The Modified pbM-estimator method and a runtime analysis technique for the RANSAC family

Stas Rozenfeld Ilan Shimshoni*
Department of Industrial Engineering and Management.
Technion, Israel Institute of Technology
Haifa, Israel 32000

Abstract

Robust regression techniques are used today in many computer vision algorithms. Chen and Meer recently presented a new robust regression technique named the projection based M-estimator. Unlike other methods in the RANSAC family of techniques, where performance depends on a user supplied scale parameter, in the pbM-estimator technique this scale parameter is estimated automatically from the data using kernel smoothing density estimation. In this work we improve the performance of the pbM-estimator by changing its cost function. Replacing the cost function of the pbM-estimator with the changed one yields the ***modified pbM-estimator****. The cost function of the modified pbM-estimator is more stable relative to the scale parameter and is also a better classifier. Thus we get a more robust and effective technique. A new general method to estimate the runtime of robust regression algorithms is proposed. Using it we show, that the modified pbM-estimator runs 2 - 3 times faster than the pbM-estimator. Experimental results of fundamental matrix estimation are presented demonstrating the correctness of the proposed analysis method and the advantages of the modified pbM-estimator.*

1 Introduction

Robust regression is the generic name for techniques which estimate a parametric regression model in the presence of outliers. In some techniques we maximize the number of inliers(RANSAC) which can be found within the provided distance(scale) from the recovered model, in the others(MLESAC, etc.) we maximize a likelihood function. The robust regression techniques most popular in computer vision RANSAC [4], MSAC [12] and MLESAC [11] use parameters provided by the user to solve the problem. The scale parameter is used in the cost function and the proportion of inliers is used to define the stopping criterion. Sometimes the latter is also used in the definition of the cost function.

In real applications often there is no reliable source that can provide us with the above mentioned parameters. Moreover, inaccurate parameters may yield erroneous results.

In all the techniques mentioned above probabilistic sampling is used to find a maximum of the optimization criterion of the robust regression. Each time we sample an *elemental subset* - subset that contain the smallest number of data points which uniquely define a *model parameter candidate* [1]. The quality of the candidate is then evaluated using all the data points and the final estimate is found by taking the model which obtained the maximal score over a number of such candidates [9]. In real applications however, the outliers dominate the data and this number becomes unfeasibly large. A possible solution is guided sampling, in which additional information about the probability of a data point being an inlier is integrated into the sampling process [10]. There are additional techniques that speed up the computations [8], [3]. The idea of these techniques is simple. There are two tests for a *model parameter candidate*. The computationally cheap one and the computationally expensive one. If the candidate passes the first test, then we run the expensive test, evaluating the final score of the candidate. Actually, on only a small part of candidates will this final evaluation be performed. There are different kinds of test pairs. In [3] the authors evaluate an elemental subset. If its value is better than any value seen so far, then the set of inliers is estimated and an additional inner RANSAC is used for the data that contains points detected as inliers. In [8] the first test is to check the model's consistency with a small number (1-3) of randomly sampled data points. If the consistency holds, then we evaluate the model using all the data.

The projection based M-estimator technique was developed recently by Chen and Meer [1]. This technique does not require the user to provide any scale parameter. For every sampled elemental subset it is estimated, using the found *model parameter candidate* and all the data points.

In this work we show, that the cost function used in [1] is

*This work was supported partly by grant 01-99-08430 of the Israeli Space Agency through the Ministry of Science Culture and Sports of Israel.

unstable as a function of the scale parameter. In other words small change of the scale parameter cause big change of the cost function. We overcome this instability by modifying the cost function. This modification also makes the cost function a better classifier.

We also propose a general way to evaluate the expected runtime of robust regression techniques. The evaluation involves kernel smoothing and maximal order statistic evaluation of non-parametric distributions. With the proposed analysis we estimate the *number* of correct elemental subsets to be sampled to ensure that the algorithm produces the correct result. This *number* depends on the quality of the cost function. From the analysis performed for the pbM-estimator and for the modified pbM-estimator a three-fold speedup was predicted as a result of using the modified pbM-estimator instead of the pbM-estimator. This result was verified empirically.

It is shown in [1], that the pbM-estimator outperforms the other RANSAC family techniques. Therefore, in this work we do not compare the modified pbM-estimator with other methods from the RANSAC family, we only demonstrate its advantages over the pbM-estimator.

2 The pbM-estimator

We start with a short review of the pbM-estimator presented in [1].

Like in most robust regression techniques, the linear errors-in-variables (EIV) model is used. Let y_{i0} be the true values of the given p-dimensional data points y_i. Then

$$\begin{aligned} y_{i0}^T\theta - \alpha = 0 \qquad & i = 1, \ldots, n. \\ y_i = y_{i0} + \delta y_i \qquad & \delta y_i \sim GI(0, \sigma^2 I_p). \end{aligned} \tag{1}$$

The normalization constraint $\|\theta\| = 1$ eliminates the ambiguity of (1) up to some multiplicative constant.

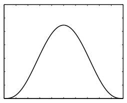
We have two parameters to estimate. A unit vector $\theta \in \mathbb{R}^p$ and a constant $\alpha \in \mathbb{R}$. We call them the *direction* and the *shift*. We also call $[\theta, \alpha]$ *a model*. Note, that this *model* defines a hyperplane, that contains y_{i0}. In some applications the true values y_{i0} are also desired. In that case $\hat{y_{i0}} = y_i - (y_i^T\theta - \alpha)\theta$ is a good way to estimate y_{i0}. In other words, each point is replaced with the closest point, that lies on the found hyperplane.

Note, that sometimes the values of the data points y_i are functions of measurements (fundamental matrix estimation [5], ellipse fitting [6], etc.). In these cases the linear EIV model is only an estimate of the correct model.

All the major robust regression techniques used in computer vision can be expressed as M-estimators [7],[2], and then [1] reformulated in the following way:

$$[\hat{\alpha}, \hat{\theta}] = \arg\max_{\alpha,\theta} \frac{1}{n} \sum_{i=1}^{n} k(\frac{y_i^T\theta - \alpha}{s}), \tag{2}$$

where k is the *M-kernel function*.



$$\text{k(u)=} \begin{cases} c(1-u^2)^3 & \text{, if } |u| \le 1 \\ 0 & \text{, if } |u| \ge 1 \end{cases}$$

is a satisfactory choice for most applications. The normalization constant c assures that $\int_{-\infty}^{\infty} k(u)du = 1$. Note, that as $k(u) = 0$ for $|u| > 1$, points further than s from the hyperplane $[\theta, \alpha]$ do not affect its value.

Remark 1

$$\hat{f}_\theta(\alpha) = \frac{1}{nh_\theta} \sum_{i=1}^{n} k(\frac{y_i^T\theta - \alpha}{h_\theta}) \tag{3}$$

is a way to estimate a density function of $1D$ *sample* $y_1^T\theta, \ldots, y_n^T\theta$ *at point* α*, known as* kernel smoothing.

The similarity of (2) and (3) motivates the replacement of the scale parameter s with the bandwidth h_θ. Thus the authors of [1] reformulate the process of finding an optimal *projection direction* $\hat{\theta}$ to:

$$\hat{\theta} = \arg\max_{\theta} \left\{ h_\theta \max_{\alpha} \frac{1}{nh_\theta} \sum_{i=1}^{n} k(\frac{y_i^T\theta - \alpha}{h_\theta}) \right\}. \tag{4}$$

This optimization problem can be divided into the *inner optimization problem*:

$$\alpha_\theta = \max_{\alpha} \frac{1}{nh_\theta} \sum_{i=1}^{n} k(\frac{y_i^T\theta - \alpha}{h_\theta}), \tag{5}$$

and the *outer optimization problem*:

$$\hat{\theta} = \arg\max_{\theta} \left\{ h_\theta \frac{1}{nh_\theta} \sum_{i=1}^{n} k(\frac{y_i^T\theta - \alpha_\theta}{h_\theta}) \right\}. \tag{6}$$

Using Remark 1, (5) and (6) can be reformulated as

$$\alpha_\theta = \arg\max_{\alpha} \hat{f}_\theta(\alpha) \quad \text{and} \quad \hat{\theta} = \arg\max_{\theta} h_\theta \hat{f}_\theta(\alpha_\theta),$$

respectively.

In the following subsections we describe the technique, used for the estimation of h_θ and the optimal *model*. h_θ will be defined as a function of θ and the data points. Finally, we describe a method to separate the inliers from the outliers using the obtained model parameters.

2.1 Inner optimization

Using Remark 1 we can conclude that if θ is given, then the inner optimization looks for the densest point of the $1D$ sample $y_1^T\theta, \ldots, y_n^T\theta$. For a more precise estimation of the density Chen and Meer derived the following formula for h_θ from the results presented in [13, Sec. 3.2.2]:

$$h_\theta = n^{-1/5} \text{med}_j |y_j^T\theta - \text{med}_i y_i^T\theta|. \tag{7}$$

To solve the inner optimization problem we start from the *coarse* sampling step. Sorting $y_i^T\theta$ we get sequence on n samples $(y^T\theta)_{(i)}$. Evaluating $\hat{f}_\theta(x)$ for ten data driven uniform samples (i.e. for $x = (y^T\theta)_{(\lfloor n*\frac{1}{11}\rfloor)},, \ldots, (y^T\theta)_{(\lfloor n*\frac{10}{11}\rfloor)}$). we let $y_{i_0}^T\theta$ to be the sample, that gave the best value of $\hat{f}_\theta(x)$. Now we calculate $\hat{f}_\theta(x)$ for ten equidistance values from the interval $[y_{i_0}^T\theta - h_\theta, y_{i_0}^T\theta + h_\theta]$. The number that gave the largest value of $\hat{f}_\theta(x)$ is defined as α_θ. Thus, to solve an inner optimization problem, we have to evaluate $\hat{f}_\theta(x)$ at 20 different points.

2.2 Outer optimization

To solve the outer optimization problem iterations are involved. At each iteration of the outer optimization p points $y_{i_1}, \ldots, y_{i_p} \in \mathbb{R}^p$ are randomly sampled, creating *an elemental subset*. It is used to find θ, the normal to the hyperplane, that contains $y_{i_1}, \ldots, y_{i_p}$. We call this part of the iteration *the RANSAC step*. Having the direction θ we find its shift α_θ together with its value by solving the inner optimization problem, as described in 2.1. Afterwards, the multidimensional *simplex direct search* is used to get some local improvement of the direction. Instead of running the direct search in dimension p on a unit sphere, the authors of [1] gave expressions for the elements of θ as polar angles. These expressions decrease the dimension of the search space from p to $p-1$ and remove the normalization constraint $\|\theta\| = 1$. Running enough iterations of the outer optimization we hope to fall close enough to the optimal $\hat{\theta}$, and get there using the direct search.

2.3 Separating inliers from outliers

After the $\hat{\theta}$ is found, the inner optimization problem is solved one more time to find $\hat{\alpha}$. The *basin of attraction* of $\hat{\alpha}$ is delimited by the nearest left and right significant local minima of $h_{\hat{\theta}}\hat{f}_{\hat{\theta}}(\alpha)$. Points whose values $y_i^T\theta$ lie between the two minima are considered inliers.

3 The modified pbM-estimator

In this section we demonstrate the instability of the pbM cost function:

$$f_1(\theta) = h_\theta \max_\alpha \left\{ \frac{1}{nh_\theta} \sum_{i=1}^{n} k(\frac{y_i^T\theta - \alpha}{h_\theta}) \right\}, \qquad (8)$$

as a function of h_θ. We will show, that a slightly different cost function - *the modified pbM cost function* is more stable and yields a better classifier separating correct and incorrect models $[\theta, \alpha]$, which results in a faster and more accurate algorithm.

3.1 Decreasing the instability in the $1D$ case

We start by demonstrating the instability of the function $\max_\alpha\{\frac{1}{n}\sum_{i=1}^{n} k(\frac{|x_i-\alpha|}{h})\}$, which is a $1D$ version of (8), as a function of bandwidth. We sample N points from the standard normal distribution and calculate the function

$$\max_\alpha\{h\frac{1}{nh}\sum_{i=1}^{n} k(\frac{|x_i-\alpha|}{h})\},$$

$$h \in [0.5h_\theta = 0.5\frac{\text{med}_j|x_j - \text{med}_i x_i|}{n^{1/5}}, 10h_\theta]$$

(dashed line on Figure 1). As can be seen, this function depends almost linearly on the bandwidth. Its final value (for $h = 10h_\theta$) is about 12 times the initial one (for $h = h_\theta/2$). The slope of this linear function, according to Remark 1, can be estimated analytically:

$$\max_\alpha\{\frac{1}{n}\sum_{i=1}^{n} k(\frac{|x_i-\alpha|}{h})\}/h = \max_\alpha\{\frac{1}{nh}\sum_{i=1}^{n} k(\frac{|x_i-\alpha|}{h})\}$$

$$\approx \max_x \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}} = \frac{1}{\sqrt{2\pi}}e^{-\frac{0^2}{2}} = 0.39.$$

This formula can be rephrased in words. According to the Remark 1, (8) is a maximal density value multiplied by the bandwidth used for the kernel smoothing. If we change the bandwidth, then the “density part” of (8) changes only slightly. Therefore, (8) depends almost linearly on the bandwidth. To “eliminate” this linearity we use a slightly different function

$$\max_\alpha\{\frac{1}{nh}\sum_{i=1}^{n} k(\frac{|x_i-\alpha|}{h})\}.$$

In other words we dropped the bandwidth and are left with the “density part”. As can be seen from the same figure this function is more stable (solid line), at least in the interval $[0.5h_\theta, 10h_\theta]$. It decreases by only 40%.

One can say, that if the elemental subset is already sampled then θ is fixed, h_θ is defined and there is no reason to argue about the stability of the cost function as a function of bandwidth. From the other hand h_θ given by 7 is only an estimation of some optimal $\hat{h}_\theta$. Therefore, the less cost function changes (near the $\hat{h}_\theta$), the more accurate evaluation we have.

Due to the increased stability, we decided to use its multidimensional version:

$$f(\theta) = \max_\alpha \left\{ \frac{1}{nh_\theta} \sum_{i=1}^{n} k(\frac{y_i^T\theta - \alpha}{h_\theta}) \right\}, \qquad (9)$$

to build the optimization problem for ***the modified pbM-estimator***:

$$[\hat{\alpha}, \hat{\theta}] = \arg\max_{\alpha,\theta} \left\{ \frac{1}{nh_\theta} \sum_{i=1}^{n} k(\frac{y_i^T\theta - \alpha}{h_\theta}) \right\}, \qquad (10)$$

that can be divided, like (2), into an inner and an outer optimization problems,

$$\alpha_\theta = \arg\max_\alpha \hat{f}_\theta(\alpha) \quad \text{and} \quad \hat{\theta} = \arg\max_\theta \hat{f}_\theta(\alpha_\theta),$$

respectively. The technique proposed in [1] and described in Section 2, can be used to solve (10) and find inliers. Using Remark 1 we conclude, that (10) solves the maximum likelihood problem of the model.

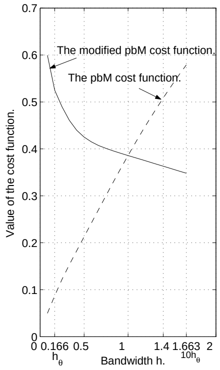


Figure 1: 1000 samples of $N(0, 1)$. Comparing the stability: The dashed line represents $\max_\alpha \frac{1}{n}\sum_{i=1}^{n} k(\frac{|x_i-\alpha|}{h})$, and the solid line represents $\max_\alpha \frac{1}{nh}\sum_{i=1}^{n} k(\frac{|x_i-\alpha|}{h})$.

In the rest of this paper we demonstrate the advantages of *the modified pbM-estimator* (the modified pbM) over *the pbM-estimator*.

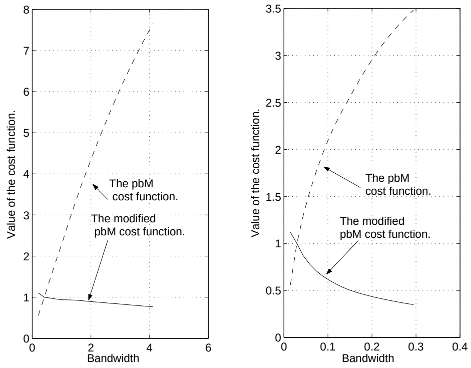
3.2 Higher dimensions

To demonstrate the stability of the modified pbM cost function in higher dimensions, we synthesized $3D$ data and fixed an arbitrary direction θ_{fix}. In such a way we got a function of the bandwidth

$$f(h) = \max_\alpha \{ \frac{1}{nh} \sum_{i=1}^{n} k(\frac{y_i^T \theta_{fix} - \alpha}{h}) \}.$$

We calculated the values of this function for $h \in [0.5h_{\theta_{fix}}, 10h_{\theta_{fix}}]$, solving the inner optimization problem for each bandwidth h. To highlight the stability of (9) we divided the received values by $f(h_{\theta_{fix}})$, setting the value 1 for bandwidth $h_{\theta_{fix}}$. For results see the solid line in Figure 2.a. For the same direction θ_{fix}, we repeat the above calculations for the pbM cost function (dashed line on Figure 2.a). Figure 2.b demonstrates the results for real $8D$ data used to recover the fundamental matrix. It can easily be seen, that (9) is more stable as a function of the bandwidth.

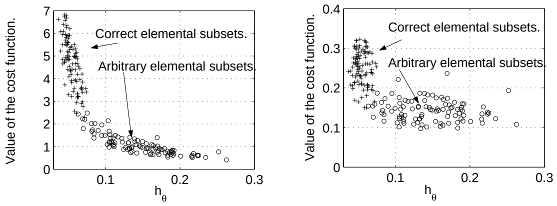
From the same figure it can be concluded, that in an "acceptable" range of bandwidths the pbM cost function (8) is an increasing function (dashed lines) and the modified pbM cost function (9) is a decreasing one (solid lines). This holds for both synthetic and real data.



a) Synthetic data. b) Real data.

Figure 2: Comparing the stability of (9) (solid lines) and (8) (dashed lines), as a functions of bandwidth for synthesized $3D$ and real $8D$ data.

From the real $8D$ data, we sampled 100 *correct elemental subsets,* i.e. elemental subsets, that contain only inliers, and 100 arbitrary elemental subsets, i.e. elemental subsets, that contain at least one outlier. Each one of these subsets defines a direction θ, a bandwidth h_θ, the value of (9) and the value of (8). We calculated these parameters for all 200 elemental subsets and present them in Figure 3. Points represent the results for arbitrary elemental subsets, and crosses represent the results for correct elemental subsets. It can be seen here, that for good *directions* h_θ is expected to be small. It is caused by hight clustering of $(\theta^T y_i)$ around α for subset of inliers, when θ is good.



a) The modified pbM. b) The pbM.

Figure 3: Comparing the bandwidths magnitudes and "structureness" of (8) and (9).

For the increasing cost function (as a function of bandwidth), values of correct elemental subsets have a tendency to be smaller, as a result of smaller bandwidths. On the other hand, values of arbitrary elemental subsets have a tendency to be larger, as a result of larger bandwidths. These tendencies can cause a correct elemental subset to have a lower value, than some arbitrary elemental subset. This phenomenon is undesirable. Therefore, being a decreasing function of bandwidth is good for the cost function.

There are two very important characteristics of the cost function. The cost function should separate properly between *correct* and *arbitrary elemental subsets*. In other words, the probability of some correct elemental subset to have a lower value of the cost function, than some arbitrary elemental subset should be as small as possible. In addition, we would like our algorithm to stop as soon as possible. I.e. we would like to decrease the number of sampled elemental subsets without affecting the algorithm's performance. In the next section we will demonstrate, that both characteristics can be improved by replacing the pbM-estimator with the modified pbM.

4 Separation property and a runtime of a robust regression

In this section we first describe a new general *technique* that allows to quantify the separation property of a robust regression cost function. Afterwards, we expand this *technique* to estimate the runtime of a robust regression algorithm. In the end of this section we use this *technique* to demonstrate that the modified pbM is a better classifier than the pbM-estimator. We also demonstrate, that using the modified pbM we can stop the algorithm earlier without effecting its performance. We would like to stress, that the *technique* is general and can be applied to any algorithm from the RANSAC family.

We are given a robust regression algorithm whose performance we would like to analyze. We start by sampling N correct elemental subsets. Each of them gives a value of the cost function. We assume, that these values are a sample of some unknown distribution. The density of this distribution $d_{corr}(x)$ can be estimated, using the *kernel smoothing* technique with the bandwidth defined by (7) (for details see [13]). Using the *kernel smoothing* technique we define $d_{corr}(x)$ by its values for the set $\{x_0^c, \ldots, x_{I_c}^c\}$, where $x_i^c - x_{i-1}^c = \delta x^c$, $x_0^c = 0$, $d_{corr}(x)$ negligible for $x > x_{I_c}^c$ and δx^c is small. Typical results are presented in Figure 4 (solid line). Area under the line should equals 1. The solid horizontal straight line represents the confidence interval of 90%. The same calculations are made for N arbitrary elemental subsets ($d_{arb}(x)$ dashed line). Using numerical integration we can find the probability of a value of an arbitrary elemental subset to be higher than a value of a correct elemental subset. The lower the probability, the better classifier we have. The confidence intervals can be used to illustrate the separation. The smaller the congruence of the confidence intervals, the better the classifier.

Now we would like to estimate the average runtime of a robust regression algorithm. Until now methods to define the number of elemental subsets to be sampled were to multiply the inverse of the probability to sample a correct elemental subset by some constant or set the number of elemental subsets to be sampled hight enough to ensure hight probability to sample at least one *correct elemental subset* [4]. This method does not take into account the properties of the cost function. For example if we have a bad classifier, then all sampled correct elemental subsets (one or two) may have lower values than the best value obtained for all sampled arbitrary elemental subsets. In this undesirable case we say that the maximal order statistic is a value of some arbitrary elemental subset and also get the algorithm's failure.

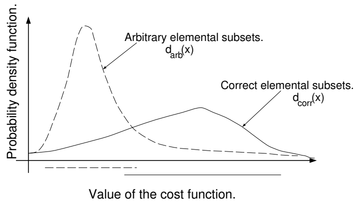


Figure 4: Density functions of the values of the cost functions for correct (solid lines) and arbitrary (dashed lines) elemental subsets. Horizontal lines represent confidence intervals of 90%.

To estimate the number of elemental subsets to be sampled, we use the found density functions ($d_{arb}(x)$ and $d_{corr}(x)$), thus taking into account the separation property of the cost function.

First, we find the expected value of the maximal order statistic of the values of correct elemental subsets as a function of sample size. To do it we use $d_{corr}(x)$. We find the distribution function on the same grid:

$$F_{corr}(x_0^c) = 0,$$

$$F_{corr}(x_i^c) = F_{corr}(x_{i-1}^c) + \int_{x_{i-1}^c}^{x_i^c} d_{corr}(t)dt \approx$$

$$F_{corr}(x_{i-1}^c) + d_{corr}(x_i^c)\delta x^c.$$

Then we find the distribution function of the maximal order statistic for a sample of size m:

$$F_{max\_m}^{corr}(x_i^c) = {F_{corr}}^m(x_i^c)$$

and its density function

$$d_{max\_m}^{corr}(x_0^c) = 0,$$

$$d_{max\_m}^{corr}(x_i^c) \approx \frac{F_{max\_m}^{corr}(x_i^c) - F_{max\_m}^{corr}(x_{i-1}^c)}{\delta x^c}.$$

Now the expected value of the maximal order statistic of values of m correct elemental subsets can be evaluated by numerical integration

$$E_{max\_m}^{corr} = \int x d_{max\_m}^{corr}(x)dx \approx \sum_{i=1}^{I_c} x_i^c d_{max\_m}^{corr}(x_i^c)\delta x^c.$$

Figure 5 (solid line) presents typical results of the calculations. According to the figure, if we sample 6 correct elemental subsets, calculate their *modified pbM* values, and take the best one, then we get on average the value of 60.

Let p be the proportion of inliers, and let n be the size of the elemental subset. Then, the probability to sample a correct elemental subset is $r = p^n$.

In real applications the proportion of sampled correct elemental subsets is very small. More precisely, to get one correct elemental subset, we have to sample, on average, $r^{-1} = p^{-n}$ elemental subsets. To incorporate this fact into our method we use the *sampling rate* for arbitrary elemental subsets. In other words, $v_{corr}(m)$ in Figure 5 gives the expected value of the maximal order statistic of values of m correct elemental subsets, while $v_{arb}(m)$ (dashed line on Figure 5) gives the expected value of the maximal order statistic of values of $[mr^{-1}] = [mp^{-n}]$ arbitrary elemental subsets. We say, that the *sampling rate* equals r. In other words, the calculations for arbitrary elemental subsets are the same as the calculations for correct elemental subsets, but we use different scales to represent the results.

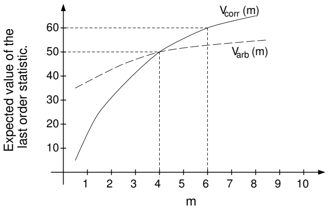


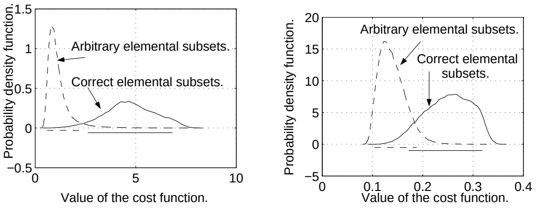
Figure 5: Expected values of the maximal order statistics for the correct (solid line) and the arbitrary (dashed line) elemental subsets.

From Figure 5 it can be seen, that the solid line *approaches* the dashed one. This *approaching* is very important for us. When the solid line crosses the dashed one, we can stop the algorithm. Indeed, if the solid line is above the dashed one, then the best received value is the cost of some correct elemental subset. The described *approaching* depends on the percentage of outliers and on the size of the elemental subset, parameters that affect the *sampling rate* r^{-1} and therefore, the expected runtime.

We can give the following interpretation to Figure 5: the algorithm can be stopped after we sampled, on averaged, 4 correct elemental subsets. I.e. we should sample, on average, $4r^{-1}$ elemental subsets.

We applied the described technique to compare between the modified pbM and the pbM-estimator.

We used real $8D$ data for fundamental matrix estimation, i.e. $n = 8$ and $r = p^8$. To build the density functions we used 100000 correct and 100000 arbitrary elemental subsets. Figure 6 shows, that the modified pbM is a better classifier.



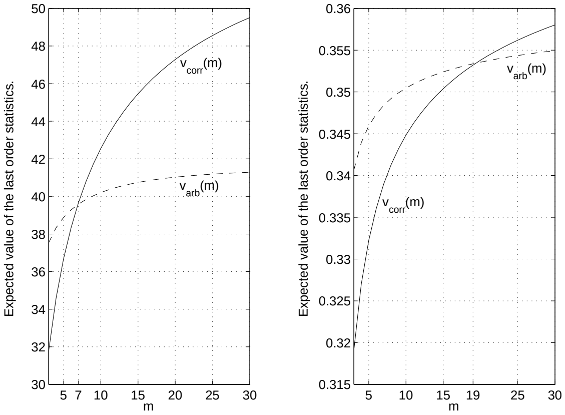
a) The modified pbM.

b) The pbM.

Figure 6: Comparing the separation property of the modified pbM and the pbM-estimator.

For the given data we had $p = 0.35$. Therefore, the probability to sample a correct elemental subset is $r = p^n = 0.35^8 \approx \frac{1}{4500}$. Therefore, we used a sampling rate of $\frac{1}{4500}$.

As Figure 7 shows, if we use the modified pbM, then the algorithm can be stopped after we sampled, on average, 7 correct elemental subsets. I.e. we should sample, on average $7r^{-1} = 7 * 4500 = 31500$ elemental subsets. When however, we use the pbM-estimator, then we should sample, on average, $19*4500 = 85500$ elemental subsets. Thus we get almost three time speed up for the modified pbM. As was mentioned before, the expected runtime depends on the *sampling rate* r^{-1}, affected by p. To investigate this effect we evaluated the expected runtime using different values of p, and changing the *sampling rate* accordingly. Figure 8 shows the results of the runtime estimation on a log-scale. As can be seen, the number of iterations required by the pbM-estimator is about 3 times the number required by the modified pbM.



a) The modified pbM.

b) The pbM.

Figure 7: Comparing the maximal order statistics for the pbM and the modified pbM. Dashed line - arbitrary elemental subsets. Solid line - correct elemental subsets.

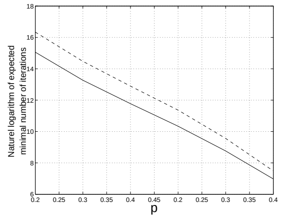


Figure 8: Comparing expected runtimes for different proportions of inliers p. Solid line - the modified pbM. Dashed line - the pbM-estimator.

5 Experimental results

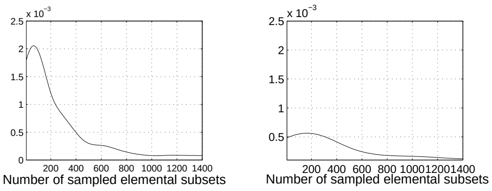
We ran two different experiments. In the first one we empirically compare the runtime of the modified pbM and the pbM-estimator verifying the conclusions of Section 4. In the second one we compared the performance of these techniques on several image pairs.

5.1 Runtime comparison

In Section 4 we showed, that the pbM-estimator requires, on average, to sample 2-3 times more elemental subsets than the modified pbM. Here we check this analytical result by running the following experiment.

We ran the modified pbM and the pbM until convergence 200 times each. If there was no convergence after we sampled 4000 elemental subsets, we stopped the algorithm. By convergence we mean, that at least 90% of the output are inliers and that at least 75% of the inliers were detected. The fundamental matrix estimation task for the *table* pair of images (Figure 10) was used. Having the results we can find empirically the median and the average of the number of elemental subsets to be sampled. We also give the number of failed trials. The closer this number is to 0 the better algorithm is. The density function of the elemental subsets sampled was found using kernel smoothing.

	mean	med	number of failed trials
pbM	1856	1384	57
modified pbM	762	188.5	15



a) The modified pbM.

b) The pbM.

Figure 9: Density function for required number of elemental subsets to be sampled.

This result justifies our conclusion, that the pbM-estimator requires 2-3 times more elemental subsets to be sampled, than the modified pbM.

5.2 Fundamental matrix estimation

As explained in [1], the role of a robust regression algorithm for the fundamental matrix estimation task should be limited to discrimination of inliers from outliers. The method does not taken into account the rank two constraint of F. In the beginning of the algorithm data points were normalized (see [5]). Afterwards, the epipolar geometry is found using a technique, that does not assume the linear EIV model. For example [6].

Unlike in the algorithm described in Section 3 and used by the authors of [1], we used the direct search only once at the end for the best received direction.

The following figures provide us with the percentage of inliers in the input data and with the ground truth. To demonstrate the ground truth 30 epipolar lines defined by 30 pairs of points from the given data were drawn.

To compare between the performances of the modified pbM and the pbM-estimator, we sampled the same number of *elemental subsets*.

The following tables give us a quantitative and qualitative description of the results. The quantitative description contains two numbers. The first one is the number of pairs of points presumed to be inliers. We would like all of them to be true inliers. Unfortunately, there are outliers in the output of the algorithm, i.e. false alarms. The second number demonstrates the number of true inliers in the set of discriminated points. The closer the ratio between these numbers is to one, the better the performance. Actually, the ratio between the second number to the first one gives the percentage of inliers in the output. To give a qualitative description of the output, the epipolar geometries was found using the technique of [6].

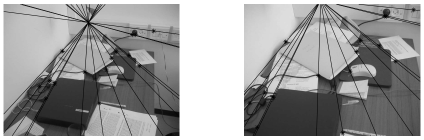
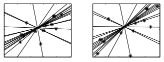
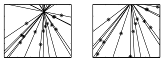


Figure 10: *Table* pair. Ground truth. 45% of inliers.

Table pair	selected points /true inliers	2000 elemental subsets
pbM	60 / 52	
modified pbM	58 / 55	

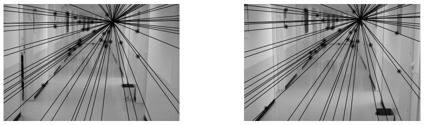
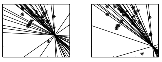
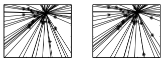


Figure 11: *Doors* pair. Ground truth. 58% of inliers.

Doors pair	selected points /true inliers	300 elemental subsets
pbM	81 / 60	
modified pbM	121 / 110	

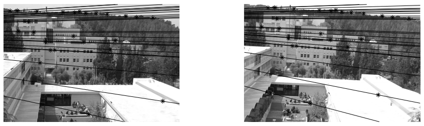
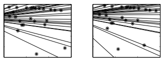
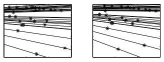


Figure 12: *Faculty*. Ground truth. 38% of inliers.

Faculty pair	selected points /true inliers	7500 elemental subsets
pbM	137 / 126	
modified pbM	113 / 95	

As can be seen, the modified pbM gives higher percentage of true inliers in the output. The epipolar geometry recovered using the inliers detected by the modified pbM is more similar to the ground truth, which is more important since the goal of the fundamental matrix estimation task is to recover the epipolar geometry. From these results we concluded, that the modified pbM outperforms its opponent.

The pbM-estimator is a worse classifier than the modified pbM, but it still separates elemental subsets quite accurately (see Figure 6.b). If we would run it enough time we would get an acceptable epipolar geometry. Actually, the number of elemental subsets to be sampled was chosen to be higher than what was required for the modified pbM to converge and lower than what was required for the pbM-estimator to converge.

The modified pbM outperformed its opponent not only for a single pair of images, but for many pairs. Results for only three of them are presented here. This fact together with the conclusions of Section 4 convinced us that our technique should be used for the robust regression task.

6 Discussion

In this work we modified the cost function proposed in [1] in order to get a more robust and effective *robust regression technique*.

In previous works from the RANSAC family, the major stopping criterion was to sample elemental subsets, until we get, on average, a *number* of correct elemental subsets. We showed, how this *number* is estimated, by analyzing the separation property of the cost function on typical image pairs. We also described, how we can compare two different cost functions for robust regression techniques.

The experiments justified empirically, that the modified pbM-estimator yields a 2-3 time speedup relative to the pbM-estimator and thus should be used in practice.

Numerical scheme for derivative and integral evaluation used in section 4 can be replaced with some other one.

The performance of our technique for the fundamental matrix estimation is improved if we replace the the algebraic distances with the geometric ones and reduce the elemental subset's size from 8 to 7 (by employing the degeneracy constraint $detF = 0$).

References

[1] H. Chen and P. Meer. Robust regression with projection based m-estimators. In *Proceedings of Ninth IEEE International Conference on Computer Vision*, volume 2, pages 878–885, October 2003.

[2] H. Chen, P. Meer, and D. E. Tyler. Robust regression for data with multiple structures. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, volume 1, page 10691075.

[3] O. Chum, J. Matas, and J.V. Kittler. Locally optimized RANSAC. In *DAGM03*, pages 236–243, 2003.

[4] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24(6):381–395, June 1981.

[5] R.I. Hartley. In defense of the eight-point algorithm. *IEEE Trans. Patt. Anal. Mach. Intell.*, 19(6):580–593, June 1997.

[6] Y. Leedan and P. Meer. Heteroscedastic regression in computer vision: Problems with bilinear constraint. *Int. J. of Comp. Vision*, 37(2):127–150, June 2000.

[7] G. Li. Robust regression. In David C. Hoaglin, Frederick Mosteller, and John W. Tukey, editors, *Exploring Data Tables, Trends, and Shapes*, pages 461 – 513. John Wiley and Sons, 1985.

[8] J. Matas and O. Chum. Randomized RANSAC with Td,d test. *Image and Vision Computing*, 22(10):837–842, September 2004.

[9] P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection*. Wiley, 1987.

[10] B. Tordoff and D.W. Murray. Guided sampling and consensus for motion estimation. In *In 7th European Conference on Computer Vision*, volume 1, pages 82–96, Copenhagen, Denmark, May 2002.

[11] P.H.S. Torr and D.W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *Int. J. of Comp. Vision*, 24(3):271–300, October 1997.

[12] P.H.S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Comp. Vis. Im. Understanding*, 78(1):138–156, April 2000.

[13] M.P. Wand and M.C. Jones. *Kernel Smoothing*. Chapman and Hall, 1995.