Parameter Estimation

Goal: We consider the problem of fitting a parameterized model to noisy data.

Model fitting arises, for example, when:

- Matching image features with a known 3D shape (the unknown parameters are M_{ext} and, perhaps, M_{int} and radial distortion);
- Fitting lines or curves to image gradient or edge data;
- Fitting the PCA model of a face, say, to an image;
- Fitting motion models to video data.

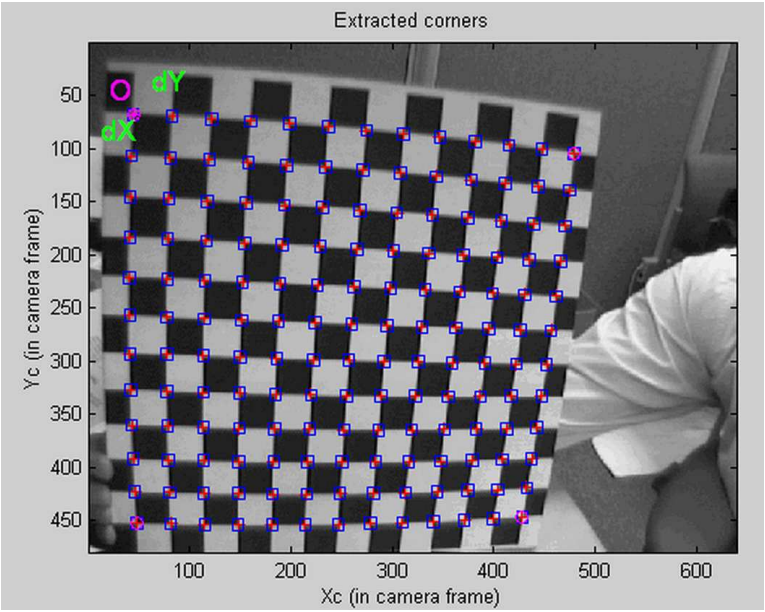
We will consider many of these specific problems later in this course.

Rule of Thumb: Never estimate parameters without at least thinking about the implied noise model.

Readings: Szeliski, Sections 6.1 and 6.2, Appendices B1 through B3.

Model Problem: Calibration using Checkerboard Corners

Example: Camera calibration. Given multiple images of a known calibration object, estimate the intrinsic and extrinsic camera parameters.



This image is from the "Camera Calibration Toolbox in Matlab", see http://www.vision.caltech.edu/bouguetj/calib_doc/.

The origin of world coordinates is at one corner point, and the checkerboard is in the plane $X_{w,3} = 0$.

Specific Case: Given the labeled checkerboard corner points in each image (e.g., the blue boxes above), estimate the 3D pose of the checkerboard in each image, and the intrinsic camera parameters.

Checkerboard Corners Positions from Camera Parameters

Let $\{\vec{z}_k\}_{k=1}^K$ denote the set of observed (and labelled) image checkerboard corner points, $\vec{z}_k \in \mathbb{R}^2$.

For each corner point, suppose

$$\vec{z}_k = \vec{f}_k(\vec{q}^0) + \vec{n}_k, \tag{1}$$

where

- $\vec{q}^0$ is the vector of unknown parameters, which includes the position and rotation of the checkerboard in the camera's coordinates, along with any unknown intrinsic parameters for the camera;
- $\vec{f}_k(\vec{q})$ is the predicted image position of the k^{th} corner point from perspective projection, given the vector of pose parameters $\vec{q}$;
- the noise vector $\vec{n}_k$ is the error, $\vec{z}_k - \vec{f}_k(\vec{q}^0)$, between the observed position and the correct position of the k^{th} corner point.

Given the observations $\{\vec{z}_k\}_{k=1}^K$, the parameters $\vec{q}$ are estimated by minimizing some measure of the implied noise vectors $\vec{n}_k$. The particular measure used is dictated by the *noise model*.

Independent Gaussian Noise

A reasonable first approximation is to assume that the noise in the observations is:

- statistically independent,
- mean zero, and
- Normally distributed.

That is, the error $\vec{n}_k = \vec{z}_k - f_k(\vec{q})$, in the k^{th} observation $\vec{z}_k$, is modelled as an independent random sample from the 2D Normal probability density function $p(\vec{n} \mid \vec{0}, \Sigma)$, where

$$p(\vec{n} \mid \vec{m}, \Sigma) = \frac{1}{2\pi|\Sigma|^{1/2}} e^{-\frac{1}{2}(\vec{n}-\vec{m})^T \Sigma^{-1}(\vec{n}-\vec{m})} \tag{2}$$

is the 2D Normal probability density function. Here the parameters are:

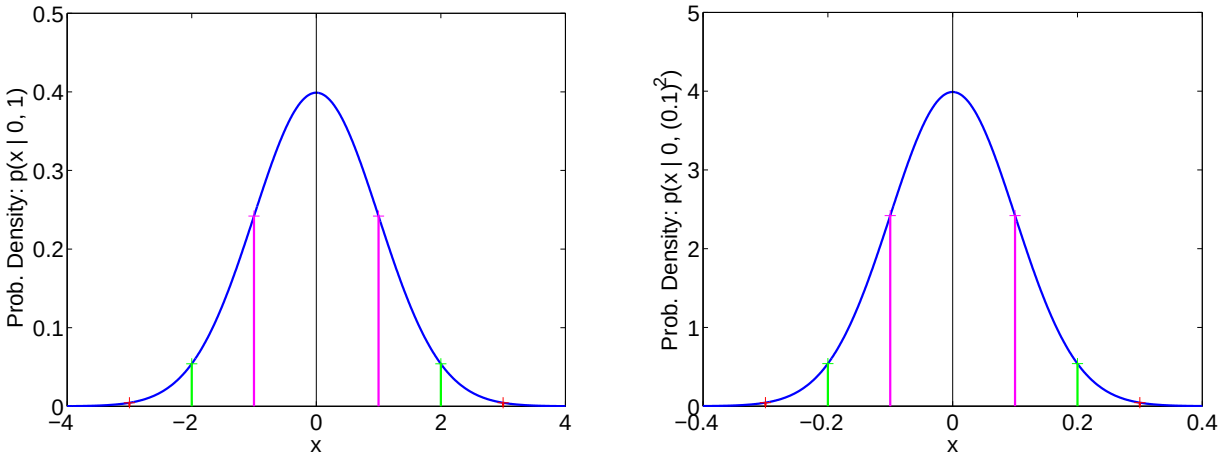
- $\vec{m}$, the mean of the distribution, and
- Σ, the 2×2 (symmetric, positive definite) covariance matrix.

See the next three slides for a quick review.

Recall the 1D Normal Distribution

The 1D probability density function for a Normal distribution with mean m and variance σ^2 is:

$$p(x \mid m, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{1}{2}\frac{(x-m)^2}{2\sigma^2}}.$$



These are "Bell curves." The left plot is for mean $m = 0$ and standard deviation $\sigma = 1$.

The right plot is for mean 0 and $\sigma = 0.1$. It is simply compressed on the horizontal axis (by a factor of 10) , and stretched on the vertical axis (by the same factor of 10). These two stretches are such that the integral of the probability density remains equal to one.

Note that for $\sigma = 0.1$, we have $p(x \mid 0, \sigma^2)|_{x=0} \approx 3.99$. Can we conclude from this that the probability of x can be bigger than 1?

Multivariate Normal Distributions

More generally, a d-dimensional Normal distribution is given by:

- $N(\vec{m}, \Sigma)$ denotes a normal distribution;
- $\vec{m} \in \mathbb{R}^d$ is the mean;
- $\Sigma \in \mathbb{R}^{d \times d}$ is the covariance matrix. As such, Σ is symmetric (i.e., $\Sigma^T = \Sigma$) and positive definite (i.e., $\vec{u}^T \Sigma \vec{u} > 0$ for all $\vec{u} \in \mathbb{R}^n \backslash \{\vec{0}\}$).
- The probability density function for $N(\vec{m}, \Sigma)$ is,

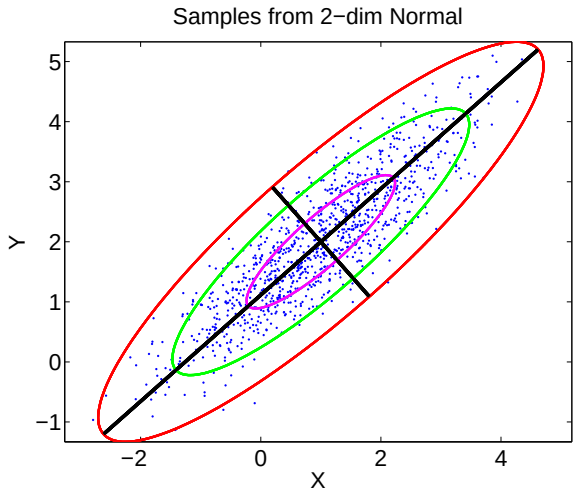
$$p(\vec{x} \,|\, \vec{m}, \Sigma) \;\equiv\; \frac{1}{(2\pi|\Sigma|))^{d/2}} e^{-\frac{1}{2}(\vec{x}-\vec{m})^T \Sigma^{-1} (\vec{x}-\vec{m})}. \tag{3}$$

  Here $|\Sigma| = det(\Sigma)$ denotes the determinant of Σ.

A two dimensional example of a Normal distribution is shown on the next slide.

Constant Standard Error Ellipsoids

Samples, $\vec{x}$, from the 2D distribution $N(\vec{m}, \Sigma)$ are shown to the right.



We define $\vec{x}$ to have a **standard error** of s if

$$(\vec{x} - \vec{m})^T \Sigma^{-1} (\vec{x} - \vec{m}) = s^2. \tag{4}$$

The purple, green and red ellipses are the curves for standard errors $s = 1, 2,$ and 3, respectively.

These elliptical shapes for constant standard error become clear by considering the eigenvalues decomposition, $\Sigma = U \Lambda U^T$, where $\Lambda = \texttt{diag}[\lambda_1, \lambda_2]$ and $\lambda_1, \lambda_2 > 0$ (recall Σ is positive definite).

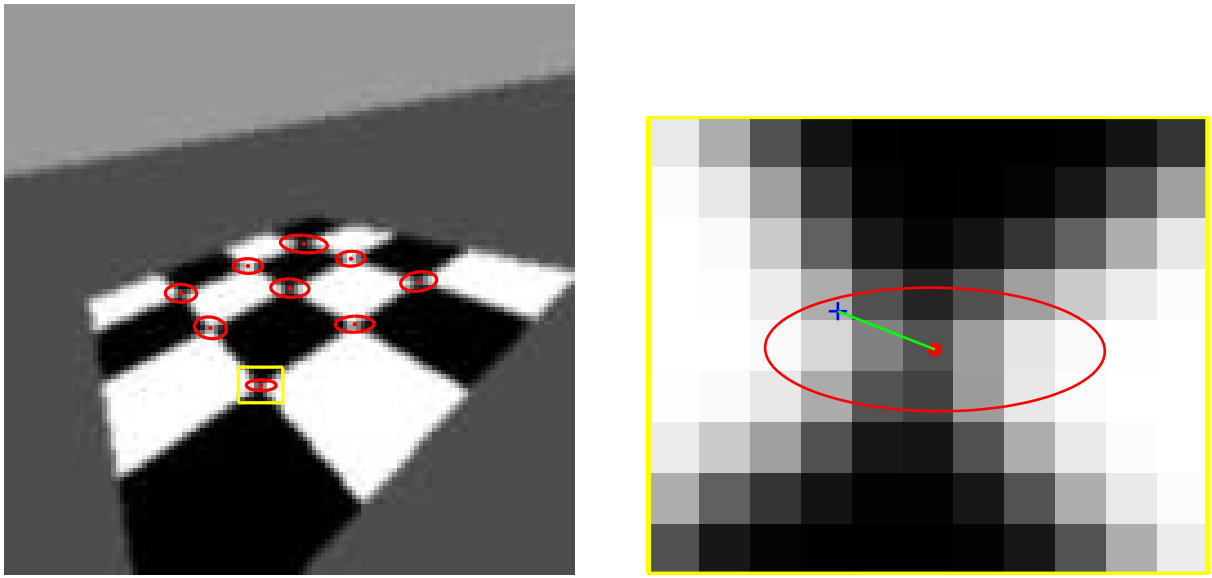
Consider the 2D coordinates $\vec{u}$ obtained from $\vec{x}$ by translating the origin to the mean point, $\vec{m}$, and rotating/reflecting by U^T, that is, $\vec{u} = U^T(\vec{x} - \vec{m})$. In these coordinates, equation (4) becomes

$$\vec{u}^T \Lambda^{-1} \vec{u} = \frac{u_1^2}{\lambda_1} + \frac{u_2^2}{\lambda_2} = s^2.$$

This is the equation of an ellipse having the principal axes aligned with the u_1 and u_2 axes. The lengths of these axes are $2s\sqrt{\lambda_1}$ and $2s\sqrt{\lambda_2}$, respectively.

Illustration of the Noise Model

Below we illustrate equation (1), namely $\vec{z}_k = \vec{f}_k(\vec{q}) + \vec{n}_k$, with $\vec{n}_k \sim N(\vec{0}, \Sigma_k)$:



The noise model $\vec{f}_k(\vec{q}) + \vec{n}_k$, with $\vec{n}_k \sim N(\vec{0}, \Sigma_k)$, is shown (left), with ellipses for standard error equal to 2 around each corner point. (These covariances are only illustrative.)

The detail figure (right) shows one observed corner point $\vec{z}_k$ (blue `'+'`), the model point $\vec{f}_k(\vec{q})$ (red `'.'`), and the error vector $\vec{e}_k = \vec{z}_k - \vec{f}_k(\vec{q})$ (green line).

The observed point is almost on the 2-standard-deviation ellipse, indicating the standard error $\sqrt{\vec{e}_k^T \Sigma_k^{-1} \vec{e}_k}$ is just less than 2.

Maximum Likelihood Estimation

Trick Question: What's the probability of observing an error $\vec{n}_k = \vec{z}_k - \vec{f}_k(\vec{q})$?[1] More helpfully, let $B(\epsilon)$ be the 2D box

$$B(\epsilon) = [-\frac{\epsilon}{2}, \frac{\epsilon}{2}] \times [-\frac{\epsilon}{2}, \frac{\epsilon}{2}].$$

Then the probability of observing noise in the region $\vec{n}_k + B(\epsilon)$ is the integral of $p(\vec{n}|\vec{0}, \Sigma_k)$ over this set. This equals $p(\vec{n}_k|\vec{0}, \Sigma_k)\epsilon^2$ plus higher order terms as $\epsilon \to 0$ (see Wikipedia, Rectangle Method).

Since the noise in each of the observations is assumed to be independent, the probability that *all* of the noise values are in the regions $\vec{n}_k + B(\epsilon)$, for $k = 1, \ldots, K$, is

$$P_\epsilon \;=\; \prod_{k=1}^{K} \Big[p(\vec{n}_k \mid \vec{0}, \Sigma_k)\epsilon^2 + O(\epsilon^4) \Big] \;=\; \epsilon^{2K} \prod_{k=1}^{K} \Big[p(\vec{z}_k - \vec{f}_k(\vec{q}) \mid \vec{0}, \Sigma_k) + O(\epsilon^2) \Big] .$$

For a fixed ϵ, we might try to choose the parameters $\vec{q}$ which maximize this probability P_ϵ. Or, more simply, we wish to choose $\vec{q}$ to maximize the *likelihood* of all the observations $\vec{z}_k$, namely

$$\text{Data Likelihood: } p(\vec{z}_1, \ldots, \vec{z}_K \mid \vec{q}) \;\equiv\; \prod_{k=1}^{K} p(\vec{z}_k - \vec{f}_k(\vec{q}) \mid \vec{0}, \Sigma_k). \tag{5}$$

[1] Answer: zero.

Recap: Maximum Likelihood Estimation

In summary, given:

- the parameterized model $\vec{f}_k(\vec{q})$ for the image position (in pixels) of the k^{th} checkerboard corner point;
- where $\vec{q}$ are the unknown parameters, including the intrinsic and extrinsic camera parameters; and
- the observed but approximate corner points $\vec{z}_k$ (in pixels); and
- the noise model for the observed $\vec{z}_k$, namely $\vec{n}_k = \vec{z}_k - \vec{f}_k(\vec{q}^{\,0}) \sim N(\vec{0}, \Sigma_k)$ are independent. Here $\vec{q}^{\,0}$ are the correct parameters, and we initially assume the noise covariances Σ_k are known.

Then we wish to maximize the data likelihood (5) with respect to $\vec{q}$:

$$\text{Data Likelihood: } p(\vec{z}_1, \ldots, \vec{z}_K \mid \vec{q}) \equiv \prod_{k=1}^{K} p(\vec{z}_k - \vec{f}_k(\vec{q}) \mid \vec{0}, \Sigma_k).$$

The resulting vector, say $\vec{q}^{\,*}$, is called a **maximum likelihood estimate** for the parameters $\vec{q}$.

Maximizing Log-Likelihood

Since $\log(L)$ is monotonically increasing for $L > 0$, maximizing the data likelihood is equivalent to maximizing the **Log-Likelihood**, $\log(p(\vec{z}_1, \ldots, \vec{z}_K \mid \vec{q}))$.

From equation (5) we have

$$\begin{aligned}
\log(p(\vec{z}_1, \ldots, \vec{z}_K \mid \vec{q})) &= \log\left[\prod_{k=1}^{K} p(\vec{z}_k - \vec{f}_k(\vec{q}) \mid \vec{0}, \Sigma_k)\right] = \sum_{k=1}^{K}\left[\log(p(\vec{z}_k - \vec{f}_k(\vec{q}) \mid \vec{0}, \Sigma_k))\right] \\
&= \sum_{k=1}^{K} \log\left(\frac{1}{(2\pi|\Sigma_k|)^{d/2}} e^{-\frac{1}{2}(\vec{z}_k - \vec{f}_k(\vec{q}))^T \Sigma_k^{-1} (\vec{z}_k - \vec{f}_k(\vec{q}))}\right) \\
&= \sum_{k=1}^{K}\left[-\frac{1}{2}(\vec{z}_k - \vec{f}_k(\vec{q}))^T \Sigma_k^{-1}(\vec{z}_k - \vec{f}_k(\vec{q})) - \frac{d}{2}\log(2\pi|\Sigma_k|)\right] \\
&= -\frac{1}{2}\left[\sum_{k=1}^{K}(\vec{z}_k - \vec{f}_k(\vec{q}))^T \Sigma_k^{-1}(\vec{z}_k - \vec{f}_k(\vec{q}))\right] + \text{Const.}
\end{aligned}$$

Here $d = 2$ and "Const." is a constant independent of $\vec{q}$.

Minimizing the Sum of Squared Standard Errors

Alternatively, it is equivalent to minimize:

$$SSSE(\vec{q}) \;=\; S^3E(\vec{q}) \;=\; \frac{1}{2}\sum_{k=1}^{K}(\vec{z}_k - \vec{f}_k(\vec{q}))^T\Sigma_k^{-1}(\vec{z}_k - \vec{f}_k(\vec{q})). \tag{6}$$

We refer to $S^3E(\vec{q})$ as the sum of squared standard errors.

Here "standard' refers (usefully, but somewhat non-standardly) to the normalization by the inverse covariance matrices Σ_k^{-1}.

Flavours of Least Squares

Since it is rare to know the noise covariances Σ_k, it is common to make two types of simplifications in the noise model,

$$\text{Isotropic: } \Sigma_k = \sigma_k^2 I, \tag{7}$$

$$\text{Identically Distributed: } \Sigma_k = \Sigma_0, \tag{8}$$

$$\text{Isotropic and Identically Distributed: } \Sigma_k = \sigma_0^2 I, \tag{9}$$

where $k = 1, 2, \ldots, K$.

For the isotropic case, the curves of constant error are circles instead of ellipses.

For identically distributed noise, the constant error curves are the same for every observed point.

Weighted Least Squares

For the isotropic case, the max-likelihood estimate (6) becomes

$$\begin{aligned} WLS(\vec{q}) &= \frac{1}{2}\sum_{k=1}^{K} w_k \, (\vec{z}_k - \vec{f}_k(\vec{q}))^T (\vec{z}_k - \vec{f}_k(\vec{q})) \\ &= \frac{1}{2}\sum_{k=1}^{K} w_k \, ||\vec{z}_k - \vec{f}_k(\vec{q}))||^2, \end{aligned} \tag{10}$$

where $w_k = 1/\sigma_k^2$ are the weights. This is called a **weighted least squares** (WLS) problem.

Note that weights $w_k = \sigma_k^{-2}$ are inversely proportional to the variance of the observed point $\vec{z}_k$.

Reweighting the data therefore implies a statement about the assumed variances of the data points, we will return to this issue later.

Ordinary Least Squares

For the isotropic and identically distributed noise model, it is equivalent to minimize

$$LS(\vec{q}) \;=\; \frac{1}{2}\sum_{k=1}^{K}(\vec{z}_k - \vec{f}_k(\vec{q}))^T(\vec{z}_k - \vec{f}_k(\vec{q})) \;=\; \frac{1}{2}\sum_{k=1}^{K}||\vec{z}_k - \vec{f}_k(\vec{q}))||^2. \tag{11}$$

Here we have omitted a constant weight $w_0 = 1/\sigma_0^2$ which appears in (10). This constant does not effect the minimization.

This is called an **ordinary or unweighted least squares** (LS) problem.

Linear and Nonlinear Least Squares

Finally, any of the previous least squares problems are said to be **linear least squares problems** if $\vec{f_k}(\vec{q})$ has only a linear dependence on the parameters $\vec{q}$. That is, $\vec{f_k}(\vec{q}) = A_k\vec{q} + \vec{b_k}$ for a constant matrix A_k and vector $\vec{b_k}$.

We show below that linear least squares problems can be reduced to solving a linear system of equations.

For the camera calibration problem, $\vec{f_k}(\vec{q})$ are nonlinear functions of $\vec{q}$. Therefore the problem of minimizing $S^3E(\vec{q})$ is called a **nonlinear least squares problem**, and similarly for the corresponding weighted or unweighted versions.

Generally we require numerical optimization software to solve nonlinear least squares problems. See, for example, the textbook by Nocedal and Wright, 2006.

Punting[2] the Calibration Problem

For now, the checkerboard fitting problem has served our purpose of:

1. Introducing maximum likelihood estimation for models with independent, multidimensional, Gaussian noise;
2. Deriving the equivalent sum of squared standard errors formulation, $S^3E(\vec{q})$, above;
3. Introducing several specific cases of minimizing the squared standard error where the approximate noise models are isotropic and/or identical for all observations.

In order to develop some general intuition for maximum likelihood estimation, we consider a simpler estimation problem next.

In particular, we consider scaled orthographic instead of perspective projection. This leads to a linear least squares problem.

[2]Here the verb "to punt" colloquially refers to accepting your current gains and minimimizing the risk of losses in the future. It comes from American football.

Scaled Orthographic Case

A scaled orthographic mapping of the checkerboard plane is given by

$$\vec{x} \;=\; s \begin{pmatrix} I_2 & \vec{0} \end{pmatrix} \hat{M}_{ex} \begin{pmatrix} X_1 \\ X_2 \\ 0 \\ 1 \end{pmatrix} \;=\; M \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} + \vec{b} \;=\; A(\vec{X})\vec{q}. \tag{12}$$

Since (12) is linear in $\vec{X} = (X_1, X_2)^T$, it can be rewritten in terms of some constant 2×2 matrix M and 2-vector $\vec{b}$.

The rightmost equation above is

$$M\vec{X} + \vec{b} \;=\; \begin{pmatrix} X_1 & 0 & X_2 & 0 & 1 & 0 \\ 0 & X_1 & 0 & X_2 & 0 & 1 \end{pmatrix} \begin{pmatrix} M_{1,1} & M_{2,1} & M_{1,2} & M_{2,2} & b_1 & b_2 \end{pmatrix}^T \;=\; A(\vec{X})\vec{q}. \tag{13}$$

Here $\vec{q}$ is the above 6-vector of coefficients, consisting of the elements of M and $\vec{b}$.

The linear mapping $\vec{x} = M\vec{X} + \vec{b} \;=\; A(\vec{X})\vec{q}$ is called an *affine transformation* of the coordinates X_1 and X_2. (See Szeliski, Sec. 6.1.)

Affine Model

For the scaled orthographic case the predicted corner points are $\vec{f}_k(\vec{q}) = A(\vec{X}_k)\vec{q}$, which is linear in the unknowns $\vec{q}$. (For simplicity, we are considering only one affine image of the checkerboard.)

The parameters $\vec{q}$ are called the *affine pose* parameters.

The sum of squared standard errors, namely $S^3E(\vec{q})$ in (6), is

$$S^3E(\vec{q}) \;=\; \frac{1}{2}\sum_{k=1}^{K}(\vec{z}_k - A_k\vec{q})^T\Sigma_k^{-1}(\vec{z}_k - A_k\vec{q}), \quad \text{where } A_k = A(\vec{X}_k). \tag{14}$$

In order for $\vec{q}^*$ to minimize $S^3E(\vec{q})$ it must be the case that the gradient of $S^3E(\vec{q})$ with respect to $\vec{q}$ must be zero at $\vec{q}^*$.

That is, we get the *normal equations*

$$\vec{0} \;=\; \frac{\partial S^3E}{\partial \vec{q}}(\vec{q}^{\,*}) \;=\; -\sum_{k=1}^{K} A_k^T\Sigma_k^{-1}(\vec{z}_k - A_k\vec{q}^{\,*}),$$

This is a linear equation in the unknown vector $\vec{q}^*$.

Linear Least Squares Solution

The normal equations can be rewritten as

$$F\vec{q}^{\,*} \;\equiv\; \left[\sum_{k=1}^{K} A_k^T \Sigma_k^{-1} A_k\right] \vec{q}^{\,*} \;=\; \left[\sum_{k=1}^{K} A_k^T \Sigma_k^{-1} \vec{z}_k\right] \;\equiv\; \vec{r}. \tag{15}$$

Here $F \in \mathbb{R}^{6\times 6}$ is the Hessian of $S^3E(\vec{q})$ (aka the Fisher information matrix).

From (15), the maximum likelihood estimate is $\vec{q}^{\,*} = F^{-1}\vec{r}$. (For the current problem, F is full rank iff the checkerboard corner points $\vec{X}_k$ are not all colinear.)

Max-Likelihood Estimation Algorithm. Construct the terms F and $\vec{r}$ in the normal equations, (15), and solve the resulting linear system.

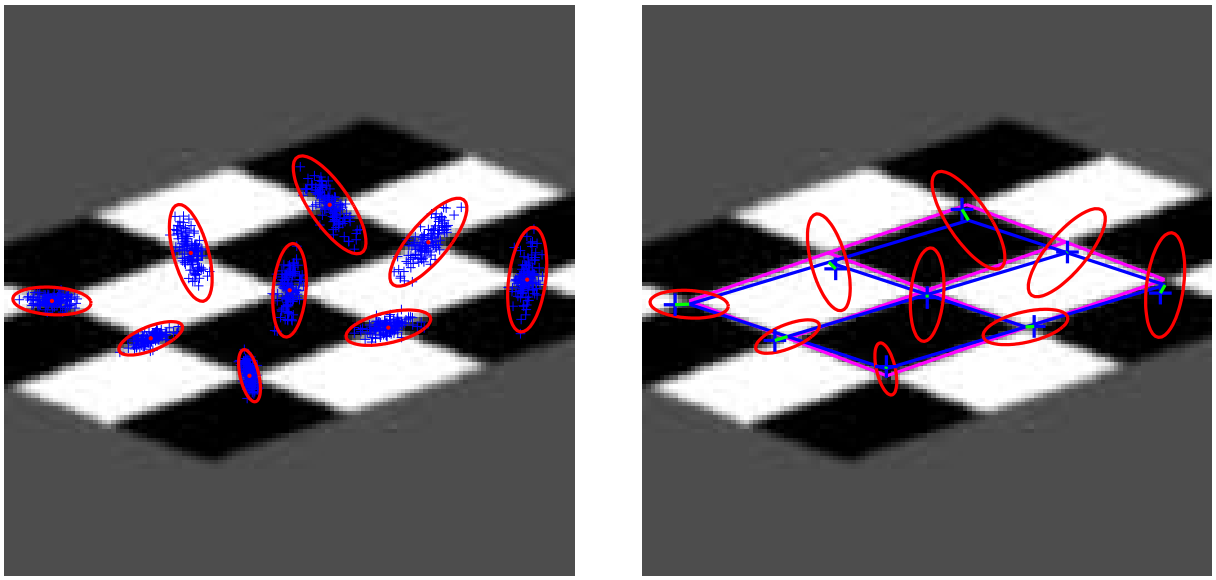
Properties of the Sum of Squares of Standard Errors

When the noise $\vec{n}_k = \vec{z}_k - A_k\vec{q}^0$ is sampled independently from the 2D Normal distribution $N(\vec{0}, \Sigma_k)$ for $k = 1, \ldots, K$, we have:

- The mean of the max-likelihood estimates $\vec{q}^*$ (using equation (15)) is equal to the true solution $\vec{q}^0$.
- Equivalently, the expectation $E[\vec{q}^*] = \vec{q}^0$.
- Here the expectation is over the distribution of noise given by the independent, mean zero, Gaussian noise model.
- An estimator is said to be unbiased iff $E[\vec{q}^*] = \vec{q}^0$, where $\vec{q}^0$ is the true value.
- Any linear least squares estimator (with mean-zero noise and non-singular normal equations) is unbiased.
- For the $S^3E(\vec{q})$ estimator, the **covariance** of $\vec{q}^*$ equals F^{-1}, the inverse of the Fisher information matrix F introduced in equation (15).

Simulating Maximum Likelihood Solutions

The Matlab code `affCheckDemo.m` simulates the max-likelihood inference of affine pose.

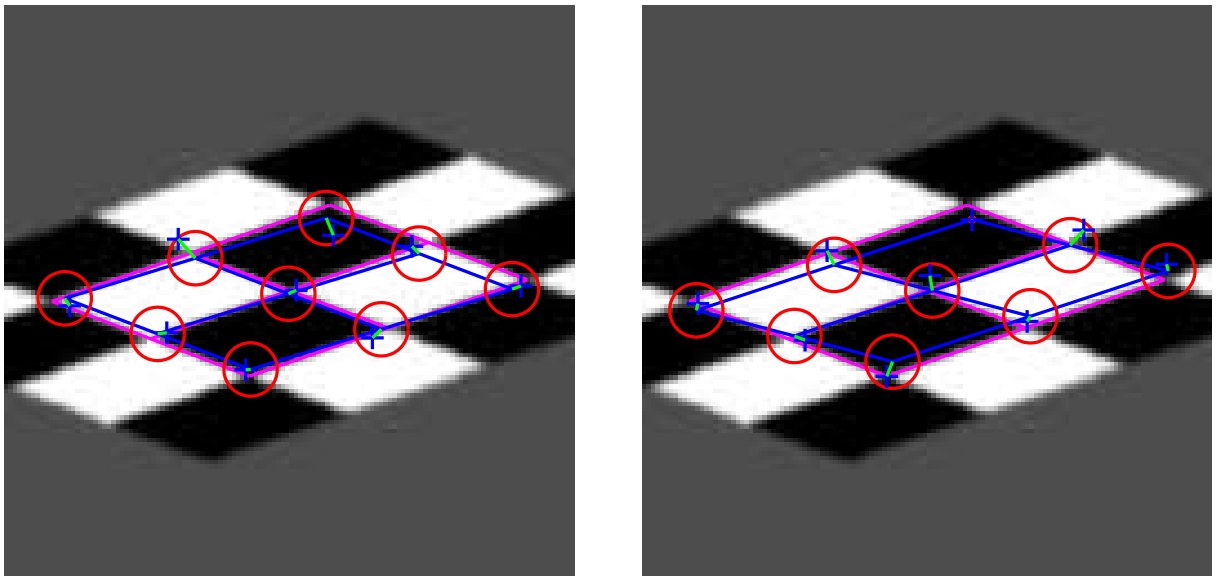


Above left shows 100 samples of the noisy corner points $\{\vec{z}_k\}_{k=1}^{K}$, where the red ellipses indicate standard errors equal to 3. Most points are seen to have a standard error less than 3.

Above right shows the solution $\vec{q}^*$ of (15) (blue grid), given the observed points $\vec{z}_k$ (blue crosses). The differences, $\vec{z}_k - \vec{f}(\vec{q}^*)$, are the inferred noises (green lines). The mauve grid shows the mean of the solutions $\vec{q}^*$ over many different noise samples.

This illustrates that, with mean-zero noise, linear least squares provides an unbiased estimate.

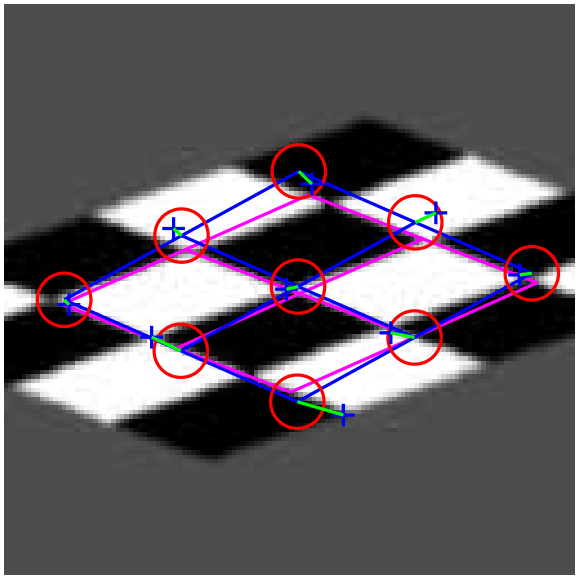
Ordinary and Reweighted Least Squares



Above left, LS: A random sample (blue crosses) from the same data set of noisy corner points is used with ordinary least squares, equation (11). The red circles indicate that the *noise model used for estimation* (only) is isotropic with identical weights, say $w_k = 1/\sigma_0^2$. The blue grid is the LS solution for this set of noisy corner points, while the mauve grid is the mean over many samples.

Above right, WLS: Similar to the LS example, except the top point has had its weight increased by a factor of 100. This increase in the weight is equivalent to a tenfold decrease in the covariance for that point (i.e., $w_k = 1/(\sigma_0/10)^2$). Note the fitted model (blue grid) closely approximates this point. Also, note the average model (mauve grid) is still unbiased.

Least Squares with an Outlier



This set of data (blue crosses) includes a single outlier (bottom blue cross), which causes large errors in the LS solution (blue grid).

The mean solution (mauve grid), over similar outlier locations, is biased away from the true solution.

Sensitivity to Outliers: For least squares, the cost of any model fit error (green lines) is quadratic in the error magnitude. Due to this quadratic cost, it is often cheaper to decrease a few very large errors and, effectively, distribute them around to the other constraints, creating many medium-sized errors.

Aspects of Estimator Performance

In choosing or designing an estimator, we should consider:

- **Bias:** Is the expected value of the estimator, say $E[\vec{q}^*]$, equal to the true solution $\vec{q}^0$? (If so, the estimator is said to be unbiased.)
- **Variance:** How large is the covariance of the estimator, $E[(\vec{q}^* - E[\vec{q}^*])(\vec{q}^* - E[\vec{q}^*])^T]$?
- **Statistical Efficiency:** Given a measure of the estimator error, such as the root mean squared (RMS) reconstruction error:

$$R(\vec{q}^*) = \left(E\left[\frac{1}{K} \sum_{k=1}^{K} ||\vec{f}_k(\vec{q}^*) - \vec{x}_k^0||^2 \right] \right)^{1/2}, \tag{16}$$

  where $\vec{x}_k^0$ is the true position of the k^{th} point, define the statistical efficiency as the ratio

$$R(\vec{q}^{opt})/R(\vec{q}^*). \tag{17}$$

  Here $\vec{q}^{opt}$ denotes the optimal estimator from the S^3E objective function in (6).

- **Tolerance to Outliers:** Does the estimator performance, say in terms of RMS error, degrade gracefully in the presence of outliers?

Statistical Efficiency

The statistical efficiencies (SE) for the previous estimators are listed below (from `affCheckerDemo.m`):

- $SE = 1.00$ for the S^3E estimator, indicating it is perfectly efficient.
- $SE = 0.67$ for the LS estimator. Recall the LS approach used above treats all errors as isotropic and identically distributed. The loss of statistical efficiency is due to the information lost in using this simple noise model. Equivalently, note the RMS error has increased by 50% from the optimal value (i.e., $1/SE = 1/0.67 \approx 1.5$).
- $SE = 0.51$ for the WLS estimator used above, for which we reweighted one point by a factor of 100 over the LS estimator. This further loss of efficiency is due to the inaccuracy of the implied noise model for the re-weighted point (i.e., $\sigma_k \leftarrow \sigma_0/10$ and $w_k \leftarrow 100 w_0 = 100/\sigma_0^2$).
- $SE = 0.26$ for the LS estimator in the presence of the outlier. This value decreases to zero as the outlier becomes more distant. The current value illustrates that the RMS error has quadrupled over the optimal estimator (w/o an outlier), indicating the sensitivity of LS to outliers.

Our next task is to consider estimation in the presence of outliers.

References

The following textbook contains sections on parameter estimation:

Richard Szeliski, Computer Vision: Algorithms and Applications, Springer; 1st Edition, November 24, 2010.

Section 6.1 and 6.1 of Szeliski's book discuss the pose estimation and alignment problems.

Appendices B1 through B3 of Szelizki's book discuss maximum likelihood estimation and least squares.

The book is available from http://szeliski.org/Book/.

For information about numerical methods for solving nonlinear optimization problems, such as nonlinear least squares, see:

J. Nocedal, and S.J. Wright, Numerical optimization, Springer series in operations research, Springer, 2006.