Report of Anton Maksimov (antonma, 16-952-137), Task 4 "Model fitting" on ETHZ course "Computer Vision".

---

1. We implement line fitting RANSAC: choosing 2 random points from the dataset and calculating number of inliers within threshold, improving model if this number increases.

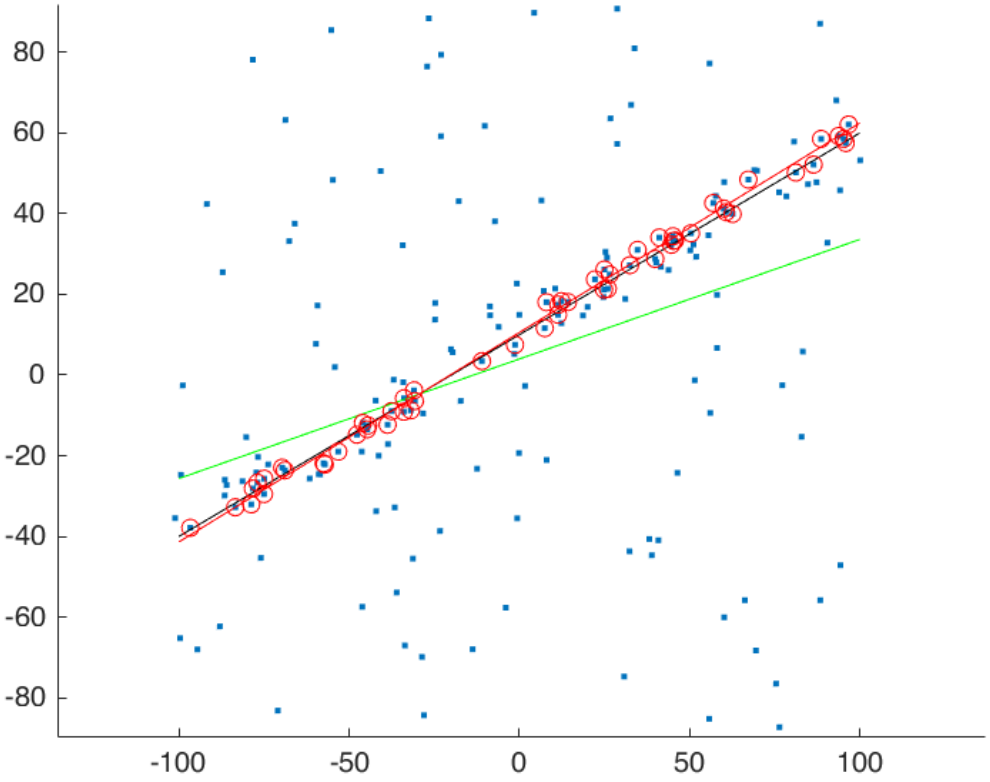

Figure 1: Using RANSAC (threshold = 3) result is much better than with least squares and almost the same as ground truth

Errors

err_ls	err_ransac	err_real
134.4	42.5	41.1

2. We implement 8-point algorithm using manually clicked points (more than 8 for better accuracy). We do everything as described in exercise task and paper «In Defense of the Eight-Point Algorithm») and show epipolar lines and epipoles (found as null vectors from SVD decomposition). Below are presented results (fig. 2 – 9).

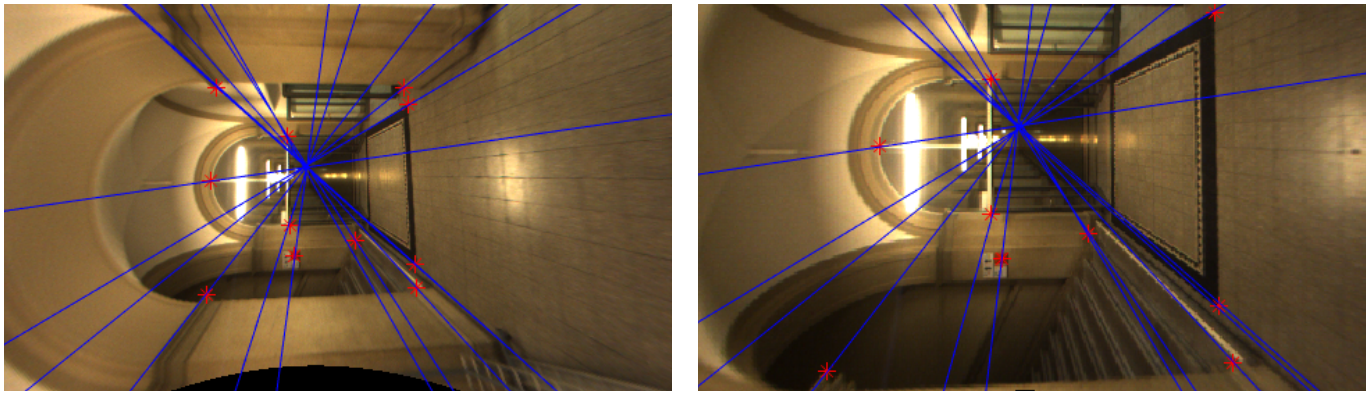

Figure 2: Epipolar lines and 11 used points with non-singular fundamental matrix

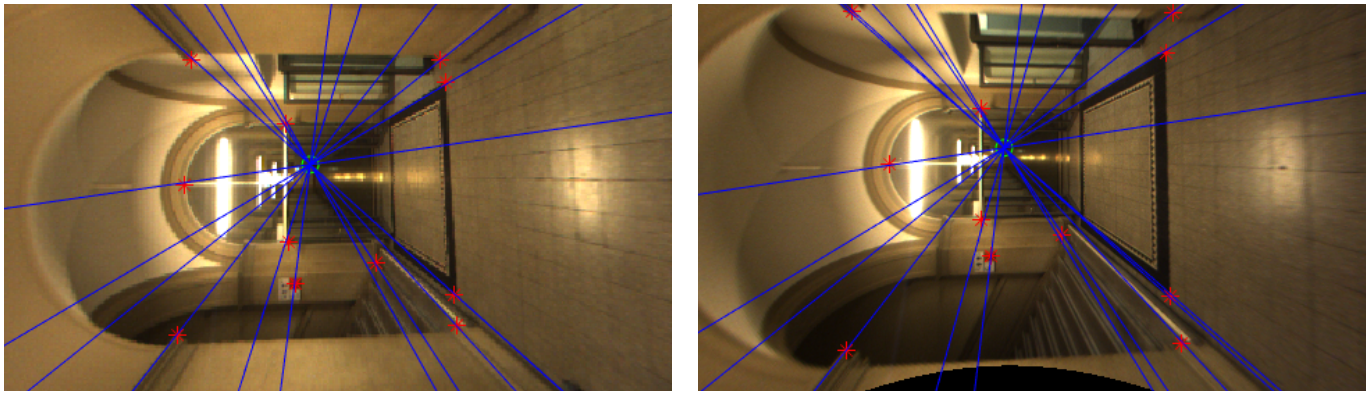

Figure 3: Epipolar lines, epipoles (green circles) and 11 used points (red crosses) with singular fundamental matrix

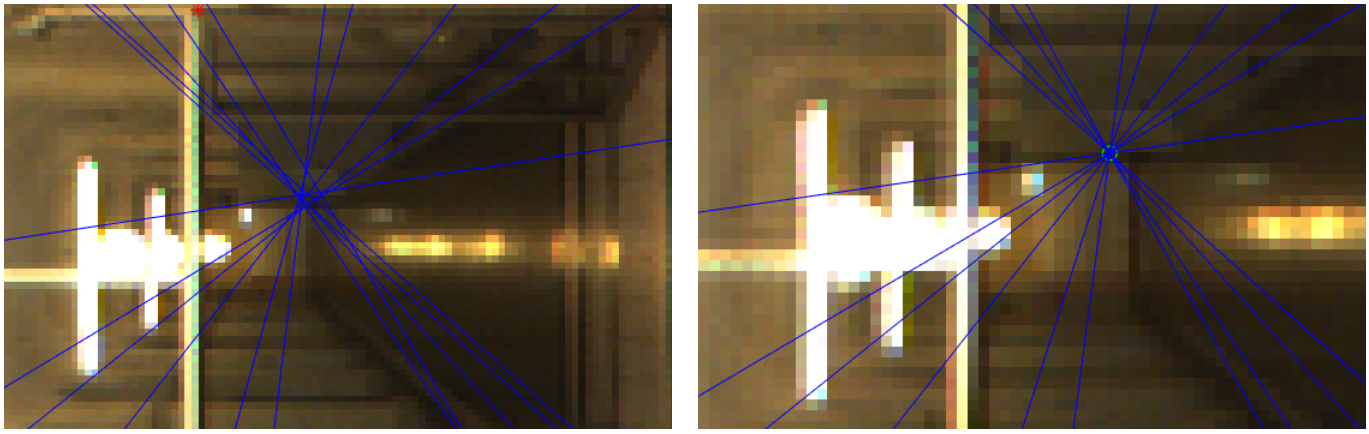

Figure 4: Blowed-up intersection of epipoles, left — with non-singularized matrix, bad; right — with singularized matrix intersection is perfect and matches with epipole, better

Figure 5: Epipolar lines and 10 used points with non-singular fundamental matrix

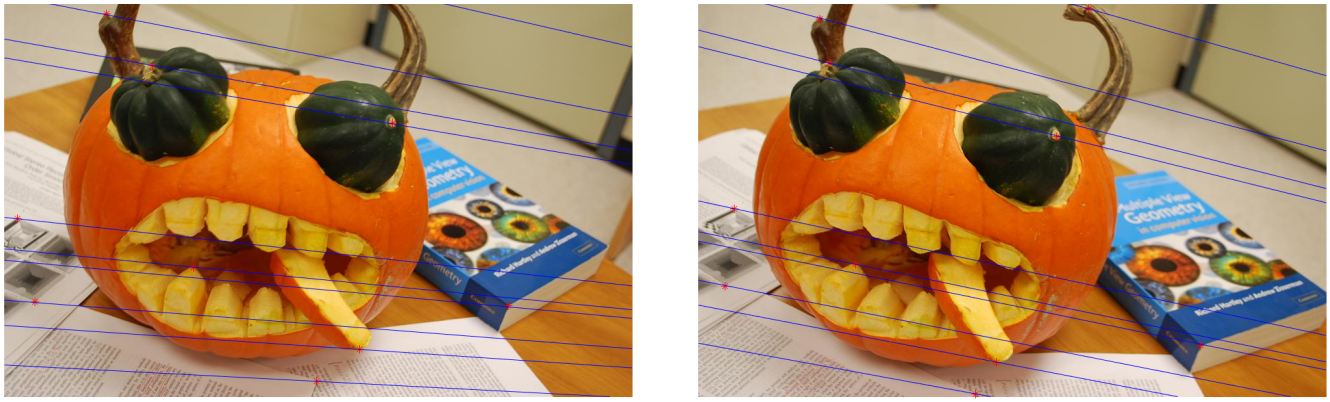

Figure 6: Epipolar lines and 10 used points with singular fundamental matrix

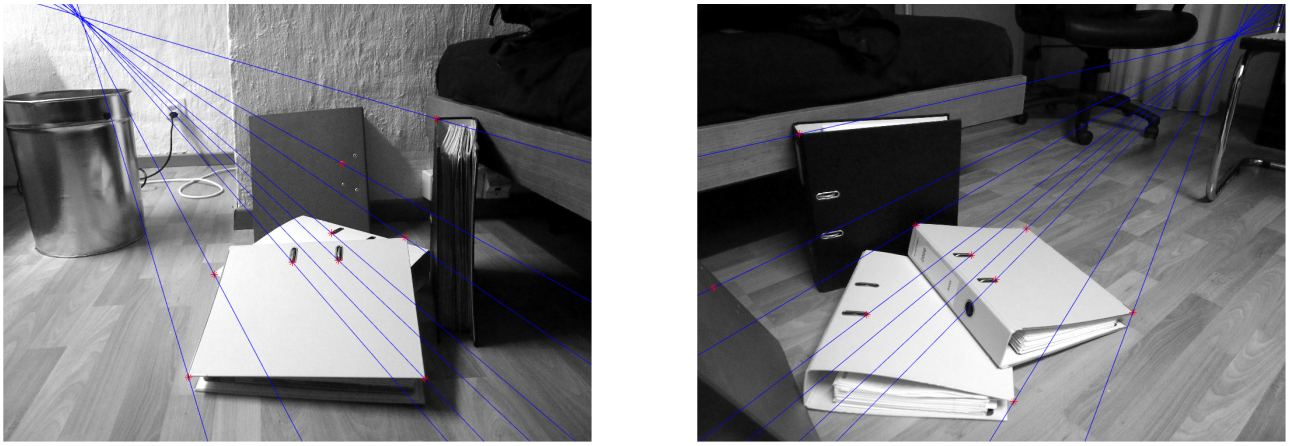

Figure 7: Epipolar lines and 9 used points with non-singular fundamental matrix

3. We download and extract vlfeat to the working directory, initialize it using
`run(strcat(pwd, '/vlfeat-0.9.21/toolbox/vl_setup'));`
and run `main_ransac8pF.m` without the second «ransac» part. Result at the fig. 10

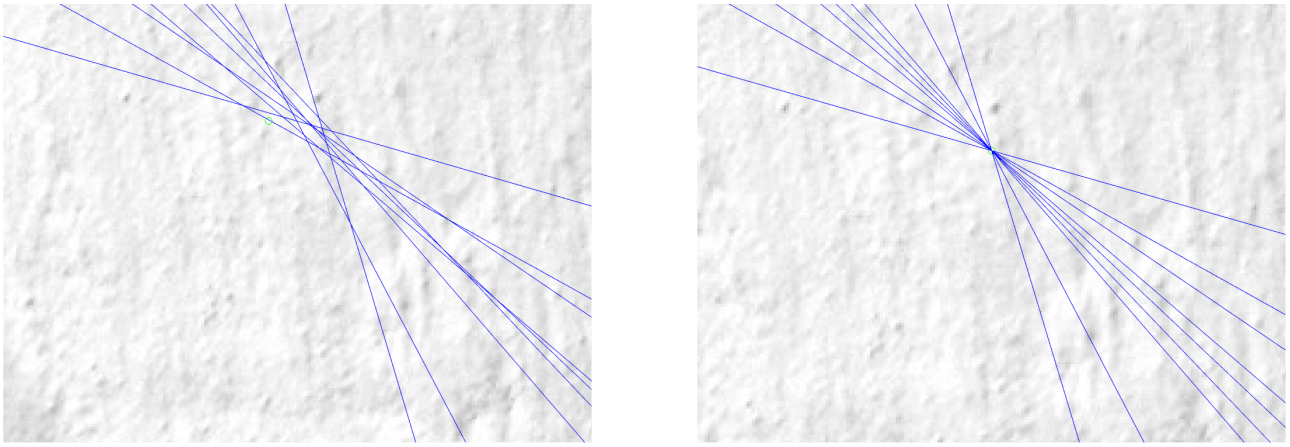

Figure 8: Blowed-up intersection of epipoles, left — with non-singularized matrix, bad; right — with singularized matrix intersection is perfect and matches with epipole, better

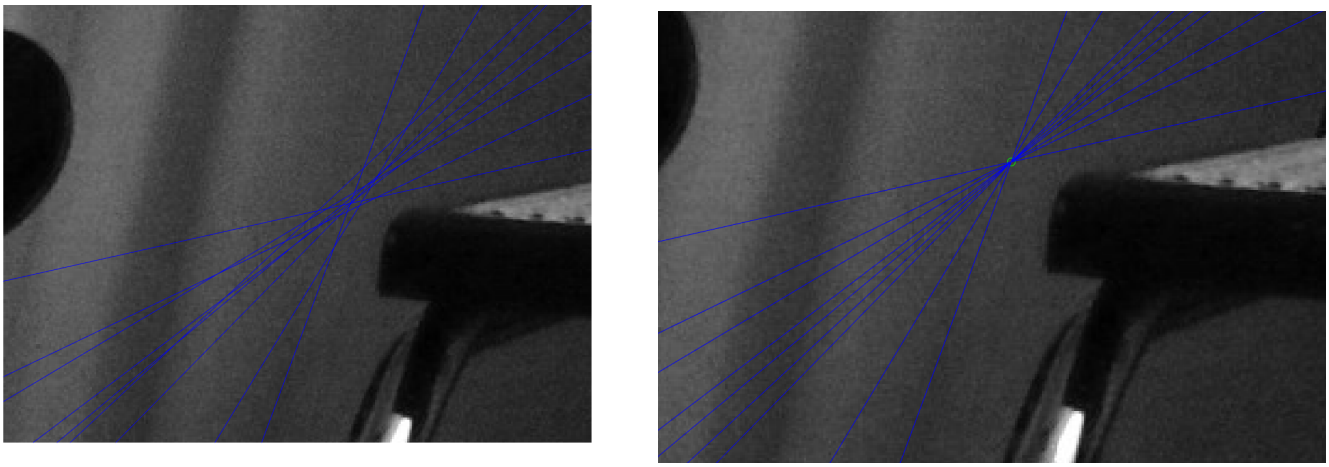

Figure 9: Blowed-up intersection of epipoles, left — with non-singularized matrix, bad; right — with singularized matrix intersection is perfect and matches with epipole, better

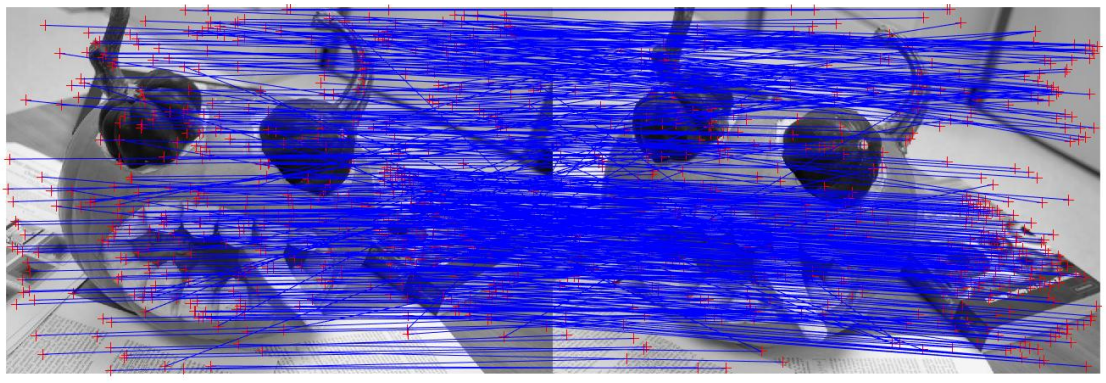

Figure 10: SIFT matching using vlfeat, many inaccuracies present (at least too diagonal lines)

4. Choose random 8 points from output of vlfeat, find singular fundamental matrix $\hat{F}$ and using it implement RANSAC algorithm on all the points of interest with different threshold. Distance is measured as sum of orthonormal distances from every point in pair to the epipolar line generated by other point from the pair:

$$\mathcal{S}(x, x') = d(x', Fx) + d(x, F^\top x'),$$

where

$$d(x_2, Fx_1) = \frac{x_2^\top F x_1}{\sqrt{a^2 + b^2}} \text{ if } Fx_1 = (a, b, c)^\top \iff \text{epipolar line } \{ax + by + c = 0\}$$

Results at the figures 11 – 27. Seeing the pictures and trying to conclude, we can propose, that the most balanced result is achieved when threshold is about 10% of dimensions of the image (if it's more or less quadratic).

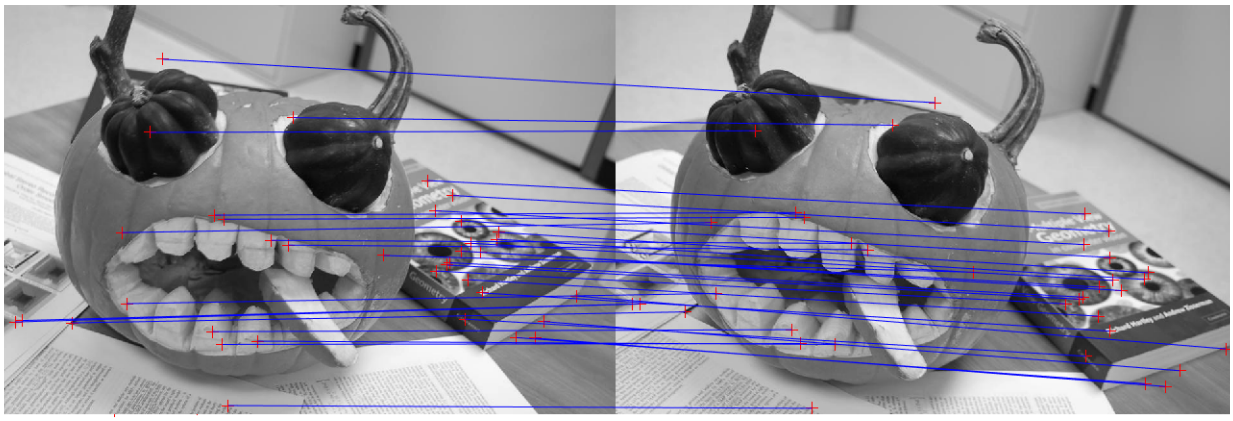

Figure 11: RANSAC, 0.2 threshold, 1000 iterations. Maybe, too small number of inliers

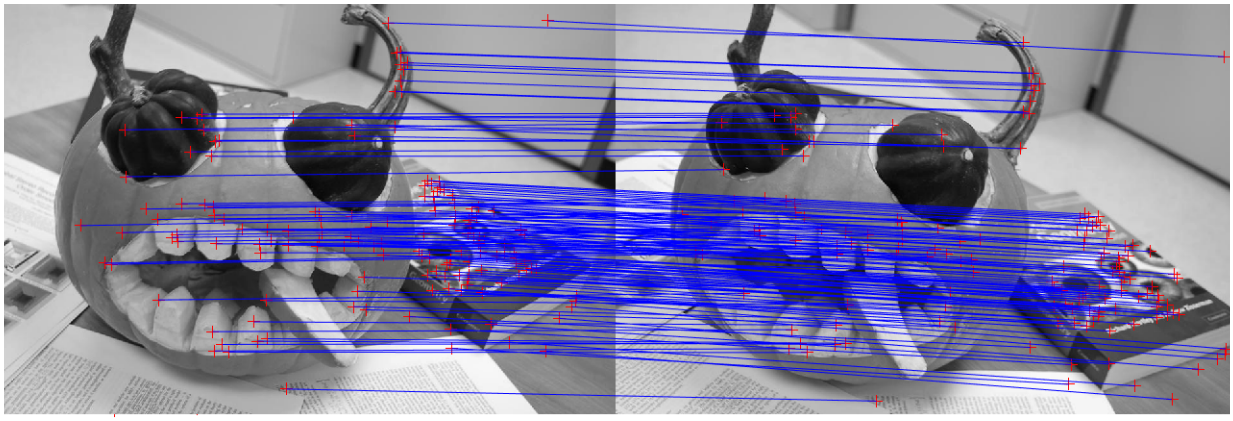

Figure 12: RANSAC, 1 threshold, 1000 iterations. Better, but still a few of inliers

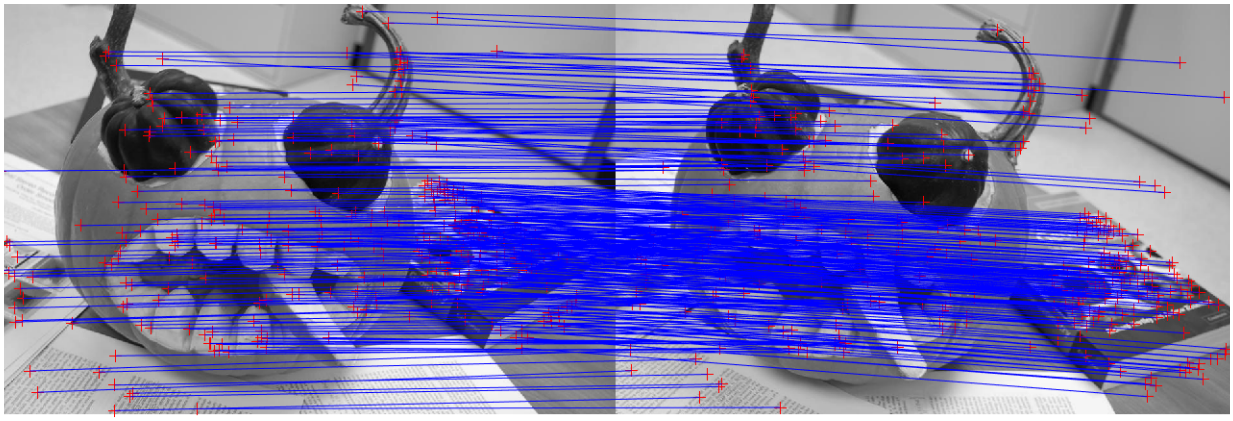

Figure 13: RANSAC, 5 threshold, 1000 iterations. Looks good

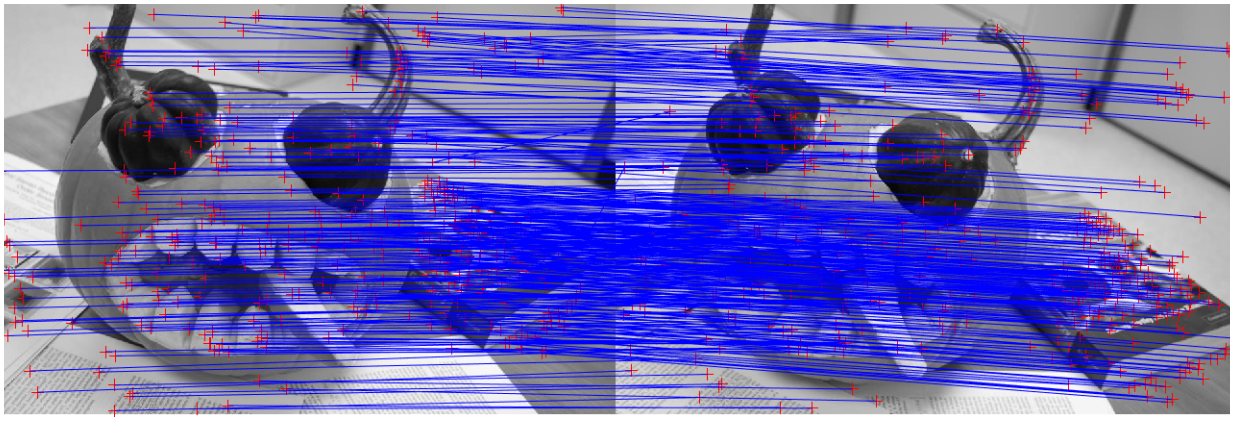

Figure 14: RANSAC, 20 threshold, 1000 iterations. Now, probably, too many errors, but ok

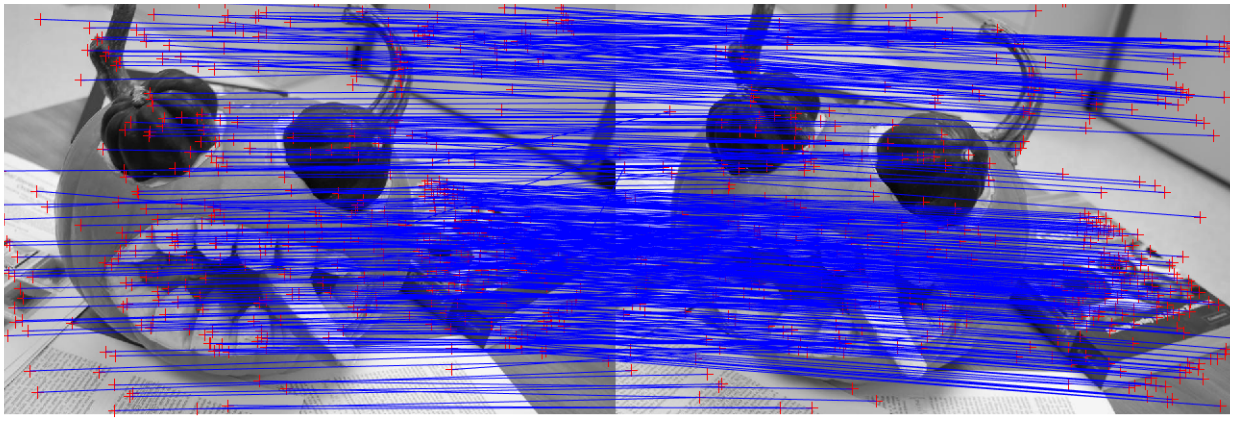

Figure 15: RANSAC, 100 threshold, 1000 iterations. Almost all points in inliers, not so good among points in clusters

Statistics of RANSAC 8 points algorithm (pumpkin),
1000 iterations, 630 points

`threshold`	0.2	1	5	20	100
inliers	43	172	353	446	532
mean Sampson dist	0.09	0.48	1.67	4.26	15.9
inliers/mean Sampson dist	486	357	212	105	33

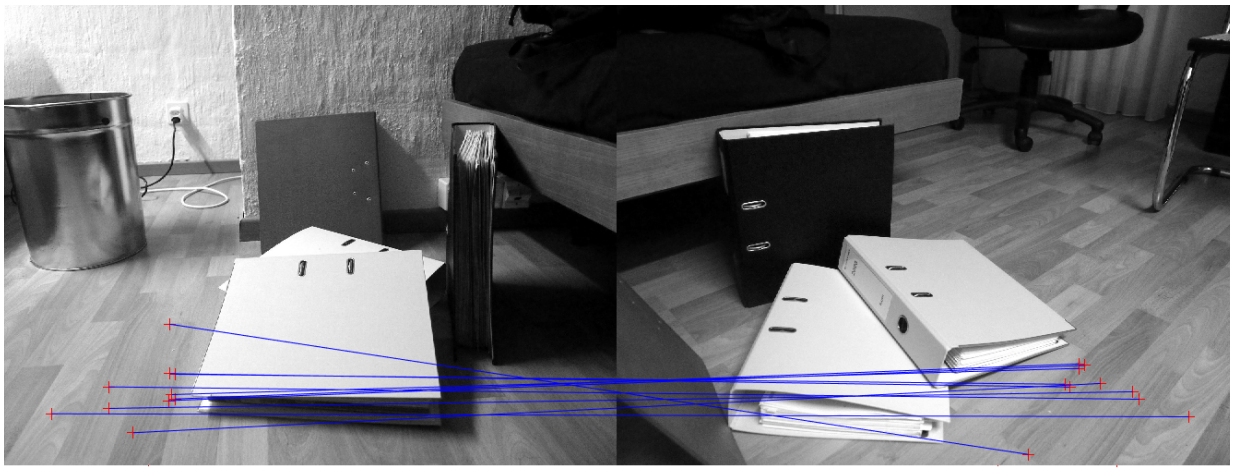

Figure 16: RANSAC, 0.2 threshold, 1000 iterations.

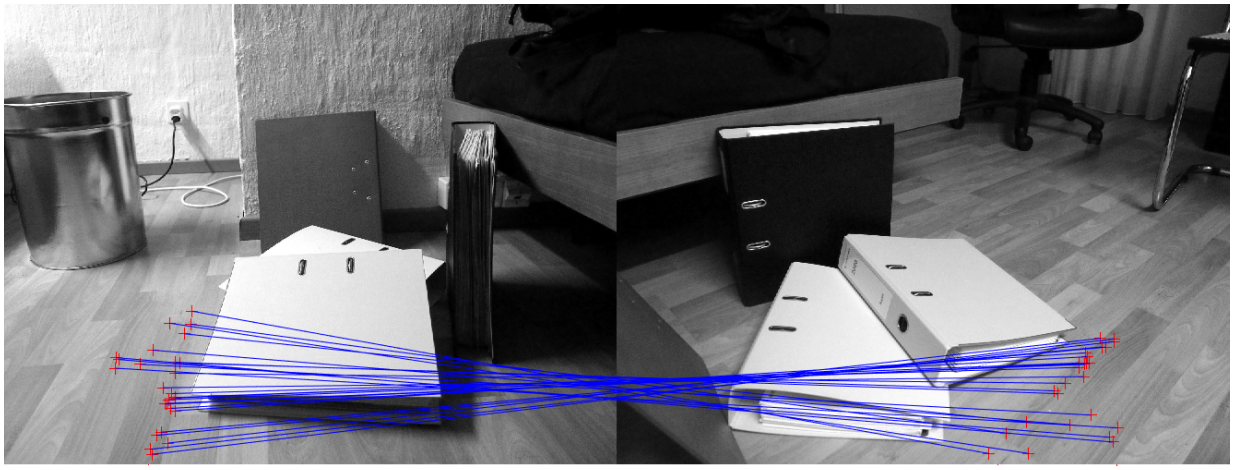

Figure 17: RANSAC, 1 threshold, 1000 iterations.

Statistics of RANSAC 8 points algorithm (rect),
1000 iterations, 709 points

`threshold`	0.2	1	5	20	100
inliers	11	28	52	83	108
mean Sampson dist	0.07	0.41	2.5	6.6	18
inliers/mean Sampson dist	150	68	21	13	6

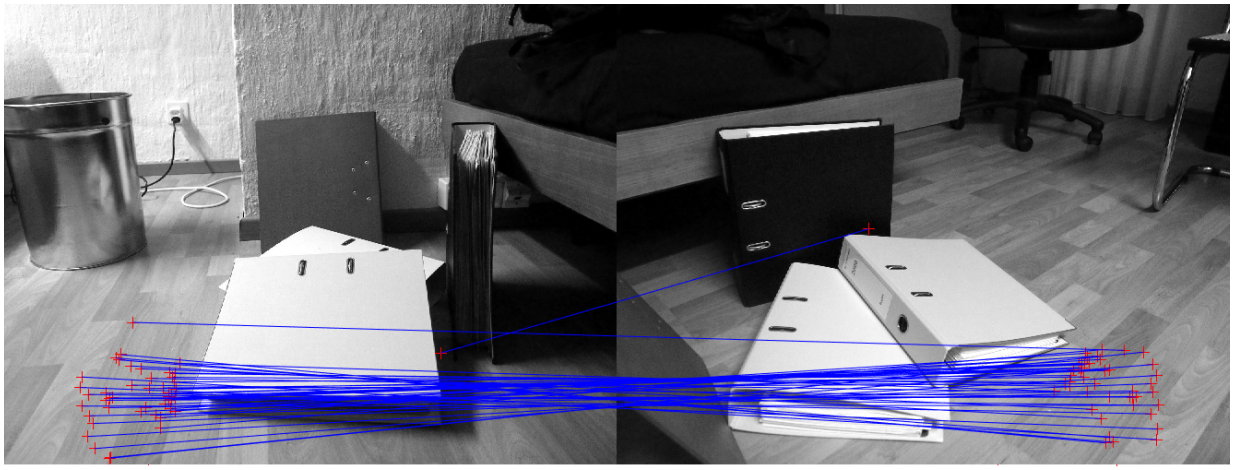

Figure 18: RANSAC, 5 threshold, 1000 iterations.

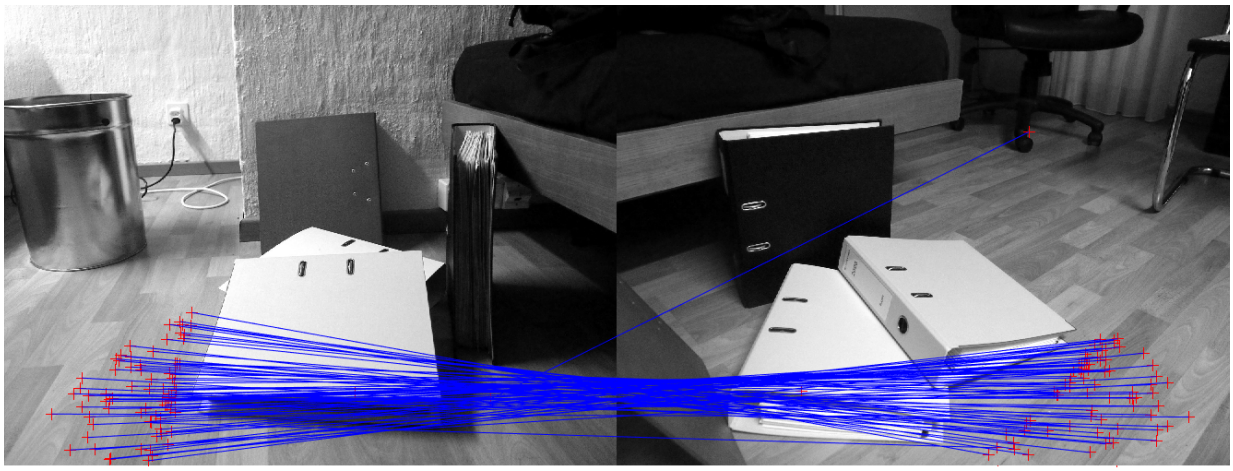

Figure 19: RANSAC, 20 threshold, 1000 iterations.

M — number of iteration of adaptive RANSAC
Limit 1000 iterations

`threshold`	0.2	1	5	20	100
pumpkin	1000	1000	472	77	18
rect	1000	1000	1000	1000	1000
ladybug	1000	1000	1000	695	137

Conclusion

1. Amount and quality of inliers are the same as was with non-adaptive algorithm, but less iterations, that is computationally beneficial.
2. At not more or less «translational» images adaptive RANSAC helps more in terms of number of iterations, because in these images present more true correspondent patches and greater number of inliers reduces number of iterations to reasonable amount. In example «rect» till threshold 100 there are still 1000 iterations, so adaptive algorithm doesn't change anything.

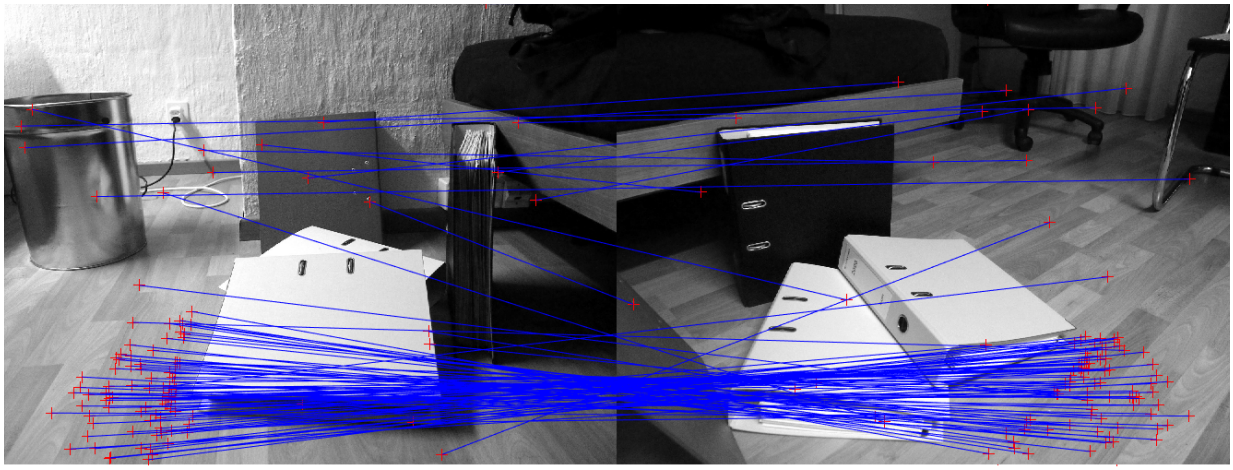

Figure 20: RANSAC, 100 threshold, 1000 iterations. Almost all points in inliers, not so good

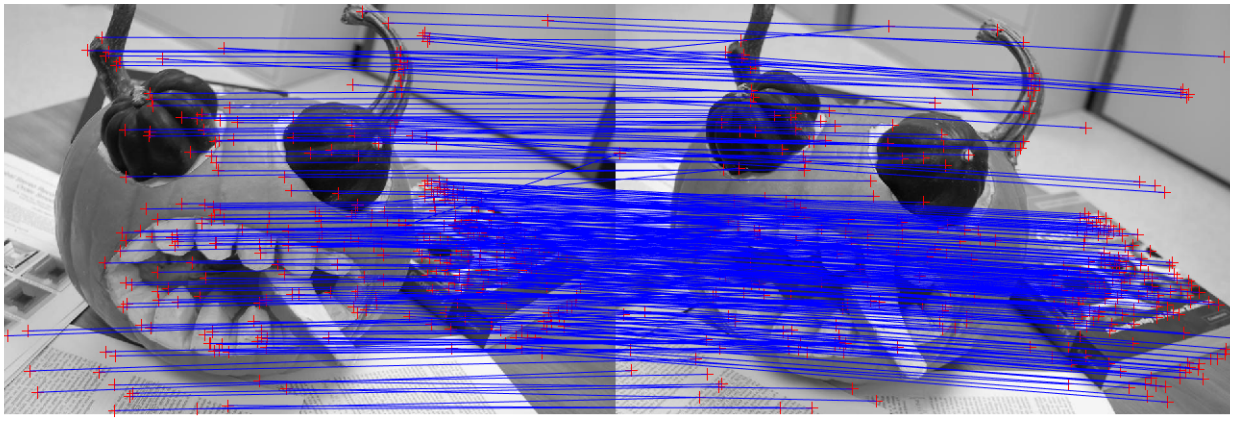

Figure 21: Adaptive RANSAC, 5 threshold, 472 iterations.

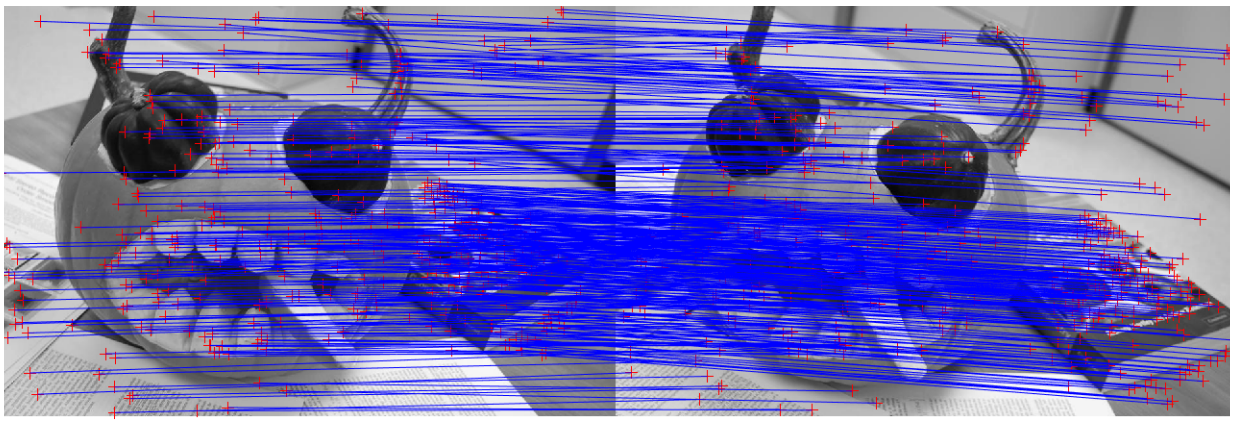

Figure 22: Adaptive RANSAC, 20 threshold, 77 iterations.

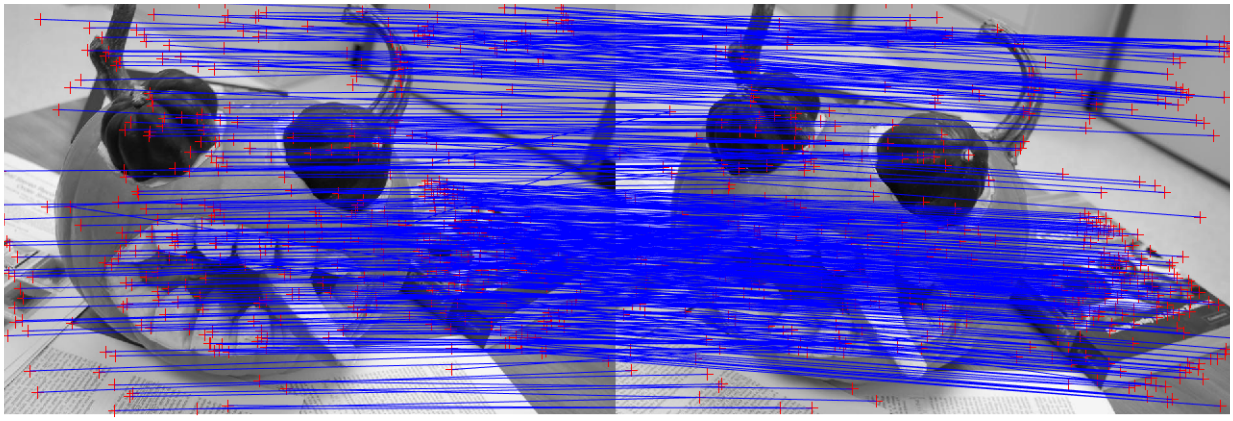

Figure 23: Adaptive RANSAC, 100 threshold, 18 iterations.

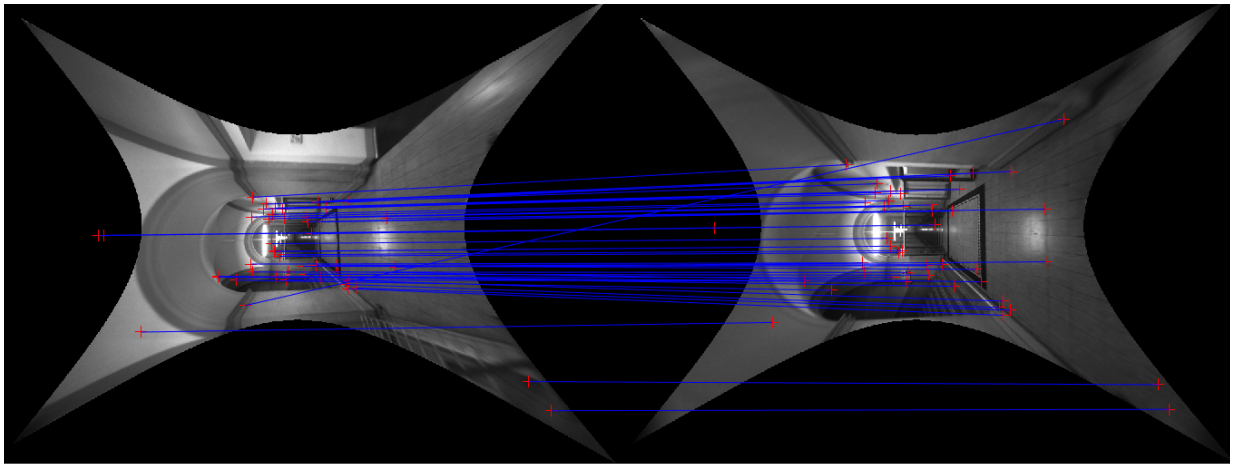

Figure 24: RANSAC, 20 threshold, 1000 iterations.

3. Also, in corridor scene error it is more possible when points are on the different «poles» in relation to the epipole (but near the epipolar line), what induces scene-specific errors.

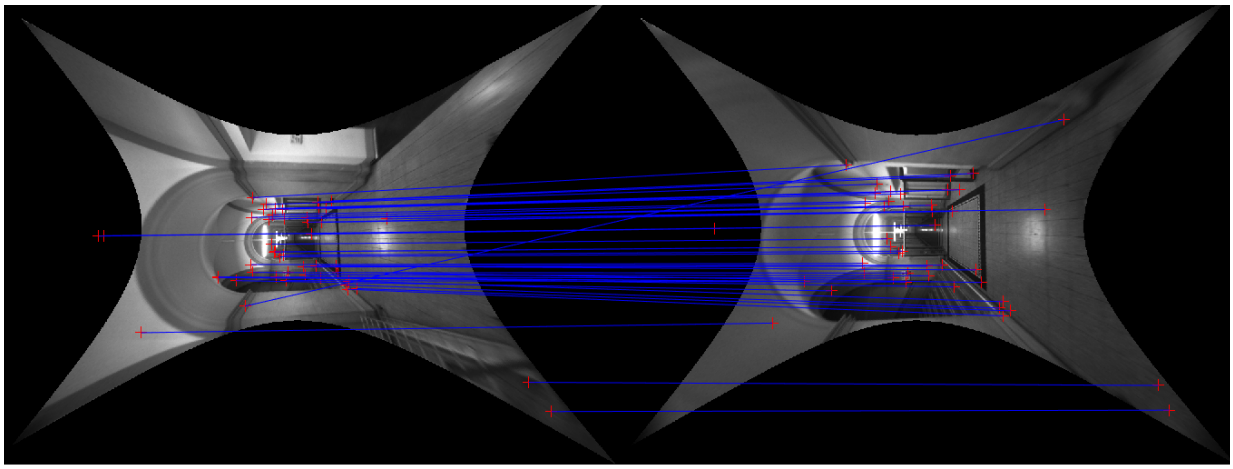

Figure 25: Adaptive RANSAC, 20 threshold, 695 iterations.

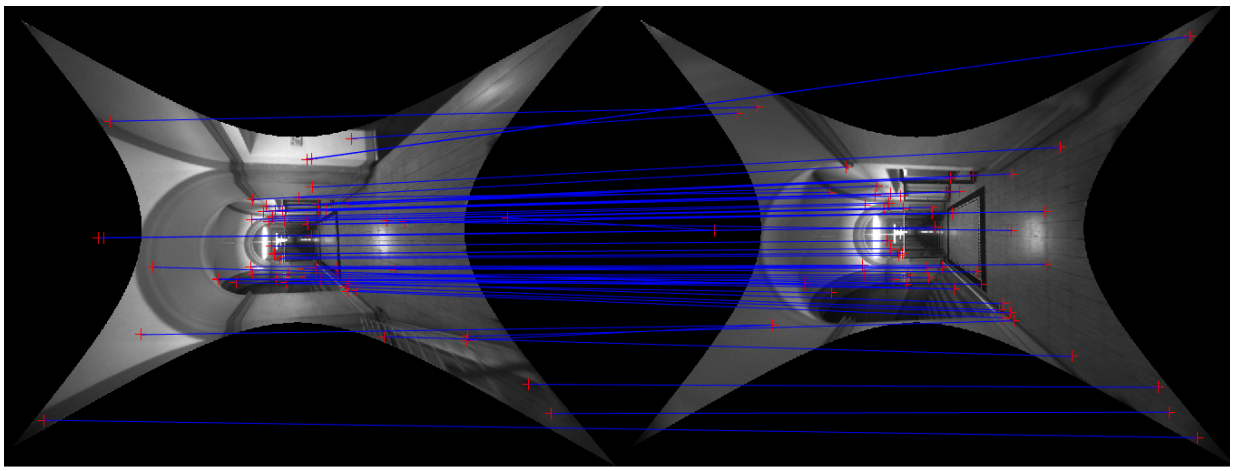

Figure 26: RANSAC, 100 threshold, 1000 iterations.

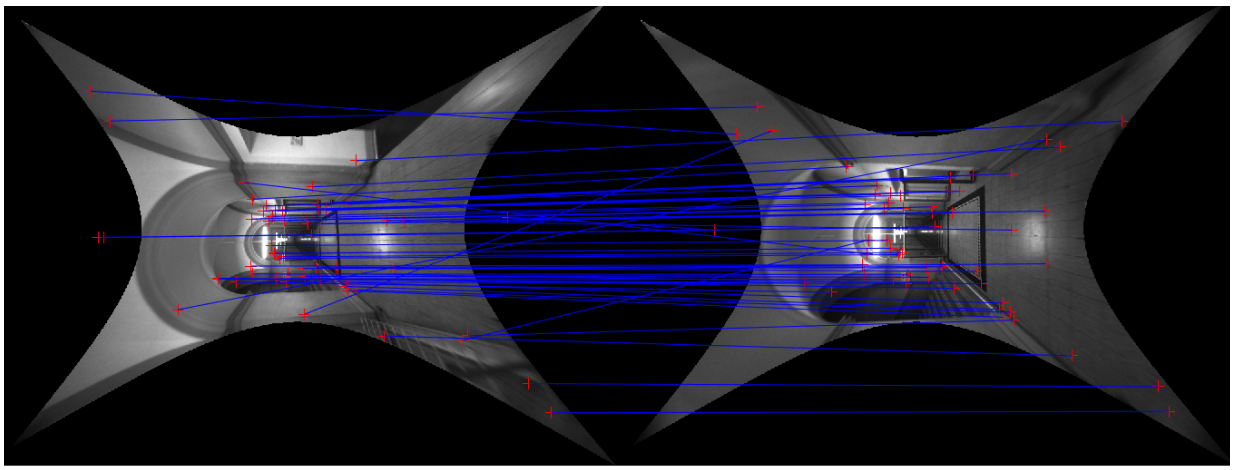

Figure 27: Adaptive RANSAC, 100 threshold, 137 iterations. Amount and quality (though bad here, too big threshold) of inliers is the same as was with non-adaptive algorithm, but less iterations, that is beneficial