Marginal Notes for Kanatani's 'Statistical Optimization for Geometric Computation'

editor: Leo Dorst

November 6, 2009

1 Introduction (Self)

Explanation of the book title:

- geometric: models and constraints
- optimization: theoretical accuracy bounds
- statistical: reliability
- computation: efficiency issues

Language:

- small noise in geometric models:
- perturbation theory on manifolds

2 Fundamentals of Linear Algebra (Leo Dorst)

Much for future reference, but let us identify newish elements.

2.1 Vector and Matrix Calculus

- **Pg. 33**: Cute notation for the *cross product operator*:

$$\text{more usual} \rightarrow \quad \mathbf{a}^{\times} \equiv \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} = \mathbf{a} \times \mathbf{I} \quad \leftarrow \text{new to me}$$

  Then

$$\mathbf{a} \times \mathbf{x} = \mathbf{a}^{\times}\,\mathbf{x} = (\mathbf{a} \times \mathbf{I})\,\mathbf{x}.$$

  Of course the cross product is very 3-D, and has rather awkward linear algebra:

$$\mathbf{a} \mapsto f(\mathbf{a}), \quad \mathbf{b} \mapsto f(\mathbf{b})$$

  but

$$f(\mathbf{a} \times \mathbf{b}) = \det(f) f^{-\top}(\mathbf{a} \times \mathbf{b}) \neq f(\mathbf{a}) \times f(\mathbf{b}).$$

  So *the transformation of a normal vector is not the normal vector of the transformation.*[1]

- **Pg. 35**: Beware of Kanatani's notation: $P_{\mathbf{n}}$ is not the projection onto $\mathbf{n}$ but the projection to the hyperplane characterized by the normal vector $\mathbf{n}$ (so perpendicular to $\mathbf{n}$). For the projection onto the $\mathbf{n}$-line he would use $P^{\mathbf{n}}$ (see just below his (2.122)).

- **Pg. 36**: In (2.56) one would expect Kanatani to use *Rodrigues' formula* for a rotation matrix, since it matches his cross product treatment. For a unit rotation axis $\mathbf{n}$, some equivalent forms are:

$$\begin{aligned} R &= \mathbf{I}\cos(\Omega) + \sin(\Omega)\,\mathbf{n}^{\times} + \big(1 - \cos(\Omega)\big)\mathbf{n}\mathbf{n}^{\top} \qquad \text{(the usual form)} \\ &= \mathbf{I} + \sin(\Omega)\,\mathbf{n}^{\times} + \big(1 - \cos(\Omega)\big)(\mathbf{n}^{\times})^2 \qquad \text{(a variation)} \\ &= \mathbf{I} + \sin(\Omega)\,(\mathbf{n} \times \mathbf{I}) + \big(1 - \cos(\Omega)\big)(\mathbf{n} \times \mathbf{I})^2 \qquad \text{(a Kanatani-like form).} \end{aligned}$$

  Still, there are more convenient rotation representations such as quaternions (fewer parameters, easier constraints), and rotors (generalized quaternions in geometric algerba)[2]. Modern optimization of rotations often uses quaternions.

[1] Buy [3], and from now on use $\mathbf{a} \wedge \mathbf{b}$ instead (since $f(\mathbf{a} \wedge \mathbf{b}) = f(\mathbf{a}) \wedge f(\mathbf{b})$ and it works in n-D). Geez, how often do I have to tell you?

[2] In Euclidean geometric algebra, the universal rotation operator (rotor) is $R = \exp(\mathbf{n}^{\star}\Omega/2)$. In conformal geometric algebra, the universal rotation operator (rotor) is $R = \exp(\Lambda^{\star}\Omega/2)$, where Λ is the unit rotation axis, not necessarily through the origin.

[[[**insert Figure**]]]

Figure 1: The four fundamental subspaces of a linear transformation.

2.2 Eigenvalue Problem

In all the treatment of linear transformations and their matrices, it it is helpful to realize that there are 4 fundamental spaces to each $m \times n$ matrix A of rank r (i.e. to each linear mapping):

- the $n - r$ dimensional *kernel* of A (aka *nullspace* of A)
- the r dimensional *image of* A (aka *column space* of A, or *range* of A)
- the $m - r$ dimensional *kernel* of $A^\top$ (aka *nullspace* of $A^\top$)
- the r dimensional *image* of $A^\top$ (aka *row space* of A, or *range* of $A^\top$)

The mapping A is invertible only for its r-dimensional image. The inverse there is the pseudo-inverse. The SVD brings the latter into a particularly convenient form, since it gives each of the spaces an orthonormal basis. We recommend reading the SVD part in section (2.3) first, since it is more general, and then going back to special matrices such as square $n \times n$ and symmetric square matrices treated in section (2.2). It gives more generic understanding.

2.3 Linear Systems and Optimization

- **Pg. 45** Remember what is behind the SVD: find a frame of orthogonal vectors $\mathbf{u}_i$ whose images under an $m \times n$ matrix A are also orthogonal. Such a frame can be found as the eigenvectors of $A^\top A$. This is easily proved:

$$(A\mathbf{u}_i) \cdot (A\mathbf{u}_j) = (A\mathbf{u}_i)^\top (A\mathbf{u}_j) = \mathbf{u}_i^\top A^\top A \mathbf{u}_j = \mathbf{u}_i^\top \sigma_j \mathbf{u}_j = \sigma_j \, \mathbf{u}_i \cdot \mathbf{u}_j. \quad (1)$$

  So the originals $\mathbf{u}_i$ and $\mathbf{u}_j$ are orthogonal precisely when the images $A\mathbf{u}_i$ and $A\mathbf{u}_j$ are.[3]

  Take the $\mathbf{u}_i$ $(i = 1, \cdots, n)$ to be unit vectors and define their images through

$$A\mathbf{u}_i = \lambda_i \, \mathbf{v}_i, \quad (2)$$

  with the unit vectors $\mathbf{v}_i$ $(i = 1, \cdots, m)$ forming an orthonormal basis. Then λ_i are the singular values, and $\lambda_i = \sqrt{\sigma_i}$. *(See remark about unusual Kanatani notation above).*

  Since $A^\top A$ is symmetric (why?), the σ_i are real; since $A^\top A$ is semi-positive definite (why?), the σ_i are non-negative; so the singular values λ_i are all

[3] *Note that Kanatani swaps the notation of λ_i and σ_i: usually the σ_i are the singular values (that is why they have an s-like symbol!) and the λ_i the eigenvalues of $A^\top A$ (it is so in [2]; he also has $\mathbf{u}$ and $\mathbf{v}$ opposite...*

[[[**insert Figure**]]]

Figure 2: The SVD visualized.

non-negative and can be ordered. We define the $m \times n$ matrix Λ to be zero except on its diagonal, where the singular values are in descending order. For a 2×3 matrix A, for instance:

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \end{pmatrix}.$$

We can convert eq.(2) to matrix form as:

$$A\,U = V\,\Lambda, \qquad \text{so} \qquad A = V\,\Lambda\,U^\top,$$

with V and U orthogonal $m \times m$ and $n \times n$ matrices, respectively, ordered to correspond to the singular values. Details see Bretscher [2]. A figure shows these aspects: A is a diagonal Λ (only a stretching of the axes) in a well-chosen representation (related to the eigenvectors of $A^\top A$, as we showed). Dimensions can also be reduced (this is the kernel of A) or added (this is the kernel of $A^\top$). One gets the original A back by rotating the domain of Λ by U, and the range of Λ by V.

- **Pg. 47**: So, now the *pseudoinverse* in these terms: the 4 fundamental spaces of the $m \times n$ matrix A of rank r (we use ONB for OrthoNormal Basis):
  - the r-D image of A ($\mathcal{R}_\mathcal{A}$) has as ONB the first r columns of V
  - the r-D image of $A^\top$ has as ONB the first r columns of U
  - the $(n-r)$-D kernel of A ($\mathcal{N}_\mathcal{A}$) has as ONB the last $n-r$ columns of U
  - the $(m-r)$-D kernel of $A^\top$ has as ONB the last $m-r$ columns of V

  Only the first r entries on the diagonal of Λ are non-zero. So we can *not* quite do the naive:

$$A^{-1} = (V\Lambda U^\top)^{-1} = U\Lambda^{-1}V^\top,$$

  but it is close; we should do that for the first r entries; and of course Λ need not be square. Therefore the pseudoinverse of A is obtained by changing entry λ_i to $1/\lambda_i$ and swapping the roles of U and V:

$$A^- = \sum_{i=1}^{r} \frac{\mathbf{u}_i \mathbf{v}_i^\top}{\lambda_i}. \qquad \text{Kanatani (2.121)}$$

  The rank-constrained inverse is just a variation on this theme, for numerical applications.

- **Pg. 49**: Referring back to a remark Kanatani makes in the introduction, least squares fitting of a line to data minimizes the sum of the squares of the vertical (y) distances and is not suitable for isotropic geometry. This method needs to be modified to minimize the perpendicular distances to the line independent of a coordinate system. He will treat line fitting in Section 10.1.

2.4 Matrix and Tensor Algebra

This is just notation, for now. To see if you understand, derive the factor of 2 in (2.199).

3 Probabilities and Statistical Estimation (Gwenn Englebienne)

Introductory remarks (see Gwenn's slides):

- Frequentist approach
  - The data comes from a distribution, let us find as best as we can which distribution that was.
  - Find *estimators* for parameters, and try to figure out how good these estimators are.
- Bayesian approach.
  - The data could have come from any number of distributions, let us find what those distributions could have been, and how likely they are.
  - Using Bayes rule does not make an approach Bayesian.

3.1 Probability Distributions

- **Pg. 61**: Expectation is a more general concept, of a function f:

  $$E[f(\mathbf{x})] = \int f(\mathbf{x})\, p(\mathbf{x})\, d\mathbf{x}.$$

  The *mean* is then the expectation of the data, setting f to the identity function.

- There is nothing yet in this chapter on estimating mean and the variance from the data. An unbiased estimator for the mean is as you would expect:

  $$\hat{\bar{\mathbf{x}}} = \tfrac{1}{N} \sum_i \mathbf{x}_i.$$

  An unbiased estimator for the variance has an unexpected normalization:

  $$\hat{V}[\mathbf{x}] = \tfrac{1}{N-1} \sum_i (\mathbf{x}_i - \bar{\mathbf{x}})\, (\mathbf{x}_i - \bar{\mathbf{x}})^\top.$$

  For a proof, see [[[**??**]]]. It should be noted that this is the estimator you use when you don't know the mean; if you do know the mean, the unbiased estimator does use the division by N.

- **Pg. 61**: Definition of *uncorrelated*: the covariance of two variables is zero. $E[(x - E[x])\,(y - E[y])] = 0$. If both variables have zero mean, they are uncorrelated if and only if $E[x\,y] = E[x]\,E[y]$.

  Definition of *independent*: the distribution of one variable does not affect that of the other: $p(x, y) = p(x)p(y)$.

  Independent always implies uncorrelated. For a normal distrution, uncorrelated implies independent, so that the two terms are equivalent then.

- **Pg. 63**: The derivative matrix occurring in (3.15) is the *Jacobian*. In (3.17) he calls '$P_{\bar{\mathbf{n}}}$ the projection along $\bar{\mathbf{n}}$', this ambiguous phrasing should be interpreted as 'the projection to the subspace perpendicular to $\bar{\mathbf{n}}$'.

- **Pg. 64**: In (3.27) that awkward projection notation again: $P_{\mathcal{N}}$ is the projection to the orthogonal complement of the null space of the tangent space, and therefore projection to the tangent space.

  Spectral decomposition of the covariance matrix is widely used. A typical application of it is the so-called Karhunen-Loève transform, better known as Principal Component Analysis (PCA). (BTW: that's *principal*, not *principle*)

3.2 Manifolds and Local Distributions

- **Pg. 67**: Equation (3.29) is an approximation. The proper way for a manifold would be to define $\bar{\mathbf{x}}$ as the location such that, for instance, the sum of squared distances *along the manifold* would be zero. $E(\mathbf{x} - \bar{\mathbf{x}})$ is not measured along, and therefore not an intrinsic measure. To solve in the proper way (which Gijs Dubbelman has executed for rotation estimation), is to define geodesic length preserving mappings between manifold and tangent space at a location (these are called the exponential and logarithmic maps), and do an iterative algorithm to find the location $\bar{\mathbf{x}}$ as defined above. This method converges under certain conditions on the manifold, such as local compactness. However, Bishop [1] also uses (3.29) [[[**(Mises distribution term, Olaf?)**]]].

- **Pg. 68**: Try to derive (3.33) from (3.32). We could not, though $[\ell^\times]\,[\ell^\times]^\top = I - \ell\,\ell^\top$ is close. What seems to be going on is that Kanatani defines a quantitiy that has the same covariance as the target quantity but has a simpler form. Still, this is geometry, so precisely the point where we would have appreciated more detail.

3.3 Gaussian Distributions and χ^2 Distributions

- **Pg. 68**: Mahalanobis distance: Σ^{-1} effectively rescales the ellipsoids to spheres, or can be viewed as the metric involved in the dot product.

- **Pg. 73**: We can interpret (3.59) and also mode$[R] = 2 - r$ as: even for large r, the χ^2 distribution remains localized, around r with standard deviation $\sqrt{2r}$. In the χ^2-test, this helps distinguish the distributions with significantly different r. But Gwenn still calls it an 'ugly hack'.

- **Pg. 77**: Misstatement: the likelihood is defined as $p(\mathbf{x}|\boldsymbol{\theta}) = f(\boldsymbol{\theta})$, that is, viewed as a function of the parameters, *not* of the data.

3.4 Statistical Estimation for Gaussian Models

- **Maximum Likelihood (ML) learning:** Try to maximize prob of data given parameters, not prob of param given data $\Pi_i p(\mathbf{y}_i|\theta)$.

  For example, if we have N i.i.d. Gaussian distributed datapoints $\mathbf{y}_{1:N}$, the probability of the data given the parameters is

$$p(\mathbf{y}_{1:N}|\boldsymbol{\theta}) = \prod_{i=1}^{N} \mathbf{n}(\mathbf{y}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{3}$$

  where $\mathbf{n}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ indicates the Gaussian distribution with parameters mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. Maximising this directly is difficult because of the product. However since e^x is a monotonically increasing function, we can simply maximise the log-likelihood instead:

$$\log p(\mathbf{y}_{1:N}|\boldsymbol{\theta}) = \sum_{i=1}^{N} \log \mathbf{n}(\mathbf{y}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{4}$$

  Now we can take the first derivative and set this equal to zero. For illustration, we do this for the mean:

$$0 = \frac{\partial}{\partial \boldsymbol{\mu}} \left[\sum_{i=1}^{N} (\mathbf{y}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{y}_i - \boldsymbol{\mu}) + \text{const} \right] \tag{5}$$

$$0 = \sum_{i=1}^{N} \frac{\partial}{\partial \boldsymbol{\mu}} (\mathbf{y}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{y}_i - \boldsymbol{\mu}) \tag{6}$$

$$0 = -2\boldsymbol{\Sigma}^{-1} \sum_{i=1}^{N} (\mathbf{y}_i - \boldsymbol{\mu}) \tag{7}$$

$$0 = \sum_{i=1}^{N} \mathbf{y}_i - \sum_{i=1}^{N} \boldsymbol{\mu} \tag{8}$$

$$\boldsymbol{\mu} = \frac{\sum_{i=1}^{N} \mathbf{y}_i}{N} \tag{9}$$

- **Maximum a posteriori (MAP) learning:** Problem ML: since we only have final amounts of data, the distribution of the observed data is never the same as the distribution the data would have if we could observe it all.

  We can therefore introduce prior knowledge about the parameters. That is, we optimise $p(\boldsymbol{\theta}|\mathbf{y})$, which we obtain by:

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathbf{y})} \tag{10}$$

$p(\mathbf{y})$ can be computed but is constant anyway, and does not affect the maximisation. Maximum a posteriori learning therefore reduces to maximising $\log p(\mathbf{y}_{1:N}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})$.

- **The Bayesian approach:** Here, we don't maximise anything. We know that the data we observe may have come from many different distributions, but not all these distributions are equally likely. Instead of finding "the most probable distribution" for given observations, we keep a distribution over distributions.

  As an example, consider coin toss giving head (Carsten's phrasing) $H(=1)$ or tail $T(=0)$.[4] We can parameterize the probability of a set of datapoints $y = \{HHT\}$ with the Bernouilli distribution:

  $$p(y_i|\theta) = \theta^{y_i}(1-\theta)^{y_i} \tag{11}$$

  This gives us that $p(H|\theta) = \theta$ and $p(T|\theta) = 1-\theta$. (As an exercise, confirm that the ML estimate of $\theta = \frac{\sum_i y_i}{N}$.) The Bayesian approach is to say that we don't know θ, and we will never know it. But we do know that some values of θ are more likely than others. For example, if we see a sequence of a hundred H and no T, we can probably infer that we have a cheater on our hands, and the coin isn't fair (because of the law of large numbers). That is, we can associate a probability with each value of θ.[5]

  So how can we compute $p(\theta|y_{1:N})$? We use Bayes' rule:

  $$p(\theta|\{H\}) = \frac{p(H|\theta)\,p(\theta)}{p(H)} \tag{12}$$

  $$p(H) = \int_\theta p(H|\theta)p(\theta)d\theta \tag{13}$$

  We first need to make assumptions about the value of θ. In this case, let us assume that we don't know anything, except that $0 \leqslant \theta \leqslant 1$. The corresponding distribution over θ is plotted in Figure 3(a). If we observe a H, we then use Bayes' rule to update the distribution over θ, resulting in Figure 3(b). As more data is observed, the distribution over θ evolves as depicted in the subsequent figures.

- Notice that *Bayesian inference*, *Maximum a posteriori* and *Maximum Likelihood* are identical in the limit of infinite amounts of data. *Maximum a Posteriori* equals *Maximum Likelihood* if we assume an uniform prior (or have infinite amounts of data).

- **Pg. 81**: Kalman filter: In (3.104) the B appears superfluous, since if $\mathbf{v}$ is Gaussian distributed, so is B in a straightforward manner, and vice versa.

[4] We do not work the example out analytically, because the functional form of the distributions becomes complicated and does not add anything to the understanding.

[5] In fact, the probability of any value of θ is zero, because θ is a continuous quantity. We therefore associate a probability *density* with it, which is normalised so that the integral between $-\infty$ and ∞ equals 1. Details can be found in any basic text on probabilities.

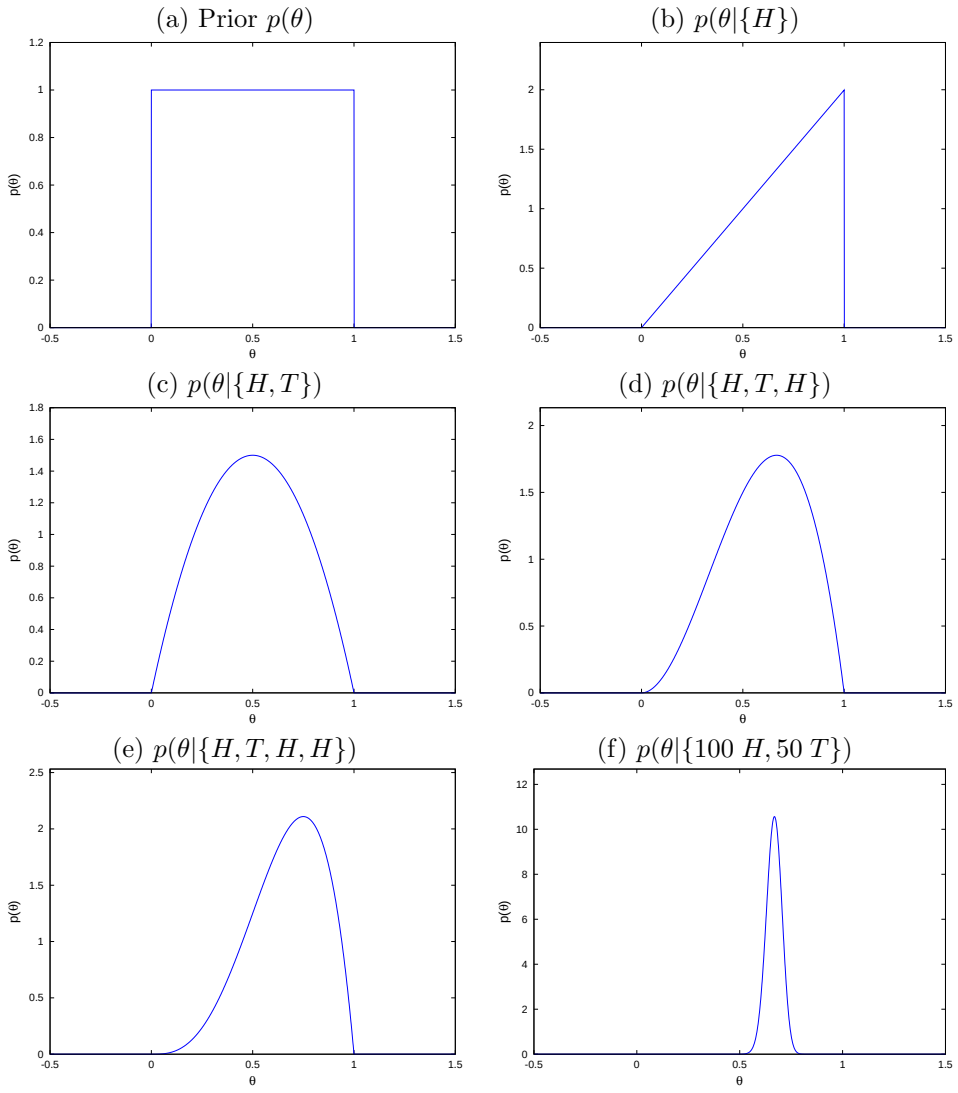

Figure 3: Update of $p(\theta|y)$ as more data is observed.

- **Pg. 81** To use Kalman, you need to know the noise distributation **v**, **w** and a model **A**, **C**. Do all people using the Kalman filter really know that? Usually not. You have to estimate them, using the EM algorithm, or the Bayesian way (closed form). For more details, Chapter 13 in [1]

  - About Kalman Filter techniques:

    A Kalman filter is just a *Bayesian Network* (though historically not so described). Notice that at each time step, the Kalman filter uses MAP (not Bayesian inference.)

    Bishop [1] Ch. 8 and 13 describe how to use a Kalman smoother, using a small amount of future data considerably improves the results. This is well-known from Linear Dynamical Systems, but few of our Kalman practitioners knew it.
  - If the model is not truly linear use an *Extended Kalman Filter* (EKF): first order approximation around the means, using Jacobians see []. [[[**Olaf, reference?**]]]
  - *Particle Filters* are similar to Kalman filters, but instead of representing the probability of the observation by the parameters of a Gaussian, they do not make any assumption on the distribution and represent it with samples (particles). This works for any "transfer function," and with any form of noise, but is more computationally intensive and requires enough samples.

- **Pg. 83**: Misleading formulation: minizing the sum squared error is really assuming (isotropic) Gaussian noise.

3.5 General Statistical Estimation

- **Pg. 84**: ℓ = Jacobian of log likelihood.
  Fischer information matrix is second derivative of log likelihood, this is the lower bound on the variance of the estimator. The *efficient estimator* is called efficient because it is the best estimate for the least amount of data.

3.6 Akaike Information Criterion

- **Pg. 91**: In (3.55), the $E^*[]$ and $E[]$ should be read as no more than the specification of the order of integration in the marginalization of distributions when computing I.

- **Pg. 93**: Expectation of the variance of the error of the parameter over the data you have not seen yet leads to:

$$AIC = 2m' - 2\sum_i \log p(x_i; \hat{\theta})$$

This is used a lot, but it is not the only information criterion. The *Bayesian Information Criterion* (BIC) penalizes complex models slightly differently (and less).

- Formal relationship to Kolmogorov complexity? Hard to compute. MDL more practical.

4 Representation of Geometric Objects (Carsten Cibura)

Kanatani could have discussed alternative representations and the consequences for the convenience of the noise modelling. After all, picking the right representation is part of getting to practical computations. Now the Chapter is more like a lookup table, and there is more at stake. The unifying representation of geometric algebra, however, has not quite had the makeover in terms of the covariances yet, though a lot is already done in [4] in tensor notation.

4.1 Image Points and Image Lines

- Geometrical representation: often handy to have more coordinates than degrees of freedom; then the element resides on a manifold.
- **pg. 95** Normalization strange of point forcing 1 (we would allow any non-zero multiple). Could have mentioned that directions are represented as vectors having 0 as last coordinate, since we will see those (as $\Delta\mathbf{x}$). Seems to miss the convenience of unification of special cases by including the points at infinity (though there are some numerical issues). No need to think of 'image plane' yet, it is just the 2D geometry of any plane.
- **pg. 98** Strangely inconstent treatment relative to the planes in Section 4.3 in the normalization; probably caused by Kanatani's desire to split off 3D geometry in all cases, rather then real versus representational dimensions. One does find papers in which a line vector is normalized to have its normal vector unity, so in (4.7) $A^2 + B^2 = 1$. In that case $\mathbf{n}$ lies on a unit cylinder in the $\mathbf{k}$-direction.
- **pg. 98**: Sign discrimination can be useful in a representation, to represent oriented lines and planes (for instance, as the locally flat approximation to the extrema of a gradient separating 'inside' and 'outside' of an object).
- **pg. 100** Working out a term like $\mathbf{n}_1 \times V[\mathbf{n}_2] \times \mathbf{n}_1$ to find out what this matrix looks like is not extremely enlightening:

$$\begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} \times \begin{pmatrix} v_{11} & v_{21} & v_{31} \\ v_{12} & v_{22} & v_{32} \\ v_{13} & v_{23} & v_{33} \end{pmatrix} \times \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} =$$

$$= \begin{pmatrix} n_2^2 v_{33} + n_3^2 v_{22} - n_2 n_3 (v_{23} + v_{32}) & -n_3^2 v_{12} + n_1 n_2 v_{33} + n_2 n_3 v_{13} - n_1 n_3 v_{23} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

- **pg. 101**: The renormalization in (4.16) leads to the projection operator in (4.17). Writing $\mathbf{x} \equiv \mathbf{x}_1 \times \mathbf{x}_2$ and $\mathbf{a}$ for $\Delta\mathbf{x}$, we find to first order in $\mathbf{a}$:

$$\begin{aligned}
\frac{\mathbf{x}+\mathbf{a}}{\|\mathbf{x}+\mathbf{a}\|} &= \frac{\mathbf{x}+\mathbf{a}}{\sqrt{\|\mathbf{x}\|^2 + 2\mathbf{a}\cdot\mathbf{x} + \|\mathbf{a}\|^2}} \\
&\approx \frac{\mathbf{x}+\mathbf{a}}{\|\mathbf{x}\|\sqrt{1 + 2\mathbf{a}\cdot\mathbf{x}/\|\mathbf{x}\|^2}} \\
&\approx \frac{1}{\|\mathbf{x}\|}(\mathbf{x}+\mathbf{a})(1 - \mathbf{a}\cdot\mathbf{x}/\|\mathbf{x}\|^2) \\
&\approx \frac{1}{\|\mathbf{x}\|}(\mathbf{x}+\mathbf{a} - (\mathbf{a}\cdot\mathbf{x})\mathbf{x}/\|\mathbf{x}\|^2) \\
&= \frac{\mathbf{x}}{\|\mathbf{x}\|} + \frac{P_{\mathbf{x}}(\mathbf{a})}{\|\mathbf{x}\|}.
\end{aligned}$$

  This is precisely what one would expect from a simple sketch: only the tangential component of $\mathbf{a}$ matters, and gets rescaled by $\|\mathbf{x}\|$.

4.2 Space Points and Space Lines

- **pg. 104:** $\mathbf{p}$ equals $\mathbf{m}$ in the line representation as geometrical concept, but has different normalization (4.34) versus (4.30).

4.3 Space Planes

- **pg. 109:** In (4.63) we find the normalization of space planes inconsistent with planar lines; even though they are both hyperplanes whose direction may be characterized by a (unit) normal vector in their resident space.

- **pg. 113:** Conspicuously absent is an error analysis of the *Joins* characterizations.

4.4 Conics

We are not going to use the conics and quadrics as geometric objects (for intersection or other operations), we ultimately only need them to describe covariances; so this chapter could have started from there. In that case, perhaps not all the aspects would need to be covered.

- **pg. 104** Eigenvalue analysis of the symmetric matrix gives principal axes and values. That could have been mentioned, after all it is in Chapter 2.

4.5 Space Conics and Quadrics

- **pg. 123** Compare (4.99) and (4.112): using σ_i for different aspects of the same quadric-like element is rather confusing, so mind the context.

- **pg. 122** Couple this back to χ^2 [[[**Olaf link?**]]]

5 Geometric Correction (Christos Dimitrikakis)

Christos has made extensive slides on his points, consult them at the webiste [5].

5.1 General Theory

For all the applications Kanatani does in this chapter, you do not need more than the one on linear constraints (5.1.6). Encode the constraint in the form (5.41) and put it in (5.44).

5.2 Corrections

All the remaining sections have the same structure, and follow mostly the same pattern. We do Image Points, the remarks apply to the other cases as well.

- **pg. 144**: Image points: now the constraint (5.49) give in (5.41): $\mathbf{A}_1 = I$, $\mathbf{A}_2 = -I$, $\mathbf{b} = 0$. Then (5.52) follows immediately.
- **pg. 144**: Note that in (5.51), the statement on $\Delta x_1, \Delta x_2$ just means that they are direction vectors, with a homogeneous coordinate component of zero.
- **pg. 145**: The use of a posteriori here and in the other treatment is not standard statistical usage. He just means 'after the corrections'.
- **pg. 145**: Christos remarks that hypothesis testing in this manner is naive. A test like χ^2 can only be used to reject a hypothesis correctly, not to confirm it. Kanatani should have formulated the hypothesis that the points are further apart than ϵ, and used a χ^2 test (or otherwise) to reject that with a specified confidence.
- **pg. 147**: How to get (5.75)? From the linearized constraint (5.74), we determine by comparison with (5.43) $\mathbf{A}_\mathbf{x} = \bar{\mathbf{n}}^\top$ and $\mathbf{A}_\mathbf{n} = \bar{\mathbf{x}}^\top$. Then $\mathbf{W}$ in (5.45) is a scalar, which is easy to invert:

$$\begin{aligned} \mathbf{W} &= (\bar{\mathbf{n}}^\top V[\mathbf{x}]\bar{\mathbf{n}} + \bar{\mathbf{x}}^\top V[\mathbf{n}]\bar{\mathbf{x}})_1^- \\ &= 1/(\bar{\mathbf{n}}^\top V[\mathbf{x}]\bar{\mathbf{n}} + \bar{\mathbf{x}}^\top V[\mathbf{n}]\bar{\mathbf{x}}) \end{aligned}$$

  We also find that the right hand side of (5.43) is $\mathbf{n} \cdot \mathbf{x}$ (it is equal to the right hand side of (5.74)). Putting it all together using (5.44), we therefore obtain:

$$\Delta\mathbf{x} = \frac{V[\mathbf{x}]\bar{\mathbf{n}}(\mathbf{n} \cdot \mathbf{x})}{\bar{\mathbf{n}}^\top V[\mathbf{x}]\bar{\mathbf{n}} + \bar{\mathbf{x}}^\top V[\mathbf{n}]\bar{\mathbf{x}}}$$

  which is almost (5.75), barring the bars. Now, we don't have $\bar{\mathbf{n}}$ available until we have made the correction to $\mathbf{n}$, and we do not have $\Delta\mathbf{n}$ until we

have $\bar{\mathbf{x}}$, which requires $\Delta\mathbf{x}$, which requires ... et cetera. Of course to first order, $\bar{\mathbf{n}} \approx \mathbf{n}$, so we can substitute $\mathbf{n}$ for $\bar{\mathbf{n}}$, and that is probably what Kanatani did; but it would be good to have that in writing.

Geometrically, in the special case that $V[\mathbf{x}] = V[\mathbf{n}] = I$, and $\mathbf{n} \cdot \mathbf{n} = \mathbf{x} \cdot \mathbf{x} = 1$, we would expect the total correction that needs to be made to get perpendicularity of $\bar{\mathbf{n}}$ and $\bar{\mathbf{x}}$ to be of the size $(\mathbf{x} \cdot \mathbf{n})$, and to be distributed equally among the $\Delta\mathbf{x}$ and $\Delta\mathbf{n}$; which is indeed what (5.75) gives: $\Delta\mathbf{x} = \frac{1}{2}(\mathbf{x} \cdot \mathbf{n})\mathbf{n}$ and $\Delta\mathbf{n} = \frac{1}{2}(\mathbf{n} \cdot \mathbf{x})\mathbf{x}$.

6 3-D Computation by Stereo Vision (Daniël Fontijne)

- **pg. 174:** The *essential matrix* gives the relationship between the gemetrical elements in terms of their coordinates in mm in the frame of one of the cameras; this is related to the *fundamental matrix*, which does essentially (or fundamentally) the same thing but expressed in image coordinates (pixels). For the correspondence between the two you need the internal calibration matrix with the internal parameters.

- **pg. 177:** Variations of local features can also be caused by the kind of structure that is matched and the noise propertties of the detection algorithms (like SIFT). They can give conflicts with the geometric reconstruction, for instance when the calibration is off. Gwenn suggests taking $\log(J)$, which is a probability, and andd terms on the mathcing certainly, using maximum likelihood to optimize the solution for the toal error.

- **pg. 182** Daniel finds (6.51:) really interesting, since it shows that no matter how big your choose the baseline h, the Z^2 always wins as limitation on your reconstruction.

- **pg. 182:** In (6.53), $P_{\mathbf{k}}$ is just taking the first 2 components, ignoring the Z, so measures the components parallel to the image plane; and $\frac{1}{4}(\hat{\mathbf{x}} + \hat{\mathbf{x}}')(\hat{\mathbf{x}} + \hat{\mathbf{x}}')^\top$ is projection onto the $\frac{1}{2}(\hat{\mathbf{x}} + \hat{\mathbf{x}}')$-direction, multiplied by the square of the norm of that vector. For smallish disparities, the noise for $\mathbf{r}$ tends to be in the direction in which it is seen (with the $P_{\mathbf{k}}$ contribution parallel to the image plane just seen as a refinement).

- **pg. 193:** In figure 6.14, the bottom lines are grids on a plane in space, not grids in space.

- **pg. 197:** Infinity testing using χ^2: as we discussed before, the test is only meaningful for rejection of things one knows are Gaussian; and meaningless in all other cases.

- **pg. 200:** Erratum: Add the appropriate primes in (6.127).

- **pg. 203:** Basically, $\hat{\mathbf{b}} = R\hat{\mathbf{x}}' \times (\hat{\mathbf{x}} \times \mathbf{h})$.

- **Zhang calibration:** Daniel used the Zhang method to establish calibration parameters on 100 images using Zhang's method. To get the variances, it is permitted to take 100 images from this set (with duplicated) to make a new ensemble, and redo Zhang. Doing that often then provides an estimate with some statistical guarantees of the variance. One is even allowed to average the outcomes to get a better mean even than is obtainable from applying Zhang once on the whole set. This is called the *bootstrap method*. For this problem it is better than *cross-validation*, in which you would use a subset of 20 or so the original 100, because you are then not estimating the variances of the 100-image Zhang method.

7 Parametric Fitting (Isaac Esteban)

7.1 Deriving Kanatani's Eq 5.52 from Eq 7.61 (by Olaf Booij)

Just a simple exercise of deriving Equation 5.52 which gives the optimal correction of two coincident image points given the more general Equation 7.61 which does the optimal fitting of an image point given N noisy image points.

First rewrite 7.61 using $N = 2$:

$$\begin{aligned} \hat{x} &= \left(V[x_1]^- + V[x_2]^-\right)^- \left(V[x_1]^- x_1 + V[x_2]^- x_2\right) && (14) \\ &= \left(V[x_1]^- + V[x_2]^-\right)^- V[x_1]^- x_1 + \left(V[x_1]^- + V[x_2]^-\right)^- V[x_2]^- x_2. && (15) \end{aligned}$$

Now use

$$\frac{a}{a+b} = 1 - \frac{b}{a+b}, \tag{16}$$

or, actually, the matrix form (see below in Sec 7.2):

$$(A+B)^{-1}A = I - (A+B)^{-1}B. \tag{17}$$

Use this on first term of Eq (15) (using $A = V[x_1]^-$, $B = V[x_2]^-$):

$$\hat{x} = x_1 - \left(V[x_1]^- + V[x_2]^-\right)^- V[x_2]^- x_1 + \left(V[x_1]^- + V[x_2]^-\right)^- V[x_2]^- x_2. \tag{18}$$

Simplifying:

$$\hat{x} = x_1 - \left(V[x_1]^- + V[x_2]^-\right)^- V[x_2]^- (x_1 - x_2). \tag{19}$$

Now use

$$\frac{\frac{1}{b}}{\frac{1}{a} + \frac{1}{b}} = \frac{a}{a+b}, \tag{20}$$

(multiply numerator and denominator by ab). In matrix form (see below in Sec 7.2):

$$\left(A^{-1} + B^{-1}\right)^{-1} B^{-1} = A\left(A+B\right)^{-1}. \tag{21}$$

Applying this on Eq (19), using $A = V[x_1]$, $B = V[x_2]$, gives:

$$\hat{x} = x_1 - V[x_1]\left(V[x_1] + V[x_2]\right)^{-}(x_1 - x_2). \tag{22}$$

Eq 5.57 from Kanatani states:

$$\hat{x_1} = x_1 - \Delta x_1, \tag{23}$$

(notice that this is subtly different from his Eq 7.58) and Eq 5.53:

$$W = \left(V[x_1] + V[x_2]\right)^{-}. \tag{24}$$

Thus Eq (22) can be rewritten into:

$$\Delta x = V[x_1]W(x_1 - x_2), \tag{25}$$

is Kanatani's Eq 5.52.

7.2 More homework (from Carsten for Olaf)

Homework exercise 1 (from Carsten, but hey I latex it so I did it):

$$(A+B)^{-1}A = I - (A+B)^{-1}B. \tag{26}$$

$$\begin{aligned}(A+B)^{-1}A &= (A+B)^{-1}A - (A+B)^{-1}(A+B) + I \\ &= (A+B)^{-1}A - (A+B)^{-1}A - (A+B)^{-1}B + I \\ &= -(A+B)^{-1}B + I \\ &= I - (A+B)^{-1}B.\end{aligned}$$

By the way, this:

$$A(A+B)^{-1} = I - B(A+B)^{-1}, \tag{27}$$

can be shown in a similar way.

Homework exercise 2 (Carsten made me do it, and again I take all the credit):

$$\left(A^{-1} + B^{-1}\right)^{-1} B^{-1} = A\left(A+B\right)^{-1}. \tag{28}$$

Using:

$$X^{-1}Y^{-1} = (YX)^{-1}, \tag{29}$$

it can be rewritten as:

$$\begin{aligned}\left(A^{-1} + B^{-1}\right)^{-1} B^{-1} &= \left(B\left(A^{-1} + B^{-1}\right)\right)^{-1} \\ &= \left(BA^{-1} + BB^{-1}\right)^{-1} \\ &= \left(BA^{-1} + I\right)^{-1}.\end{aligned}$$

Multiplying from the left with AA^{-1} and again using trick (29):

$$\begin{aligned}
\left(A^{-1}+B^{-1}\right)^{-1}B^{-1} &= AA^{-1}\left(BA^{-1}+I\right)^{-1} \\
&= A\left(\left(BA^{-1}+I\right)A\right)^{-1} \\
&= A\left(BA^{-1}A+A\right)^{-1} \\
&= A\left(B+A\right)^{-1} \\
&= A\left(A+B\right)^{-1}.
\end{aligned}$$

8 Optimal Filter (Mark de Greef)

No notes, see slides.

9 Renormalization (Gijs Dubbelman)

My raw notes:

- taking all data at once, indepent, but there may be correlation between the data points (for instance overal shift)
- (9.4) is the central equation
- closed form solutions are biased
- W needs covariance at true points, but used observed points
- Ch5 pg 135 practical compromise, but what is the order of magnitude? could it be 2nd order? could use average of data points, so perhaps can be made rather small
- IJCV paper second order still important
- fig 9.1, $P\nabla$ is the derivaticv you should use that also enforces the constraints in (9.23)
- bias always results
- We do not understand (9.28), how can a term that does not contain u correct a gradient?
- renormalization: W was replaced by constants, now want to vary W, to improve the variance. Convergence around (9.95), not global optimum
- 2004 PAMI HEIV same as renormalization?
- 2006 paper Heteroscedastic Errors in Variable: noise different in scale and shape
- renormalization is special case of HEIV

- For estimation from images BA (bundle adjustment) (slow) gives better result and has more reasonable assumptions on the noise since it works with the image points
- HEIV is quicker (uses fewer parameters)
- BA can do heteroscadistic stuff (or mahalanobis on reprojection error)
- BA needs to known covariance of image noise in advanc

10 Applications of Geometric Estimation

This was a home reading assigment.

11 3-D Motion Analysis (Olaf Booij)

Olaf has made extensive slides on this, and also included post-presentation notes and references. I refer to those; here are my short notes, for now.

- Title might include 'Plus Scene Reconstruction' since much of it deals with that rather than estimating rotation and translation.
- Only pixel noise as Kanatani assumes is not realistic, but often used; even without noise no perfect data, for matching of highlights already gives mismatch of assumed equivalence. Also, mismatches are not necessarily Gaussian.
- Olaf has nice way of counting the DOFs, see his slides.
- Interesting numerical LA techniques around 11.6-11.10
- focus of expansion = epipole of the other camera
- theoretical bound on accuracy important: here he describes to get the covariance of translation/rotation
- (11.29) is OK in the end but an extra step would have been useful, some simplifications appear to have been done at the same time and not very consistently.
- Renormalization helps to put the error of the pixels back in rather than the 'algebraic error' (HZ term).
- 11.12 Sampson Weng weights though extra $g^T g$ in the denominator leading to ϵ^2-term in 11.41, related to Sampson distance, reprojection error see HZ. But Kanatani puts the covariances of the point correspondendes and that is good. The term $g^T g$ appears to add intrinsic structural deviation on the manifold to the reprojection error.

- Renormalization is a Kanatani thing; needs constraints; HEIV has bilinear constraints, BA (bundle adjustment) more general, and OK with todays more powerful computers.
- Non-Gaussian nature of noise appears to be more important than the Gaussian bias Kanatani 2007 HEIV , FNS (fundamental numerical scheme), for some problems some better.
- Not in the book: robustness under mismatches. Use for instance RANSAC which gives the good correspondences and an initial estimate for the iterative scheme. People use Sampson distance = residual times weights. Robust weighting by removing the contribution of far away points (since likely wrong) for instance Huber weights based on the median.
- Decomposing to h and R, non-robust which Olaf calls Horn-style; robust: Kanatani uses the covariance of G to do better than $G = U\text{diag}(1, 1, 0)V^T$ to adapt the Frobenius norm; this is unusual.
- HZ give another explanation of the $9 - 3 \neq 5$, they use the SVD to explain the DOF by putting in an extra rotation on the 2D eigenspace with eigenvalue 1 (due to $\text{diag}(1, 1, 0)$ nature).
- Four solutions dues to modeling rays as lines, Kanatani enforces by positive Z (use the Chum method instead), but omnidirectional cameras don't have an image plane. Olaf: why not enforce the decomposability within the iterative scheme; but does not seem to make a difference to the outcome?
- Questions about the factorizable and robustness: use 11.31 to get the covariance of h and R, reconstruct, reproject, look up reprojection error, use for weight in iterative scheme; also can check whether in from tor behind than take them out when behind Isaac uses 'close but no cigar points' (not use the far away points)
- Olaf: why do CV people focus on planar surfaces? How does the planarity affect the results: how is it not good: Isaac says rotation can be very wrong.
- Camera rotation: can also view this as points on a plane at infinity.
- fig 11.12: general homography allows reflection, so reflection of camera is allowed (with a reflected image)

12 3-D Interpretation of Optical Flow (Dung Manh Chu)

- Actual optical flow computation unfortunately only briefly mentioned.

- (12.12) Singularity of the matrix ∇I for all the same I: it is a projection matrix so of course singular. Specifically, if $\nabla I = (x, y)^T$, then

$$\begin{pmatrix} x \\ y \end{pmatrix} \begin{pmatrix} x & y \end{pmatrix} = \begin{pmatrix} x^2 & xy \\ xy & y^2 \end{pmatrix},$$

  which clearly has determinant 0.

- $Q_x = I - xk^T$ applied to $u = (u_1, u_2, 1)^T$ with $x = (x_1, x_2, 1)^T$ results in $Q_x u = (u_1 - x_1, u_2 - x_2, 0)^T = (u - x)^T$, a homogeneous coordinate drection vector equal to $u - x$ (if both are normalized points). So $Q_x u$ can be read geometrically as: the direction from point x to point u.

- The 'indirect approach' is modelled very much on the previous chapter. Is there a physical meaning to the flow matrix F (12.42) justifying its name?

- Pg 384, why compute $E[M]$ rather than $E[F]$? Is is just more convenient to correct M, or would it be wrong to correct F directly?

- (12.92) is a clever step, easy to verify, but how to find it?

- First optimizing F and then only going to the decomposible subset is not allowed locally if one takes the Mahalanobis into account properly. (Make a drawing.)

- Pg.390 section B makes rather a subtle but essential point: one needs to correct the individual $\dot{x}$ to compute the depth properly.

- Combining (12.154) and (12.144) we see that both small and large $\hat{Z}$ are expected to have large variance (though for different reasons), so only the middle range can be trusted in general.

- Fig.12.22 is important and should have been mentioned at the beginning of the chapter: put your estimated flow vector at the correct estimation location.

- Kien says the critical surfaces really are a problem in practice, especially in indoor 3D reconstruction. Especially the planar degeneracy occurs a lot.

- Panamoric stitching based on video can use optical flow to see when a pure rotation has occurred; but even a small translation can disturb the result considerably. There is a lot of ghosting, see work by master student Wouter Suren's (now being implemented at NFI).

13 Information Criterion for Model Selection (Romain Hugues)

- Keep the definitions straight: Data points are in m-D space, but sampled on a m'-D manifold. According to the model, they should be on a d-D

manifold (d = dimension). We will need the codimension $r = m' - d$, the 'freedom' in the model.

- Matching the numbers on the example figures: (m', d, r, n'), in the descriptions these are point space (m'), dimension (d), codimension $(r = m' - d)$ and degrees of freedom (n'). Fig 13.2(a): $(2, 0, 2, 2)$, (b) $(2, 1, 1, 2)$, (c) $(2, 2, 0, 2)$. Fig 13.3(a): $(3, 0, 3, 3)$, (b) $(3, 1, 2, 4)$, (c) $(3, 2, 1, 3)$, (d) $(3, 3, 0, 0)$.

- Main idea in this chapter is that the expected residual is the important thing to minimize (to prevent overfitting to the data). Estimating $I(S)$ is impossible, chapter provides an unbiased estimator in the form of AIC. Chapter 5 situation is in Fig 13.5: fitting to a known manifold; the situation with unknown manifold is Fig 13.6.

- In (13.41), he has multiplied everything by ϵ^2; for comparison, this is OK.

- Example of (13.62): for a rigid body motion we know that it is rotation and translation (the general model), and we can estimate their parameters. But you may have the case of pure rotation, and recognizing that would give a better model for the data. That then gets onto model comparision.

- Can you compare models to each other, or only all models to the most general one? Pg. 442, take the motion example of 13.6.4, there is a sequence to testing the efficacy of the models. So data is points in 3D space, check for the coincidence on point, line or plane.

  We has some discussion about what models could be compared.

  - *If you know the noise level (somehow, by external means)*, then you can put S_1 and S_2 and compare them directly.
  - But Kanatani's method needs a correct model if you don't know the noise level (to estimate it). This requires a quantitative ordering of the models. For instance, you have to establish whether space points are on a plane before you can check if they might be on a line within *that* plane.

- The practical usage was discussed. Refinement of the models is the important issue here. General points may be better explained by lying on a plane, and the method returns the parameters of that plane. Then that becomes the hypothesis and plane within which you test whether they might be on a line. Occam's razor, (13.66) balances degrees of freedom with explainability.

- The 5-point algorithm can be used both for planar and non-planar points. Invented around the time of the Kanatani's book. Within motion estimation, you may have planar points. If your data resided in a plane, estimate the planar homography, using 4 points. But if the data is really in 3D space, you should use 8 points, it fails for planar points (and single

sheets hyperboloids) because of the critical points. Using the 5-point algorithm, you don't have to decide, but you get multiple solutions, namely 40 (10 essential matrices with each 4 solutions). The 8-point algorithm has up to 4 multiple solutions.

- What should be done in practice? (Romain's list)

  1. Collect data, decide on plausible models
  2. Estimate manifolds and true positions for each model
  3. Compute residuals for each model
  4. If a model is always correct (weakest), estimate noise levels from residual of this model
  5. Compare two models using (13.66)

  Based on the n-point algorithm example, Olaf's modification of this list is is:

  1. Collect data
  2. Choose the algorithms you are going to use and determine their degenerate surfaces.
  3. Estimate manifolds and true positions for each degenerate surface model.
  4. Compute residuals for each model
  5. If a model is always correct (weakest), estimate noise levels from residual of this model
  6. Compare two models using (13.66)
  7. Do this for all models and pick the best fit
  8. Use an algorithm that does not have the fitted model as its degenerate surface.

14 General Theory of Geometric Estimation

This is a home reading asignment, since no one could be shamed into presenting it.

References

[1] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2007. ISBN-038731073

[2] Otto Bretscher, *Linear Algebra with Applications*, 3rd Edition, Prentice-Hall 2005.

[3] Leo Dorst, Daniel Fontijne, Stephen Mann, *Geometric Algebra for Computer Science*, Morgan Kaufmann, 2007.

[4] Chr. Perwass, *Geometric Algebra with Applications in Engineering*, Springer 2009.

[5] Our reading club web site: `www.science.uva.nl/~leo/kanatani.html`