Step 4: Classifying Local Shape. Finally, the shape classification given in Chapter 4 is achieved by defining two further quantities, the *mean curvature*, H, and the *Gaussian curvature*, K:

$$H = -\frac{k_1 + k_2}{2} \qquad K = k_1 k_2.$$

One can show that *the Gaussian curvature measures how fast the surface moves away from the tangent plane around* P, and in this sense is an extension of the 1-D curvature k. The formulae giving H and K for a range surface in r_{ij} form, $(x, y, h(x, y))$ are given in Chapter 4.

References

M.P. Do Carmo, *Differential Geometry of Curves and Surfaces*, Prentice-Hall, Englewood Cliffs (NJ) (1976).

A.6 Singular Value Decomposition

The aim of this section is to collect the basic information needed to understand the *Singular Value Decomposition* (SVD) as used throughout this book. We start giving the definition of SVD for a generic, rectangular matrix A and discussing some related concepts. We then illustrate three important applications of the SVD:

- solving systems of nonhomogeneous linear equations;
- solving rank-deficient systems of homogeneous linear equations;
- guaranteeing that the entries of a matrix estimated numerically satisfy some given constraints (e.g., orthogonality).

Definition

Singular Value Decomposition

Any $m \times n$ matrix A can be written as the product of three matrices:

$$A = U D V^\top. \tag{A.6}$$

The columns of the $m \times m$ matrix U are mutually orthogonal unit vectors, as are the columns of the $n \times n$ matrix V. The $m \times n$ matrix D is diagonal; its diagonal elements, σ_i, called *singular values*, are such that $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_n \geq 0$.

☞ While both U and V are not unique, the singular values σ_i are fully determined by A.

Some important properties now follow.

Properties of the SVD

Property 1. The singular values give you valuable information on the singularity of a square matrix, *A square matrix, A, is nonsingular if and only if all its singular values are different from zero.* Most importantly, the σ_i also tell you how close A is to be singular: the ratio

$$C = \frac{\sigma_1}{\sigma_n},$$

called *condition number*, measures the *degree of singularity of A*. When $1/C$ is comparable with the arithmetic precision of your machine, the matrix A is *ill-conditioned* and, for all practical purposes, can be considered singular.

Property 2. *If A is a rectangular matrix, the number of nonzero* σ_i *equals the rank of A.* Thus, given a fixed tolerance, ϵ (typically of the order of 10^{-6}), the number of singular values greater than ϵ equals the *effective rank* of A.

Property 3. *If A is a square, nonsingular matrix, its inverse can be written as*

$$A^{-1} = V D^{-1} U^\top.$$

Be A singular or not, the *pseudoinverse* of A, A^+, can be written as

$$A^+ = V D_0^{-1} U^\top,$$

with D_0^{-1} equal to D^{-1} for all nonzero singular values and zero otherwise. If A is nonsingular, then $D_0^{-1} = D^{-1}$ and $A^+ = A^{-1}$.

Property 4. The columns of U corresponding to the nonzero singular values span the range of A, the columns of V corresponding to the zero singular value the null space of A.

Property 5. The squares of the nonzero singular values are the nonzero eigenvalues of both the $n \times n$ matrix $A^\top A$ and $m \times m$ matrix $A A^\top$. The columns of U are eigenvectors of $A A^\top$, the columns of V eigenvectors of $A^\top A$. Morevoer, $A\mathbf{u}_k = \sigma_k \mathbf{v}_k$ and $A^\top \mathbf{v}_k = \sigma_k \mathbf{u}_k$, where $\mathbf{u}_k$ and $\mathbf{v}_k$ are the columns of U and V corresponding to σ_k.

Property 6. One possible distance measure between matrices can use the *Frobenius norm*. The Frobenius norm of a matrix A is simply the sum of the squares of the entries a_{ij} of A, or

$$\|A\|_F = \sum_{i,j} a_{ij}^2. \tag{A.7}$$

By plugging (A.6) in (A.7), it follows that

$$\|A\|_F = \sum_i \sigma_i^2.$$

We are now ready to summarize the applications of the SVD used throughout this book.

Least Squares

Assume you have to solve a system of m linear equations,

$$A\mathbf{x} = \mathbf{b},$$

for the unknown n-dimensional vector $\mathbf{x}$. The $m \times n$ matrix A contains the coefficients of the equations, the m-dimensional vector $\mathbf{b}$ the data. If not all the components of $\mathbf{b}$ are null, the solution can be found by multiplying both sides of the above equation for $A^\top$ to obtain

$$A^\top A\mathbf{x} = A^\top \mathbf{b}.$$

It follows that the solution is given by

$$\mathbf{x} = (A^\top A)^+ A^\top \mathbf{b}.$$

This solution is known to be optimal in the least square sense.

It is usually a good idea to compute the pseudoinverse of $A^\top A$ through SVD. In the case of more equations than unknowns the pseudoinverse is more likely to coincide with the *inverse* of $A^\top A$, but keeping an eye on the condition number of $A^\top A$ (**Property 1**) won't hurt.

☞ Notice that linear fitting amounts to solve exactly the same equation. Consequently, you can use the same strategy!

Homogeneous Systems

Assume you are given the problem of solving a homogeneous system of m linear equations in n unknowns

$$A\mathbf{x} = 0,$$

with $m \geq n-1$ and $rank(A) = n-1$. Disregarding the trivial solution $\mathbf{x} = 0$, a solution unique up to a scale factor can easily be found through SVD. *This solution is simply proportional to the eigenvector corresponding to the only zero eigenvalue of* $A^\top A$ (all other eigenvalues being strictly positive because $rank(A) = n-1$). This can be proven as follows.

Since the norm of the solution of a homogeneous system of equations is arbitrary, we look for a solution of unit norm in the least square sense. Therefore we want to minimize

$$\|A\mathbf{x}\|^2 = (A\mathbf{x})^\top A\mathbf{x} = \mathbf{x}^\top A^\top A\mathbf{x},$$

subject to the constraint

$$\mathbf{x}^\top \mathbf{x} = 1.$$

Introducing the Lagrange multiplier λ this is equivalent to minimize the *Lagrangian*

$$\mathcal{L}(\mathbf{x}) = \mathbf{x}^\top A^\top A\mathbf{x} - \lambda(\mathbf{x}^\top \mathbf{x} - 1).$$

Equating to zero the derivative of the Lagrangian with respect to $\mathbf{x}$ gives

$$A^\top A\mathbf{x} - \lambda\mathbf{x} = 0.$$

This equation tells you that λ is an eigenvalue of $A^\top A$, and the solution, $\mathbf{x} = \mathbf{e}_\lambda$, the corresponding eigenvector. Replacing $\mathbf{x}$ with $\mathbf{e}_\lambda$, and $A^\top A\mathbf{e}_\lambda$ with $\lambda\mathbf{e}_\lambda$ in the Lagrangian yields

$$\mathcal{L}(\mathbf{e}_\lambda) = \lambda.$$

Therefore, the minimum is reached at $\lambda = 0$, the least eigenvalue of $A^\top A$. But from **Properties 4** and **5**, it follows that this solution could have been equivalently established as *the column of* V *corresponding to the only null singular value of* A (the kernel of A). This is the reason why, throughout this book, we have not distinguished between these two *seemingly* different solutions of the same problem.

Enforcing Constraints

One often generates numerical estimates of a matrix, A, whose entries are not all independent, but satisfy some algebraic constraints. This is the case, for example, of orthogonal matrices, or the fundamental matrix we met in Chapter 7. What is bound to happen is that the errors introduced by noise and numerical computations alter the estimated matrix, call it $\hat{A}$, so that its entries no longer satisfy the given constraints. This may cause serious problems if subsequent algorithms assume that $\hat{A}$ satisfies *exactly* the constraints.

Once again, SVD comes to the rescue, and allows us to *find the closest matrix to* $\hat{A}$, *in the sense of the Frobenius norm* (***Property 6***), *which satisfies the constraints exactly*. This is achieved by computing the SVD of the estimated matrix, $\hat{A} = UDV^\top$, and estimating A as $UD'V^\top$, with D' obtained by changing the singular values of D to those expected when the constraints are satisfied exactly.[4] Then, the entries of $A = UD'V^\top$ satisfy the desired constraints by construction.

References

G. Strang, *Linear Algebra and its Applications*, Harcourt Brace Jovanovich, Orlando (FL) (1988).

[4] If $\hat{A}$ is a good numerical estimate, its singular values should not be too far from the expected ones.

A.7 Robust Estimators and Model Fitting

This section sketches a few, introductory concepts behind *robust estimators*, in particular the so called *M-estimators*. Our limited aim is to support the discussion of Chapter 5 by explaining

- why least squares are a *maximum likelihood estimator* from the point of view of statistics,
- why least squares is skewed significantly by outliers, and
- why an estimator based on the least absolute value, as used in Chapter 5, tolerates outliers better than conventional least squares.

The subject has a vast literature; an initial set of further readings is provided at the end of this appendix. Several robust estimators not detailed here have become popular in computer vision. If interested, you should check out at least the *least median of squares*, discussed in the review by Meer *et al.* and detailed in Rousseeuw and Leroy's book, and the *RANSAC (Random Sample Consensus)* algorithm, introduced by Fischler and Bolles and also discussed by Meer *et al.*

Least Squares as Maximum Likelihood Estimator

Consider N data points $\mathbf{p}_i = [x_i, y_i]^\top$, $i = 1, \ldots, N$, and a *model*, $y = f(x, \mathbf{a})$, where $\mathbf{a}$ is a vector of parameters, and f is a known function. Assume that the data points are observations corrupted by noise (to be characterized better in the following). The well-known estimate of the parameter vector, $\mathbf{a}_o$, such that $f(x, \mathbf{a}_o)$ interpolates the data best in the least squares sense is

$$a_o = \min_{\mathbf{a}} \sum_{i=1}^{N} | y_i - f(x_i, \mathbf{a}) |^2 . \tag{A.8}$$

We want to show briefly that $\mathbf{a}_o$ *is the parameter vector maximizing the probability that the data are a noisy version of* $f(x, \mathbf{a})$, given appropriate assumptions on the noise.

Notice that we should estimate $\mathbf{a}$ by maximizing the probability that $\mathbf{a}$ *is correct given the data*, but we cannot estimate this probability (why?). However, if we assume that the noise corrupting each data point is *additive* and *Gaussian*, with *zero mean* and standard deviation σ, and that the amounts of noise at different data points are all independent, we *can* express the probability P that, for a given $\mathbf{a}$, all data points fall within Δy of the true value:

$$P \propto \prod_{i=1}^{N} (e^{-\frac{(y_i - f(x_i, \mathbf{a}))^2}{2\sigma^2}} \Delta y). \tag{A.9}$$

In essence, *maximum likelihood estimation* follows from the assumption that *the parameter vector which maximizes P is also the most likely one to occur given the observed data*. To obtain (A.8) from (A.9) is now easy: we just notice that P is maximized by minimizing the negative of its logarithm; that is,

$$\left[\sum_{i=1}^{N} \frac{(y_i - f(x_i, \mathbf{a}))^2}{2\sigma^2} \right] - N \log \Delta y,$$

and the constants σ, N, Δy can be ignored in the minimization.

Why Least Squares Fits are Skewed by Outliers

Since we assumed that the noise corrupting the data points is Gaussian, the probability that a noisy point lies within a distance d of its corresponding true point decreases very rapidly with d. Consequently, the least squares estimator believes that most points lie within a few standard deviations of the true (unknown) model. Now suppose that the data contain even a small percentage of points that are just *way off* and presumably not consistent with the Gaussian hypothesis.[5] Points like these are called *outliers*. In the absence of further information, a least squares estimator believes that outliers too are close to the model, as the probability of an outlier being as far as it really is from the true model is practically zero. The result is that the outliers "pull" the best fit away from the true model much more than they should.

Why Absolute Value Estimators Tolerate Outliers Better

The essence of the problem with least squares is that the Gaussian distribution decreases very rapidly as d becomes larger than σ. Therefore, a solution is to adopt a noise distribution which does not vanish as quickly as a Gaussian; that is, *which considers outliers more likely to occur*. An example of such a distribution is the *double exponential*,

$$Pr\{d\} \propto e^{-|\frac{y_i - f(x_i, \mathbf{a})}{\sigma}|}.$$

In this case, the probability P becomes

$$P \propto \prod_{i=1}^{N} (e^{-|\frac{y_i - f(x_i, \mathbf{a})}{\sigma}|} \Delta y). \tag{A.10}$$

The same reasoning which took us from (A.9) to (A.8) takes us from (A.10) to the *robust* maximum likelihood estimator

$$\min_{\mathbf{a}} \sum_{i=1}^{N} | y_i - f(x_i, \mathbf{a}) | .$$

The price we pay is that, unlike (A.8), this problem cannot be solved in closed form in most cases, and numerical methods must be employed.

[5] The reasons for these points being in the data set can be diverse and, for the purpose of this brief introduction, not quite relevant.