Memoirs of the Faculty of Engineering, Okayama University, Vol. 38, Nos. 1 & 2, pp. 39–60, March 2004

Uncertainty Modeling and Geometric Inference

Kenichi KANATANI*
Department of Information Technology, Okayama University
Okayama 700-8530 Japan

(Received February 13, 2004)

We investigate the meaning of "statistical methods" for geometric inference based on image feature points. Tracing back the origin of feature uncertainty to image processing operations, we discuss the implications of asymptotic analysis in reference to "geometric fitting" and "geometric model selection". We point out that a correspondence exists between the standard statistical analysis and the geometric inference problem. We also compare the capability of the "geometric AIC" and the "geometric MDL" in detecting degeneracy. Next, we review recent progress in geometric fitting techniques for linear constraints, describing the "FNS method", the "HEIV method", the "renormalization method", and other related techniques. Finally, we discuss the "Neyman-Scott problem" and "semiparametric models" in relation to geometric inference. We conclude that applications of statistical methods requires careful considerations about the nature of the problem in question.

1. Introduction

Statistical inference from images is one of the key components of computer vision research today. Traditionally, statistical methods have been used for recognition and classification purposes. Recently, however, there are many studies of statistical analysis for *geometric inference* based on geometric primitives such as points and lines extracted by image processing operations.

However, the term "statistical" has somewhat a different meaning for such geometric inference problems than for the traditional recognition and classification purposes. This difference has often been overlooked, causing controversies over the validity of the statistical approach to geometric problems in general. In Sec. 2, we take a close look at this problem, tracing back the origin of feature uncertainty to image processing operations. In Sec. 3, we discuss the implications of asymptotic analysis in reference to "geometric fitting" and "geometric model selection". In Sec. 4, we point out that a correspondence exists between the standard statistical analysis and the geometric inference problem. We also compare the capability of the "geometric AIC" and the "geometric MDL" in detecting degeneracy. In Sec. 5, we review recent progress in geometric fitting techniques for linear constraints, describing the "FNS method", the "HEIV method", the "renormalization method", and other related techniques. In Sec. 6, we discuss the "Neyman-Scott problem" and "semiparametric models" in relation to geometric inference. Sec. 7 presents our concluding remarks. The derivation of the geometric AIC and the geometric MDL is summarized in the Appendix.

*E-mail kanatani@suri.it.okayama-u.ac.jp

2. What is Geometric Inference?

2.1 Ensembles for geometric inference

The goal of statistical methods is not to study the properties of observed data themselves but to infer the properties of the *ensemble* from which we regard the observed data as sampled. The ensemble may be a collection of existing entities (e.g., the entire population), but often it is a hypothetical set of conceivable possibilities. When a statistical method is employed, the underlying ensemble is often taken for granted. However, this issue is very crucial for geometric inference based on feature points.

Suppose, for example, we extract feature points, such as corners of walls and windows, from an image of a building and want to test if they are collinear. The reason why we need a statistical method is that the extracted feature positions have uncertainty. So, we have to judge the extracted feature points as collinear if they are sufficiently aligned. We can also evaluate the degree of uncertainty of the fitted line by propagating the uncertainty of the individual points. What is the ensemble that underlies this type of inference?

This question reduces to the question of why the

(a) (b)

Figure 1: (a) A feature point in an image of a building. (b) Its enlargement and the uncertainty of the feature location

uncertainty of the feature points occurs at all. After all, statistical methods are not necessary if the data are exact. Using a statistical method means regarding the current feature position as sampled from a set of its possible positions. But where else could it be if not in the current position?

2.2 Uncertainty of feature extraction

Many algorithms have been proposed for extracting feature points including the Harris operator [13] and SUSAN [49], and their performance has been extensively compared [4, 44, 48]. However, if we use, for example, the Harris operator to extract a particular corner of a particular building image, the output is unique (Fig. 1). No matter how many times we repeat the extraction, we obtain the same point because no external disturbances exist and the internal parameters (e.g., thresholds for judgment) are unchanged. It follows that the current position is the sole possibility. How can we find it elsewhere?

If we closely examine the situation, we are compelled to conclude that other possibilities should exist because the extracted position is not necessarily correct. But if it is not correct, why did we extract it? Why didn't we extract the correct position in the first place? The answer is: *we cannot*.

2.3 Image processing for computer vision

The reason why there exist so many feature extraction algorithms, none of them being definitive, is that they are aiming at an intrinsically impossible task. If we were to extract a point around which, say, the intensity varies to the largest degree in such and such a measure, the algorithm would be unique; variations may exist in intermediate steps, but the final output should be the same.

However, what we want is not "image properties" but "3-D properties" such as corners of a building, but the way a 3-D property is translated into an image property is intrinsically heuristic. As a result, as many algorithms can exist as the number of heuristics for its 2-D interpretation. If we specify a particular 3-D feature to extract, say a corner of a window, its appearance in the image is not unique. It is affected by many properties of the scene including the details of its 3-D shape, the viewing orientation, the illumination condition, and the light reflectance properties of the material. A slight variation of any of them can result in a substantial difference in the image.

Theoretically, exact extraction would be possible if all the properties of the scene were exactly known, but to infer them from images is the very task of computer vision. It follows that we must make a guess in the image processing stage. For the current image, some guesses may be correct, but others may be wrong. The exact feature position could be found only by an (non-existing) "ideal" algorithm that could guess everything correctly.

This observation allows us to interpret the "possible feature positions" to be *the positions that would be located by different (non-ideal) algorithms based on different guesses*. It follows that the set of hypothetical positions should be associated with *the set of hypothetical algorithms*. The current position is regarded as produced by an algorithm sampled from it. This explains why one always obtains the same position no matter how many times one repeats extraction using that algorithm. To obtain a different position, one has to sample another algorithm.

Remark 1 We may view the statistical ensemble in the following way. If we repeat the *same* experiment, the result should always be the same. But if we declare that the experiment is the "same" if such and such are the same while other things can vary; those variable conditions define the ensemble. The conventional view is to regard the experiment as the same if the *3-D scene* we are viewing is the same while other properties, such as the lighting condition, can vary. Then, the resulting image would be different for each (hypothetical) experiment, so one would obtain a different output each time, using the same image processing algorithm. The expected spread of the outputs measures the robustness of that algorithm.

Here, however, we are viewing the experiment as the same *if the image is the same*. Then, we could obtain different results only by sampling other algorithms. The expected spread of the outputs measures the uncertainty of feature detection from *that image*. We take this view, because we are analyzing the reliability of geometric inference from a particular image, while the conventional view is suitable for assessing the robustness of a *particular algorithm*.

2.4 Covariance matrix of a feature point

The performance of feature point extraction depends on the image properties around that point. If, for example, we want to extract a point in a region with an almost homogeneous intensity, the resulting position may be ambiguous whatever algorithm is used. In other words, the positions that potential algorithms would extract should have a large spread. If, on the other hand, the intensity greatly varies around that point, any algorithm could easily locate it accurately, meaning that the positions that the hypothet-

ical algorithms would extract should have a strong peak. It follows that we may introduce for each feature point its *covariance matrix* that measures the spread of its potential positions.

Let $V[p_\alpha]$ be the covariance matrix of the αth feature point p_α. The above argument implies that we can estimate the qualitative characteristics of uncertainty but not its absolute magnitude. So, we write the covariance matrix $V[p_\alpha]$ in the form

$$V[p_\alpha] = \varepsilon^2 V_0[p_\alpha], \tag{1}$$

where ε is an unknown magnitude of uncertainty, which we call the *noise level*. The matrix $V_0[p_\alpha]$, which we call the (*scale*) *normalized covariance matrix*, describes the relative magnitude and the dependence on orientations.

Remark 2 The decomposition of $V[p_\alpha]$ into ε^2 and $V_0[p_\alpha]$ involves scale ambiguity. In practice, this scale is implicitly determined by the image process operation for estimating the feature uncertainty applied to all the feature points in the same manner (see [29] for the details). The subsequent analysis does not depend on particular normalizations, slong as they are done in such a way that ε is much smaller than the data themselves.

2.5 Covariance matrix estimation

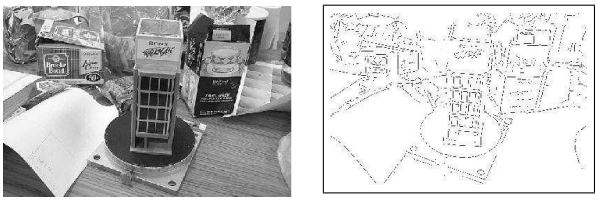
If the intensity variations around p_α are almost the same in all directions, we can think of the probability distribution as isotropic, a typical equiprobability line, known as the *uncertainty ellipses*, being a circle (Fig. 1(b)).

On the other hand, if p_α is on an object boundary, distinguishing it from nearby points should be difficult whatever algorithm is used, so its covariance matrix should have an elongated uncertainty ellipse along that boundary.

However, existing feature extraction algorithms are usually designed to output those points that have large image variations around them, so points in a region with an almost homogeneous intensity or on object boundaries are rarely chosen. As a result, the covariance matrix of a feature point extracted by such an algorithm can be regarded as nearly isotropic. This has also been confirmed by experiments [29], justifying the use of the identity as the normalized covariance matrix $V_0[p_\alpha]$.

Remark 3 The intensity variations around different feature points are usually unrelated, so their uncertainty can be regarded as statistically independent. However, if we track feature points over consecutive video frames, it has been observed that the uncertainty has strong correlations over the frames [50].

Remark 4 Many interactive applications require humans to extract feature points by manipulating a mouse. Extraction by a human is also an "algorithm", and it has been shown by experiments that humans are likely to choose "easy-to-see" points such as isolated points and intersections, avoiding points in a region with an almost homogeneous intensity or on object boundaries [29]. In this sense, the statistical characteristics of human extraction are very similar to machine extraction. This is no surprise if we recall that image processing for computer vision is essentially a heuristic that simulates human perception. It has also been reported that strong microscopic correlations exist when humans manually select corresponding feature points over multiple images [37].



(a) (b)

Figure 2: (a) An indoor scene. (b) Detected edges.

2.6 Image quality and uncertainty

The uncertainty of feature points has often been identified with "image noise", giving a misleading impression as if the feature locations were perturbed by random intensity fluctuations. Of course, we may obtain better results using higher-quality images whatever algorithm is used. However, the task of computer vision is not to analyze "image properties" but to study the "3-D properties" of the scene. As long as the image properties and the 3-D properties do not correspond one to one, any image processing inevitably entails some degree of uncertainty, however high the image quality may be, and the result must be interpreted statistically. The underlying ensemble is the set of hypothetical (inherently imperfect) algorithms of image processing. Yet, the performance of image processing algorithms has often been evaluated by adding *independent Gaussian noise* to individual pixels.

Remark 5 This also applies to *edge detection*, whose goal is to find the boundaries of 3-D objects in the scene. In reality, all existing algorithms seek *edges*, i.e., lines and curves across which the intensity changes discontinuously (Fig. 2). Yet, this is regarded by many as an objective image processing task, and the detection performance is often evaluated by adding independent Gaussian noise to individual pixels. From the above considerations, we conclude that edge detection is also a heuristic and hence no definitive algorithm will ever be found.

3. Asymptotic Analysis

3.1 What is asymptotic analysis?

As stated earlier, *statistical estimation* refers to estimating the properties of an ensemble from a finite

Figure 3: (a) For the standard statistical analysis, it is desired that the accuracy increases rapidly as the number of experiments $n \to \infty$, because admissible accuracy can be reached with a smaller number of experiments. (b) For geometric inference, it is desired that the accuracy increases rapidly as the noise level $\varepsilon \to 0$, because larger data uncertainty can be tolerated for admissible accuracy.

number of samples, assuming some knowledge, or a *model*, about the ensemble.

If the uncertainty originates from external conditions, as in experiments in physics, the estimation accuracy can be increased by controlling the measurement devices and environments. For internal uncertainty, on the other hand, there is no way of increasing the accuracy except by repeating the experiment and doing statistical inference. However, repeating experiments usually entails costs, and in practice the number of experiments is often limited.

Taking account of this, statisticians usually evaluate the performance of estimation *asymptotically*, analyzing the growth in accuracy as the number n of experiments increases. This is justified because a method whose accuracy increases more rapidly as $n \to \infty$ can reach admissible accuracy *with a fewer number of experiments* (Fig. 3(a)).

In contrast, the ensemble for geometric inference is, as we have seen, the set of potential feature positions that could be located if other (hypothetical) algorithms were used. As noted earlier, however, we can choose only *one* sample from the ensemble as long as we use a particular image processing algorithm. In other words, the number n of experiments is 1. Then, how can we evaluate the performance of statistical estimation?

Evidently, we want a method whose accuracy is sufficiently high *even for large data uncertainty*. This implies that we need to analyze the growth in accuracy as the noise level ε decreases, because a method whose accuracy increases more rapidly as $\varepsilon \to 0$ can tolerate larger data uncertainty for admissible accuracy (Fig. 3(b)).

3.2 Geometric fitting

We now illustrate the above consideration in more specific terms. Let $\{p_\alpha\}$, $\alpha = 1, ..., N$, be the extracted feature points. Suppose each point should satisfy a parameterized constraint

$$F(p_\alpha, \boldsymbol{u}) = 0 \tag{2}$$

when no uncertainty exists. In the presence of uncertainty, eq. (2) may not hold exactly. Our task is to estimate the parameter $\boldsymbol{u}$ from observed positions $\{p_\alpha\}$ in the presence of uncertainty.

A typical problem of this form is to fit a line or a curve to given N points in the image, but this can be straightforwardly extended to multiple images. For example, if a point (x_α, y_α) in one image corresponds to a point (x'_α, y'_α) in another, we can regard them as a single point p_α in a 4-dimensional joint space with coordinates $(x_\alpha, y_\alpha, x'_\alpha, y'_\alpha)$ (Fig. 4). If the camera imaging geometry is modeled as perspective projection, the constraint (2) corresponds to the *epipolar equation*; the parameter $\boldsymbol{u}$ is the *fundamental matrix* [14]. This will be discussed in more detail in Sec. 5.1.

General geometric fitting

The above problem can be stated in abstract terms as *geometric fitting* as follows. We view a feature point in the image plane or a set of feature points in the joint space as an m-dimensional vector $\boldsymbol{x}$; we call it a "datum". Let $\{\boldsymbol{x}_\alpha\}$, $\alpha = 1, ..., N$, be observed data. Their true values $\{\bar{\boldsymbol{x}}_\alpha\}$ are supposed to satisfy r constraint equations

$$F^{(k)}(\bar{\boldsymbol{x}}_\alpha, \boldsymbol{u}) = 0, \qquad k = 1, ..., r, \tag{3}$$

parameterized by a p-dimensional vector $\boldsymbol{u}$. We call eq. (3) the (*geometric*) *model*. The domain $\mathcal{X}$ of the data $\{\boldsymbol{x}_\alpha\}$ is called the *data space*; the domain $\mathcal{U}$ of the parameter $\boldsymbol{u}$ is called the *parameter space*. The number r of the constraint equations is called the *rank* of the constraint. The r equations $F^{(k)}(\boldsymbol{x}, \boldsymbol{u}) = 0$, $k = 1, ..., r$, are assumed to be mutually independent, defining a manifold $\mathcal{S}$ of codimension r parameterized by $\boldsymbol{u}$ in the data space $\mathcal{X}$. Eq. (3) requires that the true values $\{\bar{\boldsymbol{x}}_\alpha\}$ be all in the manifold $\mathcal{S}$. Our task is to estimate the parameter $\boldsymbol{u}$ from the noisy data $\{\boldsymbol{x}_\alpha\}$ (Fig. 5(a)).

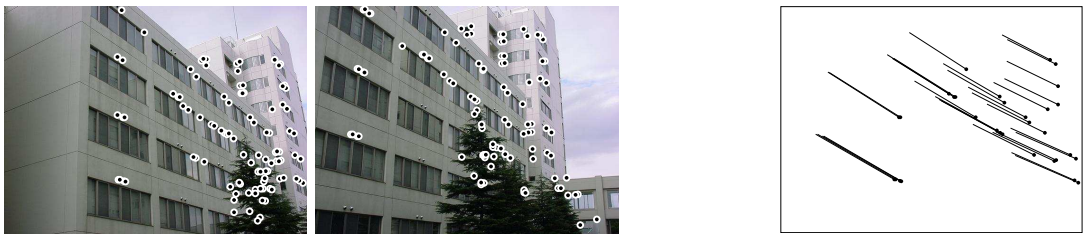
Maximum likelihood estimation

Let

$$V[\boldsymbol{x}_\alpha] = \varepsilon^2 V_0[\boldsymbol{x}_\alpha] \tag{4}$$

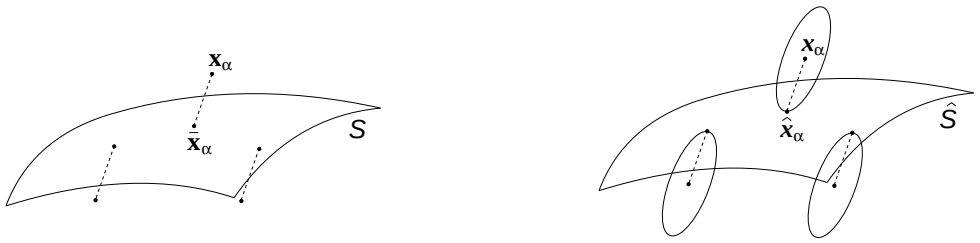
be the covariance matrix of $\boldsymbol{x}_\alpha$, where ε and $V_0[\boldsymbol{x}_\alpha]$ are the noise level and the normalized covariance matrix, respectively. If the distribution of uncertainty is Gaussian, which we assume hereafter, the probability density of the data $\{\boldsymbol{x}_\alpha\}$ is given by

$$P(\{\boldsymbol{x}_\alpha\}) = C \prod_{\alpha=1}^{N} e^{-(\boldsymbol{x}_\alpha - \bar{\boldsymbol{x}}_\alpha, V[\boldsymbol{x}_\alpha]^{-1}(\boldsymbol{x}_\alpha - \bar{\boldsymbol{x}}_\alpha))/2}, \tag{5}$$



(a) (b)

Figure 4: (a) Two images of a building and extracted feature points. (b) *Optical flow* consisting of segments connecting corresponding feature points (black dots correspond to the positions in the left image). The two endpoints can be identified with a point in a four-dimensional space.



(a) (b)

Figure 5: (a) Fitting a manifold $\mathcal{S}$ to the data $\{\boldsymbol{x}_\alpha\}$. (b) Estimating $\{\bar{\boldsymbol{x}}_\alpha\}$ and $\boldsymbol{u}$ by minimizing the sum of squared Mahalanobis distance with respect to the normalized covariance matrices $V_0[\boldsymbol{x}_\alpha]$.

where C is a normalization constant. Throughout this paper, we denote the inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ by $(\boldsymbol{a}, \boldsymbol{b})$.

Maximum likelihood (*ML*) *estimation* is to find the values of $\{\bar{\boldsymbol{x}}_\alpha\}$ and $\boldsymbol{u}$ that maximize the *likelihood*, i.e., eq. (6) into which the data $\{\boldsymbol{x}_\alpha\}$ are substituted, or equivalently minimize the sum of the squared *Mahalanobis distances* in the form

$$J = \sum_{\alpha=1}^{N} (\boldsymbol{x}_\alpha - \bar{\boldsymbol{x}}_\alpha, V_0[\boldsymbol{x}_\alpha]^{-1}(\boldsymbol{x}_\alpha - \bar{\boldsymbol{x}}_\alpha)) \qquad (6)$$

subject to the constraint (3) (Fig. 5(b)). The solution is called the *maximum likelihood* (*ML*) *estimator*. If the uncertainty is small, which we assume hereafter, the constraint (3) can be eliminated by introducing Lagrange multipliers and applying first order approximation. After some manipulations, we obtain the following form [15]:

$$J = \sum_{\alpha=1}^{N} \sum_{k,l=1}^{r} W_\alpha^{(kl)} F^{(k)}(\boldsymbol{x}_\alpha, \boldsymbol{u}) F^{(l)}(\boldsymbol{x}_\alpha, \boldsymbol{u}). \qquad (7)$$

Here, $W_\alpha^{(kl)}$ is the (kl) element of the inverse of the $r \times r$ matrix whose (kl) element is $(\nabla_{\mathbf{x}} F_\alpha^{(k)}, V_0[\boldsymbol{x}_\alpha] \nabla_{\mathbf{x}} F_\alpha^{(l)})$; we symbolically write

$$\left(W_\alpha^{(kl)}\right) = \left((\nabla_{\mathbf{x}} F_\alpha^{(k)}, V_0[\boldsymbol{x}_\alpha] \nabla_{\mathbf{x}} F_\alpha^{(l)})\right)^{-1}, \qquad (8)$$

where $\nabla_{\mathbf{x}} F^{(k)}$ is the gradient of the function $F^{(k)}$ with respect to $\boldsymbol{x}$. The subscript α means that $\boldsymbol{x} = \boldsymbol{x}_\alpha$ is substituted.

Remark 6 The data $\{\boldsymbol{x}_\alpha\}$ may be subject to some constraints. For example, each $\boldsymbol{x}_\alpha$ may be a unit vector. The above formulation still holds if the inverse $V_0[\boldsymbol{x}_\alpha]^{-1}$ in eq. (6) is replaced by the (Moore-Penrose) generalized (or pseudo) inverse $V_0[\boldsymbol{x}_\alpha]^-$ [15].

Similarly, the r constraints in eq. (3) may be redundant, say only r' $(< r)$ of them are independent. The above formulation still holds if the inverse in eq. (8) is replaced by the generalized inverse of rank r' with all but r' largest eigenvalues are replaced by zero [15].

Accuracy of the ML estimator

It can be shown [15] that the covariance matrix of the ML estimator $\hat{\boldsymbol{u}}$ has the form

$$V[\hat{\boldsymbol{u}}] = \varepsilon^2 \boldsymbol{M}(\hat{\boldsymbol{u}})^{-1} + O(\varepsilon^4), \qquad (9)$$

where

$$\boldsymbol{M}(\boldsymbol{u}) = \sum_{\alpha=1}^{N} \sum_{k,l=1}^{r} W_\alpha^{(kl)} \nabla_{\mathbf{u}} F_\alpha^{(k)} \nabla_{\mathbf{u}} F_\alpha^{(k)\top}. \qquad (10)$$

Here, $\nabla_{\mathbf{u}} F^{(k)}$ is the gradient of the function $F^{(k)}$ with respect to $\boldsymbol{u}$. The subscript α means that $\boldsymbol{x} = \boldsymbol{x}_\alpha$ is substituted.

Remark 7 It can be proved that no other estimators could reduce the covariance matrix further than eq. (9) except for the higher order term $O(\varepsilon^4)$ [15, 18]. The ML estimator is optimal in this sense. Recall that we are focusing on the asymptotic analysis for $\varepsilon \to 0$. Thus, what we call the "ML estimator" should be understood to be a first approximation to the true ML estimator for small ε.

Remark 8 The p-dimensional parameter vector $\boldsymbol{u}$ may be constrained. For example, it may be a unit vector. If it has only p' ($< p$) degrees of freedom, the parameter space $\mathcal{U}$ is a p'-dimensional manifold in $\mathcal{R}^p$. In this case, the matrix $\boldsymbol{M}(\boldsymbol{u})$ in eq. (9) is replaced by $\boldsymbol{P}_{\mathbf{u}}\boldsymbol{M}(\boldsymbol{u})\boldsymbol{P}_{\mathbf{u}}$, where $\boldsymbol{P}_{\mathbf{u}}$ is the projection matrix onto the tangent space to the parameter space $\mathcal{U}$ at $\boldsymbol{u}$ [15]. The inverse $\boldsymbol{M}(\hat{\boldsymbol{u}})^{-1}$ in eq. (9) is replaced by the generalized inverse $\boldsymbol{M}(\hat{\boldsymbol{u}})^{-1}$ of rank p' [15].

3.3 Geometric model selection

Geometric fitting is to estimate the parameter $\boldsymbol{u}$ of a a given model. If we have multiple candidate models

$$F_1^{(k)}(\bar{\boldsymbol{x}}_\alpha, \boldsymbol{u}_1) = 0, \quad F_2^{(k)}(\bar{\boldsymbol{x}}_\alpha, \boldsymbol{u}_2) = 0, \quad ..., \qquad (11)$$

from which we are to select an appropriate one for the observed data $\{\boldsymbol{x}_\alpha\}$, the problem is (*geometric*) *model selection* [15, 17, 19].

Suppose, for example, we want to fit a curve to given points in two dimensions. If they are almost collinear, a straight line may fit fairly well, but a quadratic curve may fit better, and a cubic curve even better. Which curve should we fit? A naive idea is to compare the *residual* (*sum of squares*), i.e., the minimum value $\hat{J}$ of J in eq. (6); we select the one that has the smallest residual $\hat{J}$. This does not work, however, because the ML estimator $\hat{\boldsymbol{u}}$ is so determined as to minimize the residual $\hat{J}$, and the residual $\hat{J}$ can be made arbitrarily smaller if the model is equipped with more parameters to adjust. So, the only conclusion would be to fit a curve of a sufficiently high degree passing through all the points.

Geometric AIC

The above observation leads to the idea of compensating for the negative bias of the residual caused by substituting the ML estimator. This is the principle of Akaike's *AIC* (*Akaike information criterion*) [1], which is derived from the asymptotic behavior of the *Kullback-Leibler information* (or *divergence*) as the number n of experiments goes to infinity. Doing a similar analysis to Akaike's and examining the asymptotic behavior as the noise level ε goes to zero, we can obtain the following *geometric AIC* [15, 16] (see Appendix A for the derivation):

$$\text{G-AIC} = \hat{J} + 2(Nd + p)\varepsilon^2 + O(\varepsilon^4). \qquad (12)$$

Here, d is the dimension of the manifold $\mathcal{S}$ defined by the constraint (3) in the data space $\mathcal{X}$, and p is the dimension of $\boldsymbol{u}$ (i.e., the number of unknowns). The model for which eq. (12) is the smallest is regarded as the best. The derivation of eq. (12) is based on the following facts [15, 16] (see Appendix A for the details):

- The ML estimator $\hat{\boldsymbol{u}}$ converges to its true value as $\varepsilon \to 0$.
- The ML estimator $\hat{\boldsymbol{u}}$ obeys a Gaussian distribution under linear constraints, because the noise is assumed to be Gaussian. For nonlinear constraints, linear approximation can be justified in the neighborhood of the solution if ε is sufficiently small.
- A quadratic form in standardized Gaussian random variables is subject to a χ^2 distribution, whose expectation is equal to its degree of freedom.

Geometric MDL

Another well known criterion for model selection is Rissanen's *MDL* (*Minimum description length*) [45, 46, 47], which measures the goodness of a model by the minimum information theoretic code length of the data and the model. The basic idea is simple, but the following difficulties must be resolved for applying it in practice:

- Encoding a problem involving real numbers requires an infinitely long code length.
- The probability density, from which a minimum length code can be obtained, involves unknown parameters.
- The exact form of the minimum code length is very difficult to compute.

Rissanen [45, 46, 47] avoided these difficulties by quantizing the real numbers in a way that does not depend on individual models and substituting the ML estimators for the parameters. They, too, are real numbers, so they are also quantized. The quantization width is so chosen as to minimize the total description length (the *two-stage encoding*). The resulting code length is evaluated asymptotically as the data length n goes to infinity. If we analyzes the asymptotic behavior of encoding the geometric fitting problem as the noise level ε goes to zero, we obtain the following *geometric MDL* [21] (see Appendix B for the derivation):

$$\text{G-MDL} = \hat{J} - (Nd + p)\varepsilon^2 \log\Big(\frac{\varepsilon}{L}\Big)^2 + O(\varepsilon^2). \qquad (13)$$

Here, L is a reference length chosen so that its ratio to the magnitude of data is $O(1)$, e.g., L can be taken to be the image size for feature point data. Its exact determination requires an a priori distribution that specifies where the data are likely to appear (we will discuss this more in Sec. 4.1), but it has been observed that the model selection is not very much affected by L as long as it is within the same order of magnitude [21] (see Appendix B for the details):

4. Standard vs. Geometric Analysis

We now point out that a correspondence exists between the standard statistical analysis and the geometric inference problem. We also compare the capa-

bility of the geometric AIC and the geometric MDL in detecting degeneracy.

4.1 Standard statistical analysis

The asymptotic analysis in Sec. 3 bears a strong resemblance to the standard statistical estimation problem: after observing n data $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, ..., $\boldsymbol{x}_n$, we want to estimate the parameter $\boldsymbol{\theta}$ of the probability density $P(\boldsymbol{x}|\boldsymbol{\theta})$ called the (*stochastic*) *model*, according to which each datum is assumed to be sampled independently.

Maximum likelihood (ML) estimation is to find the value $\boldsymbol{\theta}$ that maximizes $\prod_{i=1}^n P(\boldsymbol{x}_i|\boldsymbol{\theta})$, or equivalently minimizes its negative logarithm $-\sum_{i=1}^n \log P(\boldsymbol{x}_i|\boldsymbol{\theta})$. It can be shown that the covariance matrix $V[\hat{\boldsymbol{\theta}}]$ of the resulting ML estimator $\hat{\boldsymbol{\theta}}$ converges, under a mild condition, to $\boldsymbol{O}$ as the number n of experiments goes to infinity (*consistency*) in the form

$$V[\hat{\boldsymbol{\theta}}] = \boldsymbol{I}(\boldsymbol{\theta})^{-1} + O\Big(\frac{1}{n^2}\Big), \tag{14}$$

where we define the *Fisher information matrix* $\boldsymbol{I}(\boldsymbol{\theta})$ by

$$\boldsymbol{I}(\boldsymbol{\theta}) = n\mathrm{E}[(\nabla_\theta \log P(\boldsymbol{x}|\boldsymbol{\theta}))(\nabla_\theta \log P(\boldsymbol{x}|\boldsymbol{\theta}))^\top]. \tag{15}$$

The operation $E[\,\cdot\,]$ denotes expectation with respect to the density $P(\boldsymbol{x}|\boldsymbol{\theta})$. The first term in the right-hand side of eq. (14) is called the *Cramer-Rao lower bound* (*CRLB*), describing the minimum degree of fluctuations in all estimators. Thus, the ML estimator is optimal if n is sufficiently large (*asymptotic efficiency*).

If we have multiple candidate models

$$P_1(\boldsymbol{x}|\boldsymbol{\theta}_1), \quad P_2(\boldsymbol{x}|\boldsymbol{\theta}_2), \quad P_3(\boldsymbol{x}|\boldsymbol{\theta}_3), \quad ..., \tag{16}$$

from which we are to select an appropriate one for the observations $\boldsymbol{x}_1$, $\boldsymbol{x}_1$, ..., $\boldsymbol{x}_n$, the problem is (*stochastic*) *model selection*. Akaike's AIC has the following form:

$$\mathrm{AIC} = -2\sum_{i=1}^N \log P(\boldsymbol{x}_i|\hat{\boldsymbol{\theta}}) + 2k + O\Big(\frac{1}{n}\Big). \tag{17}$$

The model for which this quantity is the smallest is regarded as the best. The derivation of eq. (17) is based on the following facts [1]:

- The maximum likelihood estimator $\hat{\boldsymbol{\theta}}$ converges to its true value as $n \to \infty$ (the *law of large numbers*).
- The maximum likelihood estimator $\hat{\boldsymbol{\theta}}$ asymptotically obeys a Gaussian distribution as $n \to \infty$ (the *central limit theorem*).
- A quadratic form in standardized Gaussian random variables is subject to a χ^2 distribution, whose expectation is equal to its degree of freedom.

The Rissanen's MDL has the following form [46, 47]:

$$\mathrm{MDL} = -\sum_{i=1}^n \log P(\boldsymbol{x}_i|\hat{\boldsymbol{\theta}}) + \frac{k}{2}\log\frac{n}{2\pi} + \log\int_{\mathcal{T}} \sqrt{|\boldsymbol{I}(\boldsymbol{\theta})|}d\boldsymbol{\theta} + O(1). \tag{18}$$

Here, $\hat{\boldsymbol{\theta}}$ is the ML estimator; the symbol $O(1)$ denotes terms of order 0 in n in the limit $n \to \infty$. In order that the integration in the right-hand side of eq. (18) exists, the domain $\mathcal{T}$ of the parameter $\boldsymbol{\theta}$ must be compact. In other words, we must specify in the k-dimensional space of $\boldsymbol{\theta}$ a finite region $\mathcal{T}$ in which the true value of $\boldsymbol{\theta}$ is likely to exist. This is nothing but the *Bayesian* standpoint that requires a prior distribution for the parameter to estimate. If it is not known, we must introduce an appropriate expedient to suppress an explicit dependence on the prior. Such an expedient is also necessary for the geometric MDL, i.e., the introduction of the reference length L in eq. (18).

4.2 Dual interpretations

We have seen that the limit $n \to \infty$ for the standard statistical analysis corresponds to the limit $\varepsilon \to 0$ for geometric inference. For example, the covariance matrix of the ML estimator agrees with the Cramer-Rao lower bound up to $O(1/n^2)$ for $n \to \infty$ (see eq. (14)), while for geometric inference it agrees with the lower bound bound up to $O(\varepsilon^4)$ for $\varepsilon \to 0$ (see eq. (9)). If follows that $1/\sqrt{n}$ for the standard statistical analysis plays the same role as ε for geometric inference.

The same correspondence exists for model selection, too. The unknowns for geometric inference are the p parameters of the constraint plus the N true positions specified by the d coordinates of the d-dimensional manifold $\mathcal{S}$ defined by the constraint. If eq. (12) is divided by ε^2, we have $\hat{J}/\varepsilon^2 + 2(Nd + p) + O(\varepsilon^2)$, which is ($-2$ times the logarithmic likelihood)+2(the number of unknowns), the same form as Akaike's AIC (17). The same holds for eq. (13), which corresponds to Rissanen's MDL (18) if ε is replaced by $1/\sqrt{n}$ [21].

This correspondence can be interpreted as follows. Since the underlying ensemble is hypothetical, we can actually observe only one sample as long as a particular algorithm is used. Suppose we hypothetically sample n different algorithms to find n different positions. The optimal estimate of the true position under the Gaussian model is their sample mean. The covariance matrix of the sample mean is $1/n$ times that of the individual samples. Hence, this hypothetical estimation is equivalent to dividing the noise level ε in eq. (4) by $\sqrt{n}$.

In fact, there were attempts to generate a hypothetical *ensemble of algorithms* by randomly varying

the internal parameters (e.g., the thresholds for judgments), not adding random noise to the image [5, 6]. Then, one can compute their means and covariance matrix. Such a process as a whole can be regarded as one operation that effectively achieves higher accuracy.

Thus, the asymptotic analysis for $\varepsilon \to 0$ is equivalent to the asymptotic analysis for $n \to \infty$, where n is the number of hypothetical observations. As a result, the expression $\cdots + O(1/\sqrt{n^k})$ in the standard statistical analysis turns into $\cdots + O(\varepsilon^k)$ in geometric inference.

4.3 Noise level estimation

In order to use the geometric AIC or the geometric MDL, we need to know the noise level ε. If not known, it must be estimated. Here arises a sharp contrast between the standard statistical analysis and our geometric inference.

For the standard statistical analysis, the noise magnitude is a *model parameter*, because "noise" is defined to be *the random effects that cannot be accounted for by the assumed model*. Hence, the noise magnitude should be estimated, if not known, *according to the assumed model*. For geometric inference, on the other hand, the noise level ε is *a constant that reflects the uncertainty of feature detection*. So, it should be estimated *independently of individual models*.

If we know the true model, it can be estimated from the residual $\hat{J}$ using the knowledge that $\hat{J}/\varepsilon^2$ is subject to a χ^2 distribution with $rN - p$ degrees of freedom in the first order [15]. Specifically, we obtain an unbiased estimator of ε^2 in the form

$$\hat{\varepsilon}^2 = \frac{\hat{J}}{rN - p}. \tag{19}$$

The validity of this formula has been confirmed by many simulations.

One may wonder if model selection is necessary at all when the true model is known. In practice, however, a typical situation where model selection is called for is *degeneracy detection*. In 3-D analysis from images, for example, the constraint (3) corresponds to our knowledge about the scene such as rigidity of motion. However, the computation fails if degeneracy occurs (e.g., the motion is zero). Even if exact degeneracy does not occur, the computation may become numerically unstable in near degeneracy conditions. In such a case, the computation can be stabilized by switching to a model that describes the degeneracy [17, 22, 27, 28, 34, 42, 56].

Degeneracy means *addition* of new constraints, such as some quantity being zero. It follows that the manifold $\mathcal{S}$ degenerates into a submanifold $\mathcal{S}'$ of it. Since the general model still holds irrespective of the degeneracy, i.e. $\mathcal{S}' \subset \mathcal{S}$, we can estimate the noise level ε from the residual $\hat{J}$ of the general model $\mathcal{S}$, which we know is true, using eq. (19).

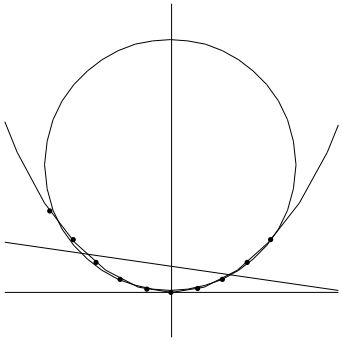


Figure 6: Fitting a line, a circle, and an ellipse.

Remark 9 Eq. (19) can be intuitively understood as follows. Recall that $\hat{J}$ is the sum of the square distances from $\{\boldsymbol{x}_\alpha\}$ to the manifold $\hat{\mathcal{S}}$ defined by the constraint $F^{(k)}(\boldsymbol{x}, \boldsymbol{u}) = 0$, $k = 1, ..., r$. Since $\hat{\mathcal{S}}$ has codimension r (the dimension of the orthogonal directions to it), the residual $\hat{J}$ should have expectation $rN\varepsilon^2$. However, $\hat{\mathcal{S}}$ is fitted by adjusting its p-dimensional parameter $\boldsymbol{u}$, so the expectation of $\hat{J}$ reduces to $(rN - p)\varepsilon^2$.

Note that we need more than $\lfloor p/r \rfloor$ data for this estimation. For example, if we know that the true model is a planar surface, we need to observe more than three points for degeneracy detection.

Remark 10 It may appear that the residual $\hat{J}$ of the general model cannot be stably computed in the presence of degeneracy. However, what is unstable is *model specification*, not the residual. For example, if we fit a planar surface to almost collinear points in 3-D, it is difficult to specify the fitted plane stably; the solution is very susceptible to noise. Yet, the residual is stably computed, since unique specification of the fit is difficult *because all the candidates have almost the same residual*.

Note that the noise level estimation from the general model $\mathcal{S}$ by eq. (19) is still valid even if degeneracy occurs, because degeneracy means shrinkage of the model manifold $\mathcal{S}'$ *within* $\mathcal{S}$, which does not affect the data deviations in the "orthogonal" directions (in the Mahalanobis sense) to $\mathcal{S}$ that account for the residual $\hat{J}$.

4.4 Comparing the geometric AIC/MDL

We now illustrate the different characteristics of the geometric AIC and the geometric MDL in detecting degeneracy.

Detection of Circles and Lines

Consider an ellipse that is tangent to the x-axis at the origin O with the minor radius 50 in the y direction and eccentricity $1/\beta$. On it, we take eleven points with equally spaced x coordinates. Adding Gaussian noise of mean 0 and variance ε^2 to the x and y coordinates of each point independently, we fit an ellipse,

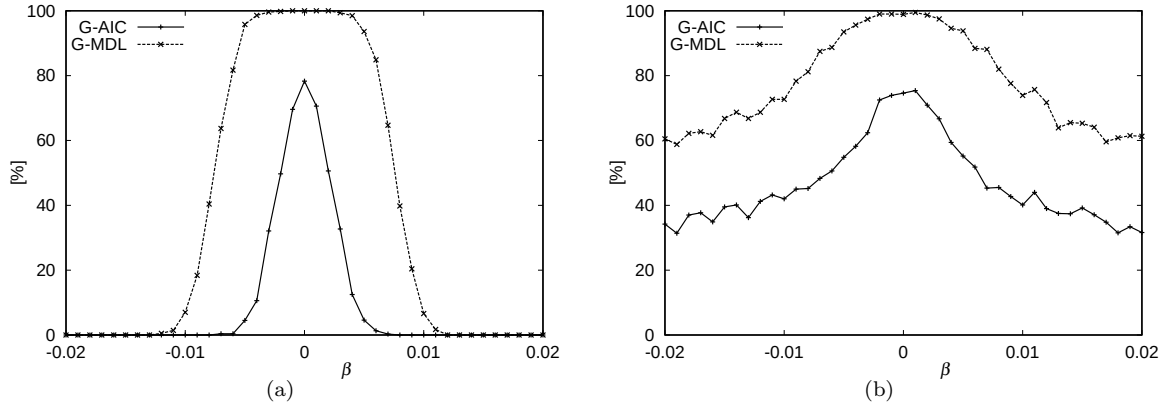


Figure 7: The ratio (%) of detecting a line by the geometric AIC (solid lines with +) and the geometric MDL (dotted lines with ×) using (a) the true noise level and (b) the estimated noise level.

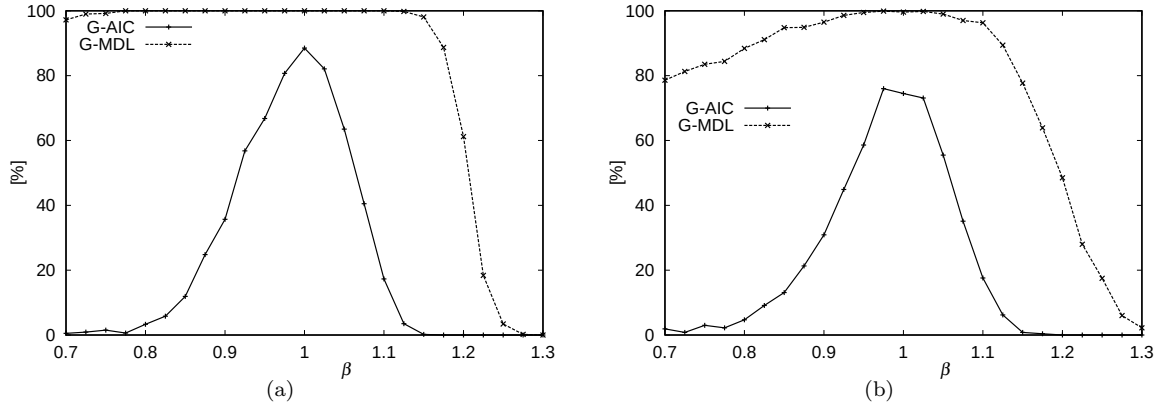


Figure 8: The ratio (%) of detecting a circle by the geometric AIC (solid lines with +) and the geometric MDL (dotted lines with ×) using (a) the true noise level and (b) the estimated noise level.

a circle, and a line in a statistically optimal manner [25, 26], using a technique called *renormalization*[1] [15] (we will discuss this in Sec. 5.6). Fig. 6 shows one instance for $\beta = 2.5$ and $\varepsilon = 0.1$. Note that a line and a circle are degeneracies of an ellipse.

Lines, circles, and ellipses define 1-dimensional (geometric) models with 2, 3, and 5 degrees of freedom, respectively. Their geometric AIC and the geometric MDL for N points are

$$\begin{aligned}
\text{G-AIC}_l &= \hat{J}_l + 2(N+2)\varepsilon^2, \\
\text{G-AIC}_c &= \hat{J}_c + 2(N+3)\varepsilon^2, \\
\text{G-AIC}_e &= \hat{J}_e + 2(N+5)\varepsilon^2, \\
\text{G-MDL}_l &= \hat{J}_l - (N+2)\varepsilon^2 \log\Big(\frac{\varepsilon}{L}\Big)^2, \\
\text{G-MDL}_c &= \hat{J}_c - (N+3)\varepsilon^2 \log\Big(\frac{\varepsilon}{L}\Big)^2, \\
\text{G-MDL}_e &= \hat{J}_e - (N+5)\varepsilon^2 \log\Big(\frac{\varepsilon}{L}\Big)^2, \qquad (20)
\end{aligned}$$

where the subscripts l, c, and e refer to lines, circles, and ellipses, respectively. For each β, we compute the geometric AIC and the geometric MDL of the fitted line, circle, and ellipse and choose the one that has the smallest value. We used the reference length $L = 1$.

Fig. 7(a) shows the percentage of choosing a line for $\varepsilon = 0.01$ after 1000 independent trials for each β. If there were no noise, it should be 0% for $\beta \neq 0$ and 100% for $\beta = 0$. In the presence of noise, the geometric AIC produces a sharp peak, indicating a high capability of distinguishing a line from an ellipse. However, it judges a line to be an ellipse with some probability. The geometric MDL judges a line to be a line almost 100%, but it judges an ellipse to be a line over a wide range of β.

In Fig. 7(a), we used the true value of ε^2. If it is unknown, it can be estimated from the residual of the general ellipse model by eq. (19). Fig. 7(b) shows the result using its estimate. Although the sharpness is somewhat lost, similar performance characteristics are observed.

Fig. 8 shows the percentage of choosing a circle for $\varepsilon = 0.01$. If there were no noise, it should be 0% for $\beta \neq 1$ and 100% for $\beta = 1$. In the presence of noise, as we see, it is difficult to distinguish a circular arc from

[1] The program is available at:
`http://www.suri.it.okayama-u.ac.jp/e-program.html`

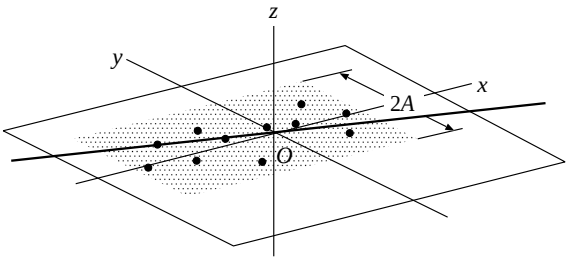


Figure 9: Fitting a space line and a plane to points in space.

an elliptic arc for $\beta < 1$. Yet, the geometric AIC can detect a circle very sharply, although it judges a circle to be an ellipse with some probability. In contrast, the geometric MDL almost always judges an ellipse to be a circle for $\beta < 1.1$.

Detection of Space Lines

Consider a rectangular region $[0, 10] \times [-1, 1]$ on the xy plane in the xyz space. We randomly take eleven points in it and magnify the region A times in the y direction. Adding Gaussian noise of mean 0 and variance ε^2 to the x, y, and z coordinates of each point independently, we fit a space line and a plane in a statistically optimal manner (Fig. 9). The rectangular region degenerates into a line segment as $A \to 0$.

A space line is a 1-dimensional model with four degrees of freedom; a plane is a 2-dimensional model with three degrees of freedom. Their geometric AIC and geometric MDL are

$$\begin{aligned} \text{G-AIC}_l &= \hat{J}_l + 2(N+4)\varepsilon^2, \\ \text{G-AIC}_p &= \hat{J}_p + 2(2N+3)\varepsilon^2, \\ \text{G-MDL}_l &= \hat{J}_l - (N+4)\varepsilon^2 \log\Big(\frac{\varepsilon}{L}\Big)^2, \\ \text{G-MDL}_p &= \hat{J}_p - (2N+3)\varepsilon^2 \log\Big(\frac{\varepsilon}{L}\Big)^2, \end{aligned} \quad (21)$$

where the subscripts l and p refer to lines and planes, respectively. For each A, we compare the geometric AIC and the geometric MDL of the fitted line and plane and choose the one that has the smaller value. We used the reference length $L = 1$.

Fig. 10(a) shows the percentage of choosing a line for $\varepsilon = 0.01$ after 1000 independent trials for each A. If there were no noise, it should be 0% for $A \neq 0$ and 100% for $A = 0$. In the presence of noise, the geometric AIC has a high capability of distinguishing a line from a plane, but it judges a line to be a plane with some probability. In contrast, the geometric MDL judges a line to be a line almost 100%, but it judges a plane to be a line over a wide range of A.

In Fig. 10(a), we used the true value of ε^2. Fig. 10(b) shows the corresponding result using its estimate obtained from the general plane model by eq. (19). We observe somewhat degraded but similar performance characteristics.

Observations

We can observe from the above examples that the geometric AIC has a higher capability for detecting degeneracy than the geometric MDL, but the general model is chosen with some probability when the true model is degenerate. In contrast, the percentage for the geometric MDL to detect degeneracy when the true model is really degenerate approaches 100% as the noise decreases. This is exactly the dual statement to the well known fact, called the *consistency of the MDL*, that the percentage for Rissanen's MDL to identify the true model converges to 100% in the limit of an infinite number of observations. Rissanen's MDL is regarded by many as superior to Akaike's AIC because the latter lacks this property.

At the cost of this consistency, however, the geometric MDL regards a wide range of nondegenerate models as degenerate. This is no surprise, since the penalty $-(Nd+p)\varepsilon^2 \log(\varepsilon/L)^2$ for the geometric MDL in eq. (13) is heavier than the penalty $2(Nd+p)\varepsilon^2$ for the geometric AIC in eq. (12). As a result, the geometric AIC is more faithful to the data than the geometric MDL, which is more likely to choose a degenerate model. This contrast has also been observed in many applications [34, 24].

Remark 11 Despite the fundamental difference of geometric model selection from the standard (stochastic) model selection, many attempts have been made in the past to apply Akaike's AIC and their variants to computer vision problems based on the asymptotic analysis of $n \to \infty$, where the interpretation of n is different from problem to problem [51, 52, 53, 54, 55]. Rissanen's MDL is also used in computer vision applications. Its use may be justified if the problem has the standard form of linear/nonlinear regression [3, 35]. Often, however, the solution having a shorter description length was chosen with a rather arbitrary definition of the complexity [12, 30, 36].

Remark 12 One may wonder why we are forced to choose one from the two asymptotic analyses, $n \to \infty$ or $\varepsilon \to 0$. Why don't we use the general form of the AIC or the MDL rather than worrying about their asymptotic expressions? The answer is that we *cannot*.

The starting principle of the AIC is the Kullback-Leibler distance of the assumed probability distribution from the true distribution. We cannot compute it exactly, because we do not know the true distribution. So, Akaike approximated it, invoking the law of large numbers and the central limit theorem, thereby estimating the true distribution from a large number of observations, while the geometric AIC is obtained by assuming that the noise is very small, thereby identifying the data as their true values to a first approximation.

Similarly, the exactly shortest code length is difficult to compute if real numbers are involved, so Ris-

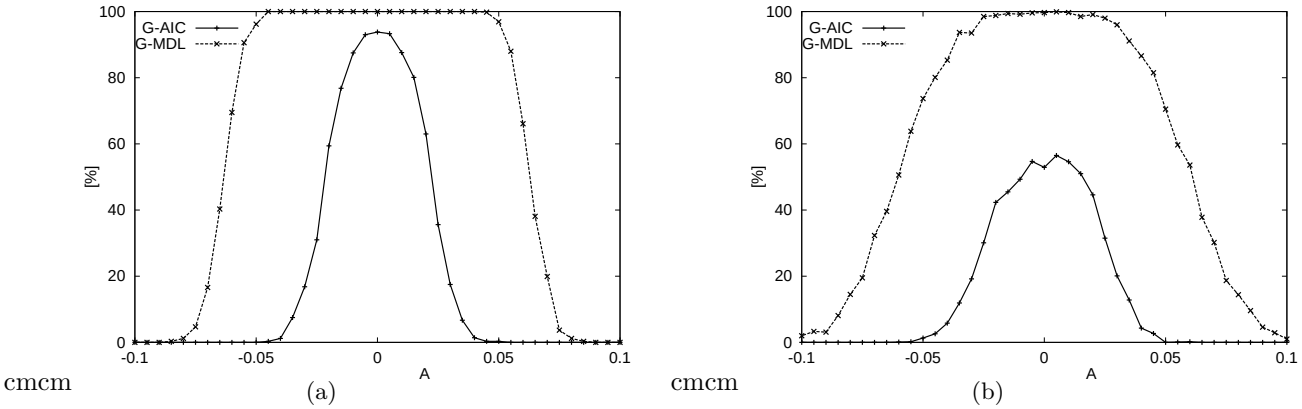


Figure 10: The rate (%) of detecting a space line by the geometric AIC (solid lines) and the geometric MDL (dashed lines) with (a) the true noise level and (b) the estimated noise level.

sanen approximated it by omitting higher order terms in the data length n. The geometric MDL is obtained by omitting higher order terms in the noise level ε.

Thus, analysis of asymptotic expressions in one form or another is inevitable if the principle of the AIC or the MDL is to be applied in practice.

Remark 13 Note that one cannot say one model selection criteria is superior to another, because each is based on its own logic. Also, if we want to compare the performance of two criteria in practice, we must formulate them in such a way that they conform to a common assumption. In this sense, one cannot compare Akaike's AIC with the geometric AIC or Rissanen's MDL with the geometric MDL, because the underlying asymptotic limits are different. Similarly, if we want to compare the geometric AIC or the geometric MDL with other existing criteria, e.g., Schwarz' BIC, derived in the asymptotic limit $n \to \infty$, they must be formulated in the asymptotic limit $\varepsilon \to 0$.

Note also that one cannot prove that a particular criterion works at all. In fact, although Akaike's AIC and Rissanen's MDL are based on rigorous mathematics, there is no guarantee that they work well in practice. The mathematical rigor is in their *reduction* from their starting principles (the Kullback-Leibler distance and the minimum description length principle), which are beyond proof. What one can tell is which criterion is more suitable for a particular application when used in a particular manner. The geometric AIC and the geometric MDL have shown to be effective in many computer vision applications [20, 23, 24, 27, 28, 34, 42, 56], but other criteria may be better in other applications.

5. Linear Geometric Fitting

Now, we consider a special type of geometric fitting problem that most frequently arises in computer vision applications: the constraint is linear in both data and unknowns. We systematically review existing methods.

5.1 Linear constraints

In many geometric inference problems of computer vision, the constraint (3) has the form

$$(\boldsymbol{\xi}(\bar{\boldsymbol{x}}_\alpha), \boldsymbol{u}) = 0, \tag{22}$$

where $\boldsymbol{\xi}(\,\cdot\,)$ is generally a nonlinear mapping from an m-dimensional vector to a p-dimensional vector. Evidently, the magnitude of $\boldsymbol{u}$ is unconstrained, so we normalize it to a unit vector: $\|\boldsymbol{u}\| = 1$.

Example 1 Suppose we are given N points $\{(x_\alpha, y_\alpha)\}$, $\alpha = 1, ..., N$, in two dimensions. Their true positions $\{(\bar{x}_\alpha, \bar{y}_\alpha)\}$ are assumed to be on a *conic* (a circle, an ellipse, a parabola, a hyperbola, or their degeneracy). Our task is to estimate the curve from the noisy data $\{(x_\alpha, y_\alpha)\}$ (see Fig. 6). The constraint on $\{(\bar{x}_\alpha, \bar{y}_\alpha)\}$ is

$$A\bar{x}_\alpha^2 + 2B\bar{x}_\alpha\bar{y}_\alpha + C\bar{y}_\alpha^2 + 2(D\bar{x}_\alpha + E\bar{y}_\alpha) + F = 0 \tag{23}$$

for some coefficients A, B, ..., D, not all being zero. This constraint reduces to eq. (22) if we put

$$\boldsymbol{\xi}(x, y) = \begin{pmatrix} x^2 & 2xy & y^2 & 2x & 2y & 1 \end{pmatrix}^\top,$$
$$\boldsymbol{u} = \begin{pmatrix} A & B & C & D & E & F \end{pmatrix}^\top. \tag{24}$$

The data space $\mathcal{X}$ is a 2-dimensional manifold in the 6-dimensional space $\mathcal{R}^6$; the parameter space $\mathcal{U}$ is the 5-dimensional unit sphere S^6 centered on the origin of $\mathcal{R}^6$.

Example 2 Suppose N points in a 3-D scene are projected to (x_α, y_α) in the first image and (x'_α, y'_α) in the second, $\alpha = 1, ..., N$. If the camera imaging geometry is perspective projection, there exists a matrix $\boldsymbol{F}$ of determinant 0 such that

$$\left(\begin{pmatrix} \bar{x}_\alpha \\ \bar{y}_\alpha \\ 1 \end{pmatrix}, \boldsymbol{F} \begin{pmatrix} \bar{x}'_\alpha \\ \bar{y}'_\alpha \\ 1 \end{pmatrix}\right) = 0, \tag{25}$$

which is called the *epipolar equation* [14]. The matrix $\boldsymbol{F}$ is known as the *fundamental matrix*. For

3-D reconstruction from the images, we need to estimate the fundamental matrix $\boldsymbol{F}$ from the noisy data $\{(x_\alpha, y_\alpha)\}$ and $\{(x'_\alpha, y'_\alpha)\}$ (see Fig. 4). Eq. (25) reduces to eq. (22) if we put

$$\boldsymbol{\xi}\ (x, y, x', y') = \big(\, xx'\ xy'\ x\ yx'\ yy'\ y\ x'\ y'\ 1\,\big)^\top,$$
$$\boldsymbol{u} = \big(\, F_{11}\ F_{12}\ F_{13}\ F_{21}\ F_{22}\ F_{23}\ F_{31}\ F_{32}\ F_{33}\,\big)^\top. \quad (26)$$

The data space $\mathcal{X}$ is a 4-dimensional manifold in the 9-dimensional space $\mathcal{R}^9$; the parameter space $\mathcal{U}$ is a 7-dimensional manifold defined by $\det \boldsymbol{F} = 0$ and $\|\boldsymbol{F}\| = 1$, where the matrix norm is define by $\|\boldsymbol{F}\| = \sqrt{\sum_{i,j=1}^3 F_{ij}^2}$.

For the linear constraint (22), the function J in eq. (7) reduces to

$$J = \sum_{\alpha=1}^N \frac{(\boldsymbol{\xi}_\alpha, \boldsymbol{u})^2}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})}, \quad (27)$$

where $V_0[\boldsymbol{\xi}_\alpha]$ is the normalized covariance matrix of $\boldsymbol{\xi}_\alpha$; we use the abbreviation $\boldsymbol{\xi}_\alpha = \boldsymbol{\xi}(\boldsymbol{x}_\alpha)$. The matrix $V_0[\boldsymbol{\xi}_\alpha]$ can be expressed to a first approximation in the form

$$V_0[\boldsymbol{\xi}_\alpha] = \nabla_{\mathbf{x}}\boldsymbol{\xi}|^\top_{\mathbf{x}=\mathbf{x}_\alpha} V_0[\boldsymbol{x}_\alpha] \nabla_{\mathbf{x}}\boldsymbol{\xi}|_{\mathbf{x}=\mathbf{x}_\alpha}, \quad (28)$$

where $\nabla_{\mathbf{x}}\boldsymbol{\xi}$ is the $m \times p$ Jacobian matrix of $\boldsymbol{\xi}(\boldsymbol{x})$:

$$\nabla_{\mathbf{x}}\boldsymbol{\xi} = \begin{pmatrix} \partial\xi_1/\partial x_1 & \cdots & \partial\xi_p/\partial x_1 \\ \vdots & & \vdots \\ \partial\xi_1/\partial x_m & \cdots & \partial\xi_p/\partial x_m \end{pmatrix}. \quad (29)$$

The covariance matrix $V[\hat{\boldsymbol{u}}]$ of the ML estimator $\hat{\boldsymbol{u}}$ given by eq. (9) now reads

$$V[\hat{\boldsymbol{u}}] = \varepsilon^2 \Big(\sum_{\alpha=1}^N \frac{\boldsymbol{P}_{\mathbf{u}} \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top \boldsymbol{P}_{\mathbf{u}}}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})} \Big)^- + O(\varepsilon^4), \quad (30)$$

where the superscript $-$ denotes the (Moore-Penrose) generalized inverse. The matrix $\boldsymbol{P}_{\mathbf{u}}$ denotes projection onto the tangent space to the parameter space $\mathcal{U}$ at $\boldsymbol{u}$ (cf. Remark 8). Since the leading term is the lower bound on the covariance matrix of any estimation (Remark 7), the ML estimator is optimal up to higher order terms in ε.

Remark 14 Since we are focusing on the asymptotic analysis for $\varepsilon \to 0$, what we call the "ML estimator" is a first approximation to the true ML estimator for small ε (Remark 7). Note that if the parameter $\boldsymbol{u}$ is not constrained, the generalized inverse in eq. (30) can be replaced by the usual inverse, and the projection matrix $\boldsymbol{P}_{\mathbf{u}}$ is not necessary. However, $\boldsymbol{u}$ is at least constrained to be a unit vector, and often additional constraints exist, e.g., $\det \boldsymbol{F} = 0$ on the fundamental matrix $\boldsymbol{F}$. If no constraints exist other than $\|\boldsymbol{u}\| = 1$, the covariance matrix $V[\hat{\boldsymbol{u}}]$ has rank $p-1$, and its null space is in the direction of $\boldsymbol{u}$. The projection matrix $\boldsymbol{P}_{\mathbf{n}}$ in this case is

$$\boldsymbol{P}_{\mathbf{u}} = \boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^\top. \quad (31)$$

5.2 Least-squares method

If $\boldsymbol{u}$ is constrained, the minimization of eq. (27) should be carried out subject to the constraint, but this is very difficult in many cases. A practical approach to this is to ignore all the constraints except the normalization $\|\boldsymbol{u}\| = 1$ and do minimization over the $(p-1)$-dimensional sphere S^{p-1} in $\mathcal{R}^p$. This expedient is motivated by the fact that if the data $\{\boldsymbol{x}_\alpha\}$ are exact, the solution should automatically satisfy the remaining constraints. It follows that if the data uncertainty is very small, which we always assume, the resulting solution $\hat{\boldsymbol{u}}$ should satisfy all the constraints up to higher order terms in ε.

However, the minimization of eq. (27) is still nonlinear even if all constraints other than $\|\boldsymbol{u}\| = 1$ are ignored. The simplest approach is to solve eqs. (22) directly by (total) least squares, minimizing

$$J_{\mathrm{LS}} = \sum_{\alpha=1}^N (\boldsymbol{\xi}_\alpha, \boldsymbol{u})^2. \quad (32)$$

If we define the second-order moment matrix

$$\boldsymbol{M} = \sum_{\alpha=1}^N \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top, \quad (33)$$

eq. (32) is rewritten as

$$J_{\mathrm{LS}} = (\boldsymbol{u}, \boldsymbol{M}\boldsymbol{u}). \quad (34)$$

The unit vector $\boldsymbol{u}$ that minimizes this is the unit eigenvector of $\boldsymbol{M}$ for the smallest eigenvalue. The resulting *LS* (*least-squares*) *solution* $\hat{\boldsymbol{u}}_{\mathrm{LS}}$ is a very crude approximation to the ML estimator $\hat{\boldsymbol{u}}$. However, because of the ease of the computation, it is often used as an initial guess for computing the ML estimator $\hat{\boldsymbol{u}}$ by iterations.

5.3 Naive method

If we define

$$\boldsymbol{M}(\boldsymbol{u}) = \sum_{\alpha=1}^N \frac{\boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})}, \quad (35)$$

eq. (27) is written as

$$J = (\boldsymbol{u}, \boldsymbol{M}(\boldsymbol{u})\boldsymbol{u}). \quad (36)$$

This inspires the following iterations for computing the ML estimator:

1. Guess an appropriate initial value $\boldsymbol{u}_0$, say the LS solution $\hat{\boldsymbol{u}}_{\mathrm{LS}}$.
2. Assuming that $\boldsymbol{u}_{i-1}$ is obtained (initially $i = 1$), let $\boldsymbol{u}_i$ be the unit eigenvector of $\boldsymbol{M}(\boldsymbol{u}_{i-1})$ for the smallest eigenvalue.
3. Return $\boldsymbol{u}_i$ if $\boldsymbol{u}_i$ is sufficiently close to $\boldsymbol{u}_{i-1}$ except for the sign. Otherwise, let $\boldsymbol{u}_{i-1} \leftarrow \boldsymbol{u}_i$, and go back to Step 2.

This scheme does not work, however, because the resulting solution $\hat{\boldsymbol{u}}$ is the value $\boldsymbol{u}$ that minimizes $(\boldsymbol{u}, \boldsymbol{M}(\hat{\boldsymbol{u}})\boldsymbol{u})$, not $(\boldsymbol{u}, \boldsymbol{M}(\boldsymbol{u})\boldsymbol{u})$. In other words,

$$(\hat{\boldsymbol{u}}, \boldsymbol{M}(\hat{\boldsymbol{u}})\hat{\boldsymbol{u}}) < (\hat{\boldsymbol{u}} + \Delta\boldsymbol{u}, \boldsymbol{M}(\hat{\boldsymbol{u}})(\hat{\boldsymbol{u}} + \Delta\boldsymbol{u})) \quad (37)$$

for any nonzero perturbation $\Delta\boldsymbol{u}$, but not

$$(\hat{\boldsymbol{u}}, \boldsymbol{M}(\hat{\boldsymbol{u}})\hat{\boldsymbol{u}}) < (\hat{\boldsymbol{u}}+\Delta\boldsymbol{u}, \boldsymbol{M}(\hat{\boldsymbol{u}}+\Delta\boldsymbol{u})(\hat{\boldsymbol{u}}+\Delta\boldsymbol{u})). \quad (38)$$

A detailed analysis shows that $\hat{\boldsymbol{u}}$ is *biased* by $O(\varepsilon^2)$ [15]. Namely, if the fluctuations of the data $\{\boldsymbol{x}_\alpha\}$ are centered on their true values $\{\bar{\boldsymbol{x}}_\alpha\}$, the corresponding fluctuations of $\hat{\boldsymbol{u}}$ are around a value different from its true value by $O(\varepsilon^2)$. This causes inadmissible errors in many practical applications.

5.4 FNS method

If the constraint on $\boldsymbol{u}$ is ignored, the solution that minimizes eq. (27) is obtained by solving $\nabla_{\mathbf{u}} J = \mathbf{0}$. Since

$$\nabla_{\mathbf{u}} J = \sum_{\alpha=1}^{N} \frac{2(\boldsymbol{\xi}_\alpha, \boldsymbol{u})\boldsymbol{\xi}_\alpha}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})} - \sum_{\alpha=1}^{N} \frac{2(\boldsymbol{\xi}_\alpha, \boldsymbol{u})^2 V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u}}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})^2}, \quad (39)$$

the equation $\nabla_{\mathbf{u}} J = \mathbf{0}$ is written in the form

$$\boldsymbol{X}(\boldsymbol{u})\boldsymbol{u} = \mathbf{0}, \quad (40)$$

where

$$\boldsymbol{X}(\boldsymbol{u}) = \sum_{\alpha=1}^{N} \frac{\boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})} - \sum_{\alpha=1}^{N} \frac{(\boldsymbol{\xi}_\alpha, \boldsymbol{u})^2 V_0[\boldsymbol{\xi}_\alpha]}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})^2}. \quad (41)$$

From this, we have the following scheme for solving eq. (40):

1. Guess an appropriate initial value $\boldsymbol{u}_0$, say the LS solution $\hat{\boldsymbol{u}}_{\mathrm{LS}}$.
2. Assuming that $\boldsymbol{u}_{i-1}$ is obtained (initially $i = 1$), solve the eigenvalue problem

   $$\boldsymbol{X}(\boldsymbol{u}_{i-1})\boldsymbol{u} = \lambda\boldsymbol{u}. \quad (42)$$

   Let $\boldsymbol{u}_i$ be the unit eigenvector for the eigenvalue λ closest to 0.
3. Return $\boldsymbol{u}_i$ if $\boldsymbol{u}_i$ is sufficiently close to $\boldsymbol{u}_{i-1}$ except for the sign. Otherwise, let $\boldsymbol{u}_{i-1} \leftarrow \boldsymbol{u}_i$, and go back to Step 2.

The resulting solution $\hat{\boldsymbol{u}}$ satisfies eq. (40). In fact, the value $\hat{\boldsymbol{u}}$ produced by the above iterations should satisfy

$$\boldsymbol{X}(\hat{\boldsymbol{u}})\hat{\boldsymbol{u}} = \lambda\hat{\boldsymbol{u}} \quad (43)$$

for some λ. Taking the inner product of $\hat{\boldsymbol{u}}$ and both sides, we have

$$(\hat{\boldsymbol{u}}, \boldsymbol{X}(\hat{\boldsymbol{u}})\hat{\boldsymbol{u}}) = \lambda. \quad (44)$$

Eq. (41) implies that

$$(\hat{\boldsymbol{u}}, \boldsymbol{X}(\hat{\boldsymbol{u}})\hat{\boldsymbol{u}}) = \sum_{\alpha=1}^{N} \frac{(\hat{\boldsymbol{u}}, \boldsymbol{\xi}_\alpha)^2}{(\hat{\boldsymbol{u}}, V_0[\boldsymbol{\xi}_\alpha]\hat{\boldsymbol{u}})} - \sum_{\alpha=1}^{N} \frac{(\boldsymbol{\xi}_\alpha, \hat{\boldsymbol{u}})^2 (\hat{\boldsymbol{u}}, V_0[\boldsymbol{\xi}_\alpha]\hat{\boldsymbol{u}})}{(\hat{\boldsymbol{u}}, V_0[\boldsymbol{\xi}_\alpha]\hat{\boldsymbol{u}})^2} = 0, \quad (45)$$

meaning that $\lambda = 0$. Thus, $\hat{\boldsymbol{u}}$ is indeed the solution of eq. (40). This method was proposed by Chojnacki et al. [7] and called the *FNS* (*fundamental numerical scheme*) *method*. Usually, the iterations converges very quickly.

Remark 15 Eq. (45) is a consequence of the fact that the right-hand side of eq. (27) is a *homogeneous function of degree 0* in $\boldsymbol{u}$. Since multiplying $\boldsymbol{u}$ by any nonzero constant does not change the value of J, the gradient $\nabla_{\mathbf{u}} J$ is necessarily orthogonal to $\boldsymbol{u}$. Thus, $(\boldsymbol{u}, \nabla_{\mathbf{u}} J) = 2(\boldsymbol{u}, \boldsymbol{X}(\boldsymbol{u})\boldsymbol{u})$ is identically 0.

5.5 HEIV method

Eq. (40) can also be written as

$$\boldsymbol{M}(\boldsymbol{u})\boldsymbol{u} = \boldsymbol{L}(\boldsymbol{u})\boldsymbol{u}, \quad (46)$$

where

$$\boldsymbol{M}(\boldsymbol{u}) = \sum_{\alpha=1}^{N} \frac{\boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})}, \quad \boldsymbol{L}(\boldsymbol{u}) = \sum_{\alpha=1}^{N} \frac{(\boldsymbol{\xi}_\alpha, \boldsymbol{u})^2 V_0[\boldsymbol{\xi}_\alpha]}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})^2}. \quad (47)$$

This implies the following scheme.

1. Guess an appropriate initial value $\boldsymbol{u}_0$, say the LS solution $\hat{\boldsymbol{u}}_{\mathrm{LS}}$.
2. Assuming that $\boldsymbol{u}_{i-1}$ is obtained (initially $i = 1$), solve the generalized eigenvalue problem

   $$\boldsymbol{M}(\boldsymbol{u}_{i-1})\boldsymbol{u} = \lambda\boldsymbol{L}(\boldsymbol{u}_{i-1})\boldsymbol{u}. \quad (48)$$

   Let $\boldsymbol{u}_i$ be the generalized eigenvector for the generalized eigenvalue closest to 1. The norm of $\boldsymbol{u}_i$ is normalized to

   $$(\boldsymbol{u}_i, \boldsymbol{L}(\boldsymbol{u}_{i-1})\boldsymbol{u}_i) = 1. \quad (49)$$

3. Return $\boldsymbol{u}_i$ if $\boldsymbol{u}_i$ is sufficiently close to $\boldsymbol{u}_{i-1}$ except for the sign. Otherwise, let $\boldsymbol{u}_{i-1} \leftarrow \boldsymbol{u}_i$, and go back to Step 2.

The resulting solution $\hat{\boldsymbol{u}}$ should satisfy

$$\boldsymbol{M}(\hat{\boldsymbol{u}})\hat{\boldsymbol{u}} = \lambda\boldsymbol{L}(\hat{\boldsymbol{u}})\hat{\boldsymbol{u}}, \quad (50)$$

for some λ. Taking the inner product of $\hat{\boldsymbol{u}}$ and both sides, we have

$$(\hat{\boldsymbol{u}}, \boldsymbol{M}(\hat{\boldsymbol{u}})\hat{\boldsymbol{u}}) = \lambda, \quad (51)$$

due to the normalization convention (49), which implies from the second of eqs. (47) that

$$1 = (\hat{\boldsymbol{u}}, \boldsymbol{L}(\hat{\boldsymbol{u}})\hat{\boldsymbol{u}}) = \sum_{\alpha=1}^{N} \frac{(\boldsymbol{\xi}_\alpha, \boldsymbol{u})^2 (\hat{\boldsymbol{u}}, V_0[\boldsymbol{\xi}_\alpha]\hat{\boldsymbol{u}})}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})^2}$$
$$= \sum_{\alpha=1}^{N} \frac{(\boldsymbol{\xi}_\alpha, \boldsymbol{u})^2}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})}. \tag{52}$$

From the first of eqs. (47), we see that

$$(\hat{\boldsymbol{u}}, \boldsymbol{M}(\hat{\boldsymbol{u}})\hat{\boldsymbol{u}}) = \sum_{\alpha=1}^{N} \frac{(\hat{\boldsymbol{u}}, \boldsymbol{\xi}_\alpha)^2}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})} = 1, \tag{53}$$

meaning that $\lambda = 1$. Thus, $\hat{\boldsymbol{u}}$ is indeed the solution of eq. (46). However, the matrix $\boldsymbol{L}(\boldsymbol{u})$ is usually singular, because the matrix $V_0[\boldsymbol{x}_\alpha]$ in the second of eqs. (47) is likely to degenerate. This is easily seen from eq. (28): the dimension p of $\boldsymbol{\xi}_\alpha$ is generally larger than the dimension m of $\boldsymbol{x}_\alpha$. Hence, the generalized eigenvalue problem (48) need to be reduced to subproblems of smaller dimensions. The reduced form (we omit the details, see [9]) was proposed by Leedan and Meer [31] and Matei and Meer [33] and called called the *HEIV* (*heteroscedastic errors-in-variables*) *method*.

5.6 Renormalization method

The reason why the solution of the naive method of Sec. 5.3 is biased is that the matrix $\boldsymbol{M}(\boldsymbol{u})$ in eq. (35) is biased. If we decompose the datum $\boldsymbol{\xi}_\alpha$ into its true value $\bar{\boldsymbol{\xi}}_\alpha$ and the noise term $\Delta\boldsymbol{\xi}_\alpha$, the expectation of eq. (35) is

$$E[\boldsymbol{\xi}_\alpha\boldsymbol{\xi}_\alpha^\top]$$
$$= E[(\bar{\boldsymbol{\xi}}_\alpha + \Delta\boldsymbol{\xi}_\alpha)(\bar{\boldsymbol{\xi}}_\alpha + \Delta\boldsymbol{\xi}_\alpha)^\top]$$
$$= E[\bar{\boldsymbol{\xi}}_\alpha\bar{\boldsymbol{\xi}}_\alpha^\top] + E[\bar{\boldsymbol{\xi}}_\alpha\Delta\boldsymbol{\xi}_\alpha^\top] + E[\Delta\boldsymbol{\xi}_\alpha\bar{\boldsymbol{\xi}}_\alpha^\top] + E[\Delta\boldsymbol{\xi}_\alpha\Delta\boldsymbol{\xi}_\alpha^\top]$$
$$= \bar{\boldsymbol{\xi}}_\alpha\bar{\boldsymbol{\xi}}_\alpha^\top + V_0[\boldsymbol{\xi}_\alpha]. \tag{54}$$

Thus,

$$E[\boldsymbol{M}(\boldsymbol{u})] = \bar{\boldsymbol{M}}(\boldsymbol{u}) + \varepsilon^2\boldsymbol{N}(\boldsymbol{u}) + O(\varepsilon^4), \tag{55}$$

where $\bar{\boldsymbol{M}}(\boldsymbol{u})$ is the value of $\boldsymbol{M}(\boldsymbol{u})$ evaluated using the true values $\{\bar{\boldsymbol{\xi}}_\alpha\}$ and

$$\boldsymbol{N}(\boldsymbol{u}) = \sum_{\beta=1}^{N} \frac{V_0[\boldsymbol{\xi}_\beta]}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\beta]\boldsymbol{u})}. \tag{56}$$

Eq. (55) implies that an unbiased solution can be obtained if the matrix $\boldsymbol{M}(\boldsymbol{u})$ in eq. (36) is replaced by

$$\hat{\boldsymbol{M}}(\boldsymbol{u}) = \boldsymbol{M}(\boldsymbol{u}) - \varepsilon^2\boldsymbol{N}(\boldsymbol{u}). \tag{57}$$

The square noise level ε^2 is unknown, but if we note that the smallest eigenvalue of $\bar{\boldsymbol{M}}(\boldsymbol{u})$ is 0, we can estimate ε^2 so that the smallest eigenvalue of $\hat{\boldsymbol{M}}(\boldsymbol{u})$ is 0. Thus, we obtain the following scheme

1. Guess an appropriate initial value $\boldsymbol{u}_0$, say the LS solution $\hat{\boldsymbol{u}}_{\rm LS}$, and let $c_0 = 0$.
2. Assuming that $\boldsymbol{u}_{i-1}$ and c_{i-1} are obtained (initially $i = 1$), solve the eigenvalue problem
$$(\boldsymbol{M}(\boldsymbol{u}_{i-1}) - c_{i-1}\boldsymbol{N}(\boldsymbol{u}_{i-1}))\boldsymbol{u} = \lambda\boldsymbol{u}. \tag{58}$$
Let $\boldsymbol{u}_i$ be the unit eigenvector for the smallest eigenvalue λ.
3. Return $\boldsymbol{u}_i$ if λ is sufficiently close to 0. Otherwise, let
$$c_i = c_{i-1} + \frac{\lambda}{(\boldsymbol{u}_i, \boldsymbol{N}(\boldsymbol{u}_{i-1})\boldsymbol{u}_i)}. \tag{59}$$
4. Let $\boldsymbol{u}_{i-1} \leftarrow \boldsymbol{u}_i$ and go back to Step 2.

Eqs. (58) and (59) imply that if c_i is close to 0 we have

$$(\boldsymbol{M}(\boldsymbol{u}_{i-1}) - c_i\boldsymbol{N}(\boldsymbol{u}_{i-1}))\boldsymbol{u}_i = \boldsymbol{0}. \tag{60}$$

In fact, the inner product of $\boldsymbol{u}_i$ and the left-hand side is

$$(\boldsymbol{u}_i, (\boldsymbol{M}(\boldsymbol{u}_{i-1}) - c_i\boldsymbol{N}(\boldsymbol{u}_{i-1}))\boldsymbol{u}_i)$$
$$= (\boldsymbol{u}_i, (\boldsymbol{M}(\boldsymbol{u}_{i-1}) - c_{i-1}\boldsymbol{N}(\boldsymbol{u}_{i-1}))\boldsymbol{u}_i)$$
$$- \frac{\lambda(\boldsymbol{u}_i, \boldsymbol{N}(\boldsymbol{u}_{i-1})\boldsymbol{u}_i)}{(\boldsymbol{u}_i, \boldsymbol{N}(\boldsymbol{u}_{i-1})\boldsymbol{u}_i)}$$
$$= \lambda - \lambda = 0. \tag{61}$$

If c_i is close to 0, the matrix $\boldsymbol{M}(\boldsymbol{u}_{i-1}) - c_i\boldsymbol{N}(\boldsymbol{u}_{i-1})$ is positive semidefinite, so eq. (61) implies that $\boldsymbol{u}_i$ is included in the null space of $\boldsymbol{M}(\boldsymbol{u}_{i-1}) - c_i\boldsymbol{N}(\boldsymbol{u}_{i-1})$, proving eq. (60). Hence, the solution satisfies

$$(\boldsymbol{M}(\hat{\boldsymbol{u}}) - c\boldsymbol{N}(\hat{\boldsymbol{u}}))\hat{\boldsymbol{u}} = \boldsymbol{0}, \tag{62}$$

and c gives an estimate of ε^2. This scheme was proposed by Kanatani [15] and called *renormalization*.

Remark 16 Historically, this method was proposed first; the HEIV and FNS methods were proposed as an refinement to it. However, the renormalization solution and the HEIV/FNS solution (FNS and HEIV produce the same value) are both optimal in the sense that their covariance matrices differ only in the term $O(\varepsilon^4)$ in eq. (30) [15]. This is confirmed by numerical simulations [7, 8, 9].

Remark 17 Renormalization tries to eliminate the bias term in eq. (55) by "subtraction" in the form of eq. (57). An alternative strategy would be to remove the bias by "division". In fact, if we let $\tilde{\boldsymbol{M}}(\boldsymbol{u}) = \boldsymbol{N}(\boldsymbol{u})^{-1/2}\boldsymbol{M}(\boldsymbol{u})\boldsymbol{N}(\boldsymbol{u})^{-1/2}$ (the negative square root is defined by replacing all its eigenvalues λ by $1/\sqrt{\lambda}$ in the canonical form), $E[\bar{\boldsymbol{M}}(\boldsymbol{u})]$ and $\tilde{\boldsymbol{M}}(\boldsymbol{u})$ share the same eigenvectors up to $O(\varepsilon^4)$. If $\tilde{\boldsymbol{u}}$ is an eigenvector of $\tilde{\boldsymbol{M}}(\boldsymbol{u})$, the corresponding eigenvector of $\boldsymbol{M}(\boldsymbol{u})$ is $\boldsymbol{N}(\boldsymbol{u})^{-1/2}\tilde{\boldsymbol{u}}$. This implies that an unbiased solution is obtained by applying the naive method of Sec. 5.3

to $\tilde{\boldsymbol{M}}(\boldsymbol{u})$. This strategy is known as *equilibration* or *whitening*. However, the matrix $\boldsymbol{N}(\boldsymbol{u})$ is often singular due to the degeneracy of $V_0[\boldsymbol{\xi}_\alpha]$ (cf. Sec. 5.1), so $\boldsymbol{N}(\boldsymbol{u})^{-1/2}$ cannot be computed. Still, it has been applied to a few problems for which $\boldsymbol{N}(\boldsymbol{u})$ does not degenerate [32, 38, 39].

5.7 Optimal correction

In deriving the FNS, HEIV, and renormalization methods, we ignored all constraints on $\boldsymbol{u}$ except $\|\boldsymbol{u}\| = 1$. Let the remaining constraints be

$$\phi^{(k)}(\boldsymbol{u}) = 0, \qquad k = 1, ..., r. \tag{63}$$

From eq. (30), the normalized covariance of the ML estimator $\hat{\boldsymbol{u}}$ is given by

$$V_0[\hat{\boldsymbol{u}}] = \Big(\boldsymbol{P}_{\hat{\mathbf{u}}}\boldsymbol{M}(\hat{\boldsymbol{u}})\boldsymbol{P}_{\hat{\mathbf{u}}}\Big)^{-}, \tag{64}$$

where $\boldsymbol{M}(\boldsymbol{u})$ is defined in eq. (35) (or in eqs. (46)). The maximum likelihood solution of $\boldsymbol{u}$ that satisfies the constraint (63) is obtained to a first approximation by minimizing

$$J = (\hat{\boldsymbol{u}} - \boldsymbol{u}, V_0[\hat{\boldsymbol{u}}]^{-}(\hat{\boldsymbol{u}} - \boldsymbol{u})) \tag{65}$$

subject to eq. (63). Introducing Lagrange multipliers and first order approximation, we obtain the following solution [15]:

$$\boldsymbol{u}^* = \hat{\boldsymbol{u}} - V_0[\hat{\boldsymbol{u}}] \sum_{k,l=1}^{r} w^{(kl)} \hat{\phi}^{(k)} \nabla_{\mathbf{u}} \hat{\phi}^{(l)}. \tag{66}$$

Here, $w^{(kl)}$ is the (kl) element of the inverse of the $r \times r$ matrix whose (kl) element is $(\nabla_{\mathbf{u}}\hat{\phi}^{(k)}, V_0[\hat{\boldsymbol{u}}]\nabla_{\mathbf{u}}\hat{\phi}^{(l)})$, i.e.,

$$\Big(w^{(kl)}\Big) = \Big((\nabla_{\mathbf{u}}\hat{\phi}^{(k)}, V_0[\hat{\boldsymbol{u}}]\nabla_{\mathbf{u}}\hat{\phi}^{(l)})\Big)^{-1}. \tag{67}$$

The hat means that the ML estimator $\hat{\boldsymbol{u}}$ is substituted for $\boldsymbol{u}$. The normalized covariance matrix of the corrected value $\boldsymbol{u}^*$ of eq. (66) is

$$V_0[\boldsymbol{u}^*] = V_0[\hat{\boldsymbol{u}}] - \sum_{k,l=1}^{r} w^{(kl)} (V_0[\hat{\boldsymbol{u}}]\nabla_{\mathbf{u}}\hat{\phi}^{(k)})(V_0[\hat{\boldsymbol{u}}]\nabla_{\mathbf{u}}\hat{\phi}^{(k)})^\top \tag{68}$$

up to $O(\varepsilon^2)$ [15]. For a single constraint, eqs. (66) and (68) reduce to

$$\boldsymbol{u}^* = \hat{\boldsymbol{u}} - \frac{\hat{\phi} V_0[\hat{\boldsymbol{u}}]\nabla_{\mathbf{u}}\hat{\phi}}{(\nabla_{\mathbf{u}}\hat{\phi}, V_0[\hat{\boldsymbol{u}}]\nabla_{\mathbf{u}}\hat{\phi})}, \tag{69}$$

$$V_0[\boldsymbol{u}^*] = V_0[\hat{\boldsymbol{u}}] - \frac{(V_0[\hat{\boldsymbol{u}}]\nabla_{\mathbf{u}}\hat{\phi})(V_0[\hat{\boldsymbol{u}}]\nabla_{\mathbf{u}}\hat{\phi})^\top}{(\nabla_{\mathbf{u}}\hat{\phi}, V_0[\hat{\boldsymbol{u}}]\nabla_{\mathbf{u}}\hat{\phi})}. \tag{70}$$

Remark 18 If the r constraints in eq. (63) are redundant, say only r' $(< r)$ of them are independent, the inverse in eq. (67) is replaced by the generalized inverse of rank r' (cf. Remark 6).

Remark 19 If all the r constraints in eq. (63) are independent, the rank of the matrix $V_0[\boldsymbol{u}^*]$ given by eq. (66) is smaller than $V_0[\hat{\boldsymbol{u}}]$ by r. Intuitively, the ellipsoid that represents the uncertainty of $\boldsymbol{u}$ in $\mathcal{R}^p$ "collapses" in the r directions in which the constraint (63) is violated, while it keeps its shape in the directions orthogonal to them. Hence, the optimality of the ML estimator is not affected by doing this type of posterior correction [15].

Remark 20 Eq. (66) enforces all the constraints only to a first approximation, so $\phi^{(k)}(\boldsymbol{u}^*)$, $k = 1, ..., r$, may not exactly be 0, and $\boldsymbol{u}^*$ may not exactly be a unit vector. Such higher order discrepancies can be eliminated by iterating eqs. (69) and (70) in the form

$$\boldsymbol{u}^* \leftarrow N[\hat{\boldsymbol{u}} - \frac{\hat{\phi} V_0[\hat{\boldsymbol{u}}]\nabla_{\mathbf{u}}\hat{\phi}}{(\nabla_{\mathbf{u}}\hat{\phi}, V_0[\hat{\boldsymbol{u}}]\nabla_{\mathbf{u}}\hat{\phi})}], \tag{71}$$

$$V_0[\boldsymbol{u}^*] \leftarrow \boldsymbol{P}_{\mathbf{u}^*}\Big(V_0[\hat{\boldsymbol{u}}] - \frac{(V_0[\hat{\boldsymbol{u}}]\nabla_{\mathbf{u}}\hat{\phi})(V_0[\hat{\boldsymbol{u}}]\nabla_{\mathbf{u}}\hat{\phi})^\top}{(\nabla_{\mathbf{u}}\hat{\phi}, V_0[\hat{\boldsymbol{u}}]\nabla_{\mathbf{u}}\hat{\phi})}\Big)\boldsymbol{P}_{\mathbf{u}^*}, \tag{72}$$

where $N[\,\cdot\,]$ denotes normalization to a unit vector $(N[\boldsymbol{v}] = \boldsymbol{v}/\|\boldsymbol{v}\|)$, and $\boldsymbol{P}_{\mathbf{u}^*}$ is the projection matrix defined by eq. (31). Eq. (72) makes the null space of the $V_0[\boldsymbol{u}^*]$ exactly compatible with $\boldsymbol{u}^*$.

6. Other Uncertainty Modelings

Finally, we discuss some new topics related to the use of statistical methods for geometric inference.

6.1 Asymptotic parameters

The number n that appears in the standard statistical analysis is the *number of experiments*. It is also called the *number of trials*, the *number of observations*, and the *number of samples*. Evidently, the properties of the ensemble are revealed more precisely as more data are sampled from it.

However, the number n is often called the *number of data*, which has caused considerable confusion. For example, if we observe a 100-dimensional vector datum in one experiment, one may think that the "number of data" is 100, but this is wrong: the number n of experiments is 1. We are observing one sample from an ensemble of 100-dimensional vectors.

For character recognition, the underlying ensemble is the set of possible character images, and the learning process concerns the number n of training steps necessary to establish satisfactory responses. This is independent of the dimension N of the vector that represents each character. The learning performance is evaluated asymptotically as $n \to \infty$, not $N \to \infty$.

For geometric inference, however, many researchers have taken the dimension of the data as the "number of data" perhaps because the ensemble is hypothetical and one cannot sample more than one datum from it. However, if we extract, for example, 50 feature points, they constitute a 100-dimensional

vector consisting of their x and y coordinates. If no other information, such as the image intensity, is used, the image is completely characterized by that vector. Applying a statistical method means regarding it as a sample from a hypothetical ensemble of 100-dimensional vectors.

6.2 Neyman-Scott problem

In the past, many computer vision researchers have analyzed the asymptotic behavior as $N \to \infty$ without explicitly mentioning what the underlying ensemble is. This is perhaps motivated by a similar formulation in the statistical literature. Suppose, for example, a rod-like structure lies on the ground in the distance. We emit a laser beam toward it and estimate its position and orientation by observing the reflection of the beam, which is contaminated by noise. We assume that the laser beam can be emitted in any orientation any number of times but the emission orientation is measured with noise. The task is to estimate the position and orientation of the structure as accurately as possible by emitting as small a number of beams as possible. Naturally, the estimation performance should be evaluated in the asymptotic limit $n \to \infty$ with respect to the number n of emissions.

The underlying ensemble is the set of all response times for all possible directions of emission. Usually, we are interested in the position and orientation of the structure but not the exact orientation of each emission, so the variables for the former are called the *structural parameters*, which are fixed in number, while the latter are called the *nuisance parameters*, which increase indefinitely as the number n of experiments increases [2]. Such a formulation is called the *Neyman-Scott problem* [40]. Since the constraint is an implicit function in the form of eq. (3), we are considering an *errors-in-variables model* [11]. If we linearize the constraint by changing variables, the noise characteristics differs for each data component, so the problem is *heteroscedastic* [31].

To solve this problem, one can introduce a parametric model for the distribution of possible laser emission orientations, regarding the actual emissions as random samples from it. This formulation is called a *semiparametric model* [2]. An optimal solution can be obtained by finding a good *estimating function* [2, 43].

6.3 Semiparametric models

Since the semiparametric model has something different from the geometric inference problem described in Sec. 3.2, a detailed analysis is required for examining if application of a semiparametric model to geometric inference will yield a desirable result [43, 41]. In any event, one should explicitly state what kind of ensemble (or ensemble of ensembles) is assumed before doing statistical analysis.

This is not merely a conceptual issue. It also affects the performance evaluation of simulation experiments. In doing a simulation, one can freely change the number N of feature points and the noise level ε. If the accuracy of Method A is higher than Method B for particular values of N and ε, one cannot conclude that Method A is superior to Method B, because opposite results may come out for other values of N and ε. Here, we have two alternatives for performance evaluation: fixing ε and varying N to see if admissible accuracy is attained for a smaller number of feature point; fixing N and varying ε to see if larger data uncertainty can be tolerated for admissible accuracy. These two types of evaluation have different meanings. Our conclusion is that the results of one type of evaluation cannot directly be compared with the results of the other.

7. Conclusions

We have investigated the meaning of "statistical methods" for geometric inference based on image feature points. Tracing back the origin of feature uncertainty to image processing operations, we discussed the implications of asymptotic analysis in reference to "geometric fitting" and "geometric model selection". We pointed out that a correspondence exists between the standard statistical analysis and the geometric inference problem. We also compared the capability of the "geometric AIC" and the "geometric MDL" in detecting degeneracy. Next, we reviewed recent progress in geometric fitting techniques for linear constraints, describing the "FNS method", the "HEIV method", the "renormalization method", and other related techniques. Finally, we discussed the "Neyman-Scott problem" and "semiparametric models" in relation to geometric inference.

From these discussions, we conclude that applications of statistical methods requires careful considerations about the nature of the problem in question and that different statistical theories are necessary for different classes of problems. In this sense, there is much room for new statistical theories to emerge as the scope of computer vision research expands. The important thing is, however, to always make clear the underlying hypotheses and assumptions, not simply using the methods in the statistical literature.

In Appendix, we summarize the derivation of the geometric AIC and the geometric MDL.

Acknowledgments: This work was supported in part by the Ministry of Education, Culture, Sports, Science and Technology, Japan, under the Grant in Aid for Scientific Research C(2) (No. 15500113), the Support Center for Advanced Telecommunications Technology Research, and Kayamori Foundation of Informational Science Advancement.

References

[1] H. Akaike, A new look at the statistical model identification, IEEE Trans. Autom. Control, **16**-6 (1977), 716–723.

[2] S. Amari and M. Kawanabe, Information geometry of estimating functions in semiparametric statistical models, *Bernoulli*, **3** (1997), 29–54.

[3] K. Bubna and C. V. Stewart, Model selection techniques and merging rules for range data segmentation algorithms, *Comput. Vision Image Understand.*, **80**-2 (2000), 215–245.

[4] F. Chabat, G. Z. Yang and D. M. Hansell, A corner orientation detector, *Image Vision Comput.*, **17**-10 (1999), 761–769.

[5] K. Cho and P. Meer, Image segmentation form consensus information, *Comput. Vision Image Understand.*, **68**-1 (1997), 72–89.

[6] K. Cho, P. Meer, J. Cabrera, Performance assessment through bootstrap, *IEEE Trans. Patt. Anal. Mach. Intell.*, **19**-11 (1997), 1185–1198.

[7] W. Chojnacki, M. J. Brooks, A. van den Hengel and D. Gawley, On the fitting of surfaces to data with covariances, *IEEE Trans. Patt. Anal. Mach. Intell.*, **22**-11 (2000), 1294–1303.

[8] W. Chojnacki, M. J. Brooks and A. van den Hengel, Rationalising the renormalisation method of Kanatani, *J. Math. Imaging Vision*, **14**-1 (2001), 21–38.

[9] W. Chojnacki, M. J. Brooks, A. van den Hengel and D. Gawley, From FNS to HEIV: A link between two vision parameter estimation methods, *IEEE Trans. Patt. Anal. Mach. Intell.*, **26**-2 (2004), 264–268.

[10] B. Efron and R. J. Tibshirani, *An Introduction to Bootstrap*, Chapman-Hall, New York, 1993.

[11] W. A. Fuller, *Measurement Error Models*, Wiley, New York, 1987.

[12] H. Gu, Y. Shirai and M. Asada, MDL-based segmentation and motion modeling in a long sequence of scene with multiple independently moving objects, *IEEE Trans. Patt. Anal. Mach. Intell.*, **18**-1 (1996), 58–64.

[13] C. Harris and M. Stephens, A combined corner and edge detector, *Proc. 4th Alvey Vision Conf.*, Aug. 1988, Manchester, U.K., pp. 147–151.

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

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

[16] K. Kanatani, Geometric information criterion for model selection, *Int. J. Comput. Vision*, **26**-3 (1998), 171–189.

[17] K. Kanatani, Statistical optimization and geometric inference in computer vision, *Phil. Trans. Roy. Soc. Lond.*, A-**356** (1998), 1303–1320.

[18] K. Kanatani, Cramer-Rao lower bounds for curve fitting, *Graphical Models Image Process.*, **60**-2 (1988), 93–99.

[19] K. Kanatani, Model selection criteria for geometric inference, in A. Bab-Hadiashar and D. Suter (eds.), *Data Segmentation and Model Selection for Computer Vision: A Statistical Approach*, Springer, 2000, pp. 91–115.

[20] K. Kanatani, Motion segmentation by subspace separation and model selection, *Proc. 8th Int. Conf. Comput. Vision*, July 2001, Vancouver, Canada, Vol. 2, pp. 301–306.

[21] K. Kanatani, Model selection for geometric inference, plenary talk, *Proc. 5th Asian Conf. Comput. Vision*, January 2002, Melbourne, Australia, Vol. 1, pp. xxi–xxxii.

[22] K. Kanatani, Motion segmentation by subspace separation: Model selection and reliability evaluation, *Int. J. Image Graphics/*, **2**-2 (2002), 179–197.

[23] K. Kanatani, Evaluation and selection of models for motion segmentation, *Proc. 7th Euro. Conf. Comput. Vision*, May 2002, Copenhagen, Denmark, Vol. 3, pp. 335–349.

[24] K. Kanatani and C. Matsunaga, Estimating the number of independent motions for multibody motion segmentation, *Proc. 5h Asian Conf. Comput. Vision*, January 2002, Melbourne, Australia, Vol. 1, pp. 7–12.

[25] Y. Kanazawa and K. Kanatani, Optimal line fitting and reliability evaluation, *IEICE Trans. Inf. & Syst.*, **E79-D**-9 (1996), 1317–1322.

[26] Y. Kanazawa and K. Kanatani, Optimal conic fitting and reliability evaluation, *IEICE Trans. Inf. & Syst.*, **E79-D**-9 (1996), 1323–1328.

[27] Y. Kanazawa and K. Kanatani, Infinity and planarity test for stereo vision, *IEICE Trans. Inf. & Syst.*, **E80-D**-8 (1997), 774–779.

[28] Y. Kanazawa and K. Kanatani, Stabilizing image mosaicing by model selection, in M. Pollefeys, L. Van Gool, A. Zisserman and A. Fitzgibbon (eds.), *3D Structure from Images–SMILE 2000*, Springer, Berlin, 2001, pp. 35–51.

[29] Y. Kanazawa and K. Kanatani, Do we really have to consider covariance matrices for image features? *Proc. 8th Int. Conf. Comput. Vision*, July 2001, Vancouver, Canada, Vol. 2, pp. 586–591.

[30] Y. G. Leclerc, Constructing simple stable descriptions for image partitioning, *Int. J. Comput. Vision*, **3**-1 (1989), 73–102.

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

[32] W. J. MacLean, Removal of translation bias when using subspace methods, *Proc. 7th Int. Conf. Comput. Vision*, September 1999, Kerkyra, Greece, Vol. 2, pp. 753–758.

[33] B. Matei and P. Meer, A generalized method for errors-in-variables problem in computer vision, *Proc. 15th Int. Conf. Patt. Recog.*, September 2000, Barcelona, Spain, Vol. 2, pp. 18–25.

[34] C. Matsunaga and K. Kanatani, Calibration of a moving camera using a planar pattern: Optimal computation, reliability evaluation and stabilization by model selection, in *Proc. 6th Euro. Conf. Comput. Vision*, June–July, 2000, Dublin, Ireland, Vol. 2, pp. 595–609.

[35] B. A. Maxwell, Segmentation and interpretation of multicolored objects with highlights, *Comput. Vision Image Understand.*, **77**-1 (2000), 1–24.

[36] S. J. Maybank and P. F. Sturm, MDL, collineations and the fundamental matrix, *Proc. 10th British Machine Vision Conference*, September 1999, Nottingham, U.K., pp. 53–62.

[37] D. D. Morris, K. Kanatani and T. Kanade, Gauge fixing for accurate 3D estimation, *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, December 2001, Kauai, Hawaii, U.S.A., Vol. 2, pp. 343–350.

[38] M. Mühlich and R. Mester, The role of total least squares in motion analysis, *Proc. 5th Euro. Conf. Comput. Vision*, June 1998, Freiburg, Germany, Vol. 2, pp. 305–321.

[39] M. Mühlich and R. Mester, A considerable improvement in pure parameter estimation using TLS and equilibration, *Patt. Recog. Lett.*, **22**-11 (2001), 1181–1189.

[40] J. Neyman and E. L. Scott, Consistent estimates based on partially consistent observations, *Econometrica*, **16**-1 (1948), 1–32.

[41] N. Ohta, Motion parameter estimation from optical flow without nuisance parameters, *3rd Int. Workshop on Statistical and Computational Theory of Vision* October 2003, Nice, France: `http://www.stat.ucla.edu/~sczhu/Workshops/SCTV2003.html`

[42] N. Ohta and K. Kanatani, Moving object detection from optical flow without empirical thresholds, *IEICE Trans. Inf. & Syst.*, **E81-D**-2 (1998), 243–245.

[43] T. Okatani and K. Deguchi, Toward a statistically optimal method for estimating geometric relations from noisy data: Cases of linear relations, *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, June 2003, Madison, WI, U.S.A., Vol. 1, pp. 432–439.

[44] D. Reisfeld, H. Wolfson and Y. Yeshurun, Context-free attentional operators: The generalized symmetry transform, *Int. J. Comput. Vision*, **14** (1995), 119–130.

[45] J. Rissanen, Universal coding, information, prediction and estimation, *IEEE Trans. Inform. Theory*, **30**-4 (1984), 629–636.

[46] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*, World Scientific, Singapore, 1989.

[47] J. Rissanen, Fisher information and stochastic complexity, IEEE Trans. Inform. Theory, **42**-1 (1996), 40–47.

[48] C. Schmid, R. Mohr and C. Bauckhage, Evaluation of interest point detectors, *Int J. Comput. Vision*, **37**-2 (2000), 151–172.

[49] S. M. Smith and J. M. Brady, SUSAN—A new approach to low level image processing, *Int. J. Comput. Vision*, **23**-1 (1997), 45–78.

[50] Y. Sugaya and K. Kanatani, Outlier removal for feature tracking by subspace separation, *IEICE Trans. Inf. & Syst.*, **E86**-D (2003), 1095–1102.

[51] P. H. S. Torr, An assessment of information criteria for motion model selection, *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, June 1997, Puerto Rico, pp. 47–53.

[52] P. H. S. Torr, Geometric motion segmentation and model selection, *Phil. Trans. Roy. Soc. Lond.*, A-**356** (1998), 1321–1340.

[53] P. H. S. Torr, Bayesian model estimation and selection for epipolar geometry and generic manifold fitting, *Int. J. Comput. Vision*, **50**-1 (2002), 35–61, 2002.

[54] P. H. S. Torr, A. FitzGibbon and A. Zisserman, Maintaining multiple motion model hypotheses through many views to recover matching and structure, *Proc. 6th Int. Conf. Comput. Vision*, January 1998, Bombay, India, pp. 485–492.

[55] P. H. S. Torr and A. Zisserman, Concerning Bayesian motion segmentation, model averaging, matching and the trifocal tensor, in *Proc. 6th Euro. Conf. Comput. Vision*, June–July, 2000, Dublin, Ireland, Vol. 1, pp. 511–528.

[56] Iman Triono, N. Ohta and K. Kanatani, Automatic recognition of regular figures by geometric AIC, *IEICE Trans. Inf. & Syst.*, **E81-D**-2 (1998), 246–248.

Appendix

A. Derivation of the Geometric AIC

A.1 Goodness of a Model

Akaike [1] adopted as the measure of the goodness of the model

$$P(\{\boldsymbol{X}_\alpha\}) = \prod_{\alpha=1}^{N} \frac{e^{-(\boldsymbol{X}_\alpha-\bar{\boldsymbol{x}}_\alpha, V_0[\boldsymbol{x}_\alpha]^{-1}(\boldsymbol{X}_\alpha-\bar{\boldsymbol{x}}_\alpha))/2\varepsilon^2}}{\sqrt{(2\pi\varepsilon^2)^m|V_0[\boldsymbol{x}_\alpha]|}}, \tag{73}$$

the *Kullback-Leibler distance* (or *divergence*) the true distribution from it:

$$\begin{aligned} D &= \int\cdots\int P_T(\{\boldsymbol{X}_\alpha\}) \log \frac{P_T(\{\boldsymbol{X}_\alpha\})}{P(\{\boldsymbol{X}_\alpha\})} d\boldsymbol{X}_1 \cdots d\boldsymbol{X}_N \\ &= E[\log P_T(\{\boldsymbol{X}_\alpha\})] - E[\log P(\{\boldsymbol{X}_\alpha\})]. \end{aligned} \tag{74}$$

Here, $E[\,\cdot\,]$ denotes expectation with respect to the true (unknown) probability density $P_T(\{\boldsymbol{X}_\alpha\})$. The assumed model is regarded as good if D is small.

Substituting eq. (73) into eq. (74) and noting that $E[\log P_T(\{\boldsymbol{X}_\alpha\})]$ does not depend on individual models, we regard the model as good if

$$\begin{aligned} &-E[\log P(\{\boldsymbol{X}_\alpha\})] \\ &\quad = \frac{1}{2\varepsilon^2} E[\sum_{\alpha=1}^{N} (\boldsymbol{X}_\alpha - \bar{\boldsymbol{x}}_\alpha, V_0[\boldsymbol{x}_\alpha]^{-1}(\boldsymbol{X}_\alpha - \bar{\boldsymbol{x}}_\alpha))] \\ &\qquad + \frac{mN}{2}\log 2\pi\varepsilon^2 + \frac{1}{2}\sum_{\alpha=1}^{N} \log |V_0[\boldsymbol{x}_\alpha]| \end{aligned} \tag{75}$$

is small. The last two terms on the right-hand side do not depend on individual models. So, multiplying the first term by $2\varepsilon^2$, we seek a model that minimizes the *expected residual*

$$E = E[\sum_{\alpha=1}^{N} (\boldsymbol{X}_\alpha - \bar{\boldsymbol{x}}_\alpha, V_0[\boldsymbol{x}_\alpha]^{-1}(\boldsymbol{X}_\alpha - \bar{\boldsymbol{x}}_\alpha))]. \tag{76}$$

A.2 Evaluation of Expectation

The difficulty of using eq. (76) as a model selection criterion is that the expectation $E[\,\cdot\,]$ must be evaluated using the *true* density, which we do not know. Here arises a sharp distinction between the standard statistical analysis, in which Akaike was interested, and the geometric inference problem, in which we are interested, as to how to evaluate the expectation.

For the standard statistical analysis, we assume that we could, at least in principle, observe as

many data as desired. If we are allowed to sample independent instances $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, ..., $\boldsymbol{x}_n$ according to a density $P_T(\boldsymbol{X})$, the expectation $E[Y(\boldsymbol{X})] = \int Y(\boldsymbol{X})P_T(\boldsymbol{X})d\boldsymbol{X}$ of a statistic $Y(\boldsymbol{X})$ can be approximated by the sample mean $(1/n)\sum_{i=1}^{n} Y(\boldsymbol{x}_i)$, which converges to the true expectation in the limit $n \to \infty$ (the *law of large numbers*). Akaike's AIC is based on this principle.

In contrast, we can obtain only *one* instance $\{\boldsymbol{x}_\alpha\}$ of $\{\boldsymbol{X}_\alpha\}$ for geometric inference, so we cannot replace expectation by the sample mean. However, we are interested only in the limit $\varepsilon \to 0$. So, the expectation $E[Y(\{\boldsymbol{X}_\alpha\})] = \int \cdots \int Y(\{\boldsymbol{X}_\alpha\})P_T(\{\boldsymbol{X}_\alpha\})d\boldsymbol{X}_1 \cdots d\boldsymbol{X}_N$ can be approximated by $Y(\{\boldsymbol{x}_\alpha\})$, because as $\varepsilon \to 0$ we have $P_T(\{\boldsymbol{X}_\alpha\}) \to \prod_{\alpha=1}^{N} \delta(\boldsymbol{X}_\alpha - \bar{\boldsymbol{x}}_\alpha)$, where $\delta(\,\cdot\,)$ denotes the Dirac delta function. It follows that we can approximate E as follows (note that $1/N$ is not necessary):

$$J = \sum_{\alpha=1}^{N} (\boldsymbol{x}_\alpha - \bar{\boldsymbol{x}}_\alpha, V_0[\boldsymbol{x}_\alpha]^{-1}(\boldsymbol{x}_\alpha - \bar{\boldsymbol{x}}_\alpha)). \tag{77}$$

A.3 Bias Removal

There is still a difficulty using eq. (77) as a criterion: the model parameters $\{\bar{\boldsymbol{x}}_\alpha\}$ and $\boldsymbol{u}$ need to be estimated. If we view eq. (77) as a measure of the goodness of the model, we should compute their ML estimators $\{\hat{\boldsymbol{x}}_\alpha\}$ and $\hat{\boldsymbol{u}}$, minimizing eq. (77) subject to the constraint (3). Substituting $\{\hat{\boldsymbol{x}}_\alpha\}$ and $\hat{\boldsymbol{u}}$ for $\{\bar{\boldsymbol{x}}_\alpha\}$ and $\boldsymbol{u}$ in eq. (77), we obtain the *residual* (*sum of squares*):

$$\hat{J} = \sum_{\alpha=1}^{N} (\boldsymbol{x}_\alpha - \hat{\boldsymbol{x}}_\alpha, V_0[\boldsymbol{x}_\alpha]^{-1}(\boldsymbol{x}_\alpha - \hat{\boldsymbol{x}}_\alpha)). \tag{78}$$

Here, a logical inconsistency arises. Eq. (3) defines not a particular model but a *class* of models parameterized by $\{\bar{\boldsymbol{x}}_\alpha\}$ and $\boldsymbol{u}$. If we choose particular values $\{\hat{\boldsymbol{x}}_\alpha\}$ and $\hat{\boldsymbol{u}}$ (i.e., the ML-estimators), we are given a particular model. According to the logic in Sec. A.1, its goodness should be evaluated by $E[\sum_{\alpha=1}^{N}(\boldsymbol{X}_\alpha - \hat{\boldsymbol{x}}_\alpha, V_0[\boldsymbol{x}_\alpha]^{-1}(\boldsymbol{X}_\alpha - \hat{\boldsymbol{x}}_\alpha))]$. According to the logic in Sec. A.2, the expectation can be approximated using a *typical* instance of $\{\boldsymbol{X}_\alpha\}$. However, $\{\hat{\boldsymbol{x}}_\alpha\}$ and $\hat{\boldsymbol{u}}$ were computed from $\{\boldsymbol{x}_\alpha\}$, so $\{\boldsymbol{x}_\alpha\}$ *cannot* be a typical instance of $\{\boldsymbol{X}_\alpha\}$. In fact, $\hat{J}$ is generally smaller than $E[\sum_{\alpha=1}^{N}(\boldsymbol{X}_\alpha - \hat{\boldsymbol{x}}_\alpha, V_0[\boldsymbol{x}_\alpha]^{-1}(\boldsymbol{X}_\alpha - \hat{\boldsymbol{x}}_\alpha))]$, because $\{\hat{\boldsymbol{x}}_\alpha\}$ and $\hat{\boldsymbol{u}}$ were so determined as to minimize $\hat{J}$.

This is the difficulty that Akaike encountered in the derivation of his AIC. His strategy for resolving this can be translated in our setting as follows.

Ideally, we should approximate the expectation using an instance $\{\boldsymbol{x}_\alpha^*\}$ of $\{\boldsymbol{X}_\alpha\}$ generated *independently* of the current data $\{\boldsymbol{x}_\alpha\}$. In other words, we should evaluate

$$J^* = \sum_{\alpha=1}^{N} (\boldsymbol{x}_\alpha^* - \hat{\boldsymbol{x}}_\alpha, V_0[\boldsymbol{x}_\alpha]^{-1}(\boldsymbol{x}_\alpha^* - \hat{\boldsymbol{x}}_\alpha)). \tag{79}$$

Let us call $\{\boldsymbol{x}_\alpha^*\}$ the *future data*; they are "another" instance of $\{\boldsymbol{X}_\alpha\}$ that *might* occur if we did a hypothetical experiment. In reality, we have the current data $\{\boldsymbol{x}_\alpha\}$ only[2]. So, we try to compensate for the bias in the form

$$\hat{J}^* = \hat{J} + b\varepsilon^2. \tag{80}$$

Both $\hat{J}^*$ and $\hat{J}$ are $O(\varepsilon^2)$, so b is $O(1)$. Since $\hat{J}^*$ and $\hat{J}$ are random variables, so is b. It can be proved [15, 16] that

$$E^*[E[b]] = 2(Nd + p) + O(\varepsilon^2), \tag{81}$$

where $E[\,\cdot\,]$ and $E^*[\,\cdot\,]$ denote expectations for $\{\boldsymbol{x}_\alpha\}$ and $\{\boldsymbol{x}_\alpha^*\}$, respectively, and $d = m - r$ is the dimension of the manifold $\mathcal{S}$ defined the constraint $F^{(k)}(\boldsymbol{x}, \boldsymbol{u}) = 0$, $k = 1$, ..., r (recall that p is the dimension of the parameter vector $\boldsymbol{u}$).

Thus, we obtain an unbiased estimator of $\hat{J}^*$ in the first order in the form

$$\text{G-AIC} = \hat{J} + 2(Nd + p)\varepsilon^2, \tag{82}$$

which is the *geometric AIC* of Kanatani [15, 16], who derived eq. (81) directly. Here, we have given a new justification by going back to the Kullback-Leibler distance (74).

B. Derivation of the Geometric MDL

B.1 Two-Stage Encoding

If the data $\{\boldsymbol{x}_\alpha\}$ are sampled according to the probability density (73), they can be encoded, after their domain is quantized, in a shortest prefix code of length

$$-\log P = \frac{J}{2\varepsilon^2} + \frac{mN}{2}\log 2\pi\varepsilon^2 + \frac{1}{2}\sum_{\alpha=1}^{N} \log |V_0[\boldsymbol{x}_\alpha]|, \tag{83}$$

up to a constant that depends only on the domain and the width of the quantization. Here, J is the sum of the square Mahalanobis distances in eq. (6). Using the natural logarithm, we take $\log_2 e$ bits as the unit of length.

Note the similarity and contrast to the geometric AIC, which minimizes the *expectation* of eq. (83) (see eq. (75)), while here eq. (83) is directly minimized with a different interpretation.

In order to do encoding using eq. (73), we need the true values $\{\bar{\boldsymbol{x}}_\alpha\}$ and the parameter $\boldsymbol{u}$. Since they are unknown, we use their ML estimators that minimize

[2] If such data $\{\boldsymbol{x}_\alpha^*\}$ actually exist, the test using them is called *cross-validation*. We can also generate equivalent data by a computer. Such a simulations is called *bootstrap* [10].

eq. (83) (specifically J). The last two terms of eq. (83) do not depend on individual models, so the minimum code length is $\hat{J}/2\varepsilon^2$ up to a constant, where $\hat{J}$ is the residual in eq. (78). For brevity, we hereafter call "the code length determined up to a constant that does not depend on individual models" simply the *description length*.

Since the ML estimators $\{\hat{\boldsymbol{x}}_\alpha\}$ and $\hat{\boldsymbol{u}}$ are real numbers, they also need to be quantized. If we use a larger quantization width, their code lengths become shorter, but the description length $\hat{J}/2\varepsilon^2$ will increase. So, we take the width that minimizes the total description length. The starting point is the fact that eq. (7) can be written as follows [15]:

$$J = \hat{J} + \sum_{\alpha=1}^{N} (\boldsymbol{x}_\alpha - \hat{\boldsymbol{x}}_\alpha, V_0[\hat{\boldsymbol{x}}_\alpha]^-(\boldsymbol{x}_\alpha - \hat{\boldsymbol{x}}_\alpha)) + (\boldsymbol{u} - \hat{\boldsymbol{u}}, V_0[\hat{\boldsymbol{u}}]^{-1}(\boldsymbol{u} - \hat{\boldsymbol{u}})) + O(\varepsilon^3). \quad (84)$$

Here, the superscript $-$ denotes the (Moore-Penrose) generalized inverse, and $V_0[\hat{\boldsymbol{x}}_\alpha]$ and $V_0[\hat{\boldsymbol{u}}_\alpha]$ are, respectively, the a posteriori covariance matrices of the ML estimators $\hat{\boldsymbol{x}}_\alpha$ and $\hat{\boldsymbol{u}}$ given as follows [15]:

$$V_0[\hat{\boldsymbol{x}}_\alpha] = V_0[\boldsymbol{x}_\alpha] - \sum_{k,l=1}^{r} W_\alpha^{(kl)} (V[\boldsymbol{x}_\alpha]\nabla_{\mathbf{x}} F_\alpha^{(k)})(V[\boldsymbol{x}_\alpha]\nabla_{\mathbf{x}} F_\alpha^{(k)})^\top,$$
$$V_0[\hat{\boldsymbol{u}}] = \Big(\sum_{\alpha=1}^{N}\sum_{k,l=1}^{r} W_\alpha^{(kl)} (\nabla_{\mathbf{u}} F_\alpha^{(k)})(\nabla_{\mathbf{u}} F_\alpha^{(l)})^\top\Big)^{-1}. \quad (85)$$

The symbol $W_\alpha^{(kl)}$ has the same meaning as in eq. (7). It is easily seen that $V_0[\hat{\boldsymbol{x}}_\alpha]^-$ is a singular matrix of rank d whose domain is the tangent space to the optimally fitted manifold $\hat{\mathcal{S}}$ at $\hat{\boldsymbol{x}}_\alpha$.

B.2 Encoding Parameters

In order to quantize $\hat{\boldsymbol{u}}$, we introduce appropriate (generally curvilinear) coordinates (u_i), $i = 1, ..., p$, into the p-dimensional parameter space $\mathcal{U}$ and quantize it into a grid of width δu_i. Suppose $\hat{\boldsymbol{u}}$ is in a (curvilinear) rectangular region of sides L_i. There are $\prod_{i=1}^{p}(L_i/\delta u_i)$ grid vertices inside, so specifying one from these requires the code length

$$\log \prod_{i=1}^{p} \frac{L_i}{\delta u_i} = \log V_u - \sum_{i=1}^{p} \log \delta u_i, \quad (86)$$

where $V_u = \prod_{i=1}^{p} L_i$ is the volume of the rectangular region. We could reduce eq. (86) using a large width δu_i, but eq. (84) implies that replacing $\hat{\boldsymbol{u}}$ by the nearest vertex would increase the description length $\hat{J}/2\varepsilon^2$ by $(\delta\boldsymbol{u}, V_0[\hat{\boldsymbol{u}}]^{-1}\delta\boldsymbol{u})/2\varepsilon^2$ in the first order in ε, where we define $\delta\boldsymbol{u} = (\delta u_i)$. So, we choose such $\delta\boldsymbol{u}$ that minimizes the sum of $(\delta\boldsymbol{u}, V_0[\hat{\boldsymbol{u}}]^{-1}\delta\boldsymbol{u})/2\varepsilon^2$ and eq. (86). Differentiating this sum with respect to δu_i and letting the result be 0, we obtain

$$\frac{1}{\varepsilon^2}\Big(V_0[\hat{\boldsymbol{u}}]^{-1}\delta\boldsymbol{u}\Big)_i = \frac{1}{\delta u_i}, \quad (87)$$

where $(\,\cdot\,)_i$ designates the ith component. If the coordinate system of $\mathcal{U}$ is so taken that $V_0[\hat{\boldsymbol{u}}]^{-1}$ is diagonalized, eq. (87) reduces to

$$\delta u_i = \frac{\varepsilon}{\sqrt{\lambda_i}}, \quad (88)$$

where λ_i is the ith eigenvalue of $V_0[\hat{\boldsymbol{u}}]^{-1}$. It follows that the volume of one grid cell is

$$v_u = \prod_{i=1}^{p} \delta u_i = \frac{\varepsilon^p}{\sqrt{|V_0[\hat{\boldsymbol{u}}]^{-1}|}}. \quad (89)$$

Hence, the number of cells inside the region V_u is

$$N_u = \int_{V_u} \frac{d\boldsymbol{u}}{v_u} = \frac{1}{\varepsilon^p}\int_{V_u} \sqrt{|V_0[\hat{\boldsymbol{u}}]^{-1}|}\, d\boldsymbol{u}. \quad (90)$$

Specifying one from these requires the code length

$$\log N_u = \log \int_{V_u} \sqrt{|V_0[\hat{\boldsymbol{u}}]^{-1}|}\, d\boldsymbol{u} - \frac{p}{2}\log\varepsilon^2. \quad (91)$$

B.3 Encoding True Values

For quantizing the ML-estimators $\{\hat{\boldsymbol{x}}_\alpha\}$, we need not quantize the entire m-dimensional data space $\mathcal{X}$, because they are constrained to be in the optimally fitted d-dimensional manifold $\hat{\mathcal{S}}$ ($\subset \mathcal{X}$) specified by $\hat{\boldsymbol{u}}$, which we have already encoded. So, we only need to quantize $\hat{\mathcal{S}}$. To this end, we introduce appropriate curvilinear coordinates in it. Since each $\hat{\boldsymbol{x}}_\alpha$ has its own normalized covariance matrix $V_0[\hat{\boldsymbol{x}}_\alpha]$ (eqs. (85)), we introduce different coordinates $(\xi_{i\alpha})$, $i = 1, ..., d$, for each α. Then, they are quantized into a (curvilinear) grid of width $\delta\xi_{i\alpha}$.

Suppose $\hat{\boldsymbol{x}}_\alpha$ is in a (curvilinear) rectangular region of sides $l_{i\alpha}$. There are $\prod_{i=1}^{d}(l_{i\alpha}/\delta\xi_{i\alpha})$ grid vertices inside, so specifying one from these requires the code length

$$\log \prod_{i=1}^{d} \frac{l_{i\alpha}}{\delta\xi_{i\alpha}} = \log V_{x\alpha} - \sum_{i=1}^{d} \log \delta\xi_{i\alpha}, \quad (92)$$

where $V_{x\alpha} = \prod_{i=1}^{d} l_{i\alpha}$ is the volume of the rectangular region. We could reduce eq. (92) using a large width $\delta\xi_{i\alpha}$, but replacing $\hat{\boldsymbol{x}}_\alpha$ by its nearest vertex would increase the description length $\hat{J}/2\varepsilon^2$. Let $\delta\bar{\boldsymbol{x}}_\alpha$ be the m-dimensional vector that expresses the displacement $\{\delta\xi_{i\alpha}\}$ on $\hat{\mathcal{S}}$ in the (original) coordinates of $\mathcal{X}$. Eq. (84) implies that the increase in $\hat{J}/2\varepsilon^2$ is $(\delta\bar{\boldsymbol{x}}_\alpha, V_0[\hat{\boldsymbol{x}}_\alpha]^-\delta\bar{\boldsymbol{x}}_\alpha)/2\varepsilon^2$ in the first order in ε, so we choose such $\{\delta\xi_{i\alpha}\}$ that minimize the sum of $(\delta\bar{\boldsymbol{x}}_\alpha, V_0[\hat{\boldsymbol{x}}_\alpha]^-\delta\bar{\boldsymbol{x}}_\alpha)/2\varepsilon^2$ and eq. (92). Differentiating this sum with respect to $\delta\xi_{i\alpha}$ and letting the result be 0, we obtain

$$\frac{1}{\varepsilon^2}\Big(V_0[\hat{\boldsymbol{x}}_\alpha]^-\delta\bar{\boldsymbol{x}}_\alpha\Big)_i = \frac{1}{\delta\xi_{i\alpha}}. \quad (93)$$

Let the coordinates $(\xi_{i\alpha})$ be such that the d basis vectors at $\hat{\boldsymbol{x}}_\alpha$ form an orthonormal system. Also, let

the coordinates of $\mathcal{X}$ be such that at $\hat{\boldsymbol{x}}_\alpha \in \hat{\mathcal{S}}$ the m basis vectors consist of the d basis vectors of $\hat{\mathcal{S}}$ plus $m-d$ additional basis vectors orthogonal to $\hat{\mathcal{S}}$. Then, the first d components of $\delta\bar{\boldsymbol{x}}_\alpha$ coincide with $\{\delta\xi_{i\alpha}\}$, $i = 1, ..., d$; the remaining components are 0. If, furthermore, the coordinates $(\xi_{i\alpha})$ are so defined that $V_0[\hat{\boldsymbol{x}}_\alpha]^-$ is diagonalized, the solution $\delta\xi_{i\alpha}$ of eq. (93) is given by

$$\delta\xi_{i\alpha} = \frac{\varepsilon}{\sqrt{\lambda_{i\alpha}}}, \tag{94}$$

where $\lambda_{1\alpha}$, ..., $\lambda_{d\alpha}$ are the d positive eigenvalues of $V_0[\hat{\boldsymbol{x}}_\alpha]^-$. It follows that the volume of one grid cell is

$$v_{x\alpha} = \prod_{i=1}^{d} \delta\xi_{i\alpha} = \frac{\varepsilon^d}{\sqrt{|V_0[\hat{\boldsymbol{x}}_\alpha]^-|_d}}, \tag{95}$$

where $|V_0[\hat{\boldsymbol{x}}_\alpha]^-|_d$ denotes the product of its d positive eigenvalues. Hence, the number of cells inside the region $V_{x\alpha}$ is

$$N_\alpha = \int_{V_{x\alpha}} \frac{d\boldsymbol{x}}{v_{x\alpha}} = \frac{1}{\varepsilon^d} \int_{V_{x\alpha}} \sqrt{|V_0[\hat{\boldsymbol{x}}_\alpha]^-|_d}\, d\boldsymbol{x}. \tag{96}$$

Specifying one from these requires the code length

$$\log N_\alpha = \log \int_{V_{x\alpha}} \sqrt{|V_0[\hat{\boldsymbol{x}}_\alpha]^-|_d}\, d\boldsymbol{x} - \frac{d}{2} \log \varepsilon^2. \tag{97}$$

B.4 Geometric MDL

From eqs. (91) and (97), the total code length for $\{\hat{\boldsymbol{x}}_\alpha\}$ and $\hat{\boldsymbol{u}}$ becomes

$$\sum_{\alpha=1}^{N} \log \int_{V_{x\alpha}} \sqrt{|V_0[\hat{\boldsymbol{x}}_\alpha]^-|_d}\, d\boldsymbol{x} + \log \int_{V_u} \sqrt{|V_0[\hat{\boldsymbol{u}}]^{-1}|} d\boldsymbol{u} - \frac{Nd+p}{2} \log \varepsilon^2 \tag{98}$$

The accompanying increase in the description length $\hat{J}/2\varepsilon^2$ is $(\delta\bar{\boldsymbol{x}}_\alpha, V_0[\hat{\boldsymbol{x}}_\alpha]^- \delta\bar{\boldsymbol{x}}_\alpha)/2\varepsilon^2 + (\delta\boldsymbol{u}, V_0[\hat{\boldsymbol{u}}]^{-1}\delta\boldsymbol{u})/2\varepsilon^2$ in the first order in ε. If we substitute eqs. (88) and (94) together with $V_0[\hat{\boldsymbol{x}}_\alpha]^- = \mathrm{diag}(1/\lambda_{1\alpha}, ..., 1/\lambda_{d\alpha}, 0, ..., 0)$ and $V_0[\hat{\boldsymbol{u}}]^{-1} = \mathrm{diag}(1/\lambda_1, ..., 1/\lambda_p)$, this increase is

$$\frac{(\delta\bar{\boldsymbol{x}}_\alpha, V_0[\hat{\boldsymbol{x}}_\alpha]^- \delta\bar{\boldsymbol{x}}_\alpha)}{2\varepsilon^2} + \frac{(\delta\boldsymbol{u}, V_0[\hat{\boldsymbol{u}}]^{-1}\delta\boldsymbol{u})}{2\varepsilon^2} = \frac{Nd+p}{2}. \tag{99}$$

Since eqs. (88) and (94) are obtained by omitting terms of $o(\varepsilon)$, the omitted terms in eq. (99) are $o(1)$. It follows that the total description length is

$$\frac{\hat{J}}{2\varepsilon^2} - \frac{Nd+p}{2} \log \varepsilon^2 + \sum_{\alpha=1}^{N} \log \int_{V_{x\alpha}} \sqrt{|V_0[\hat{\boldsymbol{x}}_\alpha]^-|_d}\, d\boldsymbol{x} + \log \int_{V_u} \sqrt{|V_0[\hat{\boldsymbol{u}}]^{-1}|} d\boldsymbol{u} + \frac{Nd+p}{2} + o(1). \tag{100}$$

Multiplying this by $2\varepsilon^2$, which does not affect model selection, we obtain

$$\hat{J} - (Nd+p)\varepsilon^2 \log \varepsilon^2 + 2\varepsilon^2 \Big(\sum_{\alpha=1}^{N} \log \int_{V_{x\alpha}} \sqrt{|V_0[\hat{\boldsymbol{x}}_\alpha]^-|_d}\, d\boldsymbol{x} + \log \int_{V_u} \sqrt{|V_0[\hat{\boldsymbol{u}}]^{-1}|} d\boldsymbol{u} \Big) + (Nd+p)\varepsilon^2 + o(\varepsilon^2). \tag{101}$$

B.5 Scale Choice

In practice, it is difficult to use eq. (101) as a criterion because of the difficulty in evaluating the third term. If we note that $-\log \varepsilon^2 \gg 1$ as $\varepsilon \to 0$, we may omit terms of $O(\varepsilon^2)$ and define

$$\text{G-MDL} = \hat{J} - (Nd+p)\varepsilon^2 \log \varepsilon^2. \tag{102}$$

This is the form suggested by Matsunaga and Kanatani [34]. However, the problem of scale arises. If we multiply the unit of length by, say, 10, both ε^2 and $\hat{J}$ are multiplied by 1/100. Since N, d, and p are nondimensional constants, G-MDL should also be multiplied by 1/100. But $\log \varepsilon^2$ reduces by $\log 100$, which could affect model selection[3]. In eq. (101), in contrast, the influence of scale is canceled between the second and third terms.

To begin with, the logarithm can be defined only for a nondimensional quantity, so eq. (102) should have the form

$$\text{G-MDL} = \hat{J} - (Nd+p)\varepsilon^2 \log \Big(\frac{\varepsilon}{L}\Big)^2, \tag{103}$$

where L is a reference length. In theory, it can be determined from the third term of eq. (101), but its evaluation is difficult. So, we adopt a practical compromise, choosing a scale L such that $\boldsymbol{x}_\alpha/L$ is $O(1)$. We may interpret this as introducing a prior distribution in a region of volume L^m in the data space $\mathcal{X}$. For example, if $\{\boldsymbol{x}_\alpha\}$ are image coordinate data, we can take L to be the image size. We call eq. (103) the *geometric MDL*.

Recall that for asymptotic analysis as $\varepsilon \to 0$, it is essential to fix the scale of the normalized covariance matrix $V_0[\boldsymbol{x}_\alpha]$ in eq. (4) in such a way that the noise level ε is much smaller than the data themselves (Remark 2). So, we have $-\log(\varepsilon/L)^2 \gg 1$. If we use a different scale $L' = \gamma L$, we have $-\log(\varepsilon/L')^2 = -\log(\varepsilon/L)^2 + \log\gamma^2 \approx -\log(\varepsilon/L)^2$ as long as the scale is of the same order of magnitude. It has been confirmed that the scale choice does not practically affect model selection in most applications. Nonetheless, the introduction of the scale is a heuristic compromise, and more studies about this will be necessary.

[3]The preference is unchanged if the candidate models have the same d and p, but we usually compare models of different d and p.