Beyond RANSAC: User Independent Robust Regression

Raghav Subbarao and Peter Meer
Department of Electrical and Computer Engineering
Rutgers University, Piscataway
NJ 08854, USA
rsubbara,meer@caip.rutgers.edu

Abstract

RANSAC is the most widely used robust regression algorithm in computer vision. However, RANSAC has a few drawbacks which make it difficult to use in a lot of applications. Some of these problems have been addressed through improved sampling algorithms or better cost functions, but an important problem still remains. The algorithms are not user independent, and require some knowledge of the scale of the inlier noise. The projection based M-estimator (pbM) offers a solution to this by reframing the regression problem in a projection pursuit framework. In this paper we derive the pbM algorithm for heteroscedastic data. Our algorithm is applied to various real problems and its performance is compared with RANSAC and MSAC. It is shown that pbM gives better results than RANSAC and MSAC in spite of being user independent.

1. Introduction

Regression tries to estimate a parameter vector given a data set and a functional relation between the data and the parameters. In many cases, it is possible that not all of the data satisfies the given relation. These points, referred to as *outliers*, interfere with the regression and lead to incorrect results if they are not handled properly. Robust regression algorithms are methods which perform regression on data sets which contain outliers without a loss of accuracy in the parameter vector estimates. Due to the complex nature of computer vision algorithms, outliers are a frequent occurrence in many applications. Any system which aims to solve even simple visual tasks, with minimal user interference, must address the problem of outliers.

The most widely used robust algorithm in computer vision is *Random Sampling and Consensus* (RANSAC) [6]. Due to its ease of implementation and its applicability RANSAC is widely used. However, RANSAC also suffers from a number of drawbacks which make it inappropriate for some real world applications. A considerable amount of research has been devoted to addressing these shortcomings e.g. [3, 11, 21, 22, 23]. Effort has also been made to understand the theoretical properties of RANSAC and its equivalence to other robust techniques [2]. It was shown that data with multiple structures (contains several instances of a model) is not equivalent to having unstructured outliers.

In this paper, we discuss an important obstruction to applying RANSAC in practice which has largely been ignored till recently, namely the sensitivity to scale. We propose a robust estimator which manages to overcome this problem, the projection based M-estimator (pbM), and does not require any user intervention. Therefore, the algorithm does not require the user to specify a scale for the noise. The algorithm proposed here uses random sampling. A number of methods have been proposed for improving on random sampling using other available information. These methods can be used in the hypothesis generation part of pbM while keeping the rest of the algorithm the same.

In Section 2 we briefly review some of the work that has been done to improve the basic RANSAC algorithm. In Section 3 the heteroscedastic errors-in-variables problem is introduced and its robust M-estimator formulation is reframed in terms of kernel density estimation. The pbM algorithm is discussed in Section 4. The results of the pbM algorithm are presented in Section 5 and its performance is compared to RANSAC and MSAC.

2. Previous Work

Random sampling and consensus (RANSAC) is the most widely used robust estimator in computer vision today. RANSAC was proposed in [6] and has since then been applied to a wide range of problems including fundamental matrix estimation [21], trifocal tensor estimation [22], camera pose estimation [11] and structure from motion [11]. Other applications of RANSAC can be found in [8].

RANSAC has been used extensively and has proven to give better performance than various other robust estimators. Some of these estimators such as LMedS [12] were developed in the statistics community, but were found to

not give good results for complex vision applications [15].

Improvements have been proposed to the basic RANSAC algorithm of [6]. In [22], it was pointed out that RANSAC treats all inliers uniformly. In other words, in the cost function that RANSAC tries to optimize, all the inliers score nothing while all outliers score a constant penalty. Better performance was obtained by using a cost function where the inlier scores a penalty depending on how well it satisfies the required functional relation. All outliers scored a constant penalty. This new method is known as MSAC (M-estimator sample consensus) and was found to give better performance than basic RANSAC. Assuming the inliers are corrupted by Gaussian noise and the scale of the noise σ is known, MSAC uses a scale of 1.96σ so that inliers are rejected only 5% of the time. A slightly different algorithm was proposed in [23], where the cost function was modified to yield the maximum likelihood (ML) estimate under the assumption that outliers are uniformly distributed. This algorithm was called MLESAC (maximum likelihood sampling and consensus). The parameters of the noise are required to be given by the user.

The drawbacks of the random sampling component of RANSAC have also been well documented. RANSAC samples points randomly and this can be improved based on other information. This idea was first proposed in [19], where the match scores from the point matching stage are used in the sampling. By replacing the random sampling in RANSAC by *guided sampling* the probability of getting a *good* elemental subset was drastically improved. In [3], RANSAC is applied to fundamental matrix estimation by giving each point match a score which indicates the systems belief whether the point is an inlier. This score is used in the sampling and after each estimate the score is recursively modified. Alternatively, the sampling step in RANSAC can be improved by combining it with importance sampling [20]. The original RANSAC algorithm is not directly applicable to real-time problems and needs to be modified to adapt to time constraints. In [11], a termination condition based on the execution time of the algorithm is used to limit the sampling so that RANSAC can be used for live structure from motion.

In spite of the large amount of research devoted to improving RANSAC, a difficulty to applying it to practical problems has often been overlooked. In general, RANSAC requires a *user-defined scale* which can be thought of as an assumption on the scale of the inlier noise. The performance of RANSAC is rather sensitive to this scale value.

Extensions of RANSAC such as MSAC [21] and MLESAC [23] also suffer from this problem. All these methods require an estimate of the scale of the noise corrupting the inliers and their performance is sensitive to the accuracy of this value. Usually there is a trade-off between the number of inliers in the model and the scale of the noise. RANSAC, MSAC and MLESAC attempt to maximize the number of inliers while placing an upper limit on the scale.

In many applications we have no knowledge of the true scale of the inlier noise and only recently that there has been interest in developing solutions to the robust regression problem which are not dependent on scale [1, 14, 16, 17, 25]. Singh et al [14] use a maximum likelihood approach under the assumption that the inlier noise is normally distributed. The cost function is then locally optimized using a provably convergent algorithm to obtain the parameter estimates. Wang and Suter [25] propose a two step method to solve the problem. In the first step, they propose a robust estimator for the scale of the inlier noise. In the second step they optimize a cost function which uses the scale found in the first step. However, the robust scale estimate found in the first step is used as an estimate of the inlier noise when deciding the inlier-outlier dichotomy. Consequently, this estimate needs to be re-estimated as better estimates of the true parameters are obtained.

Projection based M-estimators [1, 17] offer a novel solution to the scale selection problem. The dependence on scale is circumvented by exploiting an intrinsic relation between the optimization criteria and the data space. The pbM estimator can handle heteroscedastic data [17] and multiple constraints [18]. The pbM estimator works well even with data sets where the outliers dominate the inliers [1, 17].

3. Projection Based M-Estimator

The original pbM estimator was proposed as a solution to the robust linear errors-in-variables (EIV) model [1]. It was later generalized to the robust heteroscedastic-errors-in-variables (HEIV) model which is more general then the linear EIV problem [17]. The algorithm proposed here solves the robust HEIV problem but modifies the method of [17]. Our method can be generalized to handle multiple constraints although the details of this are not presented here.

The robust heteroscedastic errors-in-variables problem can be stated as follows. Let $\mathbf{y}_{io} \in \mathbb{R}^p,\ i = 1, \ldots, n_1$ represent the true value of the given data points $\mathbf{y}_i$. Given $n > n_1$ data points $\mathbf{y}_i, i = 1, \ldots, n$, we would like to estimate $\hat{\boldsymbol{\theta}} \in \mathbb{R}^p$ and $\hat{\alpha} \in \mathbb{R}$ such that,

$$\begin{aligned} \hat{\mathbf{y}}_i^T \hat{\boldsymbol{\theta}} - \hat{\alpha} = 0,\ i = 1, \ldots, n_1 \\ \mathbf{y}_i = \mathbf{y}_{io} + \delta\mathbf{y}_i \qquad \delta\mathbf{y}_i \sim GI(0, \sigma^2 \mathbf{C}_i) \end{aligned} \tag{1}$$

where, $\hat{\mathbf{y}}_i$ is an estimate of $\mathbf{y}_{io}$. The remaining points $\mathbf{y}_i,\ i = n_1 + 1, \ldots, n$ are outliers and no assumptions are made about their distribution.

The number of inliers, n_1, is unknown. Each $\mathbf{y}_i$ is allowed to be corrupted by noise of a different covariance. This makes the problem *heteroscedastic* as opposed to homoscedastic, where all the covariances are the same. Het-

eroscedasticity usually occurs in vision due to nonlinear mappings between known image data and the vectors involved in the regression. Given the covariance of the image data the covariances of the vectors can be found by error propagation. We assume that the various covariance matrices, $\mathbf{C}_i$, are known up to a common scale σ^2. A global scaling does not affect any further analysis and is ignored from now on. The multiplicative ambiguity in (1) is removed by imposing the constraint $\|\boldsymbol{\theta}\| = 1$.

The robust M-estimator formulation of this problem is,

$$\left[\hat{\alpha}, \hat{\boldsymbol{\theta}}\right] = \arg\min_{\alpha,\boldsymbol{\theta}} \frac{1}{ns} \sum_{i=1}^{n} \rho\left(\frac{\mathbf{y}_i^T\boldsymbol{\theta} - \alpha}{s\sqrt{\boldsymbol{\theta}^T\mathbf{C}_i\boldsymbol{\theta}}}\right) \quad (2)$$

where, s is a scale parameter. The term, $\mathbf{y}_i^T\boldsymbol{\theta} - \alpha$ measures the deviation of data point from the required constraint. Deviations of points with larger covariances should have less weight than points with smaller covariances. This is achieved by the dividing by $\sqrt{\boldsymbol{\theta}^T\mathbf{C}_i\boldsymbol{\theta}}$ which is the standard deviation of the projection, $\mathbf{y}_i^T\boldsymbol{\theta}$.

The *loss function* $\rho(u)$ is non-negative, even, symmetric and non-decreasing with $|u|$. It has a unique minimum of $\rho(0) = 0$ and a maximum of 1 as $u \to \pm\infty$. Therefore, it penalizes points on how much they deviate from the constraint. Greater deviations are penalized more, with the maximum possible penalty being one.

The scale, s, controls how much error the cost function is allowed to tolerate. The s also appears in the denominator of (2) outside the summation. This term is usually neglected and if the scale is held constant it does not affect the maximization. However, in many cases, including pbM, scores evaluated at multiple scales are compared and this term is necessary to account for the different scales [13, 17].

Any redescending M-estimator can be used, but in our implementation, we use the *biweight* loss function,

$$\rho(u) = \begin{cases} 1 - (1-u^2)^3 & if \ \ |u| \le 1 \\ 1 & if \ \ |u| > 1. \end{cases} \quad (3)$$

The M-estimator formulation can be rewritten as

$$\left[\hat{\alpha}, \hat{\boldsymbol{\theta}}\right] = \arg\max_{\alpha,\boldsymbol{\theta}} \frac{1}{ns} \sum_{i=1}^{n} \kappa\left(\frac{\mathbf{y}_i^T\boldsymbol{\theta} - \alpha}{s\sqrt{\boldsymbol{\theta}^T\mathbf{C}_i\boldsymbol{\theta}}}\right) \quad (4)$$

The function $\kappa(u) = 1 - \rho(u)$ in (4) is the *M-kernel function*. The M-kernel function corresponding to the biweight loss function is given by

$$\kappa(u) = \begin{cases} (1-u^2)^3 & if \ \ |u| \le 1 \\ 0 & if \ \ |u| > 1. \end{cases} \quad (5)$$

3.1. Kernel Density Estimation

Let $\boldsymbol{\theta}$ be a given direction. The projection of the data points $\mathbf{y}_i$ along this direction are given by $\mathbf{y}_i^T\boldsymbol{\theta}$ and the covariance of this projection is given by $\boldsymbol{\theta}^T\mathbf{C}_i\boldsymbol{\theta}$. The *weighted kernel density estimate* of these points, based on an even kernel function $K(u)$ and a bandwidth h is

$$\hat{f}_{\boldsymbol{\theta}}(x) = \frac{c_{K,h}}{nhw} \sum_{i=1}^{n} \frac{w_i}{\sqrt{\boldsymbol{\theta}^T\mathbf{C}_i\boldsymbol{\theta}}} K\left(\frac{\mathbf{y}_i^T\boldsymbol{\theta} - x}{h\sqrt{\boldsymbol{\theta}^T\mathbf{C}_i\boldsymbol{\theta}}}\right) \quad (6)$$

where, w_i are nonnegative weights which control the relative importance of each point. The constant $c_{K,h}$ satisfies

$$c_{K,h} \int_{\mathbb{R}} K\left(\frac{x}{h}\right) dx = 1 \quad (7)$$

and a normalization factor of $w = \sum_i w_i$ is necessary to ensure that $\hat{f}$ integrates to 1 over the real line and is a valid density function.

The mode of this distribution is defined as the global maximum of $\hat{f}_{\boldsymbol{\theta}}(x)$. Therefore

$$x_{mode} = \hat{f}_{\boldsymbol{\theta}}(x) \quad (8)$$

and this value can be computed efficiently using the mean shift algorithm [4].

Comparing (4) and (6), it can be seen that the two functions are similar. To make this similarity mathematically precise, we have do the following. The kernel function K is chosen to be the M-kernel function κ, the weights are taken as $w_i = \sqrt{\boldsymbol{\theta}^T\mathbf{C}_i\boldsymbol{\theta}}$ and the bandwidth h is taken as the scale s. With these modifications, the kernel density estimate equation of (6) becomes,

$$\hat{f}_{\boldsymbol{\theta}}(x) = \frac{c_{\kappa,s}}{nsw} \sum_{i=1}^{n} \kappa\left(\frac{\mathbf{y}_i^T\boldsymbol{\theta} - x}{s\sqrt{\boldsymbol{\theta}^T\mathbf{C}_i\boldsymbol{\theta}}}\right) \quad (9)$$

which differs from the M-kernel formulation of (4) only by the constant factor $c_{\kappa,s}/w$.

This new formulation offers a computational advantage. If $\boldsymbol{\theta}$ is close to the true value of the model and the bandwidth h is reasonably close to s, then the mode x_{mode} should be a good estimate of the intercept α. Now, any value of the bandwidth is satisfactory as long as we can reliably estimate the intercept for a given $\boldsymbol{\theta}$. This is a much easier requirement to satisfy than requiring the scale estimate to be an estimate of the unknown noise scale. Furthermore, for kernel density estimation, there exist plug in rules for bandwidth selection which are *purely data driven*. For one-dimensional kernel density estimation the bandwidth

$$h_{\boldsymbol{\theta}} = n^{-1/5} \operatorname*{med}_j \left|\mathbf{y}_j^T\boldsymbol{\theta} - \operatorname*{med}_i \mathbf{y}_i^T\boldsymbol{\theta}\right| \propto n^{-1/5}\sigma_{mad} \quad (10)$$

is proposed in [24, Sec.3.2.2]. The bandwidth is now explicitly dependent on the direction $\boldsymbol{\theta}$. This bandwidth is proportional to the robust *median absolute deviations* (MAD)

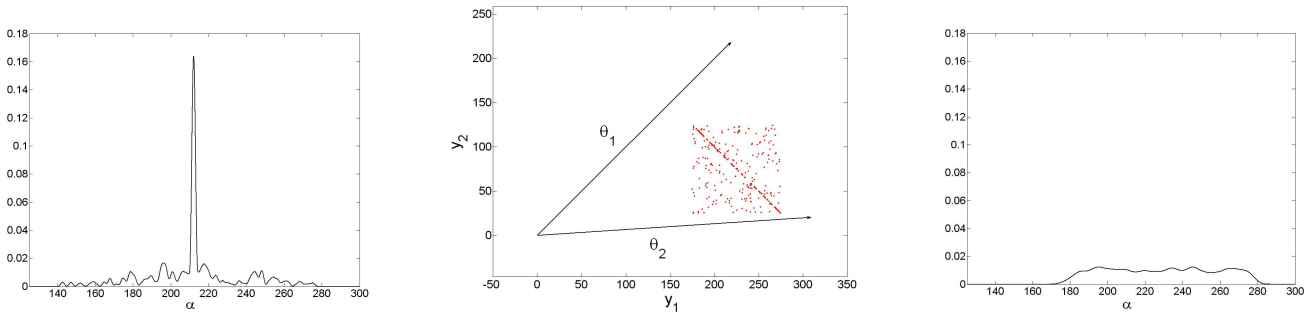

Figure 1. An example of projection pursuit. The 2D data points and the two directions, θ_1 and θ_2 are shown in the middle image. The kernel density estimate of the projections along θ_1 is shown on the left. There is a clear peak at the intercept. The projections along θ_2 give a more diffuse density, as seen in the right figure.

scale estimate, σ_{mad}, which is commonly used in vision applications. The cost function with this new bandwidth is

$$\hat{f}_{\boldsymbol{\theta}}(x) = \frac{1}{nh_{\boldsymbol{\theta}}} \sum_{i=1}^{n} \kappa \left(\frac{\mathbf{y}_i^T \boldsymbol{\theta} - x}{h_{\boldsymbol{\theta}} \sqrt{\boldsymbol{\theta}^T \mathbf{C}_i \boldsymbol{\theta}}} \right). \tag{11}$$

The constant $c_{\kappa,s}/w$ is dropped since it does not affect the function maximization, and the dependence of the cost function and bandwidth on the direction $\boldsymbol{\theta}$ is made explicit in the subscript. This cost function does not integrate to one and is not longer a kernel density. However, it is proportional to a kernel density estimate and the position of the maximum does not change. The robust heteroscedastic errors-in-variables problem becomes

$$\left[\hat{\alpha}, \hat{\boldsymbol{\theta}}\right] = \arg\max_{\boldsymbol{\theta}} \left[\max_{x} \hat{f}_{\boldsymbol{\theta}}(x) \right]. \tag{12}$$

This is a projection pursuit definition of the M-estimator problem. The *projection index* is the quantity in the brackets and accounts for the heteroscedasticity of the data. The deviation of each point from the hyperplane is $\mathbf{y}_i^T \boldsymbol{\theta} - \alpha$, and this deviation is normalized by its standard deviation, $\sqrt{\boldsymbol{\theta}^T \mathbf{C}_i \boldsymbol{\theta}}$, to account for the heteroscedasticity. The inner maximization returns the intercept as function of $\boldsymbol{\theta}$

$$\alpha = \arg\max_{x} \hat{f}_{\boldsymbol{\theta}}(x). \tag{13}$$

The projection pursuit approach toward M-estimation has a clear geometric interpretation. The direction $\boldsymbol{\theta}$ can be regarded as the unit normal of a candidate hyperplane fitted to the p-dimensional data. The bandwidth $h_{\boldsymbol{\theta}}$ defines a band along the direction $\boldsymbol{\theta}$. The band is translated in $\mathbb{R}^p$, along $\boldsymbol{\theta}$, to maximize the M-score of the orthogonal distances from the hyperplane. The M-estimate corresponds to the densest band over all $\boldsymbol{\theta}$.

These ideas are geometrically illustrated in Figure 1. The inliers, which lie close to a line, and outliers are shown in the middle figure. Their projections are taken along two directions, θ_1 and θ_2. The kernel density estimate of the projections along direction θ_1 is shown on the left and it exhibits a clear mode at the intercept. The kernel density estimate based on the projections along θ_2 is much more diffuse. The mode is not that high and consequently, θ_2 will have a much lower projection index than θ_1.

4. The pbM Algorithm

The first step of pbM consists of *probabilistic sampling*. Both RANSAC [6] and LMedS [12] also rely on probabilistic sampling. Elemental subsets containing the smallest number of data points which *uniquely* define the model parameters are chosen, without replacement, from the given data set. The elemental subsets are used to get initial estimate of $\boldsymbol{\theta}$.

Given an estimate of $\boldsymbol{\theta}$, the intercept α is estimated according to (13). This mode search is done through mean shift [4]. The density at α given by (11) is the projection index for $\boldsymbol{\theta}$. To improve this estimate, a local search is performed in the neighbourhood of $\boldsymbol{\theta}$.

The optimization criteria of (12) is non-differentiable and derivative based methods are not directly applicable to the function optimization. However, if we ignore the dependence of $h_{\boldsymbol{\theta}}$ on $\boldsymbol{\theta}$ and the relation between α and $\boldsymbol{\theta}$ as expressed in (13), the function is differentiable. These approximations are locally valid and it is now possible to compute the Jacobians of $\hat{f}_{\boldsymbol{\theta}}$ with respect to $\boldsymbol{\theta}$ and α.

By definition $\boldsymbol{\theta}$ is a unit vector and therefore we need to restrict the search to the unit sphere in $\mathbb{R}^p$ which is a particular case of a Grassmann manifold, $G_{p,1}$, consisting of vectors $\boldsymbol{\theta} \in \mathbb{R}^p$ satisfying $\boldsymbol{\theta}^T \boldsymbol{\theta} = 1$. We use the first order conjugate gradient method for Grassmann manifolds [5] which requires the computation of Jacobians. The details can be found in [5, 18]. At convergence of this local search, the mode is refined again using mean shift starting at the current estimate of α and the constraint (13) is again

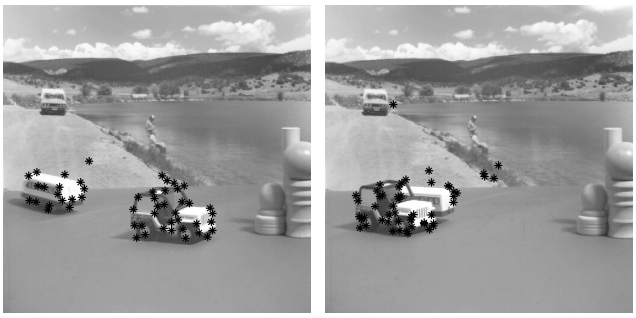

Figure 2. Images used for affine motion estimation. All the 51 points (inliers and outliers) matched in the two views are shown.

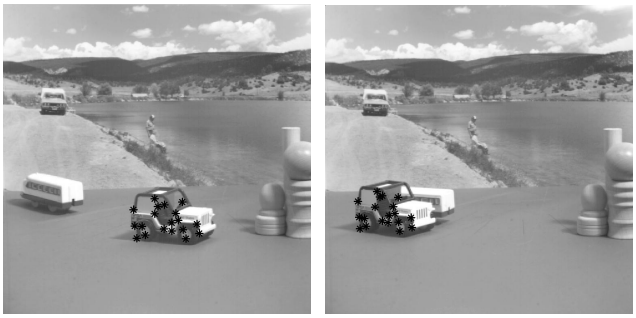

Figure 3. Results of affine motion estimation. The 19 inliers returned by pbM are shown in the two images.

enforced.

This procedure of local optimization is repeated for each elemental subset and the parameter pair $(\boldsymbol{\theta}, \alpha)$ with the highest score is taken as $(\hat{\boldsymbol{\theta}}, \hat{\alpha})$. Given $(\hat{\boldsymbol{\theta}}, \hat{\alpha})$, the inlier/outlier dichotomy estimation is based on the kernel density estimate of the projections along $\hat{\boldsymbol{\theta}}$. Starting from the mode at $\hat{\alpha}$ the first local minima of the distribution on both sides are estimated. Only points with *projections lying in this window* are taken to be inliers and all other points are classified outliers.

Replacing the scale s by the bandwidth of the kernel density estimator $h_{\boldsymbol{\theta}}$ is more than a simple substitution. For M-estimators and RANSAC, the scale determines a band, and the data points within this band are taken into consideration. For pbM the bandwidth $h_{\boldsymbol{\theta}}$, which is data driven, is only used in estimating the density of the projected points. The inlier/outlier dichotomy is determined by the shape of the density of projections along $\hat{\boldsymbol{\theta}}$ and $h_{\boldsymbol{\theta}}$ is *not* the threshold for acceptance. The acceptance or rejection of data points is based on the minima of the density around the mode, and these are not sensitive to changes of bandwidth $h_{\boldsymbol{\theta}}$.

The pbM algorithm can be extended to handle multiple constraints [18] by generalizing the M-estimator formulation of (2). The parameter values now lie on a general Grassmann manifold, $G_{p,k},\ k \leq p$, and the computational aspects of the algorithm can be directly extended to handle these spaces. The procedure to obtain the inlier/outlier dichotomy also needs to be adapted for multiple constraints.

5. Experimental Results

In this section we compare the performance of various robust estimators on real problems. Our algorithm was extensively tested and performed better than other estimators. Here we present results on a few well known images which have been commonly used to test various other methods. In our experiments, we compare pbM with RANSAC and MSAC. MSAC was compared to RANSAC [22], and was shown to do at least as well as RANSAC in all cases.

For each experiment, the ground truth was either available or computed by hand. Given the ground truth, the scale of the inlier noise σ_t was estimated. Both RANSAC and MSAC were tuned to the optimal value of $\sigma_{opt} = 1.96\sigma_t$. In a real application the ground truth is unknown and the scale of the inlier noise is also unknown. To simulate the real situation, RANSAC and MSAC were also run after tuning them to the MAD scale estimate given by

$$\sigma_{mad}(\boldsymbol{\theta}) = \underset{j}{\text{med}} \left| \mathbf{y}_j^T \boldsymbol{\theta} - \underset{i}{\text{med}}\, \mathbf{y}_i^T \boldsymbol{\theta} \right| . \tag{14}$$

The performance of pbM was compared to RANSAC and MSAC tuned to both the optimal and MAD scale estimates. The MAD scale estimate is a function of the direction $\boldsymbol{\theta}$. Adapting the scale to $\boldsymbol{\theta}$ is slightly better than using an initial TLS fit. When RANSAC and MSAC score for two different direction are being compared, the scores have been evaluated at different scales. To account for the scale variation, it is necessary to divide the score by the scale. However, RANSAC and MSAC do not do this and this affects the results. For each of the five estimators, the *same 1000 elemental subsets* were used.

5.1. Affine Motion Estimation

For 2D affine transformation associated with a moving object, $\mathbf{y}_{io} = [y_{i1o}\ \ y_{i2o}]\,,\ i = 1, 2$, are the (unknown) true coordinates of a pair of salient points in correspondence. The six parameter affine transformation between them

$$\begin{bmatrix} y_{21o} \\ y_{22o} \end{bmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{bmatrix} y_{11o} \\ y_{12o} \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} \tag{15}$$

can be decoupled into two three-dimensional problems, in a_{11}, a_{12}, t_1 and a_{21}, a_{22}, t_2 respectively, each obeying a linear model. Thus, the noisy measurements of corresponding points are distributed around two planes in two different 3D spaces. The transformation parameters can be found through two separate estimation processes, and points obeying the transformation must be inliers for *both* processes.

The images used in the experiment are shown in Figure 2. A large number of point correspondences were estab-

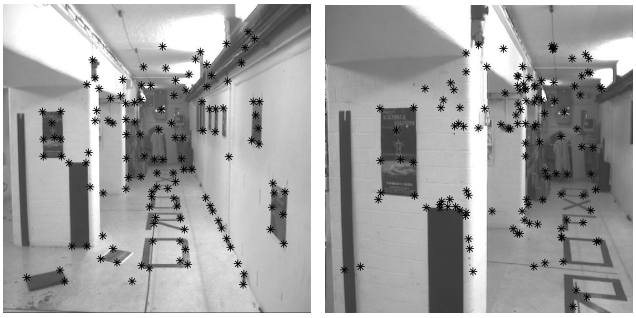

Figure 4. Fundamental matrix estimation for the *corridor* sequence. Frame 0 and frame 9 are shown along with all the 127 point matches (inliers and outliers).

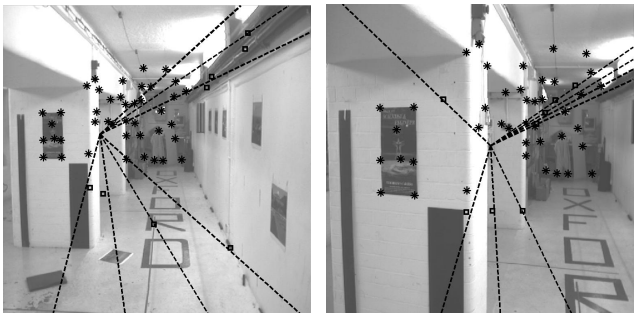

Figure 5. Results of fundamental matrix estimation for the *corridor* sequence. The 66 inliers returned by pbM and epipolar lines of the 8 outliers misclassified as inliers are shown.

lished using [7]. Next, the point matches on the static background were identified by having zero displacement and removed. The estimation process used the remaining 51 point correspondences of which 21 are inliers. The covariances of the matched points are found using the method of [9] and using a 7×7 window around each corner. Each data point has a different covariance and the data is heteroscedastic.

The performance of pbM is compared with RANSAC and MSAC in Table 1. For the ground truth, the inliers were manually selected. This was used to estimate the inlier noise standard deviation for each of the two 3-dimensional problems, $\sigma_t^{(1)} = 1.62$, $\sigma_t^{(2)} = 1.17$. The final RANSAC and MSAC estimates were obtained by applying them to the two subproblems and then combining the results. The performance of the estimators was compared based on the number of true inliers among points classified as inliers and the ratio of the standard deviation of the selected points to that of the true inlier noise. The closer the measures are to unity the better the performance. The inliers found by pbM are shown in Figure 3.

Table 1. Performance Comparison - Affine Motion Estimation

	sel./in.	$\sigma_{in}^{(1)}/\sigma_t^{(1)}$	$\sigma_{in}^{(2)}/\sigma_t^{(2)}$
RANSAC(σ_{opt})	9/9	0.30	0.28
MSAC(σ_{opt})	9/9	0.29	0.28
RANSAC(σ_{mad})	14/10	9.20	22.83
MSAC(σ_{mad})	12/10	9.76	16.52
pbM	19/19	1.13	0.92

RANSAC and MSAC give similar performances when the true scale of noise is known. The performance degrades when the MAD scale estimate is used while pbM does better then both of them even without any scale estimate.

5.2. Fundamental Matrix Estimation

For fundamental matrix estimation it is necessary to account for the heteroscedasticity of the data [10]. The fundamental matrix between two images of the same scene expresses the geometric *epipolar* constraint on corresponding points. The constraint is bilinear, and generally can be expressed as $[\mathbf{x}_{1o}^T \;\; 1]\mathbf{F}[\mathbf{x}_{2o}^T \;\; 1]^T = 0$. On linearizing we get

$$\hat{\mathbf{y}} = [\hat{\mathbf{x}}_1^T \;\; \hat{\mathbf{x}}_2^T \;\; \mathbf{vec}(\hat{\mathbf{x}}_1\hat{\mathbf{x}}_2^T)]^T \in \mathbb{R}^8 \tag{16}$$

$$[\hat{\boldsymbol{\theta}}^T \;\; \hat{\alpha}]^T = \mathbf{vec}(\hat{\mathbf{F}}) \tag{17}$$

$$\hat{\mathbf{y}}^T\hat{\boldsymbol{\theta}} - \hat{\alpha} = 0 \qquad \|\hat{\boldsymbol{\theta}}\| = 1 \tag{18}$$

where the **vec** operator transforms a matrix into its column-organized form. In the absence of any further knowledge it is reasonable to assume that the given estimates of $\mathbf{x}_{1o}$ and $\mathbf{x}_{2o}$ are corrupted by diagonal covariance matrices. However, the linearized data vectors $\mathbf{y}$ are bilinear functions of the point locations $\mathbf{x}_1$ and $\mathbf{x}_2$, and therefore the vectors $\mathbf{y}$ do *not* have the same covariances. The data vectors for the regression are heteroscedastic [10]. The linearized data vector covariances are found by error propagation.

To test our algorithm, we use two far apart frames, frames 0 and 9, from the well known *corridor* sequence. These images and the point matches are shown in Figure 4. The ground truth for this sequence is known and from this the inlier noise standard deviation is estimated as $\sigma_t = 0.88$.

Points were matched using the method of [7]. This gives 127 points with 58 inliers. Large portions of the first image are not visible in the second image and these points get mismatched. Performance is compared based on the number of true inliers among the points selected by an estimator and the ratio between the noise standard deviation of the selected points and the standard deviation of the inlier noise. The results of the various estimators are shown in Table 2.

Table 2. Performance Comparison - *Corridor* Image Pair

	selected points/true inliers	σ_{in}/σ_t
RANSAC(σ_{opt})	35/30	12.61
MSAC(σ_{opt})	11/8	9.81
RANSAC(σ_{mad})	103/52	15.80
MSAC(σ_{mad})	41/18	9.76
pbM	66/58	1.99

Figure 6. Fundamental matrix estimation for the *Valbonne* sequence. Both images are shown along with the 85 point matches (inliers and outliers).

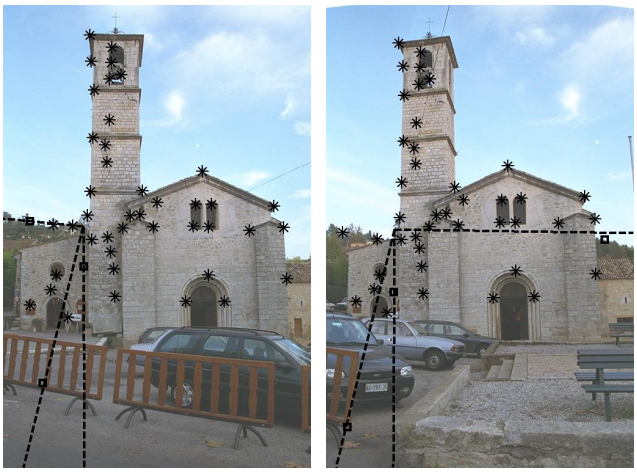

Figure 7. Results of fundamental matrix estimation for the *Valbonne* sequence. The 45 inliers returned by pbM and epipolar lines of the 3 outliers misclassified as inliers are shown.

It is clear that pbM outperforms RANSAC and MSAC in spite of being user independent. The point retained as inliers by pbM are shown in Figure 5. True inliers are shown as asterisks. Eight mismatches have been classified as inliers and these are shown as squares along with their epipolar lines. For these points, the epipolar lines pass very close to the mismatched points *or* one of the points lies close to the epipoles. In such cases the epipolar constraint (17) is satisfied. Since this is the only constraint that is being enforced, the system cannot detect such mismatches and these few mismatches are labeled inliers.

We find that pbM returns accurate estimates of the fundamental matrix. The fundamental matrix between two images should be of rank-2. This condition is usually ignored by robust regression algorithms. Once a satisfactory inlier/outlier dichotomy has been obtained, more complex estimation methods are applied to the inliers while enforcing the rank-2 constraint. Consequently, the fundamental matrix estimate returned by most robust regression algorithms are not good estimates of the true fundamental matrix. In [1] a different version of pbM, which does not account for the heteroscedasticity of the data, was used for robust fundamental matrix estimation. Even in this case, it was found that pbM gives good inlier/outlier dichotomies but incorrect estimates of the fundamental matrix.

The heteroscedastic pbM algorithm discussed here, satisfies the rank-2 constraint even though it is not explicitly enforced. This is probably because the heteroscedastic nature of the noise is accounted for and our estimate is very close to the true fundamental matrix. For the estimate returned by pbM, the ratio of the second singular value to the third singular value is of the order of 1000. The epipoles computed from the estimated fundamental matrix also matched the ground truth epipoles.

The system was tested on another image pair taken from the *Valbonne* sequence. The images and the point matches are shown in Figure 6. The true matches were selected by manual inspection to obtain the ground truth. The standard deviation of the inlier noise was fond to be $\sigma_t = 0.072$. Of the 85 points matched, 42 were inliers and the rest are mismatches. The results of the various estimators are compared in Table 3. Again, pbM does better than RANSAC and MSAC in spite of being user-independent. The rank-2 constraint is satisfied with the ratio between the second and third singular values of the fundamental matrix being of the order of 10000. The epipoles from the estimated fundamental matrix also match the true epipoles.

Table 3. Performance Comparison - *Valbonne* Image Pair

	selected points/true inliers	σ_{in}/σ_t
RANSAC(σ_{opt})	31/26	7.36
MSAC(σ_{opt})	9/6	6.93
RANSAC(σ_{mad})	46/37	17.73
MSAC(σ_{mad})	27/21	11.82
pbM	45/42	0.97

In all experiments, RANSAC and MSAC performed better when tuned to the optimal scale value. When using the MAD scale, the scores for each trial are estimated at different scales. The effect of the scale needs to be accounted for by dividing the score by the scale. Both RANSAC and MSAC do not do this. When using the same scale selection mechanism, MSAC does better than RANSAC which was expected. Among the estimators compared here, pbM clearly gives the best performance. It uses a robust data-drive scale, and accounts for the variation in scale by normalizing the score. The inlier/outlier dichotomy is separated from the scale estimation and this makes the inlier se-

lection procedure more robust.

6. Conclusion

We presented a new robust estimator, the projection based M-estimator, which is user-independent and does not require an estimate of the scale of inlier noise. Other methods like RANSAC and MSAC can be improved by accounting for the scale variation for different elemental subsets. However, pbM still outperforms them, gives better inlier/outlier dichotomies and better parameter estimates, in spite of being completely autonomous.

References

[1] H. Chen and P. Meer, "Robust regression with projection based M-estimators," in *Proc. 9th Intl. Conf. on Computer Vision,*Nice, France, volume II, (Nice, France), Oct 2003, pp. 878–885.

[2] H. Chen, P. Meer, and D. E. Tyler, "Robust regression for data with multiple structures," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Kauai, Hawaii, volume I, (Kauai, HI), December 2001, pp. 1069–1075.

[3] O. Chum and J. Matas, "Matching with PROSAC - progressive sample consensus," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* San Diego, CA, volume 1, June 2005, pp. 220–226.

[4] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, pp. 603–619, May 2002.

[5] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM Journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.

[6] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Comm. Assoc. Comp. Mach*, vol. 24, no. 6, pp. 381–395, 1981.

[7] B. Georgescu and P. Meer, "Point matching under large image deformations and illumination changes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, pp. 674–689, 2004.

[8] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[9] Y. Kanazawa and K.Kanatani, "Do we really have to consider covariance matrices for image features?," in *Proc. 8th Intl. Conf. on Computer Vision,* Vancouver, Canada, volume II, (Vancouver, CA), July 2001.

[10] B. Matei and P. Meer, "A general method for errors-in-variables problems in computer vision," in *2000 IEEE Computer Vision and Pattern Recognition Conference*, volume II, (Hilton Head Island, SC), June 2000, pp. 18–25.

[11] D. Nister, "Preemptive RANSAC for live structure and motion estimation," in *Proc. 9th Intl. Conf. on Computer Vision,*Nice, France, volume I, October 2003, pp. 199–206.

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

[13] S. Rozenfeld and I. Shimshoni, "The modified pbM-estimator method and a runtime analysis technique for the RANSAC family," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* San Diego, CA, volume I, June 2005, pp. 1113–1120.

[14] M. Singh, H. Arora, and N. Ahuja, "A robust probabilistic estimation framework for parametric image models," in T. Pajdla and J. Matas, editors, *Computer Vision – ECCV 2004*, volume I, (Prague, Czech Republic), Springer-Verlag, May 2004, pp. 508–522.

[15] C. V. Stewart, "Robust parameter estimation in computer vision.," *SIAM Reviews*, vol. 41, pp. 513–537, 1999.

[16] R. Subbarao, Y. Genc, and P. Meer, "A balanced approach to 3D tracking from image streams," in *Proc. IEEE and ACM International Symposium on Mixed and Augmented Reality*, October 2005, pp. 70–78.

[17] R. Subbarao and P. Meer, "Heteroscedastic projection based M-estimators," in *Workshop on Empirical Evaluation Methods in Computer Vision,* San Diego, CA, June 2005.

[18] R. Subbarao and P. Meer, "Subspace estimation using projection based M-estimators over Grassmann manifolds," in *Proc. European Conf. on Computer Vision,* Graz, Austria, volume I, May 2006, pp. 301–312.

[19] B. Tordoff and D. Murray, "Guided sampling and consensus for motion estimation," in *7th European Conference on Computer Vision*, volume I, (Copenhagen, Denmark), May 2002, pp. 82–96.

[20] P. H. S. Torr and C. Davidson, "IMPSAC: Synthesis of importance sampling and random sample consensus," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 3, pp. 354–364, 2003.

[21] P. H. S. Torr and D. W. Murray, "The development and comparison of robust methods for estimating the fundamental matrix," *International J. of Computer Vision*, vol. 24, no. 3, pp. 271–300, 1997.

[22] P. H. S. Torr and A. Zisserman, "Robust parameterization and computation of the trifocal tensor," *Image and Vision Computing*, vol. 15, pp. 591–605, August 1997.

[23] P. H. S. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, pp. 138–156, 2000.

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

[25] H. Wang and D. Suter, "Robust adaptive-scale parametric model estimation for computer vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 11, pp. 1459–1474, 2004.