A simplified treatment of Kanatani's renormalisation method

Wojciech Chojnacki Michael J. Brooks Anton van den Hengel
Department of Computer Science
University of Adelaide
Adelaide, SA 5005, Australia

Abstract

The renormalisation method of Kanatani is intended to find the minimisers of cost functions of a certain form. As such, it has applicability to a wide spectrum of computer vision problems that may be couched in these terms. However, despite its sophistication, the method of Kanatani has been slow to gain broad acceptance, perhaps because of the complexity of its original derivation. In this paper we present an alternative and simpler treatment of this important method.

1 Introduction

Many problems in computer vision are readily formulated as the need to minimise a cost function with respect to some unknown parameters. Such a cost function will often involve covariance matrices characterising uncertainty of the data and will take the form of a sum of quotients of quadratic forms in the parameter. Finding the values of the parameters that minimise such a cost function is often difficult.

One approach to minimising a cost function represented as a sum of fractional expressions is attributed to Sampson. Here, an initial estimate is substituted into the denominators of the cost function, and a minimiser is sought for the now scalar-weighted numerators. This procedure is then repeated with the newly obtained estimate until convergence is obtained. It emerges that this approach is biased. Noting this, Kanatani developed the *renormalisation method* that aims to undo these biasing effects.

The renormalisation method has been slow to gain wide acceptance in the vision community, perhaps due to the complexity of the mathematics associated with the original presentation. In this paper we present a simplified account of the method. It should be noted that we are not here concerned with other aspects of Kanatani's theory, such as how the cost function is derived using principle of maximum likelihood, or how we may obtain error bounds on the accuracy of estimates obtained.

We commence with a description of the class of problems to which renormalisation is applicable.

2 A class of problems

Suppose we have an item of data $\boldsymbol{x} = (x_1, \ldots, x_k)$ that constitutes an observation of some underlying 'true' value $\overline{\boldsymbol{x}} = (\bar{x}_1, \ldots, \bar{x}_k)$. The item of data might represent, for example, the measured locations of corresponding points acquired from a stereo pair of images. Suppose also that each true value is subject to the equation

$$\boldsymbol{\theta}^T \boldsymbol{u}(\overline{\boldsymbol{x}}) = 0, \tag{1}$$

where $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_l]^T$ is a vector representing unknown parameters, and $\boldsymbol{u}(\boldsymbol{x}) = [u_1(\boldsymbol{x}), \ldots, u_l(\boldsymbol{x})]^T$ is a vector with it argument transformed in such a way that: (i) each component is a quadratic form in the compound vector $[\boldsymbol{x}^T, 1]^T$, (ii) one component of $\boldsymbol{u}(\boldsymbol{x})$ is equal to 1. For some scalar-valued function ψ, let

$$\psi(\boldsymbol{\theta}) = 0 \tag{2}$$

be an ancillary constraint that may also apply, but which does not involve the data.

Our class of estimation problems can now be stated as follows: Find $\boldsymbol{\theta}$, satisfying (2), which "best fits" the data $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$. This problem statement will be formalised in the following sections.

2.1 An example problem

Conic fitting is a classical problem of the above type [2, 10]. A point $\boldsymbol{y} = [y_1, y_2]^T$ lies on a particular *conic* if

$$ay_1^2 + by_1y_2 + cy_2^2 + dy_1 + ey_2 + f = 0 \tag{3}$$

for some real numbers a, b, c, d, e not all simultaneously zero. We can group together the coefficients a, b, c, d, e, f, obtaining the vector of *parameters*

$$\boldsymbol{\theta} = [a, b, c, d, e, f]^T. \tag{4}$$

If, adopting the terminology of Meer [7], we introduce the vector of *basis functions* or *carriers*

$$\boldsymbol{u}(\boldsymbol{y}) = [y_1^2, y_1y_2, y_2^2, y_1, y_2, 1]^T, \tag{5}$$

then equation (3) can be rewritten as

$$\boldsymbol{\theta}^T \boldsymbol{u}(\boldsymbol{y}).$$

The conic itself is therefore defined as the set of all points $\boldsymbol{y}$ such that $\boldsymbol{\theta}^T \boldsymbol{u}(\boldsymbol{y}) = 0$.

Assume that we have measured data points $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ to which we wish to fit a conic. If $n = 5$ then we can solve the five equations $\boldsymbol{\theta}^T \boldsymbol{u}(\boldsymbol{x}_i) = 0$ algebraicly to determine $\boldsymbol{\theta}$. If $n > 5$, however, then it is unlikely that all the measured points will lie exactly on any particular conic. This is because the measurement process introduces noise into the data. The noise introduced into the data, and the fact that $n > l$, result in the system of equations $\boldsymbol{\theta}^T \boldsymbol{u}(\boldsymbol{x}_i) = 0$ being overdetermined. Given that there is no algebraic solution to the problem, we want to select the $\boldsymbol{\theta}$ representing the conic which, in some sense, best fits the data. A means by which such a $\boldsymbol{\theta}$ may be selected is given in Section 4.2.

Among the other problems to which the renormalisation technique is applicable are estimating coefficients of the *epipolar equation* [5], and estimating coefficients of the *differential epipolar equation* [3, 9], each involving an ancillary *cubic constraint*. The precise way in which these example problems accord with the our problem form is described in [4].

3 Cost Functions and Estimators

A vast class of techniques for solving our problem rest upon the use of *cost functions* measuring the extent to which the data and candidate estimates fail to satisfy (1). If—for simplicity—one sets aside the ancillary constraint, then, given a cost function $J = J(\boldsymbol{\theta}; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, a corresponding estimate $\widehat{\boldsymbol{\theta}}$ is defined by

$$J(\widehat{\boldsymbol{\theta}}) = \min_{\boldsymbol{\theta} \neq 0} J(\boldsymbol{\theta}; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n).$$

Cost functions such as $J(\boldsymbol{\theta}; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ are often capable of accepting covariance matrices characterising uncertainty of the data. It is, however, quite possible to proceed without such information by assuming equal variances for all the variable elements of the data vectors.

3.1 Ordinary Least Squares Estimator

A straightforward estimator is the minimiser of the cost function

$$J_{\mathrm{OLS}}(\boldsymbol{\theta}; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{i=1}^{n} \frac{\boldsymbol{\theta}^T \boldsymbol{A}_i \boldsymbol{\theta}}{\|\boldsymbol{\theta}\|},$$

where $\boldsymbol{A}_i = \boldsymbol{u}(\boldsymbol{x}_i)\boldsymbol{u}(\boldsymbol{x}_i)^T$. Here the numerator of each summand is $\boldsymbol{\theta}^T \boldsymbol{A}_i \boldsymbol{\theta}$, which is the square of the *algebraic distance* $\|\boldsymbol{\theta}^T \boldsymbol{u}(\boldsymbol{x}_i)\|$. The denominator merely ensures that the cost function is homogeneous in $\boldsymbol{\theta}$. So, for our conic fitting example, the algebraic residual for a particular data point $\boldsymbol{x}_i$ is just the absolute value of equation (3) for that point divided by the magnitude of $\boldsymbol{\theta}$. The degree to which a candidate conic $\boldsymbol{\theta}$ fits the data is thus measured by the sum of the squares of these algebraic residuals. The $\boldsymbol{\theta}$ for which J_{OLS} is minimal is termed the *ordinary least squares* (OLS) *estimate* and is denoted $\widehat{\boldsymbol{\theta}}_{\mathrm{OLS}}$. It is uniquely determined, up to a scalar factor, by an eigenvector of $\sum_{i=1}^{n} \boldsymbol{A}_i$ associated with the smallest eigenvalue.

4 Weighted Least Squares

Kanatani's renormalisation method aims at minimising cost functions of a particular form. These cost functions can be derived in several ways, but the result can be seen as being similar to the ordinary least squares cost function outlined above, with the addition of a weighting for each data point. For this reason they are labeled as weighted least squares cost functions.

4.1 A model of the data

We regard the data $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ as a sample value taken on by an aggregate of vector-valued random variables $(\mathsf{x}_1, \ldots, \mathsf{x}_n)$. Each of these vector valued random variables x_i is taken to be centred on the true value $\overline{\boldsymbol{x}}_i$ satisfying $\boldsymbol{\theta}^T \boldsymbol{u}(\overline{\boldsymbol{x}}_i) = 0$. We also assume that the data come equipped with a collection $(\boldsymbol{\Lambda}_{\boldsymbol{x}_1}, \ldots, \boldsymbol{\Lambda}_{\boldsymbol{x}_n})$ of positive definite $k \times k$ *covariance matrices*. These matrices constitute repositories of prior information about the uncertainty of the data. The random variables $(\mathsf{x}_1, \ldots, \mathsf{x}_n)$ thus satisfy

$$\mathsf{E}[\mathsf{x}_i] = \overline{\boldsymbol{x}}_i \text{ and } \mathsf{E}\left[(\mathsf{x}_i - \overline{\boldsymbol{x}}_i)(\mathsf{x}_i - \overline{\boldsymbol{x}}_i)^T\right] = \boldsymbol{\Lambda}_{\boldsymbol{x}_i}. \quad (6)$$

If covariance information is not available then default matrices $\boldsymbol{\Lambda}_{\boldsymbol{x}_i}$ can be manufactured by setting the diagonal elements corresponding to the measured elements of the data vectors $\boldsymbol{x}_i$ to 1. For our conic fitting problem the manufactured covariances would be $\boldsymbol{\Lambda}_{\boldsymbol{x}_i} = \mathrm{diag}(1, 1)$.

4.2 The Mahanalobis Distance

Using the explicit form of the probability density functions involved, one can show that the *maximum likelihood estimate* is the parameter $\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}$ at which the cost function

$$J_{\mathrm{ML}}(\boldsymbol{\theta}; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{i=1}^{n} (\boldsymbol{x}_i - \overline{\boldsymbol{x}}_i)^T \boldsymbol{\Lambda}_{\boldsymbol{x}_i}^{-1} (\boldsymbol{x}_i - \overline{\boldsymbol{x}}_i)$$

attains a minimum [4, 6]. Each term in the above summation represents the squared *Mahalanobis distance* between the measured location $\boldsymbol{x}_i$ and its true location $\overline{\boldsymbol{x}}_i$. Note that the minimiser of the cost function $\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}$ remains unchanged if the covariance matrices are multiplied by a common scalar.

4.3 Weighted Least Squares Cost Function

Unfortunately the function $J_{\mathrm{ML}}(\boldsymbol{\theta}; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ does not lend itself to explicit calculation. We can derive a good approximation to J_{ML}, however, in the form of the function

$$J_{\mathrm{WLS}}(\boldsymbol{\theta}; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{i=1}^{n} \frac{\boldsymbol{\theta}^T \boldsymbol{A}_i \boldsymbol{\theta}}{\boldsymbol{\theta}^T \boldsymbol{B}_i \boldsymbol{\theta}}. \quad (7)$$

where matrix $\boldsymbol{A}_i$ is as described above and $\boldsymbol{B}_i$ defined as $\boldsymbol{B}(\boldsymbol{x}_i, \boldsymbol{\Lambda}_{\boldsymbol{x}_i}) = \partial_{\boldsymbol{x}}\boldsymbol{u}(\boldsymbol{x}_i)\boldsymbol{\Lambda}_{\boldsymbol{x}_i}\partial_{\boldsymbol{x}}\boldsymbol{u}(\boldsymbol{x}_i)^T$. The matrix $\partial_{\boldsymbol{x}}\boldsymbol{u}(\boldsymbol{y})$ is defined by

$$(\partial_{\boldsymbol{x}}\boldsymbol{u}(\boldsymbol{y}))_{ij} = \frac{\partial u_i}{\partial x_j}(\boldsymbol{y}).$$

The J_{WLS}-based estimate of $\boldsymbol{\theta}$ will be called the *weighted least squares* (WLS) *estimate* and will be denoted $\widehat{\boldsymbol{\theta}}_{\mathrm{WLS}}$. The renormalisation method was designed to solve precisely this cost function.

5 Iteratively Reweighted Least Squares Scheme

Sampson [8] proposed a scheme aiming to minimise a cost function involving fractional expressions. Although Sampson's work did not incorporate covariances, his algorithm is readily applicable to cost functions such as J_{WLS}. Using Sampson's method an initial estimate is substituted into the denominators of the expression and a minimiser is sought for the now scalar-weighted numerators. This process is then repeated on the basis of this minimiser.

Let

$$\boldsymbol{M}_{\boldsymbol{\theta}} = \sum_{i=1}^{n} \frac{\boldsymbol{A}_i}{\boldsymbol{\theta}^T\boldsymbol{B}_i\boldsymbol{\theta}} \tag{8}$$

and let

$$J'_{\mathrm{WLS}}(\boldsymbol{\theta}, \boldsymbol{\xi}; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \boldsymbol{\theta}^T\boldsymbol{M}_{\boldsymbol{\xi}}\boldsymbol{\theta}.$$

For simplicity, we abbreviate $J'_{\mathrm{WLS}}(\boldsymbol{\theta}, \boldsymbol{\xi}; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ to $J'_{\mathrm{WLS}}(\boldsymbol{\theta}, \boldsymbol{\xi})$. Sampson's method takes an estimate $\boldsymbol{\theta}_{k-1}$ and generates an update $\boldsymbol{\theta}_k$ by minimising the cost function $\boldsymbol{\theta} \mapsto J'_{\mathrm{WLS}}(\boldsymbol{\theta}, \boldsymbol{\theta}_{k-1})$. It is for this reason that Sampson's method is labeled the *iteratively reweighted least squares* (IRWLS) *scheme*. Sampson's method takes $\widehat{\boldsymbol{\theta}}_{\mathrm{OLS}}$ for its initial estimate $\boldsymbol{\theta}_0$, so, assuming that the sequence $\{\boldsymbol{\theta}_k\}$ converges, the *iteratively reweighted least squares estimate* is defined as $\widehat{\boldsymbol{\theta}}_{\mathrm{IRWLS}} = \lim_{k\to\infty}\boldsymbol{\theta}_k$. The IRWLS scheme is summarised in Figure 1.

6 The renormalisation scheme

Unfortunately the IRWLS sequence $\lim_{k\to\infty}\boldsymbol{\theta}_k$ does not, in general, converge to $\widehat{\boldsymbol{\theta}}_{\mathrm{WLS}}$. An estimator is said to be biased if the expected value of the estimator is not the minimiser of the associated cost function. We now show the method by which renormalisation aims to counter the bias in the IRWLS scheme.

Let $(\mathsf{x}_1, \ldots, \mathsf{x}_n)$ be vector-valued random variables centred on $(\overline{\boldsymbol{x}}_1, \ldots, \overline{\boldsymbol{x}}_n)$ with covariances $(\boldsymbol{\Lambda}_{x_1}, \ldots, \boldsymbol{\Lambda}_{x_n})$. These random variables have a probability distribution satisfying equation (6). Noting that $(\overline{\boldsymbol{x}}_1, \ldots, \overline{\boldsymbol{x}}_n)$ satisfy equation (1) for some $\boldsymbol{\theta}$, let

1. Set $\boldsymbol{\theta}_0 = \widehat{\boldsymbol{\theta}}_{\mathrm{OLS}}$.
2. Assuming that $\boldsymbol{\theta}_{k-1}$ is known, compute the matrix $\boldsymbol{M}_{\boldsymbol{\theta}_{k-1}}$.
3. Compute a normalised eigenvector of $\boldsymbol{M}_{\boldsymbol{\theta}_{k-1}}$ corresponding to the smallest (non-negative) eigenvalue and take this eigenvector for $\boldsymbol{\theta}_k$.
4. If $\boldsymbol{\theta}_k$ is sufficiently close to $\boldsymbol{\theta}_{k-1}$, then terminate the procedure; otherwise increment k and return to Step 2.

Figure 1: Iteratively reweighted least squares scheme.

$\boldsymbol{\eta} = (\boldsymbol{\theta}; \overline{\boldsymbol{x}}_1, \ldots, \overline{\boldsymbol{x}}_n)$ denote the parameterisation of our distribution. Form the following random version of $\boldsymbol{M}_{\boldsymbol{\theta}}$

$$\mathsf{M}_{\boldsymbol{\eta}} = \sum_{i=1}^{n} \frac{1}{\boldsymbol{\theta}^T\boldsymbol{B}(\overline{\boldsymbol{x}}_i, \boldsymbol{\Lambda}_{\boldsymbol{x}_i})\boldsymbol{\theta}}\boldsymbol{A}(\mathsf{x}_i)$$

with 'true' value

$$\overline{\mathsf{M}}_{\boldsymbol{\eta}} = \sum_{i=1}^{n} \frac{1}{\boldsymbol{\theta}^T\boldsymbol{B}(\overline{\boldsymbol{x}}_i, \boldsymbol{\Lambda}_{\boldsymbol{x}_i})\boldsymbol{\theta}}\boldsymbol{A}(\overline{\boldsymbol{x}}_i).$$

In view of (1), $\boldsymbol{A}(\overline{\boldsymbol{x}}_i)\boldsymbol{\theta} = 0$, for each $i = 1, \ldots, n$, so $\overline{\mathsf{M}}_{\boldsymbol{\eta}}\boldsymbol{\theta} = 0$ and further $\boldsymbol{\theta}^T\overline{\mathsf{M}}_{\boldsymbol{\eta}}\boldsymbol{\theta} = 0$. On the other hand, since each rank-one matrix $\boldsymbol{A}(\mathsf{x}_i)$ is non-negative definite, and since also each $\boldsymbol{B}(\overline{\boldsymbol{x}}_i, \boldsymbol{\Lambda}_{\boldsymbol{x}_i})$ is non-negative definite $\mathsf{M}_{\boldsymbol{\eta}}$ is non-negative definite. As the $\boldsymbol{A}(\mathsf{x}_i)$ are independent, $\mathsf{M}_{\boldsymbol{\eta}}$ is generically positive definite, with $\mathsf{E}\left[\boldsymbol{\theta}^T\mathsf{M}_{\boldsymbol{\eta}}\boldsymbol{\theta}\right] > 0$.

6.1 Removing the bias

Kanatani [6] developed a renormalisation method whereby an attempt is made at each iteration to undo the bias in $\boldsymbol{\theta}^T\mathsf{M}_{\boldsymbol{\eta}}\boldsymbol{\theta}$. Many examples may be found in the literature of problems benefiting from this approach.

Using Kanatani's method, the bias in $\boldsymbol{\theta}^T\mathsf{M}_{\boldsymbol{\eta}}\boldsymbol{\theta}$ can be removed by forming the matrix

$$\sum_{i=1}^{n} \frac{\boldsymbol{\theta}^T\boldsymbol{A}(\mathsf{x}_i)\boldsymbol{\theta} - \mathsf{E}\left[\boldsymbol{\theta}^T\boldsymbol{A}(\mathsf{x}_i)\boldsymbol{\theta}\right]}{\boldsymbol{\theta}^T\boldsymbol{B}(\overline{\boldsymbol{x}}_i, \boldsymbol{\Lambda}_{\boldsymbol{x}_i})\boldsymbol{\theta}}. \tag{9}$$

The terms $\mathsf{E}\left[\boldsymbol{\theta}^T\boldsymbol{A}(\mathsf{x}_i)\boldsymbol{\theta}\right]$ can be calculated explicitly. There is a matrix-valued function $(\boldsymbol{x}, \boldsymbol{\Lambda}) \mapsto \boldsymbol{D}(\boldsymbol{x}, \boldsymbol{\Lambda})$, to be specified later, such that, for each $i = 1, \ldots, n$,

$$\mathsf{E}\left[\boldsymbol{\theta}^T\boldsymbol{A}(\mathsf{x}_i)\boldsymbol{\theta}\right] = \boldsymbol{\theta}^T\boldsymbol{D}(\overline{\boldsymbol{x}}_i, \boldsymbol{\Lambda}_{\boldsymbol{x}_i})\boldsymbol{\theta}.$$

The unbiased $\mathsf{M}_{\boldsymbol{\eta}}$ can be written as

$$\mathsf{Y}_{\boldsymbol{\eta}} = \sum_{i=1}^{n} \frac{\boldsymbol{A}(\mathsf{x}_i) - \boldsymbol{D}(\overline{\boldsymbol{x}}_i, \boldsymbol{\Lambda}_{\boldsymbol{x}_i})}{\boldsymbol{\theta}^T \boldsymbol{B}(\overline{\boldsymbol{x}}_i)\boldsymbol{\theta}}. \tag{10}$$

The random matrix $\mathsf{Y}_{\boldsymbol{\eta}}$ is a raw model for obtaining a fully deterministic modification of $\boldsymbol{M}_{\boldsymbol{\theta}}$. For each $i = 1, \ldots, n$, let $\boldsymbol{D}_i = \boldsymbol{D}(\boldsymbol{x}_i, \boldsymbol{\Lambda}_{\boldsymbol{x}_i})$. Guided by (10), we take

$$\boldsymbol{Y}_{\boldsymbol{\theta}} = \sum_{i=1}^{n} \frac{\boldsymbol{A}_i - \boldsymbol{D}_i}{\boldsymbol{\theta}^T \boldsymbol{B}_i \boldsymbol{\theta}} \tag{11}$$

for a modified $\boldsymbol{M}_{\boldsymbol{\theta}}$. Somewhat surprisingly, the definition of $\boldsymbol{D}$ in equation (9) turns out not to be satisfactory. As Kanatani has pointed out, the problem is that while the $\boldsymbol{A}_i$ do not change when the $\boldsymbol{\Lambda}_{\boldsymbol{x}_i}$ are multiplied by a common scalar, the $\boldsymbol{D}_i$ do change. A properly designed algorithm employing a modified $\boldsymbol{M}_{\boldsymbol{\theta}}$ should provide estimates that remain intact when all the $\boldsymbol{\Lambda}_{\boldsymbol{x}_i}$ are multiplied by a common scalar. This is especially important if we aim not only to estimate the parameter, but also to evaluate the goodness of fit. Therefore further change to the numerators of the fractions forming $\boldsymbol{Y}_{\boldsymbol{\theta}}$ is necessary.

The dependence of $\boldsymbol{D}(\boldsymbol{x}, \boldsymbol{\Lambda})$ on $\boldsymbol{\Lambda}$ is fairly complex. To gain an idea of what needs to be changed, it is instructive to consider a simplified form of $\boldsymbol{D}(\boldsymbol{x}, \boldsymbol{\Lambda})$. A first-order (in some sense) approximation to $\boldsymbol{D}(\boldsymbol{x}, \boldsymbol{\Lambda})$ is, as it turns out, the matrix $\boldsymbol{B}(\boldsymbol{x}, \boldsymbol{\Lambda})$ defined above. The dependence of $\boldsymbol{B}(\boldsymbol{x}, \boldsymbol{\Lambda})$ on $\boldsymbol{\Lambda}$ is simple: if $\boldsymbol{\Lambda}$ is multiplied by a scalar, then $\boldsymbol{B}(\boldsymbol{x}, \boldsymbol{\Lambda})$ is multiplied by the same scalar. This suggests that we introduce a *compensating factor* $c(\boldsymbol{\theta}; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, or $c(\boldsymbol{\theta})$ in short, with the property that if the $\boldsymbol{\Lambda}_{\boldsymbol{x}_i}$ are multiplied by a scalar, then $c(\boldsymbol{\theta})$ is multiplied by the inverse of this scalar. With the help of $c(\boldsymbol{\theta})$, we can form, for each $i = 1, \ldots, n$, a *renormalised* numerator $\boldsymbol{\theta}^T \boldsymbol{A}_i \boldsymbol{\theta} - c(\boldsymbol{\theta})\, \boldsymbol{\theta}^T \boldsymbol{B}_i \boldsymbol{\theta}$ and can next set

$$\boldsymbol{Y}_{\boldsymbol{\theta}} = \sum_{i=1}^{n} \frac{\boldsymbol{A}_i - c(\boldsymbol{\theta})\boldsymbol{B}_i}{\boldsymbol{\theta}^T \boldsymbol{B}_i \boldsymbol{\theta}} = \boldsymbol{M}_{\boldsymbol{\theta}} - c(\boldsymbol{\theta})\boldsymbol{N}_{\boldsymbol{\theta}}, \tag{12}$$

where $\boldsymbol{M}_{\boldsymbol{\theta}}$ is given in (8) and $\boldsymbol{N}_{\boldsymbol{\theta}}$ is defined by

$$\boldsymbol{N}_{\boldsymbol{\theta}} = \sum_{i=1}^{n} \frac{1}{\boldsymbol{\theta}^T \boldsymbol{B}_i \boldsymbol{\theta}} \boldsymbol{B}_i.$$

The numerators in (12) are clearly scale invariant.

We require $\boldsymbol{Y}_{\boldsymbol{\theta}}$ to be unbiased, which implies that $\boldsymbol{\theta}^T \boldsymbol{Y}_{\boldsymbol{\theta}} \boldsymbol{\theta} = 0$ and therefore that

$$c(\boldsymbol{\theta}; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \frac{1}{n} \sum_{i=1}^{n} \frac{\boldsymbol{\theta}^T \boldsymbol{A}_i \boldsymbol{\theta}}{\boldsymbol{\theta}^T \boldsymbol{B}_i \boldsymbol{\theta}} = \frac{\boldsymbol{\theta}^T \boldsymbol{M}_{\boldsymbol{\theta}} \boldsymbol{\theta}}{\boldsymbol{\theta}^T \boldsymbol{N}_{\boldsymbol{\theta}} \boldsymbol{\theta}}. \tag{13}$$

It is obvious that $c(\boldsymbol{\theta})$ thus defined has the property required of a compensating factor. Moreover, this form of $c(\boldsymbol{\theta})$ is in accordance with the unbiasedness paradigm. Indeed, if we form the random version of c

$$\mathsf{c}(\boldsymbol{\theta}; \mathsf{x}_1, \ldots, \mathsf{x}_n) = \frac{1}{n} \sum_{i=1}^{n} \frac{\boldsymbol{\theta}^T \boldsymbol{A}(\mathsf{x}_i)\boldsymbol{\theta}}{\boldsymbol{\theta}^T \boldsymbol{B}(\overline{\boldsymbol{x}}_i, \boldsymbol{\Lambda}_{\boldsymbol{x}_i})\boldsymbol{\theta}},$$

then, insofar as $\mathsf{E}\left[\boldsymbol{\theta}^T \boldsymbol{A}(\mathsf{x}_i)\boldsymbol{\theta}\right] = \boldsymbol{\theta}^T \boldsymbol{B}(\overline{\boldsymbol{x}}_i, \boldsymbol{\Lambda}_{\boldsymbol{x}_i})\boldsymbol{\theta}$, we have, abbreviating $\mathsf{c}(\boldsymbol{\theta}; \mathsf{x}_1, \ldots, \mathsf{x}_n)$ to $\mathsf{c}(\boldsymbol{\theta})$,

$$\mathsf{E}\left[\mathsf{c}(\boldsymbol{\theta})\right] = 1 \tag{14}$$

and further

$$\mathsf{E}\left[\sum_{i=1}^{n} \frac{\boldsymbol{\theta}^T \boldsymbol{A}(\mathsf{x}_i)\boldsymbol{\theta} - \mathsf{c}(\boldsymbol{\theta})\boldsymbol{\theta}^T \boldsymbol{B}(\overline{\boldsymbol{x}}_i, \boldsymbol{\Lambda}_{\boldsymbol{x}_i})\boldsymbol{\theta}}{\boldsymbol{\theta}^T \boldsymbol{B}(\overline{\boldsymbol{x}}_i, \boldsymbol{\Lambda}_{\boldsymbol{x}_i})\boldsymbol{\theta}}\right] = 0.$$

We then see that $\mathsf{Y}_{\boldsymbol{\eta}}$ given by

$$\mathsf{Y}_{\boldsymbol{\eta}} = \sum_{i=1}^{n} \frac{\boldsymbol{A}(\mathsf{x}_i) - \mathsf{c}(\boldsymbol{\theta})\boldsymbol{B}(\overline{\boldsymbol{x}}_i, \boldsymbol{\Lambda}_{\boldsymbol{x}_i})}{\boldsymbol{\theta}^T \boldsymbol{B}(\overline{\boldsymbol{x}}_i, \boldsymbol{\Lambda}_{\boldsymbol{x}_i})\boldsymbol{\theta}}$$

is unbiased, which justifies the design of $\boldsymbol{Y}_{\boldsymbol{\theta}}$.

Since, in view of (14), $c(\boldsymbol{\theta})$ is equal to 1 in the mean, the difference between

$$\sum_{i=1}^{n} \frac{\boldsymbol{\theta}^T \boldsymbol{A}_i \boldsymbol{\theta} - c(\boldsymbol{\theta})\boldsymbol{\theta}^T \boldsymbol{B}_i \boldsymbol{\theta}}{\boldsymbol{\theta}^T \boldsymbol{B}_i \boldsymbol{\theta}}$$

and

$$\sum_{i=1}^{n} \frac{\boldsymbol{\theta}^T \boldsymbol{A}_i \boldsymbol{\theta} - \boldsymbol{\theta}^T \boldsymbol{B}_i \boldsymbol{\theta}}{\boldsymbol{\theta}^T \boldsymbol{B}_i \boldsymbol{\theta}}$$

is blurred on average. Thus the refined renormalisation based on (12) and (13) is close in spirit to our original normalisation based on (11).

7 First-order renormalisation scheme

Having undertaken this process of unbiasing, we now specify a basic renormalisation method. The approach taken is simply to adopt the iteratively reweighted least squares scheme presented in Figure 1, but with the matrix $\boldsymbol{M}_{\boldsymbol{\theta}_{k-1}}$ replaced by $\boldsymbol{Y}_{\boldsymbol{\theta}_{k-1}}$, recalling that the latter may be expressed as

$$\boldsymbol{Y}_{\boldsymbol{\theta}_{k-1}} = \sum_{i=1}^{n} \frac{\boldsymbol{A}_i - c(\boldsymbol{\theta}_{k-1})\boldsymbol{B}_i}{{\boldsymbol{\theta}_{k-1}}^T \boldsymbol{B}_i \boldsymbol{\theta}_{k-1}}, \tag{15}$$

where

$$c(\boldsymbol{\theta}_{k-1}) = \frac{1}{n} \sum_{i=1}^{n} \frac{{\boldsymbol{\theta}_{k-1}}^T \boldsymbol{A}_i \boldsymbol{\theta}_{k-1}}{{\boldsymbol{\theta}_{k-1}}^T \boldsymbol{B}_i \boldsymbol{\theta}_{k-1}}. \tag{16}$$

It should be noted that several variations on this scheme may be proposed. Furthermore, some sophisticated *second-order* renormalisation schemes may be derived via more complicated choices for the $\boldsymbol{D}_i$ that arise in equation (11) [4].

Interestingly, while the renormalisation schemes generate a good approximation to $\widehat{\boldsymbol{\theta}}_{\mathrm{WLS}}$, the minimiser of J_{WLS}, they do not have $\widehat{\boldsymbol{\theta}}_{\mathrm{WLS}}$ as their theoretical limit, in contrast with the *fundamental numerical scheme* of Chojnacki et al [4]. However, an argument may be mounted that this is unimportant, given that $\widehat{\boldsymbol{\theta}}_{\mathrm{WLS}}$ is in itself a first-order approximation to $\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}$, the desired maximum likelihood minimiser, and that the various schemes approximate $\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}$ equally well in the first-order [1].

8 Experimental Results

The previously derived algorithms were tested on the conic fitting problem described above. Specifically, the fitting algorithms were applied to contaminated data arising from a portion of an ellipse. Our tests proceeded as follows. A randomly oriented ellipse was generated such that the ratio of its major to minor axes was in the range $[2, 3]$, and its major axis was approximately 200 pixels in length. One third of the ellipse's boundary was chosen as the base curve, and this included the point of maximum curvature of the ellipse. A set of *true points* was then randomly selected from a distribution uniform along the length of the base curve.

For each of the true points, a covariance matrix was randomly generated in accordance with some chosen average level of noise σ [4]. The true points were then perturbed randomly in accordance with their associated covariance matrices, yielding the *data points*. In general, the noise conformed to an inhomogeneous and anisotropic distribution. The error measure chosen represents the sum of the shortest distances of each *true point* from the estimated ellipse.

Figure 2 shows the average error obtained when each of the methods was applied to 50 sets of perturbed data for each of 10 different average levels of noise. Each set of perturbed data contained 60 points generated by the method described earlier. As would be expected, the results show that in the face of data contaminated with inhomogeneous and anisotropic noise, the OLS method, which does not make use of uncertainty (covariance) information, performs markedly worse than the other schemes. The IRWLS scheme lags a little behind the renormalisation scheme (Ren) with extent determined by problem sensitivity. These results were typical across extensive testing.

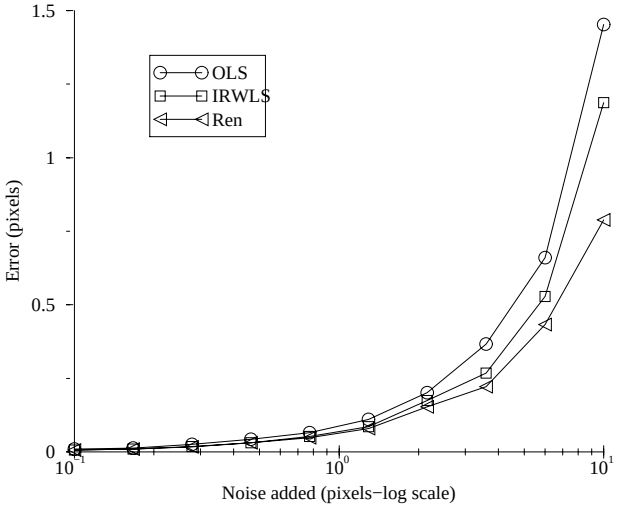

Figure 2: Error results for OLS, IRWLS and Ren.

Acknowledgements

The authors are grateful for the insightful comments of Darren Gawley, Marino Ivancic, Kenichi Kanatani, Garry Newsam and Robyn Owens. This work was in part funded by the Australian Research Council.

References

[1] *Personal communication*, K. Kanatani, May 2000.

[2] F. Bookstein, *Fitting conic sections to scattered data*, Computer Graphics and Image Processing **9** (1979), no. 1, 56–71.

[3] M. J. Brooks, W. Chojnacki, and L. Baumela, *Determining the egomotion of an uncalibrated camera from instantaneous optical flow*, Journal of the Optical Society of America A **14** (1997), no. 10, 2670–2677.

[4] W. Chojnacki, M. J. Brooks, and A. van den Hengel, *Fitting surfaces to data with covariance information: Fundamental methods applicable to computer vision*, Technical Report TR 99-03, Department of Computer Science, University of Adelaide, South Australia, August 1999, Available from `http://www.cs.adelaide.edu.au/~mjb/`.

[5] O. D. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, The MIT Press, Cambridge, Mass., 1993.

[6] K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice*, Elsevier, Amsterdam, 1996.

[7] P. Meer, *Robust techniques for computer vision*, Tutorial at the Fourteenth International Conference on Pattern Recognition, Brisbane, Australia, August 16-20, 1998.

[8] P. D. Sampson, *Fitting conics sections to 'very scattered' data: An iterative refinement of the Bookstein algorithm*, Computer Graphics and Image Processing **18** (1982), no. 1, 97–108.

[9] T. Viéville and O. D. Faugeras, *Motion analysis with a camera with unknown, and possibly varying intrinsic parameters*, Proceedings of the Fifth International Conference on Computer Vision (Cambridge, MA), IEEE Computer Society Press, Los Alamitos, CA, June 1995, pp. 750–756.

[10] Z. Zhang, *Parameter estimation techniques: a tutorial with application to conic fitting*, Image and Vision Computing **15** (1997), no. 1, 57–76.