A Modified RANSAC mechanism: Multiple models extraction algorithm

Yingdi Xie†, Jun Ohya‡

xieyingdi@ruri.waseda.jp ohya@waseda.jp

? Waseda University, Graduate School of Global Information and Telecommunication Studies (GITS), Doctoral Student

? Waseda University, Graduate School of Global Information and Telecommunication Studies (GITS), Faculty

Abstract:As one of the most frequently used regression methods, RANSAC is advanced in its effectiveness and efficiency, but RANSAC cannot extract multiple models due to its exclusivity. To extract multiple models, this paper proposes a new regression method, which is a modified version of RANSAC. To fit a model to data points, a labeling process classifies each data point into proper inlier, quasi-inlier or outlier. The model is obtained from the proper and quasi inliers. After eliminating the proper inliers, another model fitting is performed. These operations are repeated till no more model is fitted. The effectiveness of the proposed method is shown by experiments on extracting multiple lines from images.

Keywords: RANSAC, regression method

1. Introduction

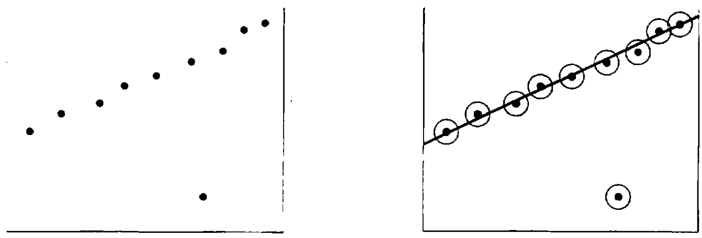
In this paper, we propose a regression method based on Random Sample Concensus (RANSAC, here and after) [1], which could be applied to a wide range applications, [2]. In single model construction problem, especially in multi-dimensional regression, RANSAC is advantageous in its effectiveness of eliminating noise factors in a efficient way. The method discussed in this paper is propsed motivated by the difficulties within the original RANSAC, that it is unable to extract multiple models. In many applications, however, single model is insufficient to characterize the desired information. For instance, in image processing, one image could contain multiple objects with similar shapes or features. Although, previous literatures have been proposed to a certain problem, such as Hough Transform applied to shape detection, they suffer various limitations, such as computational efficiency, especially applying to the problem in a high dimention space.

From this point of view, we adopt RANSAC for its efficiency to compute a model in a high dimensional space, and robustness to noise factors. The modified version of RANSAC is advantage in the ability of multiple models? calculation, through a labeling mechanism. During the process that fits a model to data points, the labeling process classifies each data point into proper inlier, quasi-inlier or outlier. The model is obtained from the proper and quasi inliers if the number of the two kinds of inlier exceeds a pre-defined threshold. After eliminating the proper inliers from the set of the data points, another model fitting is performed for the remaining data points. These operations are repeated till no more model is fitted. The rest of this paper is organized as follows: In section 2, the traditional RANSAC is briefly described, an example of line extraction is expained to make the algorithm easily understandable. Section 3 gives out our proposal. The improvement is introduced by illustrating the same line extraction problem by comparing to the one mentioned in section 2. In section 4, the computational efficiency and the robustness of handling noise factor are demonstrated. The conclusion and future work is given in section 5.

2. RANSAC

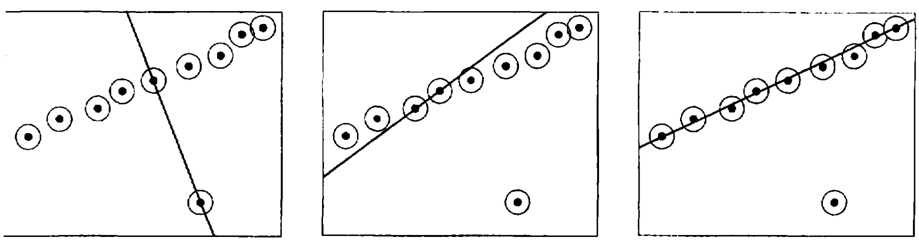
RANSAC is a general regression method, developed by M. Fischler el at in 1981. Up to the 25th anniversary, RANSAC has been cited in over a thounsand literatures in various research area, for its efficiency and effectiveness. The basic idea of computing a valid model from a known data set could be described as follows:

1. Randomly selecting a minimal set of data to compute a model.
2. Compute the distance between each data to the estimated model. Treat the data as an inlier when its distance is under a predefined threshold.



a. b.

Fig.1 Line extraction example. a. original data points image, b. one desirable



a. b. c.

Fig.2 Line detection in a noisy enviorment example. RANSAC mechanism. a. Randomly selecting two points to construct a line model. By counting the inlier, this model is invalid for its inadequety of inlier number. b. repeat the same step. c. a desirable line model to be selected as final result, because it contains the maximal inlier number.

3. If the inlier number for all data to this model is over a preset least inlier threshold, the estimation in this step is recorded as a valid model.
4. Repeat the above steps. The competetion according to the inlier number among estimated valid models could calculate the final result with a maximal inlier.
5. Terminate the above repetition when no better result could be obtained in a certain times of trail. Treat all the non-inliers as outliers.

Under this mechanism, one effective model could be computed if recompute the model with all the inliers selected previously. The inlier labeling process menifest its capability of eliminating undesirable factors, such as noise. In order to explicitly describe its mechanism, we demonstrate a line regression problem illustrated as figure 1 and 2. Assume, a original image contains 10 data points, Fig.1.a, one desirable result is demonstrated as Fig.1.b, where the outlier is denoted as red point. By implementing the RANSAC mechanism, firstly two points are randomly selected to construct a line. Since we could not ensure the validity of the selection, the inlier number is count for every point. As demonstrated in Fig.2.a, the circles around data points, are threshold for evaluating whether a point is inlier to a line model, and the selected data points are represented as yellow background. Obviously, the first trail of selection is not valid, for its inadequety of valid data points. Repeat the same process to select a valid model which contains the maximal inlier number, Fig.2.c.

3. Modified RANSAC

As described in Section 2, the conventional RANSAC has the limitation that all data can generate only one valid model. Nevertheless, various actual applications require to compute multiple models as opposed to only one model. To conquer this kind of problem, we proposed a modifed version of RANSAC, described as follows:

1. Randomly select a minimum data set required to compute a model.
2. Compute the model and count the data number whose distance in disparity space is less than a predefined lower threshold TH_1. Meanwhile, count the data number whose distance is less than a higher threshold TH_2. The inlier data within both thresholds are labeled with the computed model扣mark. Inliers within TH_1 are then labeled 損roper inliers? Inlier data within TH_2 are then labeled 揵oard sense inliers? Both are treated as inlier data.
3. Record the computed model if inlier number over a predefined threshold TH_3. Update the data set by eliminating the 損roper inliers? from the universal data set. Randomly select another minimum datum from the updated set..
4. Repeat step 2 and 3 until no valid model could be found in a certain times of test. The 損roper inlier? label has higher priority than 揵road sense inlier? label.
5. Group the data with more than two model label to the group with more inlier data. Label the data without any model label as invalid data, errors.

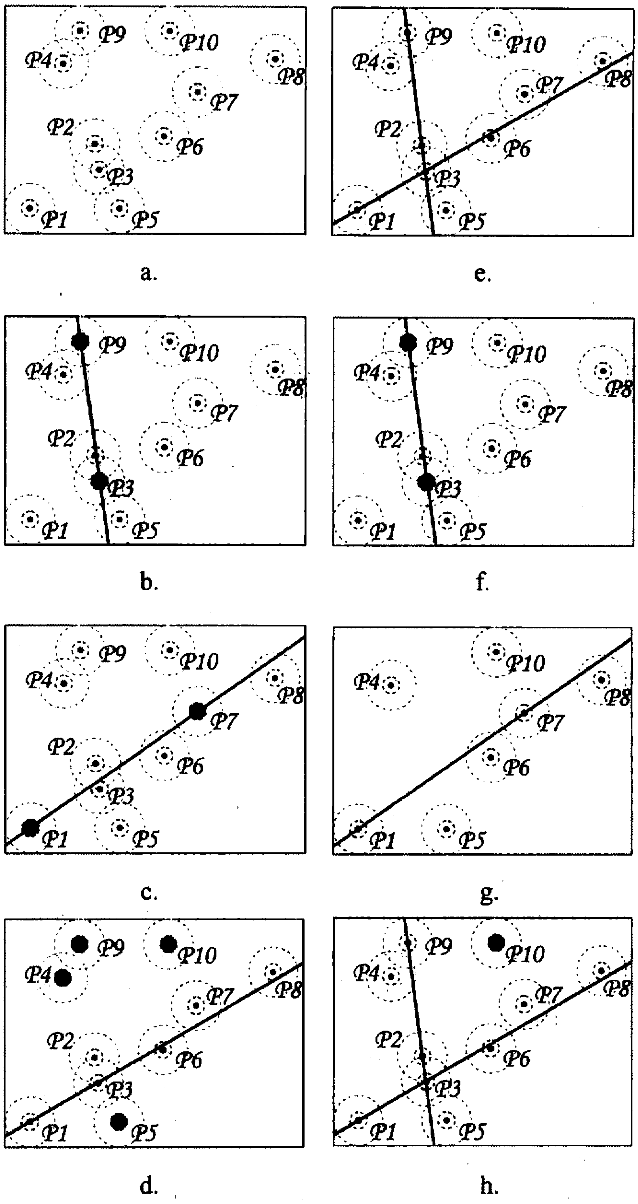
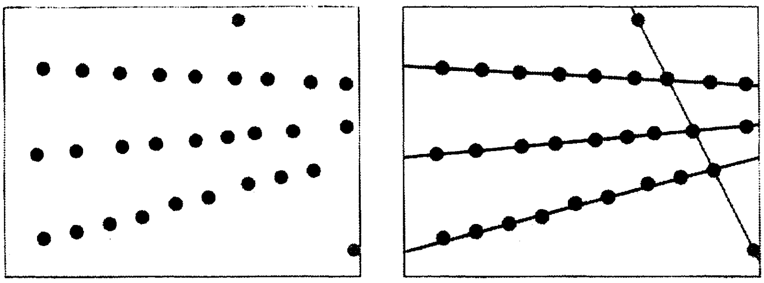


Fig.3. Comparison between traditional RANSAC mechanism a. ? f.; and our proposal. e. ? h.

Here, we show an example of line detection from points, by comparing with traditional RANSAC. As shown in Fig.2.a, suppose there are 10 points in this image. One desirable result of line detection is demonstrated as Fig.2.b. In this example, gray circle and black circle with dashed line around each point denote the threshold of the distance from a line model, TH_1 to label 損roper inlier? and TH_2 to label 揵road sense inlier? respectively. Assume, the minimal inlier for a valid line model is 4.

The computation begins from randomly selecting two points to construct a line model. Notice, the selected points in Fig.3 are labeled as with gray background. Comparison of RANSAC and its modified version are demonstrated as left and right column in this figure respectively. At the initial state of point selection and valid model computation, modified RANSAC implements in the same way as RANSAC, Fig.3.(1) and (2). Where, line model from *P4* and *P7* is not valid for its deficiency of inlier number, while line constructed from *P3* and *P9* is valid. After found a valid model, our proposal will eliminate the proper inliers from the original data set. For the next computation step, points? random selection is implemented in the updated data set. This mechanism could eliminate the unnecessary computational cost. As the results, RANSAC mechanism could only construct one line model in this example, Fig.3.(4), and label all the other non-inlier points as 搊utlier? denotes in red. While our proposal could generate all desirable lines, in this example 2. Because, in the modified version of RANSAC, models are labeled as valid only get enough inlier number, instead of competing with other models.

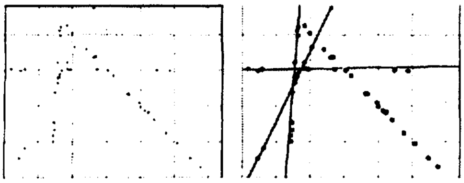


a. b.

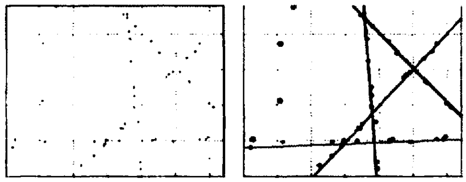
Fig.4, a. original image. b. Line model抯 validity is suffering an uncertainty if its characteristics is weak.

Here is a problem, whether elimination of proper inliers will result in model missing. The answer is yes, but it will only occur where model's characteristics is not strong, and part of the datum, which are inliers of this model, are eliminated in previous computation. Back to the line detection example, as shown in Fig.3.a, three black lines in Fig.4.b could be recognized as valid for their strong line characteristics. As for the gray line, however, we could hardly recognize whether points with gray color belong to a line model, or outlier of the black lines. Under this condition, the gray line may be detected when points two points of this line are chosen in advance. Otherwise, they will be treated as outliers, because the three red points are selected as other lines? inliers and eliminated after model fitting. Thus, model with weak charactistics is suffering an uncertainty, until it has more datum that could prove its validity.

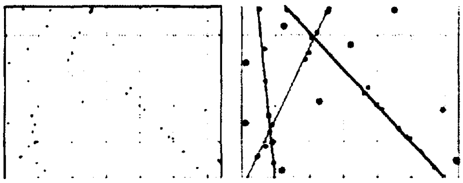
4. Experiments and Evaluation



a. Noise factor = 0

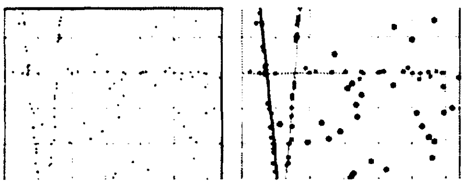


b. Noise Factor = 0.2

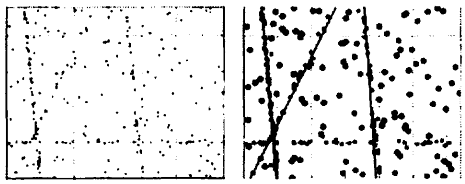


c. Noise factor = 0.4

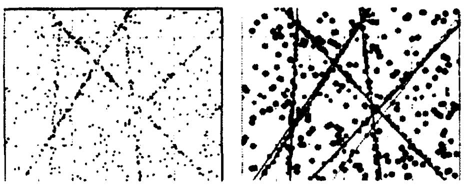
b. Noise Factor = 0.6



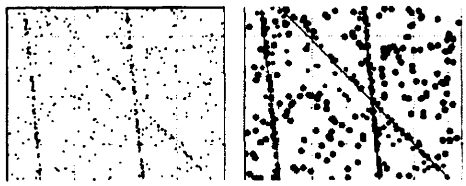
c. Noise factor = 0.8



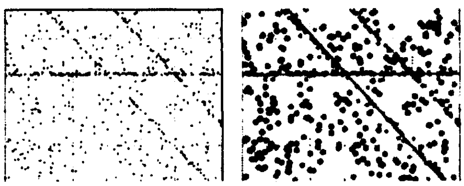
b. Noise Factor = 1.0



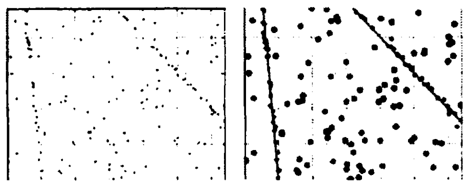
c. Noise factor = 1.2



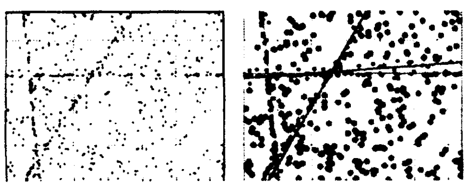
b. Noise Factor = 1.4



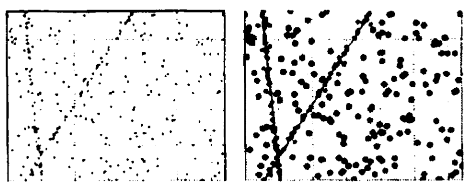
c. Noise factor = 1.6



b. Noise Factor = 1.8



c. Noise factor = 1.8



b. Noise Factor = 2.0

Fig.5 Experiments of multiple line extraction within various noise factor, which ranges from 0 to 2.0

In order to verify the proposed method卌effectiveness, experiments of line extraction within a noisy environment are implemented. For the first step, points with certain variation to multiple lines are generated. Noise points are then genereated randomly scattering all over the image. Notice, in order to explicitly evaluate the result, we arrange certain predefined point numbers to one line, by multiplying which with a predefined noise factor the noise points?number is then decided, as demonstrated in figure 5. In this figure, we only demonstrate some effective results. Nevertheless, such desirable results could not be achieved for a hundred percent. Therefor, for each noise factor condition, we implement the experiment for a hundred times, and the percentage of effectiveness (PoE) with each case are shown as table 1, by which it could be concluded that as the noise factor increases, PoE declines. Notice, in this table, the trail number denotes the number of computation while implementing line detection.

NoiseFactor	PointsNumForEachLine	LineNum	Trail Number	Time (ms)	Accuracy
0	20	3	N	27.500	100%
0.1	20	3	N	6.100	95% -1
0.2	20	3	N	12.700	95% -1
0.3	20	3	N	11.400	90% -1
0.4	20	3	N	9.000	95% -1
0.5	20	3	N	10.200	95% -1
0.6	20	3	N	16.700	90% -1
0.7	20	3	N	14.800	90% -1
0.8	20	3	N	14.100	85% -1
0.9	20	3	N	24.900	80% -1
1	20	3	N	17.800	85% -1
1.1	20	3	N	20.200	70% -2
1.2	20	3	N	23.200	75% -2
1.3	20	3	N	24.000	60% -3
1.4	20	3	N	43.300	50% -2
1.5	20	3	N	31.000	45% -2
1.6	20	3	N	36.800	35% -3
1.7	20	3	N	37.000	35% -2
1.8	20	3	N	39.800	35% -2
1.9	20	3	N	55.400	30% -3
2	20	3	N	43.700	30% -4

Table.1 Effectiveness tesitification ? against Noise Factor

NoiseFactor	PointsNumForEachLine	LineNum	Trail Number	Time (ms)	Accuracy
0.2	50	3	N^2	210.373	98% -1
0.2	50	3	N	13.988	98% -1
0.2	50	3	N/2	11.528	96% -1
0.2	50	3	sqrt(N)	11.100	85% -2
0.2	50	4	N^2	306.317	96% -1
0.2	50	4	N	19.016	96% -1
0.2	50	4	N/2	16.223	98% -1
0.2	50	4	sqrt(N)	13.400	78% -2
0.2	50	8	N^2	1,625.898	40% -3
0.2	50	8	N	51.001	49% -3
0.2	50	8	N/2	43.820	52% -2
0.2	50	8	sqrt(N)	35.650	25% - 4

Table.2 Computational efficiency verification ? against Trail Number

This is a result matches the fact. Because, in a different point of view, the noise factor may result in weaken the lines characteristics, which is the problem discussed in section 3.2. As for the computational efficiency, we testified different trail number for the line models extraction. As shown in table 2, it is concluded that setting a trail number to the value equals to the points number is a reasonable solusion. Notice, the experiments are conducted with a personal computer with a Intel Pentium 4 3.0GHz CPU, by implemeting our proposal to multiple line models extraction. Figure 5 demonstrated the results.

By analyzing the undesirable results, it is obviously that as the noise factor increases, it will more and more confuse the model detection, since the threshold for inlier detection is preset to a constant value. In the line detection problem, the false result is distinct for its low dimension: every point composes of 2 coordinate, x and y. Therefor, the distance between a line model and one point also composes of 2 dimensional summation. Consider a further apllication, to detect line within 3D space, the distance above will increase since its dimension increased. However, thus increment will result in more distinguishable between the inlier and outlier. This could be imagined as a exponetial effect. As a conclusion, application to a problem within more dimension will increase its robustness to noise factor.

5. Conclusion and Future work

In this paper, a modified version of the RANSAC regression method which is capable to handle the multiple model extraction problem is proposed. In the experiment, its effectiveness and computational efficiency are testified. Please note, this modified RANSAC mechanism is NOT a specified method only could handle shape detecion, but also could applied to a wide range of applications, as the traditional RANSAC did. In the near future, we are goint to do further research on the application of the modified RANSAC mechanism.

Reference

[1] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. Commun. ACM., 24:381? 95, 1981.

[2] M. Agrawal, K. Konolige, and L. Iocchi. Real-time detection of independent motion using stereo. In IEEE workshop on Motion (WACV/MOTION), January 2005.