EUROGRAPHICS 2020 / U. Assarsson and D. Panozzo
(Guest Editors)

Volume 39 (2020), Number 2

Fast and Robust QEF Minimization using Probabilistic Quadrics

P. Trettner[1] and L. Kobbelt[2]

[1]RWTH Aachen University, Germany

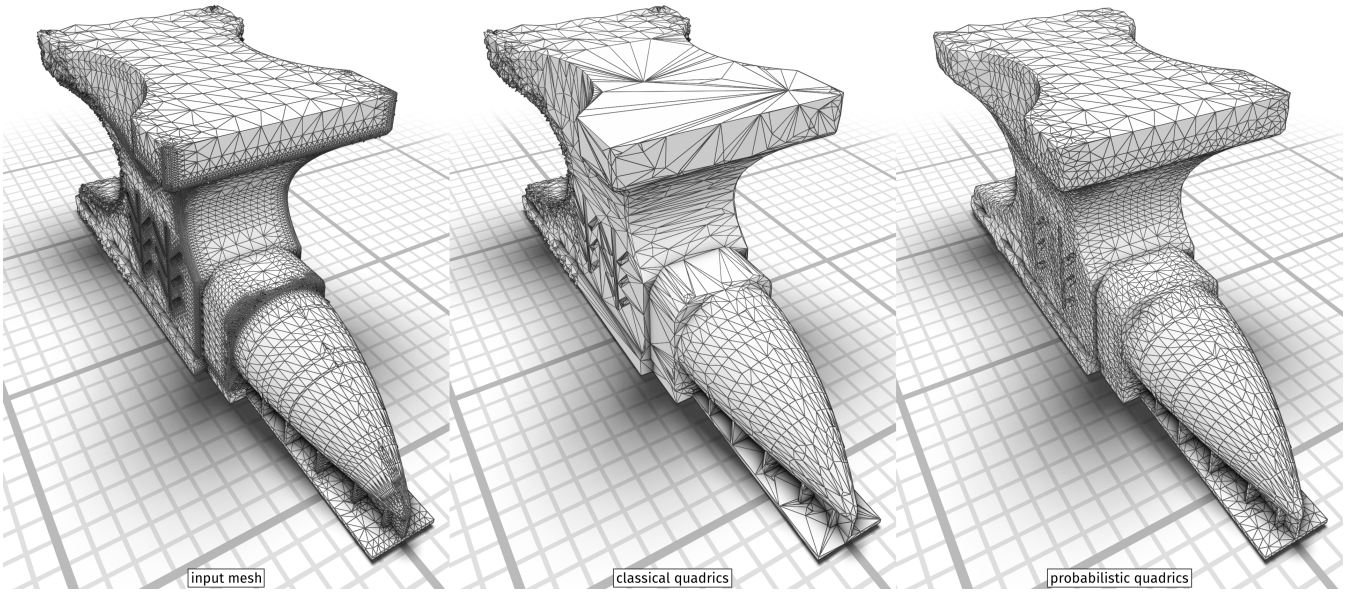

Figure 1: *Classical quadrics assume ground truth input data. Many algorithms, like the incremental decimation shown here, fail to account for noisy input and perform suboptimally in the presence of uncertain data. We introduce a new class of quadrics,* probabilistic quadrics*, that are inherently modeled around noisy data and can take arbitrary, spatially-varying distributions into account. The center and right mesh are each decimated by* 85%*. The mesh is more noisy in the back than in the front. Classical quadrics waste many vertices in noise, misclassifying it as sharp features, and produce irregular triangulations in planar regions. In contrast, probabilistic quadrics can adapt to the noise, favor regular triangulations, and still preserve sharp features. Model taken from [ZJ16].*

Abstract

Error quadrics are a fundamental and powerful building block in many geometry processing algorithms. However, finding the minimizer of a given quadric is in many cases not robust and requires a singular value decomposition or some ad-hoc regularization. While classical error quadrics measure the squared deviation from a set of ground truth planes or polygons, we treat the input data as genuinely uncertain information and embed error quadrics in a probabilistic setting ("probabilistic quadrics") where the optimal point minimizes the expected *squared error. We derive closed form solutions for the popular plane and triangle quadrics subject to (spatially varying, anisotropic) Gaussian noise. Probabilistic quadrics can be minimized robustly by solving a simple linear system — 50× faster than SVD. We show that probabilistic quadrics have superior properties in tasks like decimation and isosurface extraction since they favor more uniform triangulations and are more tolerant to noise while still maintaining feature sensitivity. A broad spectrum of applications can directly benefit from our new quadrics as a drop-in replacement which we demonstrate with mesh smoothing via filtered quadrics and non-linear subdivision surfaces.*

CCS Concepts

• ***Computing methodologies*** → ***Mesh models; Mesh geometry models;***

1. Introduction

Quadratic error functions (QEFs) are a popular choice for defining distance, loss, energy, or fitness functions in various optimization settings. In the three-dimensional case, a quadric can be represented by a symmetric matrix $Q \in \mathbb{R}^{4\times4}$ and the corresponding QEF is defined as $f(x) = x^T Q x$ for x in homogeneous coordinates. Q has the following structure:

$$Q = \begin{bmatrix} A & -b \\ -b^T & c \end{bmatrix} \tag{1}$$

Thus, f can also be written as $x^T A x - 2x^T b + c$, with a symmetric positive semidefinite matrix $A \in S^3_+$, $b, x \in \mathbb{R}^3$, and $c \in \mathbb{R}$. The point minimizing the quadratic error (the minimizer of Q) can be computed via

$$x_{\min} = A^{-1} b \tag{2}$$

Often, individual quadrics are formulated for certain objects or sub-problems. The final function to be optimized consists of the sum of various quadratic functions. Due to the linearity of the matrix product, instead of optimizing a sum of quadratic functions, one can sum the individual quadric matrices (e.g. Q_a and Q_b) into a new quadric Q' and optimize $x^T Q' x$:

$$f(x) = x^T Q_a x + x^T Q_b x = x^T (Q_a + Q_b) x = x^T Q' x \tag{3}$$

This insight lies at the heart of many quadric-based optimizations such as incremental decimation [LT98, GH97], vertex clustering [Lin00], isosurface extraction [KBSS01, JLSW02, SW04], and mesh-filtered quadrics [VNMC10, LTB19]. While often interpreted geometrically, quadrics are a versatile tool and can be used for basically any continuous attribute [Hop99].

While many geometric problems are intuitively formulated and optimized by QEFs, this process is not always robust. Many quadrics correspond to singular matrices, most notably the quadric describing the quadratic distance to a plane. To uniquely determine the minimizer of $f(x)$, it requires at least the sum of three linearly independent plane quadrics. Even then, the result is not robust if any planes are close to being coplanar, a problem often noted in decimation and isosurface extraction. This problem is magnified if the quadrics are affected by aliasing or noise, e.g. if they originate from sampled measurements or other forms of discretization.

The conceptual idea behind classical error quadrics is to consider the input geometry (e.g. input planes and triangles) as ground truth and measure the deviation of a point from it by the sum of the squared distances. The optimal point is then the least squares minimizer of the quadratic error function. For noisy (or uncertain) input data, the least squares solution is expected to implicitly "average out" the noise.

The biggest issue with classical error quadrics is that (near) degenerate (i.e. rank deficient) configurations are quite common such that pseudo-inverses have to be computed via SVD in order to guarantee a robust solution of the least squares problem.

In our approach we follow a different rationale. We assume that the input quadrics are samples drawn from a certain, potentially complex distribution. In particular, we will examine the case of planes built from point and normal, both subject to Gaussian noise, and triangles built from three points, each subject to Gaussian noise. The optimal point is then the minimizer of the expected quadratic error or, equivalently, the minimizer of the expected error quadric. We call this expected error quadric the "probabilistic error quadric".

1.1. Contribution

In summary, our method is a principled and robust way to incorporate arbitrarily distributed noise directly into the creation of error quadrics and thus let any quadric-based technique benefit from this probabilistic approach.

We contribute:

- the insight that the expected value over a distribution of quadrics can be recast into an ordinary quadratic form using a special "probabilistic" quadric.
- closed-form solutions for the popular plane and triangle quadrics under (potentially anisotropic) Gaussian noise.
- a theoretical foundation for the validity of common regularization techniques that can be applied to minimize quadrics robustly.

After describing our method in Section 3, Section 4 examines the properties of our quadrics in various experiments. Probabilistic quadrics are useful in a broad spectrum of applications which we demonstrate with isosurface extraction, mesh smoothing via filtered quadrics, and subdivision surfaces in Section 5.

A reference implementation of our probabilistic quadrics is provided as a C++ header under the permissive MIT license at https://graphics.rwth-aachen.de/probabilistic-quadrics.

2. Related Work

Quadratic error functions are a popular tool with a long history in geometry processing.

One of the first use of error quadrics was for mesh decimation. Garland and Heckbert [GH97] present an *incremental decimation* scheme using arbitrary vertex-pair collapses. They store per-vertex quadrics that are summed up on each contraction, choosing the new vertex position as the minimizer of the quadric sum. The quadrics are constructed as the sum of the quadratic distance from the planes defined by the adjacent triangles. Salinas *et al.* [SLA15] take additional plane proxies into account to better preserve planar structures. Li and Zhu [LZ08] further incorporate local curvature.

Vertex clustering also benefits greatly from quadrics. Lindstrom [Lin00] presents an out-of-core simplification system based on vertex clustering where the space complexity only depends on the output complexity. The representative points per cell are computed by minimizing an error quadric. They also propose the singular value decomposition as a robust way to minimize quadrics.

Error quadrics are not restricted to 3D geometry: Hoppe [Hop99] presents quadrics for appearance attributes and Garland and Zhou [GZ05] generalize quadric-based simplification to higher dimension.

A common problem in isosurface extraction is the preservation of sharp features. This can be done by incorporating plane-based error quadrics. Based on this idea, Kobbelt *et al.* [KBSS01] extend the *Marching Cubes* algorithm. Ju *et al.* [JLSW02] introduce *Dual Contouring* which stores Hermite data on each edge of a voxel grid and the vertices of the extracted geometry are found by minimizing an error quadric associated with each voxel. Schaefer and Warren [SW04] present *Dual Marching Cubes* where they apply a feature preserving Marching Cubes to the dual graph of an octree, informing the subdivision level by the quadratic error.

Recently, averaging and filtering quadrics across the surface of a mesh to perform tasks like smoothing and clustering gained in popularity. Vieira *et al.* [VNMC10] compute averaged quadrics over some surface area and move each vertex to its quadric minimizer to perform denoising. Legrand *et al.* [LTB19] use a bilateral filtering of quadrics for smoothing and clustering.

Quadrics are also popular in shape approximation. Yan *et al.* [YWLY12] directly fit quadrics to the surface to perform variational mesh segmentation. Thiery *et al.* [TGB13] introduce *Sphere-Meshes* for mesh approximation based on spherical error quadrics. They later use these meshes for animation [TGBE16]. Calderon and Boubekeur use quadrics as part of their bounding proxy generation [CB17].

3. Method

We will start by deriving probabilistic quadrics for planes and triangles under Gaussian noise. These quadric are formed by taking a classical quadric, choosing a distribution of input parameters, and then computing the expected value of Q (in the form of $\mathbb{E}[A]$, $\mathbb{E}[b]$, and $\mathbb{E}[c]$). Afterwards we discuss the stability of these quadrics and present the general case.

3.1. Planes under Gaussian Noise

Given a plane defined by position q and normal n, we compute the quadratic distance to this plane via

$$\begin{aligned} d(x) &= \langle x-q, n\rangle^2 \\ &= x^T nn^T x - 2q^T nn^T x + q^T nn^T q. \end{aligned} \tag{4}$$

This is a quadric with $A = nn^T$, $b = nn^T q$, and $c = q^T nn^T q$.

If normal and position are estimates from measurements, they could be modeled by independent multivariate Gaussian distributions with means $\bar{n}, \bar{q} \in \mathbb{R}^3$ and variances $\Sigma_n, \Sigma_q \in \mathbb{R}^{3\times 3}$:

$$n \sim \mathcal{N}(\bar{n}, \Sigma_n), \quad q \sim \mathcal{N}(\bar{q}, \Sigma_q) \tag{5}$$

Recalling that the second moment of a Gaussian is given by $\mathbb{E}_x[xx^T] = \bar{x}\bar{x}^T + \Sigma_x$ we can derive the probabilistic quadric $Q_{p(n,q)}$:

$$\mathbb{E}[A] = \mathbb{E}_{n,q}[nn^T] = \bar{n}\bar{n}^T + \Sigma_n \tag{6}$$

Because n and q are independent, $\mathbb{E}[b]$ can be computed as

$$\begin{aligned} \mathbb{E}[b] &= \mathbb{E}_{n,q}[nn^T q] = \mathbb{E}_n[nn^T]\mathbb{E}_q[q] = (\bar{n}\bar{n}^T + \Sigma_n)\bar{q} \\ &= \bar{n}\bar{n}^T\bar{q} + \Sigma_n\bar{q}. \end{aligned} \tag{7}$$

For quadratic forms it can be shown that $\mathbb{E}_x[x^T Mx] = \bar{x}^T M\bar{x} + \mathrm{Tr}[M\Sigma_x]$ [MP92] where $\mathrm{Tr}[\cdot]$ is the trace of a matrix. Finally, using $\mathrm{Tr}[xx^T M] = x^T Mx$ we can compute $\mathbb{E}[c]$ as:

$$\begin{aligned} \mathbb{E}[c] &= \mathbb{E}_{n,q}[q^T nn^T q] = \mathbb{E}_q[q^T \mathbb{E}_n[nn^T]q] \\ &= \mathbb{E}_q[q^T(\bar{n}\bar{n}^T + \Sigma_n)q] \\ &= \bar{q}^T(\bar{n}\bar{n}^T + \Sigma_n)\bar{q} + \mathrm{Tr}[(\bar{n}\bar{n}^T + \Sigma_n)\Sigma_q] \\ &= \bar{q}^T(\bar{n}\bar{n}^T + \Sigma_n)\bar{q} + \mathrm{Tr}[\bar{n}\bar{n}^T\Sigma_q] + \mathrm{Tr}[\Sigma_n\Sigma_q] \\ &= \bar{q}^T\bar{n}\bar{n}^T\bar{q} + \bar{q}^T\Sigma_n\bar{q} + \bar{n}^T\Sigma_q\bar{n} + \mathrm{Tr}[\Sigma_n\Sigma_q]. \end{aligned} \tag{8}$$

Compared to the original quadric, we add Σ_n to A and $\Sigma_n q$ to b. Two insightful interpretations of this probabilistic quadric are:

1. The probabilistic plane quadric is the sum of the classical plane quadric and the quadric for $||x-q||^2$, weighted by Σ_n.
2. The probabilistic plane quadric is the classical plane quadric, solved with generalized Tikhonov regularization in the form of $||Ax-b||^2 + ||x-q||^2_{\Sigma_n}$.

Note that Σ_q does not affect A and b and thus does not contribute to the quadric minimizer $x_{\min}$.

Usually, the noise level Σ_n is small. Using the first interpretation we could argue that adding $||x-q||^2$ has little effect due to the weight Σ_n. This is correct, as long as A is not close to being singular. However, for singular or almost singular A, adding $||x-q||^2_{\Sigma_n}$ has the desired property of choosing $x_{\min}$ close to q.

Another significant advantage is that, when summing the probabilistic quadrics over multiple planes, each Σ_n can be chosen differently. For example, if the planes result from laser scans, samples farther away from the scanner have higher noise levels and thus should have larger Σ_n. Furthermore, the noise is not isotropic: the depth error is usually larger than horizontal or vertical error. During normal estimation, the distribution of neighboring points can be taken into account to construct an anisotropic Σ_n.

When drawn from the Gaussian distribution, normals become non-normalized resulting in expected values weighted by normal length. While it is possible to use distributions that produce only unit-length vectors, the formulas become disproportionate more complex. In our experiments, the differences are negligible when Σ_n is small or large and isotropic. Only highly anisotropic large Σ_n produce noticeable deviations.

3.2. Triangles under Gaussian Noise

Lindstrom *et al.* [LT98,Lin00] proposed the following error quadric for triangles $t = (p,q,r)$ with $p,q,r \in \mathbb{R}^3$:

$$\begin{aligned} Q &= \begin{bmatrix} A & -b \\ -b^T & c \end{bmatrix} = nn^T \\ n &= \begin{pmatrix} p\times q + q\times r + r\times p \\ -|p,q,r| \end{pmatrix} \end{aligned} \tag{9}$$

In this formulation, n is a 4-vector and $|p,q,r|$ the scalar triple product. These quadrics are automatically area-weighted. Small or highly anisotropic triangles are often problematic. Even with area weights they lead to problems, especially when the normals deviate strongly or in the presence of fold-overs (cf. Figure 2).

For the probabilistic quadric we will again model the uncertainty as a multivariate Gaussian, this time as three independently distributed positions:

$$p \sim \mathcal{N}(\bar{p}, \Sigma_p), \quad q \sim \mathcal{N}(\bar{q}, \Sigma_q), \quad r \sim \mathcal{N}(\bar{r}, \Sigma_r) \tag{10}$$

The full derivation is in Appendix A (Theorem A.8, A.9, and A.10). There we show that:

$$\begin{aligned}
\mathbb{E}[A] = \; & \mathbb{E}[(p\times q + q\times r + r\times p)(p\times q + q\times r + r\times p)^T] \\
= \; & (\bar{p}\times\bar{q} + \bar{q}\times\bar{r} + \bar{r}\times\bar{p})(\bar{p}\times\bar{q} + \bar{q}\times\bar{r} + \bar{r}\times\bar{p})^T \\
& + [\bar{p}-\bar{q}]_\times \Sigma_r [\bar{p}-\bar{q}]_\times^T \\
& + [\bar{q}-\bar{r}]_\times \Sigma_p [\bar{q}-\bar{r}]_\times^T \\
& + [\bar{r}-\bar{p}]_\times \Sigma_q [\bar{r}-\bar{p}]_\times^T \\
& + \mathrm{Ci}[\Sigma_p, \Sigma_q] + \mathrm{Ci}[\Sigma_q, \Sigma_r] + \mathrm{Ci}[\Sigma_r, \Sigma_p]
\end{aligned} \tag{11}$$

where $[\cdot]_\times$ is the cross product matrix. We call $\mathrm{Ci}[\cdot,\cdot]$ the cross-interference matrix. It is a symmetric 3×3 matrix and captures second-order uncertainty effects, defined as follows:

$$\begin{aligned}
\mathrm{Ci}[A,B]_{xx} = \; & A_{yy}B_{zz} - 2A_{yz}B_{yz} + A_{zz}B_{yy} \\
\mathrm{Ci}[A,B]_{xy} = \; & -A_{xy}B_{zz} + A_{xz}B_{yz} + A_{yz}B_{xz} - A_{zz}B_{xy} \\
\mathrm{Ci}[A,B]_{xz} = \; & A_{xy}B_{yz} - A_{xz}B_{yy} - A_{yy}B_{xz} + A_{yz}B_{xy} \\
\mathrm{Ci}[A,B]_{yy} = \; & A_{xx}B_{zz} - 2A_{xz}B_{xz} + A_{zz}B_{xx} \\
\mathrm{Ci}[A,B]_{yz} = \; & -A_{xx}B_{yz} + A_{xy}B_{xz} + A_{xz}B_{xy} - A_{yz}B_{xx} \\
\mathrm{Ci}[A,B]_{zz} = \; & A_{xx}B_{yy} - 2A_{xy}B_{xy} + A_{yy}B_{xx}
\end{aligned} \tag{12}$$

given symmetric matrices $A, B \in \mathbb{R}^{3\times 3}$. It has a simpler form if all Gaussians are isotropic with variance σ^2 (i.e. $A = B = \sigma^2 \cdot \mathbf{I}$):

$$\mathrm{Ci}[A,B] = 2\sigma^4 \cdot \mathbf{I} \tag{13}$$

For b, we can derive the following:

$$\begin{aligned}
\mathbb{E}[b] = \; & \mathbb{E}[(p\times q + q\times r + r\times p)\cdot|p,q,r|] \\
= \; & (\bar{p}\times\bar{q} + \bar{q}\times\bar{r} + \bar{r}\times\bar{p})\cdot|\bar{p},\bar{q},\bar{r}| \\
& + (\bar{q}-\bar{p})\times\Sigma_r(\bar{p}\times\bar{q}) \\
& + (\bar{r}-\bar{q})\times\Sigma_p(\bar{q}\times\bar{r}) \\
& + (\bar{p}-\bar{r})\times\Sigma_q(\bar{r}\times\bar{p}) \\
& + \mathrm{Ci}[\Sigma_p,\Sigma_q]\bar{r} + \mathrm{Ci}[\Sigma_q,\Sigma_r]\bar{p} + \mathrm{Ci}[\Sigma_r,\Sigma_p]\bar{q}
\end{aligned} \tag{14}$$

The value of c is again more complex and not needed for computing $x_{\min}$ but important for computing the value of the quadric, e.g. for greedy selection in incremental decimation:

$$\begin{aligned}
\mathbb{E}[c] = \; & \mathbb{E}[|p,q,r|^2] \\
= \; & |\bar{p},\bar{q},\bar{r}|^2 \\
& + (\bar{p}\times\bar{q})^T\Sigma_r(\bar{p}\times\bar{q}) \\
& + (\bar{q}\times\bar{r})^T\Sigma_p(\bar{q}\times\bar{r}) \\
& + (\bar{r}\times\bar{p})^T\Sigma_q(\bar{r}\times\bar{p}) \\
& + p^T\,\mathrm{Ci}[\Sigma_q,\Sigma_r]p \\
& + q^T\,\mathrm{Ci}[\Sigma_r,\Sigma_p]q \\
& + r^T\,\mathrm{Ci}[\Sigma_p,\Sigma_q]r \\
& + \mathrm{Tr}[\Sigma_r\,\mathrm{Ci}[\Sigma_p,\Sigma_q]]
\end{aligned} \tag{15}$$

This quadric is more sophisticated than the plane quadric and harder to interpret. However, there is at least one interesting pattern. $\mathbb{E}[A]$ and $\mathbb{E}[b]$ both consist of $\mathcal{O}(1)$, $\mathcal{O}(\Sigma)$, and $\mathcal{O}(\Sigma^2)$ terms, and $\mathbb{E}[c]$ additionally of an $\mathcal{O}(\Sigma^3)$ term:

- The $\mathcal{O}(1)$ term corresponds to the classical quadric formed by the means of the positions.
- The $\mathcal{O}(\Sigma)$ terms are linear in Σ_p, Σ_q, Σ_r and capture the interaction between the covariance of one vertex with the means of the other two vertices, i.e. the opposing edge.
- The $\mathcal{O}(\Sigma^2)$ terms contain the interaction between two covariance matrices (in the shape of the $\mathrm{Ci}[\cdot,\cdot]$ matrices).
- Finally, $\mathbb{E}[c]$ contains the $\mathcal{O}(\Sigma^3)$ term quantifying the combined uncertainty of all covariances.

As expected, all terms have an obvious symmetric structure. The only non-obvious part is $\mathrm{Tr}[\Sigma_r\,\mathrm{Ci}[\Sigma_p,\Sigma_q]]$ which can be shown to be commutative in all Σs.

3.3. Stability

A common problem with classical quadrics is that they are not invertible in planar regions. $A = nn^T$ has always rank one and thus a two-dimensional kernel. Trying to compute $x_{\min} = A^{-1}b$ leads to undefined results which is why using an SVD is recommended. This problem is not unique to planar regions but occurs in any rank-deficient quadric.

Our probabilistic quadrics do not suffer from this problem. For any positive definite covariance, the probabilistic plane quadric $A = nn^T + \Sigma$ is full rank and invertible (see Theorem A.11). Similarly, the probabilistic triangle quadrics are also positive definite and thus invertible given non-degenerate covariance matrices (see Theorems A.12 and A.13).

3.4. General Probabilistic Quadric

The derivations in Section 3.1 and 3.2 can easily be generalized to arbitrary configurations. Let $Q(s) \in \mathbb{R}^{4\times 4}$ be the individual quadric corresponding to object $s \in S$ (for example, s could be a plane and S the set of all planes). Given a probability distribution $p(s)$ over objects, we define the expected value formulation of QEF minimization:

$$\begin{aligned}
f(x) &= \mathbb{E}_{p(s)}[x^T Q(s) x] \\
&= \int_{s\in S} x^T Q(s)\, x \cdot p(s)\mathrm{d}s \\
&= x^T \left[\int_{s\in S} Q(s)p(s)\mathrm{d}s\right] x \\
&= x^T\,\mathbb{E}_{p(s)}[Q(s)]\,x \\
&= x^T Q_{p(s)}\, x
\end{aligned} \tag{16}$$

This provides the foundation of our method: The expected quadratic error is equal to the quadratic error of the expected quadric. We call $Q_{p(s)}$ the probabilistic quadric corresponding to the distribution $p(s)$. Since $Q_{p(s)}$ is just another 4×4 matrix (independent of x), we can apply our technique to any geometry processing task involving quadrics, handling data uncertainty in a more principled and robust manner.

For many distributions, $Q_{p(s)}$ can be computed analytically. The plane and triangle quadric only contain multi-variate second order polynomials when fully expanded. Thus, in theory, it is simple to plug in any distributions as long as second moments and covariances are known. The practical challenge arises from re-assembling the terms into an efficient form.

Even for involved probability distributions without closed-form solutions for their expected values, the fact that $Q_{p(s)}$ can be pre-computed means that numerical integration methods can be applied. In a sense, methods that average or sum classical quadrics over large regions are already computing a numerical approximation of the expected quadric.

One might wonder which distribution to use for a given problem. A good rule of thumb is that the distribution should roughly model how the data was obtained or what form of uncertainty can be expected. For example, points from a laser scan have a somewhat quantifiable positional uncertainty making the triangle quadric a good choice while it is not clear how this uncertainty translates to a Σ_n for a plane quadric. However, it is not critical to exactly match the input distribution as can be seen in Figure 2 where the Gaussian quadrics perform well, even for spiky non-Gaussian noise.

4. Experiments

4.1. Performance

The benchmarks were performed on a 3.60 GHz Intel Core i9-9900K (4.8 GHz single core turbo) using a single CPU thread. All algorithms were implemented in C++, compiled with Clang 7 using the flags `-O3`, `-march=native`, and `-ffast-math`.

Table 1 summarizes the central performance metrics. Constructing classical quadrics is pretty close to being limited by memory bandwidth on a modern CPU (processing more than 10 GB/s). The same can be observed for the probabilistic plane quadrics. Our probabilistic triangle quadric requires considerably more computation due to the various occurrences of the covariances. It is about $10\times$ slower than the classical version, $5\times$ when assuming isotropic noise. However, with about 20 million matrix operations per second on a single core, it is rarely the bottleneck.

Computing a singular value decomposition, even on 3×3 matrices, is not a cheap operation. Minimizing the probabilistic quadrics using $x_{\min} = A^{-1}b$ can be evaluated close to memory bandwidth at over 250 million operations per second. In contrast, we tested some highly optimized SVD algorithms [GJ*10, Jan14, MST*11] for classical quadrics with the best one reaching about 4.5 million operations per second, over $50\times$ slower than the matrix inversion. Especially for clean inputs with almost no noise, using the SVD is basically mandatory for classical quadrics as A^{-1} will run into numerical problems in rank-deficient (e.g. planar or edge) regions. For any positive covariance, our probabilistic quadrics have full rank and can always be minimized using $A^{-1}b$.

4.2. Properties

Figure 2 shows the results of various experiments performed on the different quadrics. In each scenario, fixed percentage of the vertices were removed via incremental decimation using edge collapses. We exemplarily depict 70% as a compromise to show a high level of decimation but not too high, which would limit the degrees of freedom too much. Given an edge with vertices v_1 and v_2, the new vertex position is chosen as $x_{\min}$ of the quadric $Q' = Q_{v_1} + Q_{v_2}$. Collapses are performed in ascending order of $x_{\min}^T Q' x_{\min}$, skipping those that would lead to flips. The classical quadrics were minimized with an SVD using the edge center as reference for the least-norm solution.

operation	throughput	cycles
constructing plane quadrics		
classical (A, b)	399 000 000 / s	12 / op
classical (A, b, c)	368 000 000 / s	13 / op
probabilistic (A, b)	273 000 000 / s	17 / op
probabilistic (A, b, c)	137 000 000 / s	35 / op
probabilistic (A, b, isotropic)	399 000 000 / s	12 / op
probabilistic (A, b, c, isotropic)	310 000 000 / s	15 / op
constructing triangle quadrics		
classical (A, b)	236 000 000 / s	20 / op
classical (A, b, c)	221 000 000 / s	21 / op
probabilistic (A, b)	22 000 000 / s	216 / op
probabilistic (A, b, c)	18 700 000 / s	257 / op
probabilistic (A, b, isotropic)	55 400 000 / s	86 / op
probabilistic (A, b, c, isotropic)	49 500 000 / s	96 / op
computing $x_{\min}$		
solving $A^{-1}b$	261 000 000 / s	17 / op
`Eigen::JacobiSVD` [GJ*10]	4 445 000 / s	1168 / op
`Eigen::BDCSVD` [GJ*10]	3 070 000 / s	1560 / op
`svd3` [Jan14, MST*11]	3 280 000 / s	1460 / op

Table 1: *Performance of different quadric-related operations on a single* 4.8 GHz *core. Measured metrics are throughput in operations per second (higher is better) and time taken in CPU cycles per operation (lower is better). The probabilistic plane and triangle quadrics are from Section 3.1 and Section 3.2 respectively. Isotropic means simplified formulas using* $\Sigma_{[\cdot]} = \sigma^2 \boldsymbol{I}$. *Computation was done in single precision. Note that classical quadrics often require SVDs while probabilistic ones can be solved via* $A^{-1}b$.

Point quadrics refer to $\|x - v\|^2$ for vertex position v. Their minimizer is the center of gravity of all summed up vertices. These quadrics are always stable and favor regular triangulations but obviously do not preserve sharp features.

The regularly triangulated cube with no noise demonstrates the feature preservation of plane-based quadrics. However, it also shows the instability of classical quadrics: every collapse has zero error leading to a highly irregular triangulation. In contrast, our probabilistic quadrics lead to quite regular triangulations while preserving features.

In the second experiment the cube has a small amount of vertex noise and a non-uniform triangulation. The vertex noise amplifies the uncertainty in the normal in regions of smaller triangles. Plane quadrics are not area-weighted and sensitive to the noise in the normal. Thus, they are more easily affected by irregular triangulation and tend to preserve regions of smaller triangles even if there is no geometric reason to do so. The area-weighted triangle quadrics lead to more regular triangulations.

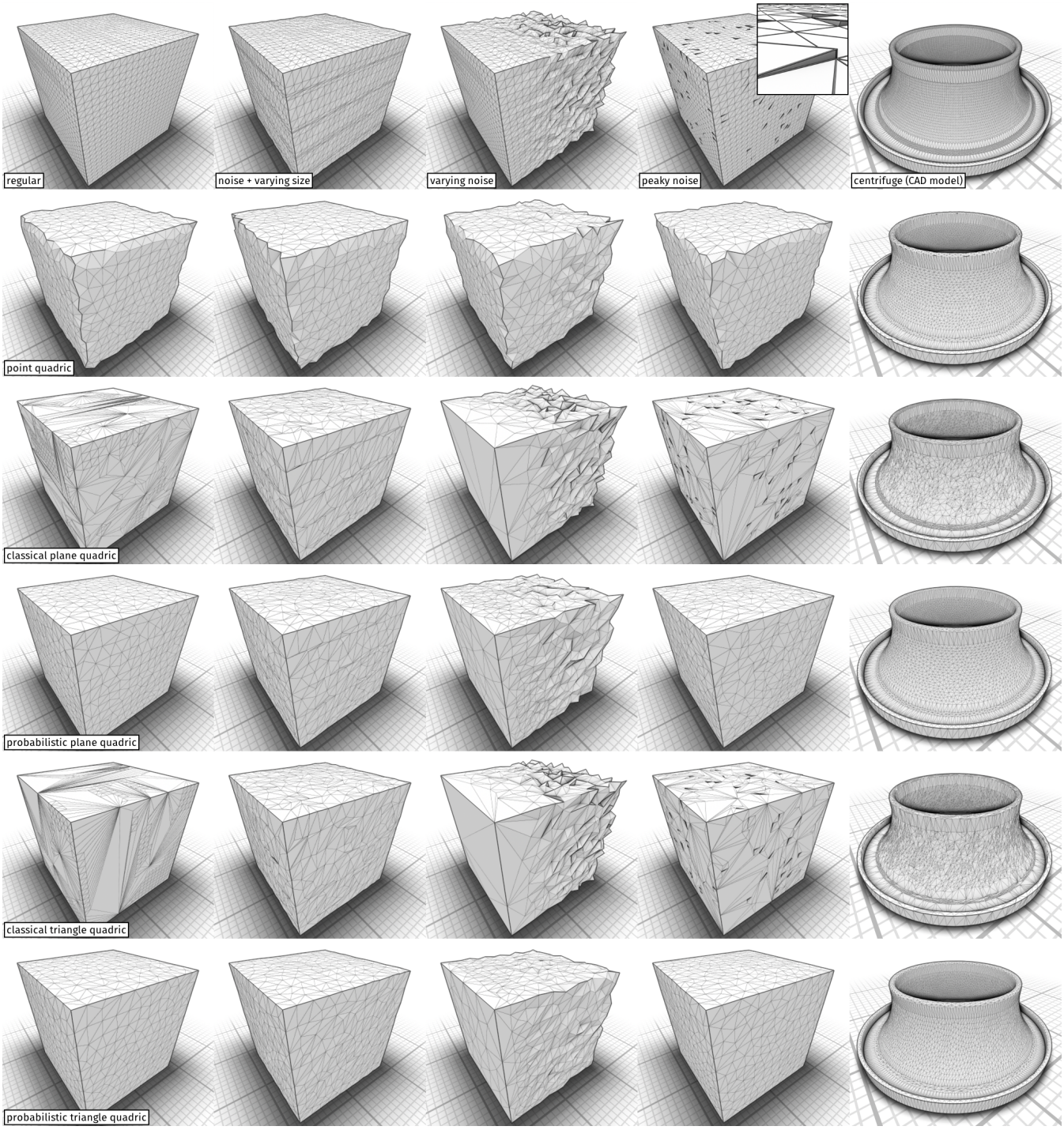

Figure 2: *Various experiments demonstrating the properties of our probabilistic quadrics in comparison to classical quadrics under different conditions. The top row shows the input mesh. The other rows show the results of removing* 70% *of the vertices via incremental decimation using different quadrics. Our probabilistic quadrics naturally preserve features, favor regular triangulations, and can adapt to noise.* Σ *were chosen to correspond to* 5% *of normal/edge length, except in the middle case where it went up to* 50% *on the noisy end. See Section 4.2 for a detailed discussion. The rendering uses an outline shader based on normal to improve readability. Best viewed in the digital version. Centrifuge model from [ZJ16].*

The third experiment has spatially varying noise levels. We vary Σ_n and Σ_p with the same intensity to show that our probabilistic quadrics can exploit prior information about the data uncertainty if available. Given the magnitude of the noise they are unable to fully smooth the mesh while preserving the features but still perform better than the classical quadrics. Furthermore, the probabilistic triangle quadrics outperform the plane quadrics as they model the noise source more accurately (noise in the vertices vs. noise in the normal).

In the fourth experiment a spiky noise was applied, similar to salt-and-pepper noise in images. It has a small magnitude but because the perturbed vertices were also moved towards a neighbor, the normal can change drastically. The classical quadrics are unable to remove these spikes as they consider them features, regardless of how small they are. Our probabilistic quadrics have no difficulties smoothing over this noise even though they were designed for Gaussian noise. In particular, the triangle quadric is very resilient against unreliable normals of small triangles.

The final experiment shows a CAD object with irregular triangulation, curved and flat regions, and some features. While both preserve features, the resulting triangulation caused by the classical quadrics is more irregular compared to the ones produced by our probabilistic quadrics.

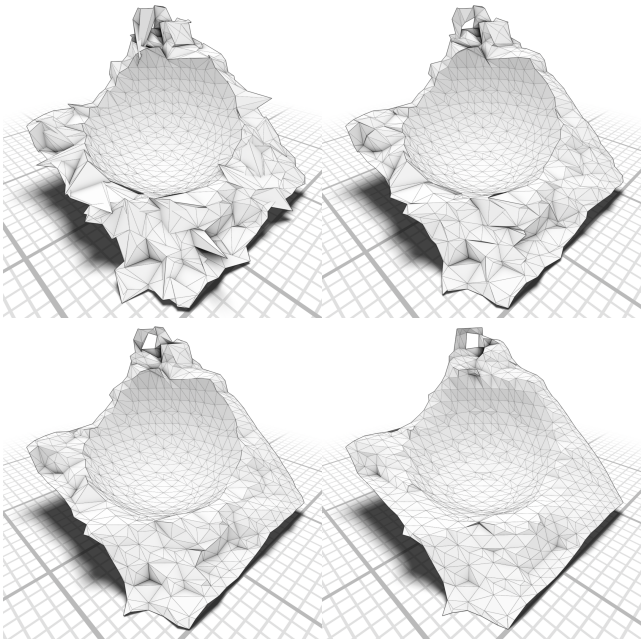

Figure 3: *Using probabilistic plane quadrics with* dual contouring *for isosurface extraction. The volume data was generated from an implicit function consisting of fractal simplex noise with a sphere cut out. The images show different values for σ_n (probabilistic plane quadric, isotropic noise in the normals). From top left to bottom right:* 0%, 10%, 25%, ∞. *With the right σ_n, probabilistic quadrics preserve features while not amplifying noise.*

5. Applications

Our probabilistic quadrics, although specially constructed, are still simple 4×4 matrices and thus can be used in any quadric-based application as a drop-in replacement for classical quadrics. In the previous section we already show their effect on incremental decimation. Vertex clustering benefits similarly except that the output topology is fixed by the method. In the following we showcase a few other applications.

5.1. Isosurface Extraction

Isosurface extraction is an important tool, especially in medical imaging. A sizeable amount of research has been devoted to using quadrics to preserve sharp features in the extraction of surfaces from volumetric input data [KBSS01, JLSW02, SW04]. Figure 3 shows the effect of probabilistic plane quadrics when used in *dual contouring*. Varying σ_n, the noise level assumed in the normals, allows fine control over the extracted surface: $\sigma_n = 0$ is equivalent to the classical quadric. While feature preserving it also amplifies noise in the data. At $\sigma_n \to \infty$, probabilistic quadrics degenerate to point quadrics. Normals are disregarded and the minimizer is the center of gravity of the points used to construct the plane quadrics. This produces overly smooth surfaces. In the intermediate cases, probabilistic quadrics achieve the best of the two worlds: Clear features are preserved but the uncertainty in the normals does not lead to overly amplified (spiky) noise.

5.2. Adaptive Mesh Smoothing

In Figure 4 we demonstrate mesh smoothing and denoising in the spirit of [VNMC10] and [LTB19]. We initialize each face with its probabilistic triangle quadric. These quadrics are then smoothed over multiple iterations by averaging them with their neighboring faces. Finally, for each vertex we compute the sum of adjacent face quadrics and move the vertex into the minimizer of the quadric.

In the example we chose a mesh with spatially strongly varying noise. Classical quadrics are unable to remove noise above a certain level and start to treat the peaks as features to be preserved. Using a global Σ is not granular enough to account for the spatially varying nature of the noise: the value must be high enough to smooth over the noise but that negatively affects features at the same time.

If the amount of noise is known a priori (even if only approximately), then this can be taken into account to construct probabilistic quadrics with a spatially varying Σ. Even without this information we can make an "educated guess". In this example we chose a simple heuristic: the per-vertex absolute angle defect. Lower angle defect indicates less noise. While not perfect, this already lets us smooth over the noisy parts while still reasonably preserving features.

5.3. Subdivision and Interpolation

Inspired by Gaussian-product Subdivision Surfaces [PBW19], we use quadrics to define an experimental non-linear surface interpolation scheme, shown in Figure 5. Instead of interpolating surface positions or applying a subdivision stencil on vertex positions, we

Figure 4: *Quadric-based mesh smoothing. The input mesh shows strongly spatially varying noise. Classical quadrics (top right) preserve features and smooth over some noise but are unable to suppress stronger noise completely. With probabilistic triangle quadrics and a high value for Σ (bottom right), the noise can be almost eliminated but smaller features are smoothed over. Using heuristics (such as the angle defect) to estimate the local noise level, adaptively changing Σ smooths over the noise while preserving features (bottom left). Model from [ZJ16].*

interpolate error quadrics. More concretely, we annotate each vertex with a probabilistic plane quadric with normal n and covariance Σ_n. We then interpolate the quadrics and compute new vertex positions as the minimizer of the interpolated quadrics. This scheme permits a spectrum of surface properties, from smooth with low curvature (high Σ) to sharp features (low Σ). Anisotropic covariances can be used to apply these properties direction-dependently. This approach leads to an equivalent formulation as [PBW19] and can be seen as an alternative interpretation of their method.

6. Future Work

We presented the derivation of the probabilistic plane and triangle quadric assuming independent, potentially anisotropic Gaussian noise for each input parameter. An obvious future avenue of research is to investigate various other probabilistic quadrics and their feasibility. This includes different types of noise such as uniform, chi-square, gamma, or pareto distributed inputs but also cases where the linear independence assumption does not hold.

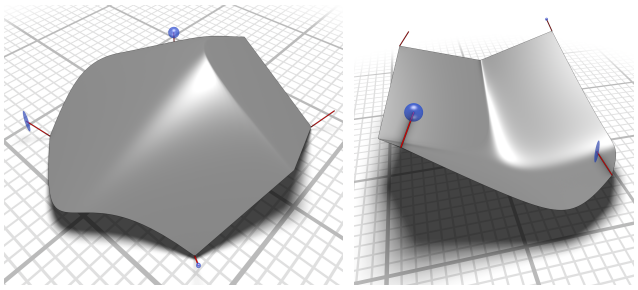

Figure 5: *Non-linear surface interpolation scheme using quadrics. Vertices are annotated with probabilistic plane quadrics (normals shown in red, covariance matrices as blue ellipsoids). Instead of surface positions, the quadrics are bilinearly interpolated. The interpolated position is then the minimizer of the interpolated quadric.*

Even when analytical solutions are intractable, we mentioned that probabilistic quadrics can still be computed via numerical integration. In a sense, averaging a large number of classical quadrics is already a numerical integration, although a spatial one. If the input distribution can be simulated it is also possible to construct local quadrics via Monte Carlo integration.

In Section 5.2 we used the angle defect as a simple measure for local noise. This method can be refined by investigating more sophisticated measures that could for example quantify the anisotropy of the noise. Our subdivision and interpolation scheme of Section 5.3 is a simple proof-of-concept and deserves a proper investigation. One particular issue that arose was that weighting interpolated quadrics behaves non-linearly. Finding a mapping that makes it more linear would help for intuitive control.

7. Conclusion

Classical quadrics treat the input geometry as ground truth. Any noise and uncertainty present in the data can only be incorporated in an ad hoc fashion by averaging the quadrics over large regions. In contrast, our *probabilistic quadrics* embrace the uncertain nature of most inputs and model the expected quadratic error over the input distribution. We show how to efficiently construct probabilistic versions for two popular quadrics: the plane quadric built by position and normal under Gaussian noise and the triangle quadric built from three positions, each under Gaussian noise. This probabilistic framework lets us deal with uncertainty in a granular and principled manner.

A welcome side effect is that all our quadrics are full rank by construction (for non-zero covariance). This means that finding the minimizer of a quadric can be done robustly by solving a simple 3×3 linear system instead of a singular value decomposition which which is faster by a factor of 50 in our benchmarks.

Our experiments also show that the probabilistic quadrics naturally favor regular triangulations while still being feature preserving. Their ability to incorporate local information about the shape of the noise enables us to deal with spatially varying noise.

Because our probabilistic quadrics are still quadratic functions in form of a symmetric positive definite 4×4 matrix, they can be used as a drop-in replacement in any application using classical quadrics. We demonstrate their versatility by showing how they improve decimation, isosurface extraction, and smoothing. More exploratory, we construct a non-linear subdivision and interpolation scheme based on quadric minimization.

References

[CB17] CALDERON S., BOUBEKEUR T.: Bounding proxies for shape approximation. *ACM Transactions on Graphics (Proc. SIGGRAPH 2017) 36*, 5 (july 2017). 3

[GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1997), SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., pp. 209–216. 2

[GJ*10] GUENNEBAUD G., JACOB B., ET AL.: Eigen v3. http://eigen.tuxfamily.org, 2010. 5

[GZ05] GARLAND M., ZHOU Y.: Quadric-based simplification in any dimension. *ACM Trans. Graph. 24*, 2 (Apr. 2005), 209–239. 2

[Hop99] HOPPE H.: New quadric metric for simplifying meshes with appearance attributes. In *Proceedings of the 10th IEEE Visualization 1999 Conference (VIS '99)* (Washington, DC, USA, 1999), VISUALIZATION '99, IEEE Computer Society, pp. –. 2

[Jan14] JANG E. V.: Fast 3x3 svd. https://github.com/ericjang/svd3, 2014. 5

[JLSW02] JU T., LOSASSO F., SCHAEFER S., WARREN J.: Dual contouring of hermite data. *ACM Trans. Graph. 21*, 3 (July 2002), 339–346. 2, 3, 7

[KBSS01] KOBBELT L. P., BOTSCH M., SCHWANECKE U., SEIDEL H.-P.: Feature sensitive surface extraction from volume data. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2001), SIGGRAPH '01, ACM, pp. 57–66. 2, 3, 7

[Lin00] LINDSTROM P.: Out-of-core simplification of large polygonal models. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2000), SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., pp. 259–262. 2, 3

[LT98] LINDSTROM P., TURK G.: Fast and memory efficient polygonal simplification. In *Proceedings Visualization '98 (Cat. No.98CB36276)* (Oct 1998), pp. 279–286. 2, 3

[LTB19] LEGRAND H., THIERY J.-M., BOUBEKEUR T.: Filtered quadrics for high-speed geometry smoothing and clustering. *Computer Graphics Forum 38*, 1 (2019), 663–677. 2, 3, 7

[LZ08] LI Y., ZHU Q.: A new mesh simplification algorithm based on quadric error metrics. In *2008 International Conference on Advanced Computer Theory and Engineering* (Dec 2008), pp. 528–532. 2

[MP92] MATHAI A., PROVOST S.: *Quadratic Forms in Random Variables: Theory and Applications*, vol. 87. 12 1992. 3, 9

[MST*11] MCADAMS A., SELLE A., TAMSTORF R., TERAN J., SIFAKIS E.: Computing the singular value decomposition of 3x3 matrices with minimal branching and elementary floating point operations. 5

[PBW19] PREINER R., BOUBEKEUR T., WIMMER M.: Gaussian-product subdivision surfaces. *ACM Trans. Graph. 38*, 4 (July 2019), 35:1–35:11. 7, 8

[SLA15] SALINAS D., LAFARGE F., ALLIEZ P.: Structure-Aware Mesh Decimation. *Computer Graphics Forum* (Jan. 2015), 20. 2

[SW04] SCHAEFER S., WARREN J.: Dual marching cubes: primal contouring of dual grids. In *12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings.* (Oct 2004), pp. 70–76. 2, 3, 7

[TGB13] THIERY J.-M., GUY E., BOUBEKEUR T.: Sphere-meshes: Shape approximation using spherical quadric error metrics. *ACM Transaction on Graphics (Proc. SIGGRAPH Asia 2013) 32*, 6 (2013), Art. No. 178. 3

[TGBE16] THIERY J.-M., GUY E., BOUBEKEUR T., EISEMANN E.: Animated mesh approximation with sphere-meshes. *ACM Trans. Graph. 35*, 3 (May 2016), 30:1–30:13. 3

[VNMC10] VIEIRA A. W., NETO A. A., MACHARET D. G., CAMPOS M. F. M.: Mesh denoising using quadric error metric. In *2010 23rd SIBGRAPI Conference on Graphics, Patterns and Images* (Aug 2010), pp. 247–254. 2, 3, 7

[YWLY12] YAN D.-M., WANG W., LIU Y., YANG Z.: Variational mesh segmentation via quadric surface fitting. *Comput. Aided Des. 44*, 11 (Nov. 2012), 1072–1082. 3

[ZJ16] ZHOU Q., JACOBSON A.: Thingi10k: A dataset of 10,000 3d-printing models. *arXiv preprint arXiv:1605.04797* (2016). 1, 6, 8

Appendix A: Derivation of Probabilistic Quadric for Triangles under Gaussian Noise

All derivations were checked with a custom program that performs vector arithmetic symbolically using polynomials in canonical form (in which equality is decidable and reasonably efficient).

Definition A.1 (Cross product matrix) $[\cdot]_\times$ denotes the skew-symmetric matrix corresponding to the cross product, i.e.

$$a \times b = [a]_\times b = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{17}$$

Due to its skew-symmetry, $[a]_\times = -[a]_\times^T$.

Definition A.2 (Cross-interference matrix) $\mathrm{Ci}[,] : S^3_+ \times S^3_+ \to S^3_+$ is a symmetric matrix capturing second-order terms in the probabilistic quadrics:

$$\begin{aligned}
\mathrm{Ci}[A,B]_{xx} &= A_{yy}B_{zz} - 2A_{yz}B_{yz} + A_{zz}B_{yy} \\
\mathrm{Ci}[A,B]_{xy} &= -A_{xy}B_{zz} + A_{xz}B_{yz} + A_{yz}B_{xz} - A_{zz}B_{xy} \\
\mathrm{Ci}[A,B]_{xz} &= A_{xy}B_{yz} - A_{xz}B_{yy} - A_{yy}B_{xz} + A_{yz}B_{xy} \\
\mathrm{Ci}[A,B]_{yy} &= A_{xx}B_{zz} - 2A_{xz}B_{xz} + A_{zz}B_{xx} \\
\mathrm{Ci}[A,B]_{yz} &= -A_{xx}B_{yz} + A_{xy}B_{xz} + A_{xz}B_{xy} - A_{yz}B_{xx} \\
\mathrm{Ci}[A,B]_{zz} &= A_{xx}B_{yy} - 2A_{xy}B_{xy} + A_{yy}B_{xx}
\end{aligned} \tag{18}$$

Lemma A.1 Given $x \sim N(\bar{x}, \Sigma_x)$, it holds that

$$\mathbb{E}[xx^T] = \bar{x}\bar{x}^T + \Sigma_x, \tag{19}$$

because $\mathbb{E}[x_i x_j] = \bar{x}_i\bar{x}_j + \Sigma_{x,ij}$ (second moment of Gaussian).

Lemma A.2 Given $M \in \mathbb{R}^{3\times 3}$ and $v \in \mathbb{R}^3$, it holds that

$$\mathrm{Tr}[Mvv^T] = v^T M v \tag{20}$$

because $\mathrm{Tr}[ba^T] = a^T b$ for all $a, b \in \mathbb{R}^3$.

Lemma A.3 Given $x \sim N(\bar{x}, \Sigma_x)$ and a symmetric matrix A, [MP92] show that

$$\mathbb{E}[x^T A x] = \bar{x}^T A \bar{x} + \mathrm{Tr}[A\Sigma_x] \tag{21}$$

Lemma A.4 Given $a \sim N(\bar{a}, \Sigma_a)$ and $b \sim N(\bar{b}, \Sigma_b)$, it holds that

$$\mathbb{E}_a[[a]_\times \Sigma_b [a]_\times^T] = [\bar{a}]_\times \Sigma_b [\bar{a}]_\times^T + \mathrm{Ci}[\Sigma_a, \Sigma_b]. \tag{22}$$

This follows from using $\mathbb{E}_a[a_i a_j] = a_i a_j + \Sigma_{a,ij}$ and expanding all terms. Everything without Σ_a can be reassembled into $[\bar{a}]_\times \Sigma_b [\bar{a}]_\times^T$ and the rest forms $\mathrm{Ci}[\Sigma_a, \Sigma_b]$.

Lemma A.5 Given $x \in \mathbb{R}^3$ and $A, B \in S_+^3$, it holds that

$$\mathrm{Tr}[A \cdot [x]_\times \cdot B \cdot [x]_\times^T] = x^T \,\mathrm{Ci}[A, B]\, x. \tag{23}$$

We simply used our symbolic computation tool to verify equality of both sides.

Lemma A.6 Given $a \sim N(\bar{a}, \Sigma_a)$ and $b \sim N(\bar{b}, \Sigma_b)$, it holds that

$$\begin{aligned}\mathbb{E}_{a,b}[(a \times b)(a \times b)^T] = {} & (\bar{a} \times \bar{b})(\bar{a} \times \bar{b})^T + [\bar{b}]_\times \Sigma_a [\bar{b}]_\times^T \\ & + [\bar{a}]_\times \Sigma_b [\bar{a}]_\times^T + \mathrm{Ci}[\Sigma_a, \Sigma_b]\end{aligned} \tag{24}$$

Proof Using Lemma A.1 and Lemma A.4:

$$\begin{aligned}\mathbb{E}_{a,b}[(a \times b)(a \times b)^T] &= \mathbb{E}_{a,b}[[a]_\times b b^T [a]_\times^T] \\ &= \mathbb{E}_a[[a]_\times \mathbb{E}_b[b b^T][a]_\times^T] \\ &= \mathbb{E}_a[[a]_\times (\bar{b}\bar{b}^T + \Sigma_b)[a]_\times^T] \\ &= \mathbb{E}_a[[a]_\times \bar{b}\bar{b}^T [a]_\times^T] + \mathbb{E}_a[[a]_\times \Sigma_b [a]_\times^T] \\ &= \mathbb{E}_a[[b]_\times \bar{a}\bar{a}^T [b]_\times^T] + [\bar{a}]_\times \Sigma_b [\bar{a}]_\times^T + \mathrm{Ci}[\Sigma_a, \Sigma_b] \\ &= (\bar{a} \times \bar{b})(\bar{a} \times \bar{b})^T + [\bar{b}]_\times \Sigma_a [\bar{b}]_\times^T \\ &\quad + [\bar{a}]_\times \Sigma_b [\bar{a}]_\times^T + \mathrm{Ci}[\Sigma_a, \Sigma_b]\end{aligned} \tag{25}$$

□

Lemma A.7 Given $a \sim N(\bar{a}, \Sigma_a)$, $b \sim N(\bar{b}, \Sigma_b)$, and $c \sim N(\bar{c}, \Sigma_c)$, it holds that

$$\mathbb{E}_{a,b,c}[(a \times b)(b \times c)^T] = (\bar{a} \times \bar{b})(\bar{b} \times \bar{c})^T + [\bar{a}]_\times \Sigma_b [\bar{c}]_\times^T \tag{26}$$

Proof Using Lemma A.1:

$$\begin{aligned}\mathbb{E}_{a,b,c}[(a \times b)(b \times c)^T] &= \mathbb{E}_{a,b,c}[[a]_\times b b^T [c]_\times^T] \\ &= \mathbb{E}_{a,c}[[a]_\times (\bar{b}\bar{b}^T + \Sigma_b)[c]_\times^T] \\ &= (\bar{a} \times \bar{b})(\bar{b} \times \bar{c})^T + [\bar{a}]_\times \Sigma_b [\bar{c}]_\times^T\end{aligned} \tag{27}$$

□

Theorem A.8 Given $p \sim N(\bar{p}, \Sigma_p)$, $q \sim N(\bar{q}, \Sigma_q)$, and $r \sim N(\bar{r}, \Sigma_r)$, it holds that

$$\begin{aligned}\mathbb{E}[A] = {} & \mathbb{E}[(p \times q + q \times r + r \times p)(p \times q + q \times r + r \times p)^T] \\ = {} & (\bar{p} \times \bar{q} + \bar{q} \times \bar{r} + \bar{r} \times \bar{p})(\bar{p} \times \bar{q} + \bar{q} \times \bar{r} + \bar{r} \times \bar{p})^T \\ & + [\bar{p} - \bar{q}]_\times \Sigma_r [\bar{p} - \bar{q}]_\times^T \\ & + [\bar{q} - \bar{r}]_\times \Sigma_p [\bar{q} - \bar{r}]_\times^T \\ & + [\bar{r} - \bar{p}]_\times \Sigma_q [\bar{r} - \bar{p}]_\times^T \\ & + \mathrm{Ci}[\Sigma_p, \Sigma_q] + \mathrm{Ci}[\Sigma_q, \Sigma_r] + \mathrm{Ci}[\Sigma_r, \Sigma_p]\end{aligned} \tag{28}$$

Proof Expanding the outer product yields three uniform terms of form $(a \times b)(a \times b)^T$ and six mixed terms of form $(a \times b)(b \times c)^T$. The claim results from applying Lemma A.6 and Lemma A.7 together with the simplification $[a]_\times \Sigma_c [a]_\times^T - [a]_\times \Sigma_c [b]_\times^T - [b]_\times \Sigma_c [a]_\times^T + [b]_\times \Sigma_c [b]_\times^T = [a-b]_\times \Sigma_c [a-b]_\times^T$. □

Theorem A.9 Given $p \sim N(\bar{p}, \Sigma_p)$, $q \sim N(\bar{q}, \Sigma_q)$, and $r \sim N(\bar{r}, \Sigma_r)$, it holds that

$$\begin{aligned}\mathbb{E}[b] = {} & \mathbb{E}[(p \times q + q \times r + r \times p) \cdot |p, q, r|] \\ = {} & (\bar{p} \times \bar{q} + \bar{q} \times \bar{r} + \bar{r} \times \bar{p}) \cdot |\bar{p}, \bar{q}, \bar{r}| \\ & + (\bar{q} - \bar{p}) \times \Sigma_r (\bar{p} \times \bar{q}) \\ & + (\bar{r} - \bar{q}) \times \Sigma_p (\bar{q} \times \bar{r}) \\ & + (\bar{p} - \bar{r}) \times \Sigma_q (\bar{r} \times \bar{p}) \\ & + \mathrm{Ci}[\Sigma_p, \Sigma_q]\bar{r} + \mathrm{Ci}[\Sigma_q, \Sigma_r]\bar{p} + \mathrm{Ci}[\Sigma_r, \Sigma_p]\bar{q}\end{aligned} \tag{29}$$

Proof The determinant can be written as a triple product $|p, q, r| = (p \times q)^T r = (q \times r)^T p = (r \times p)^T q$. Expanding the product and choosing a suitable triple product yields three terms of form $(a \times b)(a \times b)^T c$. The claim results from applying Lemma A.6 and re-assembling some cross products. □

Theorem A.10 Given $p \sim N(\bar{p}, \Sigma_p)$, $q \sim N(\bar{q}, \Sigma_q)$, and $r \sim N(\bar{r}, \Sigma_r)$, it holds that

$$\begin{aligned}\mathbb{E}[c] = {} & \mathbb{E}[|p, q, r|^2] \\ = {} & |\bar{p}, \bar{q}, \bar{r}|^2 \\ & + (\bar{p} \times \bar{q})^T \Sigma_r (\bar{p} \times \bar{q}) + (\bar{q} \times \bar{r})^T \Sigma_p (\bar{q} \times \bar{r}) + (\bar{r} \times \bar{p})^T \Sigma_q (\bar{r} \times \bar{p}) \\ & + p^T \,\mathrm{Ci}[\Sigma_q, \Sigma_r] p + q^T \,\mathrm{Ci}[\Sigma_r, \Sigma_p] q + r^T \,\mathrm{Ci}[\Sigma_p, \Sigma_q] r \\ & + \mathrm{Tr}[\Sigma_r \,\mathrm{Ci}[\Sigma_p, \Sigma_q]]\end{aligned} \tag{30}$$

Proof

$$\begin{aligned}\mathbb{E}_{p,q,r}[|p, q, r|^2] &= \mathbb{E}_{p,q,r}[r^T (p \times q)(p \times q)^T r] \\ &= \mathbb{E}_r[r^T \mathbb{E}_{p,q}[(p \times q)(p \times q)^T] r]\end{aligned} \tag{31}$$

Set $M = \mathbb{E}_{p,q}[(p \times q)(p \times q)^T]$ and apply Lemma A.3:

$$\begin{aligned}\mathbb{E}_r[r^T \mathbb{E}_{p,q}[(p \times q)(p \times q)^T] r] &= \mathbb{E}_r[r^T M r] \\ &= \bar{r}^T M \bar{r} + \mathrm{Tr}[M \Sigma_r]\end{aligned} \tag{32}$$

The claim now results from expanding M (Lemma A.6), using the linearity of the trace, and applying Lemma A.2 and A.5. □

Theorem A.11 For any positive definite matrix $\Sigma \in S_+^3$ and any vector $n \in \mathbb{R}^3$ the matrix $A = nn^T + \Sigma$ is invertible.

Proof nn^T is a positive semidefinite matrix because $v^T nn^T v = (v^T n)^2 \geq 0$ for all $v \in \mathbb{R}$. The sum of a positive semidefinite and a positive definite matrix is positive definite. Any positive definite matrix is invertible. Thus $nn^T + \Sigma$ is invertible. □

Theorem A.12 For any positive definite matrix $\Sigma \in S_+^3$ and any vector $n \in \mathbb{R}^3$ the matrix $A = [n]_\times \Sigma [n]_\times^T$ is positive definite and thus invertible.

Proof Let v be a vector in $\mathbb{R}^3 \setminus 0$. Then $v^T A v = v^T [n]_\times \Sigma [n]_\times^T v = (v \times n)^T \Sigma (v \times n) > 0$ due to Σ being positive definite. □

Theorem A.13 For any two positive definite matrix $A, B \in S_+^3$ the matrix $\mathrm{Ci}[A, B]$ is also positive definite.

Proof Let v be a vector in $\mathbb{R}^3 \setminus 0$. Then $v^T \,\mathrm{Ci}[A, B] v = \mathrm{Tr}[A \cdot [v]_\times \cdot B \cdot [v]_\times^T]$ via Lemma A.5. From Theorem A.12 we know that $M := [v]_\times \cdot B \cdot [v]_\times^T$ is positive definite. The square root of a positive definite matrix is also positive definite. Thus $\mathrm{Tr}[AM] = \mathrm{Tr}[A\sqrt{M}\sqrt{M}] = \mathrm{Tr}[\sqrt{M} A \sqrt{M}] > 0$, because $\sqrt{M} A \sqrt{M}$ is positive definite and the trace of a positive definite matrix is positive. □