A Sampler of Useful Computational Tools for Applied Geometry, Computer Graphics, and Image Processing

By: Daniel Cohen-Or, Chen Greif, Tau Ju, Niloy J. Mitra, Ariel Shamir, Olga Sorkine-Hornung, Hao (Richard) Zhang, and Gil Hoffer

Reviewed by: Yotam Gingold

August 11, 2016

This book provides a tour of approaches to problem solving for computer graphics and geometry, image, and video processing. The book takes a computational point of view towards solving the kind of 2D and 3D problems that arise. By "computational," I mean problems are solved constructively, as steps to perform, with considerations of efficiency. The list of topics covered by the book is quite broad: geometry, linear algebra, least squares fitting, solving linear systems, principal component analysis, spectral analysis, Laplace and Poisson equations, curvature, curve and surface topology, the maximum-flow/minimum-cut problem, and skewing schemes. The applications include distances, aligning shapes, smoothing and compression, dimensionality reduction, scattered data interpolation, image remixing, image segmentation, image and video resizing, shape matching, and even efficient memory access.

This is a technical book and not an intuition-only, "hand-wavy" book. It takes some mathematical maturity and comfort with geometry, algebra, and calculus. It is perfect for the novice or intermediate practitioner (such as a graduate student) who stands to gain connective tissue between topics seen as an undergraduate and their chosen field of study. None of the topics are likely to have been encountered in an undergraduate computer science course on computer graphics, yet the topics and problem solving approaches are prevalent in computer graphics research.

The book is not long, yet covers all of these topics. To accomplish this, the authors assume outside knowledge. The authors assume a working knowledge of (vector) calculus and, occasionally, additional knowledge such as the Laplace operator, frequency domain, Fourier transform, back substitution, and condition number. How to go about programming the mathematics is also assumed.

The book is structured rather unlike a mathematics text book, in which later chapters build upon definitions and theorems from earlier chapters. As a sampler, or grab bag, the chapters can be read more-or-less independently. The chapters only rarely refer to each other. (There are a few missed opportunities.)

1 The Chapters

Chapter 1 Analytical Geometry explains, with examples, the benefits of using geometric insight to solve problems rather than jumping straight to algebraic manipulations of symbols. In one example, the distance between two lines in 3D is converted into the distance between two points in 2D. The process is magnificent. The reader may wonder whether computing the distance via the geometric simplification is computationally more efficient. It is unclear. In this case, a purely symbolic approach may be more efficient than computing the needed projections. That is besides the point. The true power of this style of thinking is the insights that come with it, which may allow a problem solver to discover an equivalent yet simpler geometric representation with a conceptually simpler and sometimes dramatically more efficient solution. These insights can be missed when attacking pools of symbols with algebraic tools.

Chapter 2 Linear Algebra is a refresher of the linear algebra concepts that are used throughout the book. To its credit, the book favors intuitive or convenient explanations over rigor. It closes with a section on homogeneous coordinates. These are a simple and useful technique to apply translations with a linear operator (matrix), but may be unfamiliar to readers without a computer graphics background.

Chapter 3 Least-Squares Solutions considers fitting curves and surfaces to data points, where the curves and surfaces are linear combinations (scaled sums) of arbitrary functions and the linear coefficients are the unknown fitting parameters. Least squares solutions are common throughout computer graphics, geometry, and image processing. The chapter also considers weighting data points differently to account for outliers. The chapter does well to point out that linear least-squares does not measure orthogonal distances. The chapter describes a reasonable compromise for 3D surface fitting and derives a complete solution for the special case of fitting a straight line.

Chapter 4 PCA and SVD does a wonderful job explaining Principal Component Analysis and Singular Value Decomposition and their power. The explanation includes figures showing how matrices takes hyper-spheres to hyper-ellipses. The chapter also explains an efficient trick for computing the PCA of a large number of points without creating an enormous matrix. (It is not clear if this trick is more efficient than using SVD, as also suggested by the chapter.) These tools are applied to find tight-fitting, oriented bounding boxes for a shape; surface normals for point cloud data; and the optimal rigid alignment (translation and rotation) between two corresponding sets of points. A complete derivation for the optimal rigid alignment is included. The derivation could be improved. The proof that the rigid transformation aligns centers of mass seems backwards to me. It sets them equal, and then shows what the translation must be to make it so. The sentence following the proof suggests an alternative that is logically correct and immediately obvious, by differentiating the least squares expression with respect to the translation. Some errors in the equations make the derivation impossible to follow. (See my errata for the corrections.) An additional sentence summarizing the $\text{Trace}(M) \geq \text{Trace}(BM)$ theorem's implications would clarify the derivation. I propose, "In our setting, this means that if $M = RH$ is symmetric positive definite, then $\text{Trace}(RH)$ is always larger than a rotation of RH. There is no better rotation than the one which makes H symmetric positive definite."

Chapter 5 Spectral Transform introduces a framework for decomposing curves and surfaces into frequency components, akin to a Fourier transform, by analyzing the behavior of the Laplace operator. This is applied to compressing and smoothing curves and surfaces by discarding high frequency information, leading to smoother shapes that require less storage. This is analogous to how images are often compressed. The connection between spectral analysis of the Laplacian and the Fourier transform is introduced gently, which I appreciate. No time is spent on what the frequency domain is, so the reader should come in with that knowledge (and gain the knowledge of how it relates to the Laplace operator). The section on surface meshes may leave the reader scratching their head, since the Laplacian matrix defined in equation 5.10 is not symmetric. (As an alternative, $T = D - W$ is symmetric.) The cotangent Laplace operator defined on page 79 will lead the typical reader to assume that the terms should be computed via trigonometry, the cotangents of angles. In practice, each cotangent can be expressed in terms of a dot product and triangle area—or entirely in terms of edge lengths—via the law of sines, law of cosines, and Heron's formula. This has confused my students in the past. An introductory book such as this would do a service to the reader to point this out.

Chapter 6 Linear Systems introduces algorithms for solving linear systems of equations. The chapter is an overview of direct (closed form) methods and iterative methods. The chapter focuses on computational issues such as running time, memory usage, and numerical stability. The chapter covers many methods, and the uninitiated reader will learn something useful about each. In comparison with the rest of the book, this chapter is about nuts and bolts rather than intuitive explanations. Near the end of the chapter is a section, “What method should you choose?” This section would be even more useful presented as a decision tree, since some of the decision criteria overlap and the preference order isn’t clear. Probably one should try all applicable methods.

There are a few omissions. The reader must look elsewhere for a definition of backward substitution and, therefore, why an upper or lower diagonal matrix is so easy to solve. The chapter misses an opportunity when defining a 2D Laplacian (page 85), to refer to chapters 5 (Spectral Transform) or 7 (Laplace and Poisson) where such a matrix is used. Bandwidth and fill-in are not defined where first used (page 85), though the “narrow-banded matrices” subsection immediately prior would be appropriate for defining bandwidth, and the subsequent section (“Iterative Methods”) does define fill-in. The section “Basic Stationary Methods” (page 88) could explain in a few more words what the iteration is doing: the right hand side is computed directly with a matrix-vector product; the left hand side matrix M is the linear system to be solved to obtain the next iteration of x. I would have liked more intuition about the conjugate gradient directions or the Krylov subspace, e.g. why is the Krylov subspace a good or reasonable subspace? The condition number of a matrix is not defined (page 92).

Chapter 7 Laplace and Poisson presents an easy-to-grasp view of gradient-based image manipulation. These approaches are based on solutions to Laplace and Poisson equations. They have proven very useful in removing blemishes or seamlessly copying-and-pasting a piece of one photograph into another. The presentation is quite gentle. The problem is first explained (discretely) and solved in terms of image pixel values, which readers with a computer science background will appreciate. Then, the problem is re-explained and solved in terms of functions on a continuous domain, which readers with a mathematics background will appreciate. (The continuous explanation would be improved by skipping fewer steps when substituting into the Euler-Lagrange equation between equations 7.6 and 7.7.) The connection between the discrete and continuous domains is made more apparent when the continuous setting is discretized and a solution found by constructing and solving a linear system of equations. The last portion of the chapter, “Laplace on meshes,” briefly presents an extension of the ideas to triangulated surfaces in 3D. It may whet readers’ appetite, or it may simply leave them scratching their heads, wondering where Poisson went and why we need the bi-Laplace rather than Laplace equation.

Chapter 8 Curvatures discusses the “amount of bending of a curve or a surface.” The chapter uses two motivating examples: approximating a surface with smaller or larger polygons according to its curvature, and finding a “signature” for a small patch of surface to be used as a hash function for finding similar surface patches. Explanations are intuitive and simple to grasp. (For rigor, the chapter points to the book by do Carmo.) The chapter first defines curvature for curves in 2D and then surfaces in 3D. Curvature is more complicated on a surface, where the area around a point can be flat (curvature = 0) in one direction and very curved in another. The chapter states that we only need to study the maximum and minimum curvatures, and then goes on to define Gaussian and mean curvature, developable surfaces, and minimal surfaces. (The paragraph on page 128 describing the intrinsic nature of Gaussian curvature is probably too dense without accompanying figures.) The chapter contains formulas for evaluating curvature on continuous curves and surfaces. To create a continuous surface from a discrete one, the chapter directs readers to fit a local quadratic polynomial using the least squares fitting in Chapter 3. The chapter closes with an optional section on the Gauss-Bonnet theorem relating the integral of Gaussian curvature over a geodesic triangle to the sum of angles at its vertices. This optional section appears intended to inspire the reader to study curvature in more depth, rather than to present a practical tool.

Chapter 9 Dimensionality Reduction does a nice job of tying in material from the earlier chapter on Principal Component Analysis (PCA), showing how it can be used to create a linear subspace of face images. The subspace has few dimensions compared to the space of all possible face images. The chapter then explains with the example and visualizations of human motion why linear dimensionality reduction is insufficient. The next example connects dimensionality reduction to clustering, the task of grouping similar points together. (The text is aware that the connection is unclear, and indeed the connection remains unintuitive.) By first increasing the number of dimensions, linear dimensionality reduction and clustering algorithms can work. The chapter describes Kernel PCA, a technique to do just this. At the end of the explanation of Kernel PCA, an example is suggested for the kernel matrix, the Gaussian of the Euclidean distance. It would crystallize the explanation to provide an explicit example for the entries of K, to show that no high dimensional ϕ mapping or even actual dot products are ever computed. Figure 9.5 provides such an explicit definition for K's entries in its caption, but the reader is not told this when the Kernel PCA section's text refers to the figure. There is a section on multidimensional scaling (MDS). The motivating example is texture mapping, which is perfect for a computer graphics reader. The explanation hints but doesn't state explicitly that MDS is the same as kernel PCA but with a different scaling. The chapter concludes with a discussion of pose normalization (finding corresponding points on the surface of an animal in different poses, such as running versus sitting) and describes a few techniques which make use of the chapter's concepts.

Chapter 10 Scattered-Data Interpolation walks the reader through progressively more complex yet more satisfying solutions to this fundamental problem. The simplest solutions work, but are not smooth. The first smooth solution, Shepard's interpolation, is too wavy. (The text doesn't say it, but the numerator of Shepard's bump function, equation 10.3, is simply one over the Euclidean distance when $\alpha = 2$. The denominator is the same for all bump functions, chosen so that the sum of bump functions $\sum f_i = 1$.) A naive, very smooth solution using sums of Gaussians doesn't interpolate the original data, but it's easy to scale them so that they do—by solving a linear system of equations. The text makes clear that the Gaussian bump, or radial basis, functions can be replaced with other functions, but doesn't explain why we would want to. Radial basis functions can't reproduce polynomials without a "patch," however, so they are not a silver bullet. Thankfully, the patch is described in detail. (One detail is provided without explanation, except for the cryptic statement that it enforces square integrable second derivatives.) The chapter closes with an overview of practical issues when using radial basis function interpolation given irregular or large quantities of points, and references to consult for further reading. No mention is made of data interpolation in a domain with a boundary. Readers will have to consult a more advanced text.

Chapter 11 Topology introduces readers to definitions and properties of shapes that are global in nature and related only to the connectedness of the shape. The text describes the historical genesis of the field of topology, Euler's Seven Bridges of Königsberg. The topological concepts introduced are those which are relevant to curves and surfaces in 2D and 3D, manifolds with and without boundaries. The chapter covers the definition of a manifold, connected components, genus, orientability, and the Euler characteristic. Although the coverage of each is brief, definitions and examples are provided. It is as much as most practitioners need to know. The chapter closes by mentioning how two problems, surface reconstruction and shape matching, benefit from taking topological information into account.

Some changes would improve clarity. The text should inform the reader that in topology, the term sphere refers to the surface of a ball, and torus refers to the surface of a solid torus or donut. (Confusingly, solid and hollow spheres and torii are briefly mentioned later in the chapter. The solid versions of these shapes should rightly lead the reader to scratch their heads about whether the distinction matters for any of the topological definitions and examples; it does.) The continuous definition of orientability is confusing. Figure 11.12 illustrating the Euler characteristic could show a tetrahedron, as that is the example used in the text. The average vertex valence argument could use a figure or counting argument to explain why (approximately) 3 edges per vertex implies an average valence of 6.

Chapter 12 Graphs and Images relates the abstract concept of graphs (nodes and edges) with images. The first example relates a flood fill algorithm on an image to a region growing algorithm on a graph. The second example converts binary image segmentation, which has a naive exponential-time solution, into a max-flow min-cut problem, for which there are polynomial time solutions. The third algorithm resizes images (and, at the end of the chapter, videos) for viewing on screens with different aspect ratios. The chapter explains how to construct equivalent graphs to use as input to a max-flow min-cut algorithm. This is important. The chapter also explains the relationship between max-flow and min-cut, two seemingly unrelated graph problems, and then describes multiple algorithms to compute solutions. Some readers may want to skim these passages. An elegance of the graph approach to image segmentation and resizing is that it can also apply to videos (3D image volumes) or 3D meshes, where the graph's nodes are the vertices or faces and the graph's edges link them to their neighbors.

Considering images to be graphs is worthwhile for problem solving. It is worth remembering that the resulting graphs are a special case of graphs in general. While general graph algorithms can be directly applied, in practice, specialized implementations of the algorithms are often used for performance reasons. The chapter describes such a specialization (the Boykov-Kolmogorov algorithm) for the maximum flow problem. Sometimes, an efficient implementation of a graph algorithm that takes the regular structure of an image into account ends up looking more like FloodFill than the general graph RegionGrow.

Chapter 13 Skewing Schemes draws readers in with the queens problem in chess (placing queens so that they cannot attack each other). A version of this problem is equivalent to Latin squares, placing numbers on a grid such that no number repeats along any row or column (or diagonal). These arrangements of numbers in a grid are called skewing schemes. The chapter then shows how to create perfect, easy-to-compute schemes by choosing relatively prime coefficients. The chapter asserts that this has applications to efficient memory lookups along rows or columns or diagonals of a matrix or volumetric grid. I don't believe modern single-CPU architectures would benefit, where sequential arrangement is preferred for cache locality. However, the approaches are still relevant for GPU and distributed computing, and Latin squares are useful for experiment design.

2 Conclusion

I enjoyed reading this book. I will now venture to guess whether you will enjoy reading this book or whether you can make use of it. If you wish to teach a graduate course on problem solving in geometry or computer graphics, this book will make for a very enjoyable semester. Because of the computational outlook of the book, it would be straightforward to create a rewarding programming assignment for almost every chapter. If you are a beginning graduate student or a practitioner looking to solve problems—the kind of person who would benefit from the aforementioned course—you will learn a lot from reading this book. If you enjoy recreational mathematics and have the requisite background, this book will inspire you with its approaches to problem solving and reward you with connections between visual problems and more abstract mathematics. If you are interested in a theoretical understanding of the techniques, this book will point you in the right direction but will not take you there. If you are looking to implement computer graphics or geometry processing algorithms, this book will not tell you how.

Errata

Chapter 1 Analytical Geometry, page 8: $l = \frac{\langle v,w \rangle}{\|v\|}$.

Chapter 2 Linear Algebra, page 16, last paragraph: "if their product is zero" should be "if their dot product is zero."

Chapter 2 Linear Algebra, page 27: The equations following "and thus we have" are erroneous. They should follow the pattern: $\langle \mathbf{a}, \mathbf{v_i} \rangle = \alpha_\mathbf{1} \langle \mathbf{v_1}, \mathbf{v_i} \rangle + \alpha_\mathbf{2} \langle \mathbf{v_2}, \mathbf{v_i} \rangle + \alpha_\mathbf{3} \langle \mathbf{v_3}, \mathbf{v_i} \rangle$ for each row i. The matrix version is not wrong, but the matrix should be transposed for the entries to match the equations.

Chapter 4 PCA and SVD, pages 60 and 61: The transposes are missing on R in the $q_i^T R p_i$ terms. The left-most or right-most term in the first equation on page 61 should be $q_i^T R^T p_i$.

Chapter 5 Spectral Transform, page 74, paragraph after equation 5.9: I believe the matrix should be defined as $G_{j,k}$ so that the k-th column is the k-th eigenvector.

(typography) Chapter 6 Linear Systems, page 92: In the equation $x = M_2^{-1}y$, the M should be italic, not bold, for consistency.

Chapter 8 Curvatures, page 123: "nominator" should be "denominator."

(grammar) Chapter 8 Curvatures, page 128: "if Monge patch representation" should be "if a Monge patch representation"

Chapter 9 Dimensionality Reduction, page 137, Figure 9.4 caption: "(b) Linear PCA properly reflects the curve structure" should be "(b) Linear PCA improperly reflects the curve structure."

Chapter 9 Dimensionality Reduction, page 141: The second paragraph introduces x_j but then uses x_i for the remainder of the paragraph.

Chapter 9 Dimensionality Reduction, page 141, equation 9.3: The last minus sign should disappear so that the last term becomes $1_n K 1_n$, and the term should be added not subtracted, so that the entire equation becomes $\bar{K} = K - 1_n K - K 1_n + 1_n K 1_n$.

(clarity) Chapter 10 Scattered-Data Interpolation: Figure 10.10, which shows the waviness, doesn't specify which radial basis function is used.

Chapter 11 Topology, page 169, first line: The text states that "the surface shown on the right in Figure 11.9" is the surface with boundary edges. However, the surface shown on the left is the one with boundary edges.

Chapter 12 Graphs and Images, Figure 12.1: The cut in (b) partitions the vertices into three disjoint sets, not two as the caption states. (The text later, on page 186, refers to this figure and describes a cut as partitioning a graph into two disjoint sub-graphs.)

Chapter 12 Graphs and Images, page 180: The RegionGrow pseudocode's for loop says "neighbors of u of v" when it should be "neighbors u of v."

Chapter 12 Graphs and Images, page 182: There is an extra open brace on the second line of the Watershed pseudocode.

(grammar) Chapter 12 Graphs and Images, page 185: In "the cost of pixels similarity," an apostrophe is missing or else "pixel" (singular) should be used.

(grammar) Chapter 12 Graphs and Images, page 189: In the Ford-Fulkerson pseudocode, there is a space missing before "node" in the parameter list. "for each edges" should be "for each edge." There are two rather than one hyphens between Ford and Fulkerson in the algorithm name.

(typography) Chapter 12 Graphs and Images, page 191: In the Boykov-Kolmogorov pseudocode, there is a space missing before "node" in the parameter list.

(grammar) Chapter 12 Graphs and Images, page 194, Figure 12.13: The last sentence has pluralization inconsistency. Either remove the first word ("A") or don't pluralize "seam."

Chapter 12 Graphs and Images, page 197, Figure 12.14: The bottom row (a) and (b) subfigures have incorrect indices.

(clarity) Chapter 12 Graphs and Images: The text on page 203 describing the forward energy graph construction uses different indices for $\pm LU$ than the node labels in Figure 12.17 (e).

Chapter 13 Skewing Schemes, page 209: The text states that Figure 13.5 (b) shows the skewing scheme $m = 2x + 6y \pmod 7$, but Figure 13.5 (b) actually shows $m = x + 3y \pmod 8$.

Chapter 13 Skewing Schemes, page 209: The equations' less-than-or-equal-to signs are improperly typeset as <=. The first equation has an inequality $m < n$ and some extra space between that and x, y, z. I think the $m < n$ is an accident and the inequality portion of the first equation should be the same as the inequality portion of the second equation. The second equation writes (mod N) where it should be (mod n).

(clarity) Chapter 13 Skewing Schemes, page 210: "there must be an odd number of coefficients" should say "there must be odd-numbered coefficients" or "the coefficients must be odd."