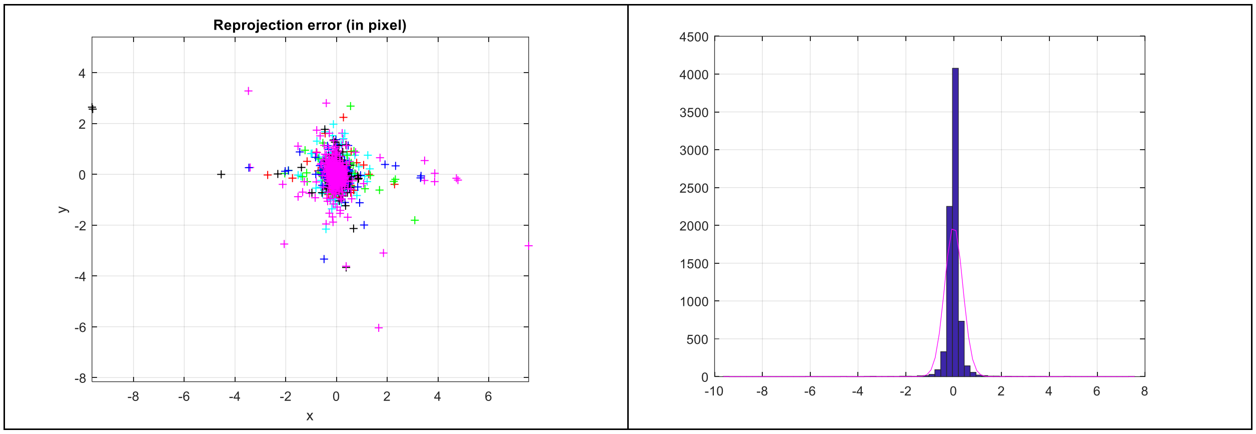


As we can see using more cameras increased the error after sectioning. That is due to the fact, that we used only 2 cameras to create the 3D object, and in the resectioning step we added cameras, that were not used to create that. And as we perform the Bundle Adjustment the error gets smaller that it was initially, because we correct them based on all cameras.

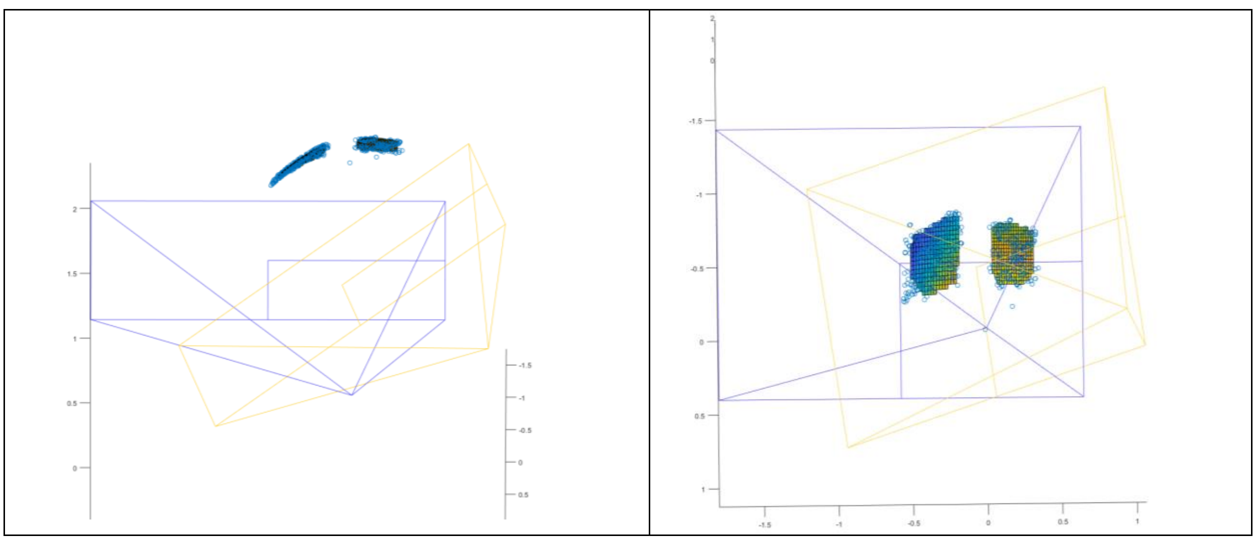
4: Mean reprojection error**:** 0.10267

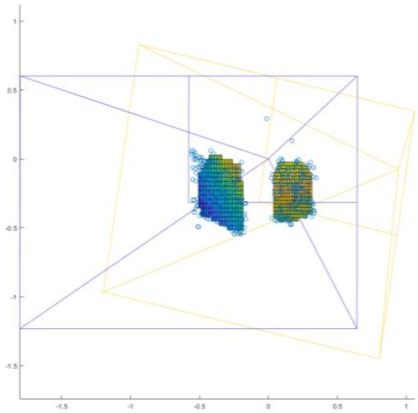
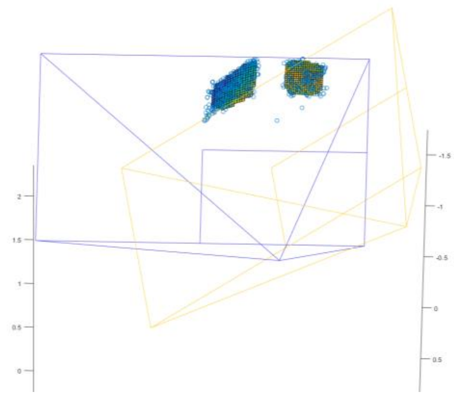
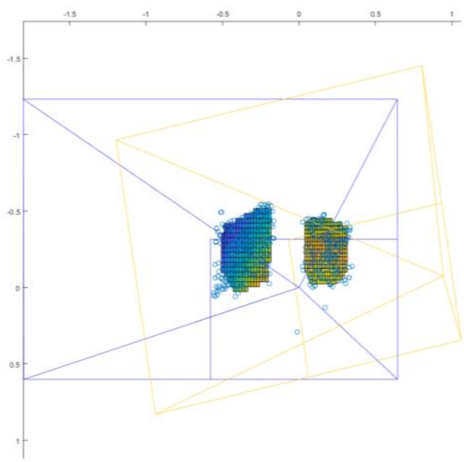
Pixel error: mean = [-0.00000 -0.00000] (same)

Pixel error: std = [0.41218 0.34574] (same)



Visualization of results:



	
	(for the visualization I projected the points to 2 dimension, and then applied K-means to separate the points. Then a surface were fitted on the points)

Peter Szabo
2020.05.17

In addition, I also colored the 2 different objects: