An Efficient Preconditioner and a Modified RANSAC for Fast and Robust Feature Matching

Anders Hast
Uppsala University,
Uppsala, Sweden
anders.hast@it.uu.se

Andrea Marchetti
IIT, CNR
Pisa, Italy
andrea.marchetti@iit.cnr.it

ABSTRACT

Standard RANSAC does not perform very well for contaminated sets, when there is a majority of outliers. We present a method that overcomes this problem by transforming the problem into a 2D position vector space, where an ordinary cluster algorithm can be used to find a set of putative inliers. This set can then easily be handled by a modified version of RANSAC that draws samples from this set only and scores using the entire set. This approach works well for moderate differences in scale and rotation. For contaminated sets the increase in performance is in several orders of magnitude. We present results from testing the algorithm using the Direct Linear Transformation on aerial images and photographs used for panographs.

Keywords

RANSAC, Preconditioner, Homography, Clustering, Feature Matching, Image Stitching.

1 INTRODUCTION

RANSAC was introduced by Fischler and Bolles more than 30 years ago [FB81] and is one of the far most used algorithms for finding corresponding pairs of feature points in images. Distinguishing these so called true matches or inliers from the outliers or non matching pairs is essential for many applications of computer vision, such as image stitching [Sze10], 3D reconstruction [Pol00] and point-cloud shape detection [SWK07], just to mention a few. Many variants have been proposed since then, trying to enhance performance of the algorithm in different ways, as will be shown in the end of this section.

One disadvantage with standard RANSAC is that it handles contaminated sets poorly. In fact, many implementations of RANSAC do not perform well when the number of inliers is less than 50% [Low04]. RANSAC is based on random sampling, as the name itself suggests: RANdom Sample Consensus and the probability of finding an initial sample containing inliers only, decreases when the amount of outliers increases. Furthermore, RANSAC usually terminates when the probability of finding more inliers is low or rather when an outlier free set has been picked with some predefined probability. Nonetheless, for heavily contaminated sets, the output is not useful as it usually contains too few inliers if any. Moreover, the output set of putative inliers is often contaminated with outliers. Another consequence for highly contaminated sets is that the stopping criterion might indicate that there is not yet a consensus, while most of the inliers are already found.

Contributions and Delimitations

We propose a naive preconditioner that eliminates the majority of outliers before running a modified version of LO-RANSAC [CMK03] on the set. The preconditioner transforms the problem of finding the consensus set to a position vector space, where an ordinary clustering algorithm can be used to find the cluster that contains the putative inliers. It will be show in examples that the approach works well if the differences in rotation and scale are moderate, which they usually are for matching of images with mainly sideway camera translations. The modified RANSAC will draw samples from this set only and whenever a larger set is found the local optimization step samples this set at least 4 times while using the homography to score the whole set. This approach will be many times faster for contaminated sets than ordinary RANSAC, as the transformation is simple and clustering is relatively fast. Moreover, the modified RANSAC will find consensus in very few iterations as it works on a set with a large majority of inliers. The preconditioner will therefore reduce the number of iterations in the modified RANSAC by orders of magnitude for contaminated sets. Since clustering can be done with $O(n)$ complexity it could also be used for sets with low contamination as it will be fast. However, In this paper an $O(n^2)$ algorithm was used.

The proposed approach will be compared to standard RANSAC only, as many of the already proposed extensions of RANSAC could be used to enhance the modified RANSAC. Especially, MultiRANSAC [ZKM05] could be used when there are multiple planes in the images. Nonetheless, the proposed approach is able to handle such cases too and it will be discussed how.

Furthermore, we have chosen to delimit ourselves in this paper to use a perspective transformation based on at least four points, the so called Direct Linear Transformation (DLT)[HZ03], which throughout the text we will be referred to as the homography. This transformation can be used for a number of applications such as image stitching of aerial images and panographs.

RANSAC and some of its Variants

Standard RANSAC proceeds in the following way: first a minimal number of points is selected, which is required to determine the homography [BL07] [HZ03] [VL01], which is the projective transformation between the images. Then the set is scored so that the inliers that falls below a certain predefined tolerance ε are counted. After transforming using the homography, these points are close enough to its corresponding match and are therefore regarded as true inliers. The algorithm terminates when the probability of finding a better model falls under a predefined threshold, otherwise it starts all over.

RANSAC generally treats all correspondences equally and draws random samples uniformly from the full set. However there are some approaches that tries to exclude probable outliers early on or alternatively determining which ones are probable inliers. Just to mention a few: MLESAC [TZ00] performs non-uniform, i.e. guided sampling of correspondences and PROSAC [CM05] draw samples from a progressively larger set of top-ranked correspondences. GODSAC [MvHK*06] use an assessment driven selection of good samples, instead of random sampling. Fuzzy RANSAC [LK07] divides the input data into categories depending on the residual error and sampling is done in the good set. Another approach [ZK06] transforms the whole problem into classification of the residual distribution. SCRAMSAC [SLK09] tries to reduce the number of outliers using a spatial consistency check. R-RANSAC [CM08], was proposed for the situation when the contamination of outliers is known, using a randomized model verification strategy. Cov-RANSAC [RFP09] incorporates the inherent uncertainty of the estimation procedure in order to achieve a more efficient algorithm. GroupSAC[NJD09] take advantage of additional grouping information between features provided by optical flow based clustering.

Other approaches, designed for real-time tracking take into account that there are similarities between a series of images captured by a camera. Hence, the order of the scoring of the pairs of matches can be planned in order to avoid scoring useless pairs. KALMANSAC [VJFS05] was designed for estimation of structure from motion (SFM). It is derived from pseudo-Bayesian filtering algorithms in a sampling framework and can handle sequences containing large number of outliers. Other examples from robotics are Preemptive RANSAC [Nis03] and Iterative RANSAC [KK06].

Other important contributions to RANSAC use different strategies. MultiRANSAC [ZKM05] is a parallel version that allows to deal with multiple models and have the advantage of being able to cope with a high percentage of outliers. GASAC [RH06] is another parallel approach using a genetic algorithm approach. Moreover, RANSAC has a low probability to find the correct solution when the data is quasi degenerate and QDEGSAC [FP05] was proposed for use in such cases. NAPSAC[MTN*02] was proposed for problems with high noise and takes advantage of the fact that if an inlier is found then any point close to that point will have a high probability to be an inlier.

There are many more versions proposed in literature and a performance evaluation of some of the more important variants of RANSAC is done by Choi et al. [CKY09] and a comparative analysis of RANSAC is given by Raguram et al. [RFP08]. Lowe [Low04] proposed to use the Hough transform [DH72] for clustering data instead of RANSAC, and there are even hybrids [HH07]. Nevertheless, RANSAC is after more than 30 years still used and improved for computer vision applications.

2 THE PRECONDITIONER

The idea is to use a preconditioner that transforms the problem of finding the consensus set to finding a cluster in a position vector space. Generally, a vector can be constructed from two points and each matching pair consists of exactly two points. Hence, it can be regarded as a vector from image a to image b, just like how the final homography transforms each point in image a to its corresponding point in image b, within a certain threshold ε. The main advantage is that a position vector can be treated as a 2D point rather than a 2D vector. The position vector will be scaled in the range $[0..1]$ so that the cluster algorithm can be given a tolerance ε_c similar to the tolerance ε used for the modified RANSAC. This is done by dividing the vector by the length of the sum of the sides in each direction, where image b is translated in each direction using the lengths of image a, so that there is no spatial overlap between the images.

Let the position vector between the feature point at (x_0, y_0) in image a and the corresponding point (x_1, y_1) in image b be:

$$\mathbf{v} = \left[\frac{(a_x + x_1) - x_0}{a_x + b_x}, \frac{(a_y + y_1) - y_0}{a_y + b_y} \right], \qquad (1)$$

where a_x, a_y and b_x, b_y are the sizes in the x and y direction for image a and b, respectively.

The nice result of such an approach is that true matches will yield points in the 2D space that are forming a cluster, while outliers will be spread out in a more random fashion. The search for true matches can therefore be done using any appropriate 2D clustering algorithm, since the vectors are regarded as points rather than vectors, i.e. they are position vectors. This is true also for cases when the images are taken from a sequence of a forward camera motion. Some prefer to visualize the matching by showing only one of the images using lines that start in the feature points in that image. The line ends in the point corresponding to the feature points of the second image, which is not shown but is supposed to overlap the first image. If the images are taken from a sequence of a forward camera motion this approach will yield lines that can point in independent directions. However, if the proposed approach is used, where the images are put so that they do not overlap and they do not share an edge, then they will all have a similar direction compared to the outliers. The direction and length of these lines or vectors, will not be exactly the same and can vary. Nonetheless, they will usually vary a lot less than compared to the outliers even if the cluster will be less dense.

Clustering

There are many clustering algorithms [CRW91] [JMF99] that could be used and some of the more popular are k-means clustering [Mac67] and the mean shift algorithm [CM02]. In our tests it was chosen to use a simple approach that for each point (position vector) in the set computes the distance to all other points. The point that have most neighbors closer than the threshold ε_c will be chosen as the cluster center and all points in the cluster are considered putative inliers. Obviously a better algorithm could be used, especially for situations where the points lie in different planes giving different homographies. However, focus in this paper does not lie on the clustering algorithm as it is a well studied area. Hence, we will instead focus on the preconditioner that transforms the matches to the new position vector space and on how to treat the clustered points using a modified version of LO-RANSAC.

The cluster will contain a majority of the inliers and also some outliers depending both on the tolerance ε_c and how well the cluster algorithm performs. Nonetheless, it is not of vital importance that the cluster will contain inliers only, as the modified RANSAC will clean it up. In all our tests we used the same value of ε_c for the clustering as the tolerance ε for the modified RANSAC.

The computational cost for the preconditioner is rather low. We used a simple approach to find the cluster center. First the vectors are computed and then the clustering algorithm needs to find the cluster. The cost of computing the distance between all points in the space for a brute force algorithm is $n(n-1)/2$, hence the complexity is $O(n^2)$. Then all points sufficiently near the point with most neighbors need to be found. Nevertheless, this cost could be reduced by dividing the space using for instance quad trees [FB74] or kd-trees [TBK08]. Moreover, binning would reduce the complexity to $O(n)$ as each point is classified to belong to a bin depending on its spatial location, in a linear search. The bin with most points will be chosen as the cluster. Nonetheless, the borders of the bins may divide the cluster and this can easily be handled by overlapping bins. Once again the bin with most points are the putative inliers. The size of the bins would be proportional to the tolerance ε_c.

A Modified RANSAC

A modified version of the LO-RANSAC [CMK03] [CMO04] algorithm is here proposed, which utilizes a local optimization step. Both the cluster and the whole set are input parameters to the algorithm, which samples from the cluster only. As the cluster contains the set, which is close to the final solution and therefore pretty free from outliers, it was chosen to sample using up to half of the matches in the cluster but obviously never less than four. This usually lead to consensus faster than sampling just four samples every time, which give less support compared to using up to half of them. This set is used to estimate the homography and scoring the number of inliers.

Every time scoring gives a maximum set of inliers the local optimization step samples iteratively from this set and estimates the homography from it. However, scoring is done using the whole set. Once again it is more efficient to use up to half the size of the set, when doing re-estimation and re-scoring. Whenever a larger set is found it uses this set to sample from and restarts the local optimization loop. We have found that about 4 iterations is usually enough for the proposed approach, while Chum et al used 10 iterations. This is of course a value that can be increased if necessary. The algorithm terminates when the probability is 99% that we have picked an outlier free set and the parameters for this test are constructed using the set belonging to the cluster. Generally, N iterations are need in order to find an outlier free set with the probability p as:

$$N = \frac{\log(1-p)}{\log(1-\gamma^s)}, \tag{2}$$

where γ is the inlier ratio, i.e. number of inliers divided by number of points in the cluster and s is the number of samples drawn each time.

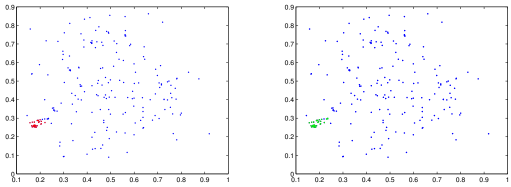

Figure 1: The result of using the preconditioner of a quasi degenerate set, where all points in blue are outliers. Left: the cluster found by the clustering algorithm (red). Right: the points in the space that corresponds to the true inliers found by the modified RANSAC (green).

If N is larger than the number of iterations of the main loop the algorithm starts all over and samples the cluster set from the preconditioner once again.

3 RESULTS

Several tests using different images were conducted in order to prove the efficiency of the proposed preconditioner and the modified RANSAC. The Harris corner detector [HS88] was chosen to detect features in most of the tests instead of the more accurate SIFT detector [Low04]. As the proposed method will be of interest especially for sets with rather high contamination, at least 50%, Harris is preferable as it is less accurate than SIFT.

A Quasi Degenerate Set

A quasi degenerate set with an inlier ratio of 0.1651 was chosen and according to equation 2 it would need 6196 iterations to find an outlier free set. The perspective distortion is small in these aerial images. However, they are rotated in a way that it becomes very hard for standard RANSAC to find the consensus set. A test was performed 10 000 times measuring how many iterations were needed to find *all* inliers and the result was on average 32 887 iterations. Moreover, a test was done 10 000 times counting both number of iterations and number of inliers using the preconditioner and the modified RANSAC. The result is shown in Table 1 on the first row. The preconditioner finds the set (red) in Fig. 1 at the left. In the right is the same points with the inliers (green) that are found by the modified RANSAC.

The images used for the result in Fig. 1 are shown in Fig. 2. The true inliers are connected by yellow lines and the outliers with red ones. The set is quasi degenerate because the true matches covers a small area, which is rather elongated. Furthermore, the images are not perfectly aligned with each other as there is about 18 degrees of rotation between them. This causes standard RANSAC to find just a portion of the inliers in most runs. This problem is overcome by the preconditioner as it finds the major part of the inliers and the modified RANSAC draws sample from this set.

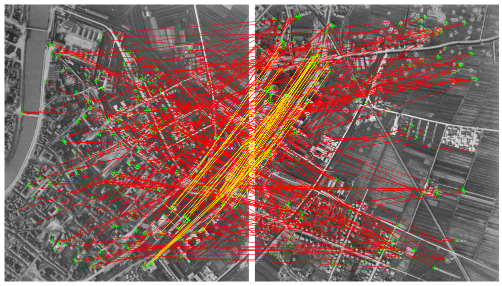

Figure 2: ©MiBAC-ICCD, Aerofototeca Nazionale, fondo RAF. Two historical photos taken over Pisa during WWII, with the true inliers connected with yellow lines (16.5%) and the false matches with red lines (83.5%)

	Iterations			Inliers	
	N	μ	σ	μ	σ
1	6196	6.144	0.452	36.000	0.000
2	1123000	17.904	6.967	81.005	0.261
3	87	6.000	6.000	211.000	0.000
4	8271	11.103	5.747	46.999	0.150
5	26	7.000	0.000	225.000	0.000
6	731	8.817	2.878	76.723	0.828
7	9	6.000	0.000	137.000	0.000
8	90	7.934	3.573	11.954	2.646
9	18	6.005	0.071	90.000	0.000

Table 1: The number of iterations (theoretical) and the mean and standard deviation for number of iterations and inliers for different matchings and images.

A Heavily Contaminated Set

A heavily contaminated set with just an inlier ratio of 0.045 was obtained by increasing the number of feature points and the ratio for the matching. Figure 3 shows how the preconditioner finds the cluster (red) in the image at the left. In the right is a close-up of the inliers (green). Standard RANSAC would, according to equation 2, need about 1 123 000 iterations to find the majority of inliers. After the preconditioner, the probability is increased to 0.9759, which corresponds to 1.93 iterations on average. The modified RANSAC could easily find almost all inliers in every run in just about 17 iterations as shown on the second row in Table 1 which is an enormous increase in performance compared to the theoretical 1 123 000 iterations.

An Almost All Inlier Set

A set that is almost outlier free with an inlier ratio is 98.40% was tested and Figure 4 shows how the preconditioner finds the whole set (red). The theoretical number of iterations are just 1.66 and the modified RANSAC needs 6 iterations to find the set, which

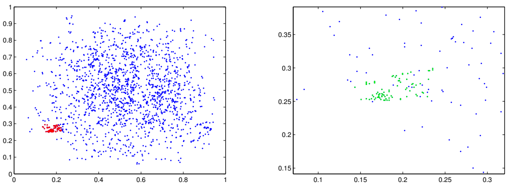

Figure 3: The result of using the preconditioner of a highly contaminated set, where all points in blue are outliers. Left: the cluster found by the clustering algorithm (red). Right: a close up of the points in the space that corresponds to the true inliers found by the modified RANSAC (green).

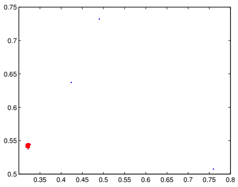

Figure 4: A set with an inlier ratio of 98.40% **The preconditioner finds the cluster (red), which is the same set as the modified RANSAC will find.**

could be reduced by diminishing the number of iterations in the local optimization.

Multiple Planes

A set of photos taken on ground were used to test the algorithm for stitching of panographs. The preconditioner is also able to find the correct cluster for images where the perspective distortion is greater and as in this case, where there are several planes. The preconditioner and modified RANSAC was used for the set of images shown in Figure 5 and Figure 6 shows how the preconditioner finds the cluster (red) in the image to the left and on the right is a close-up of the inliers (green). The cluster becomes elongated and curved because of perspective distortions and the three planes in the image. Nevertheless, the clustering algorithm is able to find the cluster containing all three planes. A more sophisticated clustering algorithm might be able to separate it into three clusters. However, this task could also be handled by a version of RANSAC that finds multiple planes.

The modified RANSAC finds all inliers in every run in just 6 iterations as shown on the third row in Table 1 compared to the theoretical value of 87 iterations for finding 99% of the inliers. This is not a huge increase in performance. However, keep in mind that the modified RANSAC finds all inliers in every run for this set, which is not the case for standard RANSAC. Increasing the rate to 99.99% would double the theoretical number of iterations needed.

Another set of images were used in the next test and Figure 7 shows how the preconditioner finds the cluster (red) at the left. In the right is a close-up of the

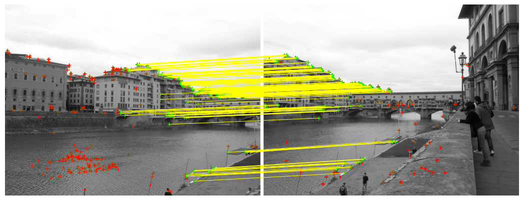

Figure 5: ©Anders Hast. Two images of the "Ponte Vecchio" in Florence, Italy. The inliers are connected with yellow lines (47.63%) and the false matches are depicted with red crosses (52.37%)

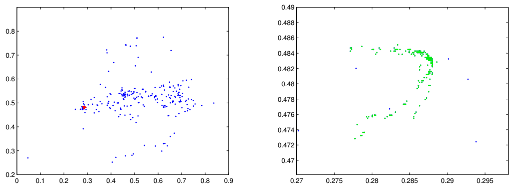

Figure 6: A moderately contaminated set with an inlier ratio of 47.63% **Left: The preconditioner finds the cluster (red). Right: A close up of the inliers found by the modified RANSAC (green).**

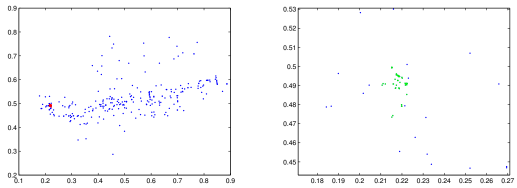

Figure 7: A contaminated set with an inlier ratio of 15.36% **Left: The preconditioner finds the cluster (red). Right: A close up of the inliers found by the modified RANSAC (green).**

inliers (green). The modified RANSAC finds almost all inliers in every run in just 11 iterations as shown on the fourth row in Table 1 compared to the theoretical value of 8271 iterations.

Yet another set of images where used and Figure 8 shows how the preconditioner finds the cluster (red) at the left. In the right is a close-up of the inliers (green). The modified RANSAC finds almost all inliers in every run in just 7 iterations as shown on the fifth row in Table 1 compared to the theoretical value of 26 iterations. Standard RANSAC found all inliers in an average of 157.6 iterations with a standard deviation of 156.4. Hence, the proposed method have the advantage of being less variable when it comes to the number of iterations, also for medium contaminated sets.

Finally we made a test using SIFT on a pair of images where there are two separate planes as shown in Figure 9. This time SIFT was used since Harris was not able to detect the points we were interested in. One set of inliers is connected with yellow lines and the other with blue ones. The cluster found for the yellow

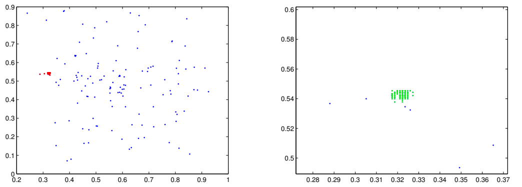

Figure 8: A set with an inlier ratio of 63.38% Left: The preconditioner finds the cluster (red). Right: A close up of the inliers found by the modified RANSAC (green).

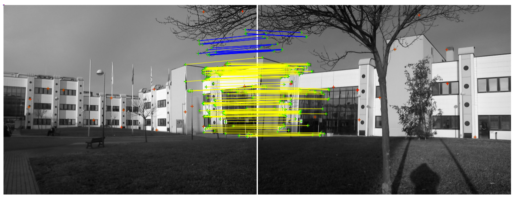

Figure 9: ©Anders Hast Two images with two clearly separable planes. The major set of inliers are connected with yellow lines and the minor set of inliers with blue ones.

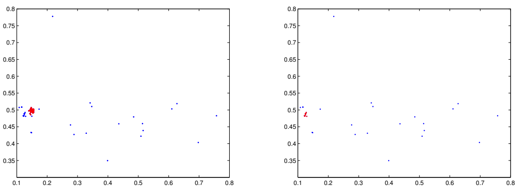

Figure 10: The preconditioner finds the first plane corresponding to the cluster (left). RANSAC finds at the inliers corresponding to that cluster and it is removed from the set. The preconditioner finds the second plane (right) corresponding to the cluster (red)

ones is shown in the left of Figure 10 and the other cluster in the right. Since the clusters were easily separated the preconditioner and modified RANSAC was run to find the first set of inliers. Then these were removed from the whole set and the procedure was repeated. The next set of inliers was easily found by the proposed approach. The result is on row six and seven in the Table 1 for each set.

Scale Differences

One aerial image was scaled down to 75% of its size to examine the impact on the cluster. As can be seen from Figure 11 the shape of the cluster is different from the one in Figure 8 even if they are exactly the same images. Depending on the cluster approach used and the tolerance ε_c, the preconditioner will find such sets as well and the modified RANSAC has no problems of handling them. (See row eight in Table 1.)

In the second test a pair of images, shown in Figure 12 were matched using SIFT in order to obtain

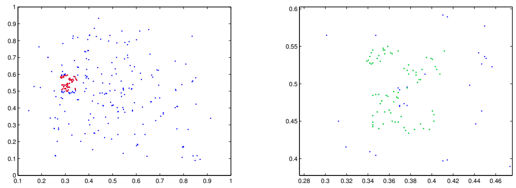

Figure 11: The Cluster (left) becomes larger in size when the scale in the input images are different. To the right is the inliers (green).

Figure 12: ©Anders Hast. Two images with taken on different distances from the main object (the chuch towers). The set of inliers are connected with yellow lines and the outliers are depicted with red crosses.

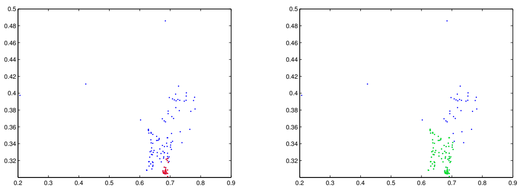

Figure 13: A set with an inlier ratio of 63.38% Left: The preconditioner finds the cluster (red). Right: A close up of the inliers found by the modified RANSAC (green).

better matches of the images that were taken on different distances to the object. Hence, the same problem of scale will occur. Figure 13 shows how the preconditioner finds just a part of the set (depending on the tolerance ε_c), which once again becomes more spread over a larger area. Anyhow, the modified RANSAC will find the whole set and the result is on row nine in Table 1.

Efficiency

Some further testing were done to test the efficiency of the method and the results are shown in Table 2. Four sets of aerial images (four first rows) and six sets of photos taken on the ground (rows five to ten) were used. The tests were once again performed 10 000 times. In the first column is the size of the clusters obtained by the preconditioner. Next is the theoretical number of iterations (N). Then follows the mean μ and standard deviation σ of the number of iterations needed by the modified RANSAC to find the inliers.

	Cluster Size	Iter. N	μ	σ	Inl. μ	σ
1	559	19	7.5	1.5	551.0	0.09
2	82	569	28.9	11.9	73.6	4.9
3	37	9144	7.9	2.3	36.7	0.48
4	50	$2.3 \cdot 10^4$	14.5	6.2	49.0	0.7
5	221	61	10.9	1.9	196.9	0.5
6	141	214	19.6	5.1	119.8	0.97
7	105	891	25.7	10.5	101.0	0.25
8	61	6109	20.6	7.7	53.1	1.3
9	44	$6.0 \cdot 10^4$	6.0	0.03	56.0	0.0
10	56	$2.4 \cdot 10^6$	6.2	0.5	67.0	0.09

Table 2: Four test runs for aerial images and six for images with multiple planes, with the cluster size, the mean and standard deviation for number of iterations and inliers.

The next values in the end of the row is the μ and σ of number of inliers.

The proposed approach is able to find most of the inliers with low deviation, except for the case on row 2 and 8, which have a σ greater than 1.0. Remember that only 4 iterations are done in the local optimization step and the σ could be decreased by increasing this number, which of course would increase the number of iterations in total.

Obviously, the proposed approach is very efficient as it reduces the number of iterations while still maintaining a high accuracy in terms of number of inliers found. Most remarkably is that the preconditioner makes it, not only possible, but even easy and fast to find the consensus set when the theoretical number of iterations exceeds tenth's of thousands and even millions. The result on row ten is from a set with an inlier ratio of 0.0372 and the theoretical number of iterations exceeds 2.4 million iterations. By using the preconditioner the number of iterations were 6.2 on average with only 0.5 in deviation. All 67 inliers were found in almost every run with a deviation of only 0.09. When the proposed approach does not find all inliers in every run one could increase the number of iterations in the local optimization step to increase the probability of finding more inliers.

4 DISCUSSION

It is important to set an appropriate tolerance ε for RANSAC and likewise it is important that the tolerance ε_c is set properly for the preconditioner. By scaling the position vector into the range [1..0] it is possible to use the same tolerance for both. Nonetheless, care must be taken so that the tolerance is proportional to the size of the image. Moreover, one must take into account the scale differences as it will affect the size of the cluster and the tolerance must be set accordingly. A similar case is when the images are taken during a forward camera movement, which yields images with different scales. It has been shown that the preconditioner is able to handle moderate changes in scale, even if only a part of the cluster is found because the cluster becomes proportionally larger, i.e it is spread out.

When there are multiple planes in the image, the cluster will be a bit different and sometimes it is even separable in space, but not always. Here some more sophisticated clustering algorithm could be used in order to separate the clusters in a more accurate way. Nonetheless, the preconditioner was able to find the main cluster in all our tests and the modified RANSAC extracted all inliers from the set. Hence, it is possible to modify and use some other version of RANSAC that is able to yield separate planes such as Multi-RANSAC [ZKM05]. Otherwise, one could also in many cases extract one cluster at a time and run the modified RANSAC on each of them.

The size of the cluster will also affect the result and different ε_c could be tested. Moreover, it is possible to change the performance by changing how many samples are drawn. Usually four samples are drawn in standard RANSAC. However, by increasing this number to half of the current consensus set, but obviously never less than four, performance was increased for the modified RANSAC. One could experiment further with what is actually the optimal number to use.

We delimited ourselves to use the four point DLT. Nonetheless, there is nothing that prevent using other types of homographies. In any case, the output of the preconditioner is independent of the homography. It is just the result of the modified RANSAC that might change depending on what homography is being used. Moreover, we used a clustering algorithm that was easy to implement but is not the fastest one. Nevertheless, what clustering algorithm to use is not so important. The important thing is that it finds the cluster and preferably does that fast.

5 CONCLUSION AND FUTURE WORK

Standard RANSAC handles highly contaminated sets poorly as the probability of drawing samples giving an outlier free set after scoring becomes very small. This problem can easily be overcome by the proposed preconditioner that transforms the problem to a position vector space where each vector is a scaled vector representing the matches. An ordinary clustering algorithm can be used to find the cluster of putative inliers. This set is then processed by a modified version of RANSAC that draws from this set exclusively but scores using the whole set. This approach will increase performance substantially for contaminated sets. The preconditioner can be used also for sets with low contamination as the clustering algorithm is relatively fast compared to estimation and scoring in the RANSAC procedure.

The preconditioner can be modified in such way that more powerful clustering methods are used in order to find more than one projection plane. Moreover, it should be determined how large differences in scale and rotation the preconditioner can handle and also what could be done to handle the more extreme cases.

6 REFERENCES

[BL07] Brown M., Lowe D. G.: Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision 74*, 1 (2007), 59–73.

[CKY09] Choi S., Kim T., Yu W.: Performance evaluation of ransac family. In *British Machine Vision Conference* (2009), pp. 1–12.

[CM02] Comaniciu D., Meer P.: Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis Machine Intelligence (PAMI) 24*, 5 (2002), .603–619.

[CM05] Chum O., Matas J.: Matching with prosac - progressive sample consensus. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2005), pp. 220–226.

[CM08] Chum O., Matas J.: Optimal randomized ransac. *IEEE Transactions on Pattern Analysis and Machine Intelligence 30*, 8 (2008), 1472–1482.

[CMK03] Chum O., Matas J., Kittler J.: Locally optimized ransac. In *the Annual Pattern Recognition Symposium of the German Association for Pattern Recognition (DAGM)* (2003), pp. 236–243.

[CMO04] Chum O., Matas J., Obdrzalek S.: Enhancing ransac by generalized model optimization. In *Asian Conference on Computer Vision (ACCV)* (2004).

[CRW91] Capoyleas V., Rote G., Woeginger G.: Geometric clusterings. *Journal of Algorithms 12* (1991), 341–356.

[DH72] Duda R. O., Hart P. E.: Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM 15* (1972), 11–15.

[FB74] Finkel R. A., Bentley J. L.: Quad trees a data structure for retrieval on composite keys. *Acta Informatica 4*, 1 (1974), 1–9.

[FB81] Fischler M. A., Bolles R. C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM 24* (1981), 381–395.

[FP05] Frahm J. M., Pollefeys M.: Ransac for (quasi-) de-generate data (qdegsac). In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2005), pp. 220–226.

[HH07] Hollander R. J. M. D., Hanjalic A.: A combined ransac-hough transform algorithm for fundamental matrix estimation. In *British Machine Vision Conference* (2007).

[HS88] Harris C., Stephens M.: A combined corner and edge detection. In *Alvey Vision Conference* (1988), pp. 147–151.

[HZ03] Hartley R. I., Zisserman A.: *Multiple View Geometry â 2nd edition*. Cambridge University Press, 2003.

[JMF99] Jain A., Murty M., Flynn P.: Data clustering - a review. *ACM Computing Surveys 31*, 3 (1999), 264–323.

[KK06] K K. T., Kondo E.: Incremental ransac for online relocation in large dynamic environments. In *IEEE International Conference on Robotics and Automation (ICRA)* (2006), pp. 1025–1030.

[LK07] Lee J. J., Kim G.: Robust estimation of camera homography using fuzzy ransac. In *Proceedings of the 2007 international conference on Computational science and its applications - Volume Part I* (Berlin, Heidelberg, 2007), ICCSA'07, Springer-Verlag, pp. 992–1002.

[Low04] Lowe D. G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2 (2004), 91–110.

[Mac67] MacQueen J. B.: Some methods for classification and analysis of multivariate observations. In *5-th Berkeley Symposium on Mathematical Statistics and Probability* (1967), vol. 1, Berkeley, University of California Press, pp. 281–297.

[MTN*02] Myatt D., Torr P., Nasuto S., Bishop J., Craddock R.: Napsac: High noise, high dimensional robust estimation - its in the bag. In *British Machine Vision Conference* (2002), vol. 2, pp. 458–467.

[MvHK*06] Michaelsen E., von Hansen W., Kirchhof M., Meidow J., Stilla U.: Estimating the essential matrix: Goodsac versus ransac. In *Photogrammetric Computer Vision* (2006), pp. 1–6.

[Nis03] Nister D.: Preemptive ransac for live structure and motion estimation. In *International Conference on Computer Vision (ICCV)* (2003), pp. 109–206.

[NJD09] Ni K., Jin H., Dellaert F.: Groupsac: Efficient consensus in the presence of groupings. In *ICCV* (2009), IEEE, pp. 2193–2200.

[Pol00] Pollefeys M.: Automated reconstruction of 3d scenes from sequences of images. *ISPRS Journal of Photogrammetry and Remote Sensing 55*, 4 (2000), 251–267.

[RFP08] Raguram R., Frahm J.-M., Pollefeys M.: A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. In *European Conference on Computer Vision (ECCV)* (2008), pp. 500–513.

[RFP09] Raguram R., Frahm J.-M., Pollefeys M.: Exploiting uncertainty in random sample consensus. In *International Conference on Computer Vision (ICCV)* (2009), pp. 2074–2081.

[RH06] Rodehorst V., Hellwich O.: Genetic algorithm sample consensus (gasac) - a parallel strategy for robust parameter estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)* (2006), pp. 1–8.

[SLK09] Sattler T., Leibe B., Kobbelt L.: Scramsac: Improving ransac's efficiency with a spatial consistency filter. In *International Conference on Computer Vision (ICCV)* (2009), pp. 2090–2097.

[SWK07] Schnabel R., Wahl R., Klein R.: Efficient ransac for point-cloud shape detection. *Computer Graphics Forum 26*, 2 (2007), 214–226.

[Sze10] Szeliski R.: Computer vision : Algorithms and applications. *Computer 5*, 3 (2010), 832.

[TBK08] Tsakok J. A., Bishop W., Kennings A.: kd-tree traversal techniques. *2008 IEEE Symposium on Interactive Ray Tracing 44*, 1 (2008), 190–190.

[TZ00] Torr P. H. S., Zisserman A.: Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding 78* (2000), 138–156.

[VJFS05] Vedaldi A., Jin H., Favaro P., Soatto S.: Kalmansac: Robust filtering by consensus. In *International Conference on Computer Vision (ICCV)* (2005), pp. 633–640.

[VL01] Vincent E., Laganiere R.: Detecting planar homographies in an image pair. *Image and Signal Processing and Analysis* (2001), 182–187.

[ZK06] Zhang W., Kosecka J.: A new inlier identification scheme for robust estimation problems. In *Proceedings of Robotics: Science and Systems* (Philadelphia, USA, August 2006).

[ZKM05] Zuliani M., Kenney C., Manjunath B.: The multiransac algorithm and its application to detect planar homographies. In *The International Conference on Image Processing (ICIP)* (2005), vol. 3, pp. 153–156.