Computational Visual Media
DOI 10.1007/s41095-xxx-xxxx-x Vol. x, No. x, month year, xx–xx

Article Type (Research/Review)

Fitting quadrics with a Bayesian prior

Daniel Beale [1] ✉, Yong-Liang Yang [1], Neill Campbell[1], Darren Cosker[1], Peter Hall[1]

© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract Quadrics are a compact mathematical formulation for a range of primitive surfaces. A problem arises when there are not enough data-points to compute the model but knowledge of the shape is available. This paper presents a method for fitting a quadric with a Bayesian prior. We use a matrix normal prior in order to favour ellipsoids on ambiguous data. The results show the algorithm to cope well when there are few points in the point cloud, competing with contemporary techniques in the area.

Keywords Geometry, statistics, graphics, computer vision.

1 Introduction

Surface fitting is one of the most important research areas in computer graphics and geometric modelling. It studies how to approximate unorganized geometric data using regular surfaces with explicit algebraic forms, which benefits a lot of important graphics applications, including shape approximation, surface reconstruction, local geometric feature analysis, etc. In the fitting process, the choice of the target surface form usually depends on the application, where algebraic surfaces with different orders, or freeform surfaces, such as B-Splines, would be required.

Among all types of surfaces; quadrics and hyper-surfaces represented by a quadratic polynomial in the embedding space, may be the most popular form in surface fitting for several reasons. First, they can represent a variety of common primitive surfaces, such as planes, spheres, cylinders, etc. Second, a lot of real world shapes, ranging from industrial products to architectures can be represented by a union of quadrics. Third, it is the form with the least order to estimate second order differential properties, such as curvatures. Fourth, the simple quadratic form allows for efficient numerical computations which are usually in closed-form.

In this work, we present a novel probabilistic quadric fitting method. Alternative to previous algorithms, we assume an additive Gaussian noise model on the algebraic distance. A Bayesian prior is placed on the parameters, allowing certain shapes to be favoured when not much data is available. We test our quadric fitting method on various datasets, both synthetic and from the real world. The experiments and comparisons show that our method is not only efficient, but also robust to contaminated data.

2 Related Work

There are a large number of examples of fitting both implicit and parametric surfaces to unordered point clouds. We focus the review on existing methods in quadric fitting and then go on to show how they can be used in a variety of applications in Computer Graphics.

Quadric fitting. The popularity of quadrics has led to many research papers on fitting them to unordered data points. Methods in this area can be broken in to two main areas, either minimising the *algebraic* or *geometric* distance of each point to the surface. The former is simpler, since it is the value computed by evaluating the implicit equation itself (defined in Equation 1). Geometric distances are formulated in terms of Euclidean distances between the points and the surface, methods in this area produce a fit which favours each point equally, although methods tend to be iterative due to the complex relationship between the implicit equation and the solution set.

One of the most popular methods to use a geometric distance is due to Taubin [25], who approximates the true distance in terms of the normalised Euclidean distance to the quadric surface. Knowledge of the explicit parametric equations of the surface can be useful for more compact least squares parameter estimates [1–3, 8, 15, 18, 22]. Fast

1 University of Bath, Claverton Down, Bath, BA2 7AY. E-mail: d.beale@bath.ac.uk, y.yang2@bath.ac.uk, n.campbell@bath.ac.uk, d.p.cosker@bath.ac.uk, p.m.hall@bath.ac.uk

Manuscript received: 2014-12-31; accepted: 2015-01-30.

estimation of the closest point the surface to a general point in space is described in [21], who also make use of RANSAC to add robustness to outliers.

Finding a least squares estimate based on the algebraic distance yield fast and numerically stable solutions which can be solved using matrix algebra. In this case, literature in the area seeks to constrain the surface to lie on a particular surface shape, such as an ellipsoid e.g. [7, 9, 12, 20]. The work of Li et al. [12] shows how to fit ellipsoids and Dai et al. [5] present a similar method for fitting to paraboloids.

Bayesian probabilistic methods exist for fitting general quadrics, for example, Subrahmonia et al. [24] show how to fit by casting a probability distribution over the geometric Taubin errors. Their use of prior is to avoid overfitting or parameters which produce a minimal error but undesirable results.

Quadric fitting applications. Quadric fitting has been applied to many important graphics applications. In the context of shape approximation, Cohen-Steiner et al. [4] presented a variational shape approximation (VSA) algorithm, where a set of planar proxies are iteratively fitted to the input mesh based on Lloyd's clustering on mesh faces, resulting in a simplified mesh representation.

To better approximate curved parts and sharp features, Wu and Kobbelt [27] extended the previous approach by using proxies other than planes, where various types of quadrics, such as spheres or cylinders, are allowed. Another extension of the VSA algorithm was presented in [28] by using general quadric proxies. For surface reconstruction, quadric fitting has been used to better represent the underlying geometry and improve the reconstruction quality. Schnabel et al. [23] presented a RANSAC-based approach to fit different types of quadrics to noisy data. Li et al. [13] further improved the reconstruction quality by optimizing the global regularities, such as symmetries, of the fitted surface arrangements. Further from joining the geometric primitives, it is also possible to create an implicit surface using a Multi-level partition [16]. This kind of approach can help to smooth the resulting surface in regions which have not been accurately modelled.

In shape analysis, local geometric features (e.g., curvatures) which are invariant under certain transformations benefit a lot of from quadric fitting. A common approach is to fit quadrics to a local neighbourhood so that differential properties can be estimated with the help of the fitted surface [10].

The use of computer vision for robotic application allows the use geometry fitting algorithms for estimating object properties for physical interaction, such as the curvature of an object [26]. Segmenting objects with primitive surfaces is also possible [19], allowing a robot to discriminate between surfaces before planning.

Our method assumes a probability distribution over the algebraic errors, and results in a compact matrix computation for the maximum a-priori estimate of the parameters given the data points. The use of a prior allows us to choose specific primitives from the collection of quadric surfaces.

3 Bayesian Quadrics

We begin by defining what a quadric is, and then extend it with the use of a noise model. We define a conditional distribution which provides the likelihood of the data given the parameters, from which we infer the probability of the parameters with the use of a suitable prior.

Quadrics are polynomial surfaces of degree two, which are a variety given by the implicit equation,

$$\boldsymbol{z}^T A \boldsymbol{z} + \boldsymbol{b}^T \boldsymbol{z} + c = 0 \tag{1}$$

with A a $M \times M$ matrix $\boldsymbol{b}$ an vector, and c a scalar, for values of $\boldsymbol{z} \in \mathbb{R}^M$.

Before defining more notation we note that A can be assumed symmetric, without loss of generality, the details of this are explained in Appendix A. Positive definiteness is determined from the eigenvalues. Letting $\boldsymbol{\psi}$ denote the each element of A, $\boldsymbol{b}$ and c. Further, let $\boldsymbol{\theta}_A$ denote the values in $\boldsymbol{\psi}$ that correspond to the values in A, i.e.,

$$\boldsymbol{\theta}_A = [A_{11}, \ldots, A_{MM}, A_{12}, \ldots, A_{(M-1)M}]^T \tag{2}$$

and then,

$$\boldsymbol{\psi} = \begin{bmatrix} \boldsymbol{\theta}_A \\ \boldsymbol{b} \\ c \end{bmatrix} \tag{3}$$

The data from the input point cloud is collected in to a data matrix $X = [\boldsymbol{x}_0, \ldots, \boldsymbol{x}_\ell, \ldots, \boldsymbol{x}_N]^T$, so that,

$$\boldsymbol{x} = (z_1^2, \ldots, z_M^2, 2z_1z_2, \ldots, z_1, \ldots, z_M, 1)^T \tag{4}$$

where $\boldsymbol{x}$ is an arbitrary column of X, and,

$$\boldsymbol{x} \cdot \boldsymbol{\psi} = \boldsymbol{z}^T A \boldsymbol{z} + \boldsymbol{b}^T \boldsymbol{z} + c \tag{5}$$

In order to use Bayesian statistics, an additive noise model is assumed on each of the data-points i.e. we assume that

$$X\boldsymbol{\psi} = \boldsymbol{\varepsilon} \tag{6}$$

where $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, I)$ is a single draw from a unit multivariate Gaussian distribution.

In order to solve the matrix equation $X\boldsymbol{\psi} = \mathbf{0}$, without a prior on $\boldsymbol{\psi}$ one can find the eigenvector corresponding to the smallest eigenvalue of X^TX. If we consider the following eigenvalue problem,

$$X^TX\boldsymbol{v} = s\boldsymbol{v} \tag{7}$$

it is sufficient to find an eigenvector $\boldsymbol{v}$ with associated eigenvalue s, with $s = 0$. Existence of such a solution

is guaranteed if the matrix X^TX is singular. In practise, however, we take the smallest eigenvalue.

The solution to Equation 7 is a least squares solution of the equation $||X\boldsymbol{\psi}||^2 = \mathbf{0}$ since the value $||X\boldsymbol{\psi}||^2$ is minimal when all of the partial derivatives with respect to $\boldsymbol{\psi}$ are zero. Considering,

$$\frac{\partial}{\partial\boldsymbol{\psi}}(||X\boldsymbol{\psi}||^2) = \frac{\partial}{\partial\boldsymbol{\psi}}(\boldsymbol{\psi}^TX^TX\boldsymbol{\psi}) \tag{8}$$

$$= 2X^TX\boldsymbol{\psi} \tag{9}$$

Setting Equation 9 equal to zero and dividing out the constant, one can see that a minimal solution using eigenvalue decomposition is a least-squares estimate.

The least squares estimate introduced above places no constraint on the matrix A, and therefore can produce a matrix with negative eigenvalues. If one wishes to ensure that the matrix is an Ellipsoid, for example, A is required to have a full set of positive eigenvalues. This is what motivates our use of Bayesian statistics. Not only does the noise model explicitly consider Gaussian errors, but it also allows us to enforce prior knowledge on the parameters. We extend the least-squares solution by introducing the model in Equation 6 and then use Bayes law, which allows us to combine the likelihood of the data $p(X|\boldsymbol{\psi})$ with a prior $p(\boldsymbol{\psi})$ in order to produce the posterior.

$$p(\boldsymbol{\psi}|X) \propto p(X|\boldsymbol{\psi})p(\boldsymbol{\psi}) \tag{10}$$

The distribution $p(X)$ is constant with respect to the parameters $\boldsymbol{\psi}$ and so we replace it with the proportionality operator ($\propto$). This is useful because traditionally, one would compute $p(X) = \int_{\boldsymbol{\psi}} p(X|\boldsymbol{\psi})p(\boldsymbol{\psi})$ which can be intractable. Certain properties are maintained across proportionality, for example, the maximum of $p(\boldsymbol{\psi}|X)$ is the same as the maximum of $\kappa p(\boldsymbol{\psi}|X)$ for any κ. Since we only seek a maximum a-priori estimate of the parameters, we only use equivalence of distributions up to proportionality.

With a careful choice of prior (discussed in Section 4) one can find the maximum a-priori estimate of the parameters given the data by finding zeros of the log-posterior derivatives. In the context of fitting a general quadric, and following from Equation 6, we assume additive Gaussian noise on equation (1) in which case the likelihood is taken to be,

$$p(X|\boldsymbol{\psi}) = \mathcal{N}(X\boldsymbol{\psi}|\mathbf{0}, I) \tag{11}$$

In other words, the errors in computing Equation 1 are jointly Gaussian, with a spherical unit covariance I. The assumption of a spherical covariance is equivalent to saying that the variables are independent, provided the parameters $\boldsymbol{\psi}$. This is true, since the error from the original surface of one point tells us nothing about the error of a neighbouring point. The covariances are assumed to be identical as a matter of simplicity. The identity is chosen rather than some scalar multiple, since the parameter becomes superfluous when the prior is introduced. The hyper parameter σ (introduced later in the text) provides all of the information necessary to model the surface.

It would be possible at this point use a prior for the vector $\boldsymbol{\psi}$. Rather than doing this we introduce a bijection between A and $\boldsymbol{\psi}$ through the vector $\boldsymbol{\theta}$ defined in Section 3. This allows the hyperparameters to be set in terms of the matrix rather than the somewhat unintuitive values of $\boldsymbol{\psi}$. In fact, using a matrix normal on A with scalar parameters is equivalent to assuming a multivariate Gaussian on the first elements of $\boldsymbol{\psi}$.

We only place a prior distribution over the matrix A since it contains information about the shape of the surface. This is discussed in detail in Section 7, with the choice of hyper-parameters discussed more in Section 4.

The Matrix Normal distribution is defined as,

$$\mathcal{MN}(A|\mathrm{W}, U, V) = \kappa e^{-\frac{1}{2}tr(V^{-1}(A-\mathrm{W})^TU^{-1}(A-\mathrm{W}))} \tag{12}$$

where, $\mathrm{W}, U, V \in \mathbb{R}^{M\times M}$ and,

$$\kappa = \left((2\pi)^{np/2}|V|^{n/2}|U|^{n/2}\right)^{-1} \tag{13}$$

The posterior is then found by multiplying Equation 11 by Equation 12 to give,

$$p(\boldsymbol{\psi}|X) \propto \mathcal{N}(X\boldsymbol{\psi}|\mathbf{0}, I)\mathcal{MN}(A|\mathrm{W}, U, V) \tag{14}$$

$$\propto e^{-\frac{1}{2}||X\boldsymbol{\psi}||^2}e^{-\frac{1}{2}tr(V^{-1}(A-\mathrm{W})^TU^{-1}(A-\mathrm{W}))} \tag{15}$$

If we then choose $U = V = \sigma I$, and ensure that W is diagonal and symmetric, the posterior can be simplified to,

$$p(\boldsymbol{\psi}|X) \propto e^{-\frac{1}{2}||X\boldsymbol{\psi}||^2}e^{-\frac{1}{2\sigma^2}tr(A^TA-2\mathrm{W}A+W^2)} \tag{16}$$

$$\propto e^{-\frac{1}{2}||X\boldsymbol{\psi}||^2}e^{-\frac{1}{2\sigma^2}(tr(A^TA)-tr(2A\mathrm{W})+tr(W^2))} \tag{17}$$

Taking the log natural and then the first partial derivatives yields,

$$\frac{\partial lnp(\boldsymbol{\psi}|X)}{\partial\boldsymbol{\psi}} = -\frac{1}{2}\frac{\partial}{\partial\boldsymbol{\psi}}||X\boldsymbol{\psi}||^2 - \frac{1}{2\sigma^2}\frac{\partial}{\partial\boldsymbol{\psi}}tr(A^TA) - \frac{1}{\sigma^2}\frac{\partial}{\partial\boldsymbol{\psi}}tr(\mathrm{W}^TA) \tag{18}$$

Before solving the vector derivatives, we first take note of the matrix derivatives for a symmetric A and W (see [17] for more details),

$$\frac{d(tr(W^TA))}{dA} = W + W^T - W\circ I \tag{19}$$

$$\frac{d(tr(A^TA))}{dA} = 2(A + A^T - A\circ I) \tag{20}$$

The operator $\circ$ is the Hadamard product for matrices, and produces a matrix of equal size with element wise multiplication, for example, if $S = Q\circ R$ then $S_{ij} = Q_{ij}R_{ij}$ for each i and j. More intuitively, Equations 19 and 20 represent a doubling of all of the the off diagonal

elements and leaving the diagonals as they were. Since A is symmetric we change notation for the derivatives and set,

$$\hat{A} = A + A^T - A \circ I \tag{21}$$

$$\hat{W} = W + W^T - W \circ I \tag{22}$$

The final derivative can then be calculated as,

$$\frac{\partial ln p(\boldsymbol{\psi}|X)}{\partial \boldsymbol{\psi}} = -X^T X \boldsymbol{\psi} - \frac{1}{\sigma^2} \begin{bmatrix} \boldsymbol{\theta}_{\hat{A}} \\ \mathbf{0} \end{bmatrix} + \frac{1}{\sigma^2} \begin{bmatrix} \boldsymbol{\theta}_{\hat{W}} \\ \mathbf{0} \end{bmatrix} \tag{23}$$

In the final step we separate $\boldsymbol{\psi}$ from X and σ by the distribution of multiplication over addition. Here $\boldsymbol{\psi}$ depends on A and $\boldsymbol{\theta}_{\hat{A}}$ on $\hat{A}$. Let the matrix J be a diagonal matrix which takes a value of 1 on the diagonal when the corresponding value of $\boldsymbol{\theta}_{\hat{A}}$ is a diagonal element, and 2 otherwise. Setting the partials to zero it then follows that,

$$\left(X^T X + \frac{1}{\sigma^2} \begin{bmatrix} J & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \right) \boldsymbol{\psi} - \frac{1}{\sigma^2} \begin{bmatrix} \boldsymbol{\theta}_{\hat{W}} \\ \mathbf{0} \end{bmatrix} = 0 \tag{24}$$

Rearranging yields,

$$\boldsymbol{\psi} = \left(\sigma^2 X^T X + \begin{bmatrix} J & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \right)^{-1} \begin{bmatrix} \boldsymbol{\theta}_{\hat{W}} \\ \mathbf{0} \end{bmatrix} \tag{25}$$

4 Choice of prior

We use a matrix normal prior rather than a Wishart or inverse Wishart prior for the matrix A. The Wishart distributions are commonly used, well principled, and suitable for use with empirical data. However, computing a maximum a-priori estimate in a similar fashion to the previous section leads to a solution that must be determined by search, as opposed to the closed form of Equation 25. The requirement that Wishart and inverse Wishart matrices are symmetric positive definite is another limitation, particularly if cylinders or hyperboloids are necessary. Section 7 discusses how to construct a mean matrix W in order to favour different shapes of surface.

The prior is only cast over the matrix A since the parameters $\boldsymbol{b}$ and c contain information about the center and also the width of the resulting surface. These values are also coupled, in the sense that a change in center affects both $\boldsymbol{b}$ and c. Section 6 discusses how to extract the center and width from the parameters, but their relationship is pathological. Constraining A alone provides just enough information on the shape of the surface to ensure that it does not become a hyperboloid, without affecting the quality of the fit.

The matrices U and V determine the variance of the matrix A from the mean W. Smaller values more strongly favour the mean as opposed to the data. For our experiments we simply choose the parameter $\sigma \in \mathbb{R}^+$, which gives control over how close the matrix A is to the identity. The value σ is free, in the sense that it can be chosen arbitrarily or depending on a larger dataset. The results in Section 8 describe how it is chosen for our experiments, however, it could also be computed using a training set, which contained prior information about the variance of the observed quadric from the mean.

5 3D data

The results in Section 3 are theoretical, in the sense that they can be applied to data of arbitrary dimension. This section provides explicit formula for 3 dimensions. We note that the same principles apply for 2 dimensional data, allowing us to model ellipses and hyperbola.

The parameters A and $\boldsymbol{b}$ can be written as follows,

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{26}$$

In this case $\boldsymbol{\psi}$ is defined as,

$$\boldsymbol{\psi} = \begin{bmatrix} \boldsymbol{\theta}_A \\ \boldsymbol{b} \\ c \end{bmatrix} \tag{27}$$

where,

$$\boldsymbol{\theta}_A = [a_{11}, a_{22}, a_{33} a_{12}, a_{13}, a_{23}]^T \tag{28}$$

The matrix X is the quadratic data matrix,

$$X = [\boldsymbol{x}_0, \ldots, \boldsymbol{x}_N] \tag{29}$$

where,

$$\boldsymbol{x}_i = (z_{1i}^2, z_{1i}^2, z_{3i}^2, 2z_{1i}z_{2i}, \ldots, z_{1i}, z_{2i}, z_{3i}, 1)^T \tag{30}$$

The matrix J is then taken to be,

$$J = \begin{bmatrix} I & 0 \\ 0 & 2I \end{bmatrix} \tag{31}$$

and,

$$\boldsymbol{\theta}_{\hat{W}} = \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix} \tag{32}$$

6 Parametrisation

The quadric can be parameterised by finding the transformation between the canonical shape, such as the unit sphere, and the resulting surface.

Consider the following,

$$(\boldsymbol{x} - \boldsymbol{\mu})^T \Lambda (\boldsymbol{x} - \boldsymbol{\mu}) = \tau \tag{33}$$

Multiplying out and rearranging leads to,

$$\boldsymbol{x}^T \Lambda \boldsymbol{x} - \boldsymbol{\mu}^T (\Lambda + \Lambda^T) \boldsymbol{x} + \boldsymbol{\mu}^T \Lambda \boldsymbol{\mu} - \tau = 0 \tag{34}$$

If we then take,

$$A = \Lambda \tag{35}$$

$$\boldsymbol{b} = -(\Lambda + \Lambda^T) \boldsymbol{\mu} \tag{36}$$

$$c = \boldsymbol{\mu}^T \Lambda \boldsymbol{\mu} - \tau \tag{37}$$

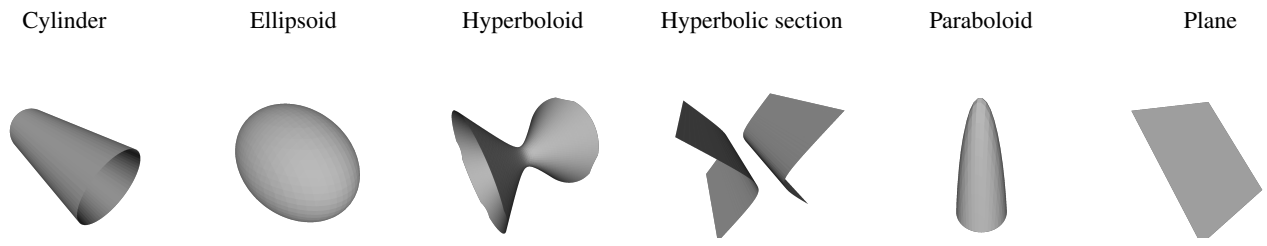


Fig. 1 A collection of canonical quadric shapes.

it can be seen that Equation 1 takes on a canonical form of $\boldsymbol{y}^T W \boldsymbol{y} = 1$ after an affine transformation of the data. Where W is a diagonal matrix with entries contained in the set $\{1, 0, -1\}$.

For an ellipsoid, the matrix W is the identity. Faces and vertices on a unit sphere are generated using a spherical coordinate system, and we transform only the vertices. Letting $\boldsymbol{v}$ be an arbitrary vertex on the unit sphere, and $USU^T = A$ be the eigen decomposition of A (since A is Hermitian $U^T = U^{-1}$). It follows that,

$$\hat{\boldsymbol{v}} = U \left(\frac{S}{\tau} \right)^{-\frac{1}{2}} \boldsymbol{v} + \boldsymbol{\mu} \tag{38}$$

where $\hat{\boldsymbol{v}}$ is the required ellipsoid vertex. The same principle can be taken for any canonical shape, including cylinders, ensuring that the principal axis is aligned with the zero entry of W.

Alternative methods exist for computing the solutions of Equation 1, isosurface algorithms for example. A parametric approach is beneficial for its speed, however it then also requires algebraic methods for merging with other surfaces; which can become complicated.

7 Eigen decomposition

The shape of the quadric can be determined by analysing the eigenvalues of A.

Since A is real and symmetric it is a special case of a Hermitian matrix, and so it follows that all of the eigenvalues are real. In the case of a symmetric positive definite matrix all of the eigenvalues are positive, and so the quadric forms an ellipsoid. If any of the eigenvalues are zero then the axis of the corresponding eigenvector is indeterminate, in the case of 2 positive eigenvalues this is a cylinder, for example. A single negative eigenvalue leads to a hyperboloid.

The final classifications can be obtained by studying the canonical form of the quadric introduced in Section 6, $\boldsymbol{y}^T W \boldsymbol{y} = 1$, where $\boldsymbol{y}$ is an affine transformation of the arbitrary data values $\boldsymbol{z}$, and W is a diagonal matrix. Multiplying this equation out for different combinations of sign on the diagonal of W reduces to a scalar equation which can be compared to studied quadratic forms, as presented in Andrews and Sequin [2].

It is important to note that Equation 1 is not only invariant to scale, but also negation, in the sense that multiplying the whole equation by a scalar $\alpha \in \mathbb{R}$ does not change the solution set. Consider the following expansion,

$$\boldsymbol{z}^T A \boldsymbol{z} + \boldsymbol{b}^T \boldsymbol{z} + c = 0 \tag{39}$$

$$\iff \alpha(\boldsymbol{z}^T A \boldsymbol{z} + \boldsymbol{b}^T \boldsymbol{z} + c) = 0 \tag{40}$$

$$\iff \boldsymbol{z}^T (\alpha A) \boldsymbol{z} + \alpha \boldsymbol{b}^T \boldsymbol{z} + \alpha c = 0 \tag{41}$$

Further, let $A = USU^T$ be the eigen decomposition of A, then $\alpha A = U^T(\alpha S)U$. This means that a scaling and negation of the eigenvalues of A only affects the other parameters, but not the solutions, e.g. a full set of negative eigenvalues yields the same solution as a full set of positive eigenvalues. Since the sign of the eigenvalues of A are the same as the sign of the diagonal entries of W we can relate them to the shape of the surface.

Shape	Positive	Zero	Negative
Ellipsoid	3	0	0
Cylinder	2	1	0
Paraboloid	2	1	0
Plane	0	0	0
Hyperboloid	2	0	1
Hyperbolic section	1	1	1

Tab. 1 A table showing the relationship between eigenvalues and shape of the corresponding quadric.

A range of different surface shapes are attainable in this way, for example, a hyperboloid can be made from two negative and one positive eigenvalue, which is the same as two positive and one negative eigenvalue. A table showing the classification of the surface with the eigenvalues is shown in Table 1, and represents only a sample of the possible combinations. We note that the parameters $\boldsymbol{b}$ and c do also have an affect on the shape, for example, if $\boldsymbol{b} = 0$ and we have one positive eigenvalue, and two zero, the resulting shape is not a paraboloid but a pair of planes. A

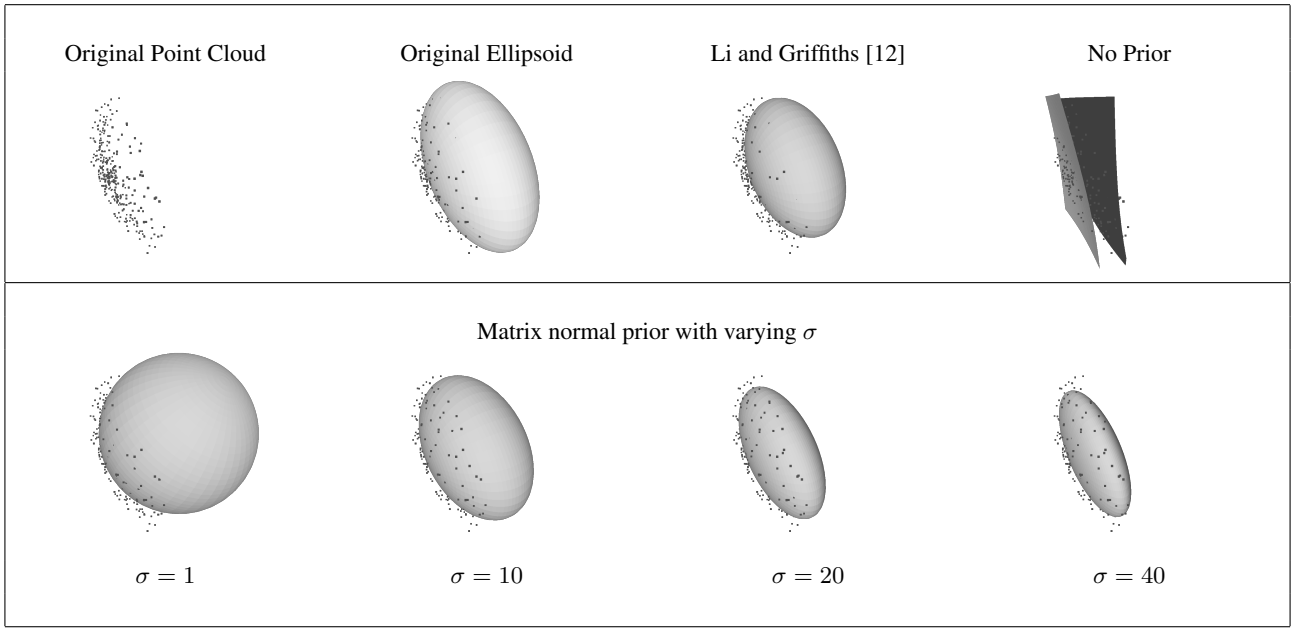


Fig. 2 A collection of ellipses that have been fit to the data. The bottom row shows the results of fitting with a prior, and the top right shows the results of fitting with no prior. The first two images are the point cloud and the ellipsoid that they were generated from.

collection of examples of these shapes can be seen in Figure 1.

8 Results

The results are broken in to two scenarios. Firstly it is shown that the method is able to perform on data drawn from simulated quadrics, ellipsoids with known parameters. The algorithm is shown to perform on empirical data from a collection of 3D point clouds. The results are compared to quadric fitting without a prior and also against the work of Li and Griffiths [12], who solve a generalised eigenvalue problem similar to that of [7].

The work presented by Li and Griffiths is a candidate example of a modern ellipsoid fitting algorithm, from the larger set of algorithms available (e.g. [1–3, 8, 15, 18, 22]). Our method is a Bayesian model, which allows us to provide a parameter which expresses a measure of how close the shape should be to a canonical shape, such as a sphere. Rather than compare to all other algorithms we simply show that a hard constraint in to an ellipse does not always fit the data, particularly when there is a large percentage of it missing or excess noise.

In the final section of the results we give an example of a fit to a simulated hyperboloid, primarily to show that our algorithm also performs on different surface types, it also demonstrates the value in choosing different diagonals on the mean matrix W.

8.1 Algorithmic complexity

One of the most attractive attributes of a direct method for fitting, such as ours and [12] or [7], is its efficiency. A well known alternative is to compute the squared Euclidean (or *geometric*) distance between each point and the surface, and then minimise the sum of squares [21]. This is difficult since computing the Euclidean distance is a non linear problem, see [6, 14] for more details, and the complexity is dependent on a polynomial root finding algorithm. Minimising the sum of squared errors in this way requires an non-linear least squares algorithm such as Levenberg-Marquardt. Computing the error alone is in the order of $\mathcal{O}(NM^3)$, and it is a lower bound on the final fitting algorithm which involves iteratively computing the error and parameter gradients at each step. It is possible, however, to approximate the distance function, as shown in [25] and [15], and fit using generalised eigenvalues, or using a linear least squares estimate; these methods have a much lower complexity, but do not allow us to use a Bayesian prior when fitting.

The complexity of the algorithm presented in the paper is represented in the computation of the scatter matrix $X^T X$ in Equation 25, which is $\mathcal{O}(NM^2)$, since the subsequent matrix operations are much lower ($\mathcal{O}(M^3)$ at worst). This makes it an efficient option, particularly since the scatter can be computed independently of the value σ. Faster algorithms exist for computing covariances, for example [11], at the expense of memory, although it may be generally quicker to use parallel computation, such as a GPU, to

improve performance.

8.2 Simulated data

We show the results on a simulated dataset. Data is drawn from an ellipsoid of known dimensions, some of the data is removed, and spherical Gaussian noise is added. We show that with the correct choice of hyper parameters, the Bayesian method fits best.

Figure 2 provides an example of fitting an ellipsoid to a simulated data set, with missing data. In this experiment we only use 30% of the training data from the original ellipsoid and add Gaussian noise with a standard deviation of 0.01. It can be seen from the results that a value of $\sigma = 10$ produces a result most plausibly from the original, where the result of fitting with no prior produces a hyperboloid. It is clear from this example that for lower values of σ produce a result that is closer to a sphere.

In order to determine the quality of the fit, we compare the vertices on the original quadric with the fitted quadric using the Hausdorff distance. Firstly, the distance between a point $\boldsymbol{x}_i \in X$ and a point cloud $Y = \{\boldsymbol{y}_0, \ldots, \boldsymbol{y}_N\}$ is taken to be,

$$\varrho(\boldsymbol{x}_i, Y) = \min_{j \in \{1,\ldots,N\}} (|\boldsymbol{x}_i - \boldsymbol{y}_j|) \tag{42}$$

The Hausdorff distance is then,

$$H(X,Y) = \max(\max_{i \in \{1,\ldots,M\}} (\varrho(\boldsymbol{x}_i, Y)), \max_{j \in \{1,\ldots,N\}} (\varrho(\boldsymbol{y}_j, X))) \tag{43}$$

The first experiment is to show how each of the algorithms perform when there is missing data. We generate an arbitrary ellipsoid and remove $N\%$ of the data, and add spherical Gaussian noise with $1e-2$ standard deviations. The parameters of the ellipsoid are drawn as follows,

$$\Lambda \sim \mathcal{W}(\omega^2 I, 20) \tag{44}$$

$$\boldsymbol{\mu} \sim \mathcal{N}(\mathbf{0}, \omega^2 I) \tag{45}$$

$$\tau \sim \mathcal{N}(1, \omega^2) \tag{46}$$

With $\omega^2 = 1e-2$. The parameters A, $\boldsymbol{b}$ and c are computed as in Section 6. The percentage of data removed is varied between 10 and 100, and we choose the hyperparameter $0.1 < \sigma < 30$ which produces the best Hausdorff distance. Using a multivariate matrix normal prior is compared with (1) using no prior at all and (2) the ellipsoid specific, least squares, method of Li and Griffiths [12].

The results of the first experiment can be seen in Figure 3. Our method consistently outperforms both methods, although the distance drops below the standard deviation in Gaussian noise in all methods after 60% of the data is observed. Fitting without a prior also gives reasonable results if enough of the data is observed.

In the second simulation experiment we demonstrate how the method is robust to noise by varying the amount of additive Gaussian noise added to each sampled ellipsoid, denote the noise level as $\hat{\tau}$. There is some relationship between of size of the original ellipse to noise ratio and the quality of fit. In order to demonstrate the quality of fit over a range of values we sample 10 quadrics and compute error statistics. In this experiment we use 100% of the points, i.e. there is no missing data. The hyperparameter is fixed at $\sigma = 10$. The results from the second experiment can be seen in Figure 4. Our method produces a Hausdorff distance which is comparable to [12] for lower values of $\hat{\tau}$, as $\hat{\tau}$ increases the use of a prior proves to be beneficial. Although fitting without a prior gives reasonable results when $\hat{\tau}$ is small, it becomes very inaccurate when $\hat{\tau}$ is greater than 0.06. This tends to be due to negative eigenvalues in the matrix A which produces a hyperboloid.

In order to show that the algorithm works for other types of quadric surface we simulate a hyperboloid and remove a percentage of the points. Three quadrics are fit to the data

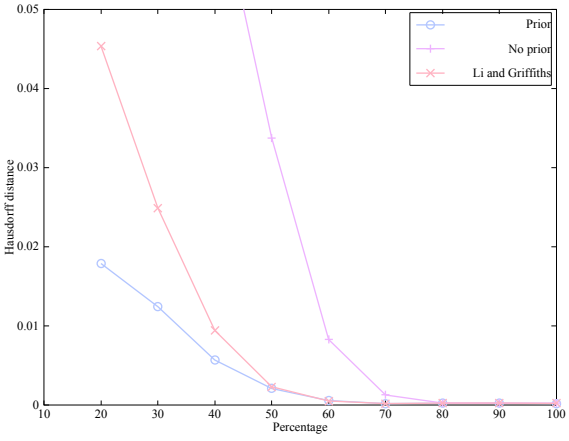


Fig. 3 The Hausdorff distance for a varying amount of missing data.

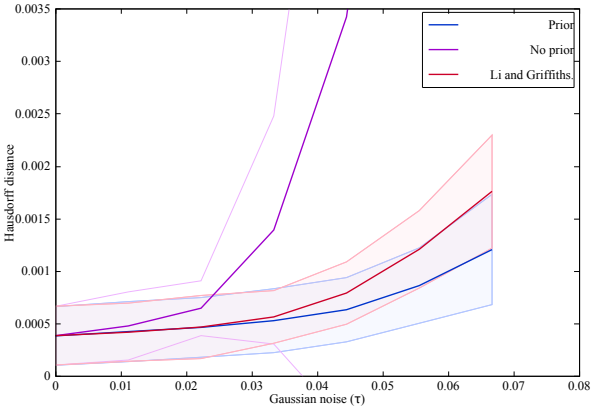


Fig. 4 The Hausdorff distance for a varying amount of Gaussian noise. The shaded regions represent one standard deviation from the mean.

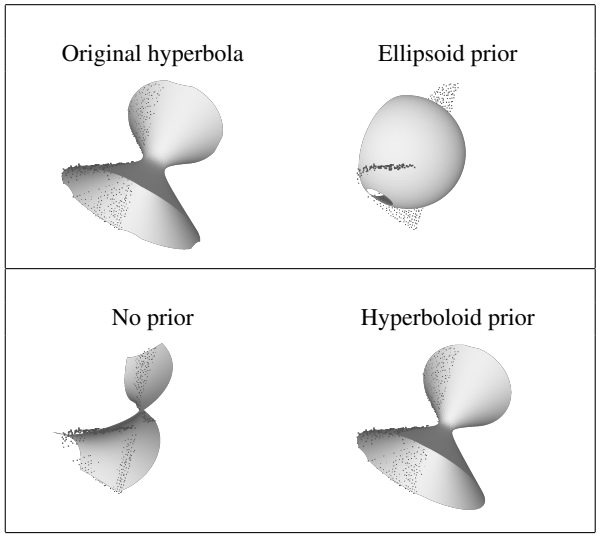


Fig. 5 A table showing the fitting of a hyperbola using a variety of different priors. The original points are shown as black dots.

using different priors for each of an ellipsoid, no prior and a hyperbola. For the ellipsoid the $W = I$ and $\sigma = 1e^{-6}$; and for the hyperbola $W = diag(-1, 1, 1)$ and $\sigma = 10$. Notice that the variance on the ellipsoid needs to be low enough or it will favor a different shape. The hyperboloid hyper-parameter is aligned on the x-axis, this also true of the simulated data, and represents our knowledge of the shape. Fitting without a prior chooses the hyperboloid's medial axis to be aligned with the z-axis. The results of this experiment can be seen in Figure 5. The original quadric is in the top left of the figure. It is clear that using a hyperboloid prior is the best option for this data.

8.3 Empirical data

We provide some examples which demonstrate our algorithm on real data from point clouds extracted from a visual structure from motion pipeline. In this scenario, regions of the point cloud can be left without any reconstructed points. A candidate example is shown in Figure 6. A smooth surface must be fit to the roof of the phone box, a challenge for most surface fitting algorithms due the the noisy and shapeless extracted points. Previous algorithms either do not fit at all, such as in Li and Griffiths [12], or do not allow control over the curvature of the data, which is the case when using no prior. The surface gradient can be chosen to match the roof curvature by varying σ and produces a final result as shown.

Some examples of the algorithm on point cloud data are shown in Figure 7. The point clouds are computed from a collection of photographs comprising a post box, rugby ball and roof. The surfaces are visually plausible even when

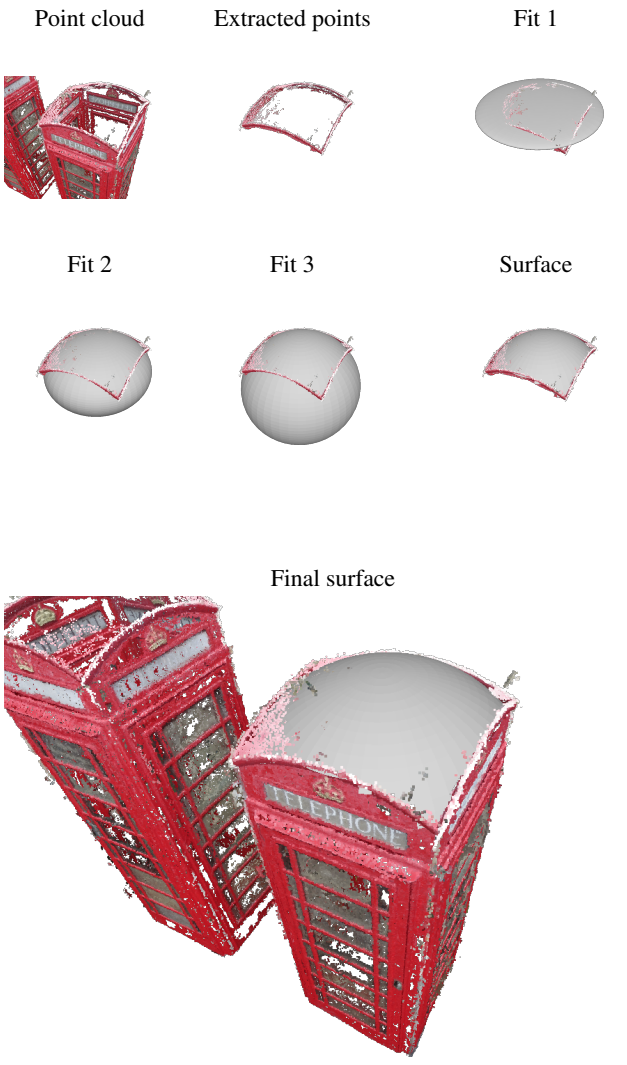


Fig. 6 Phone box data set. The top left shows the original point cloud and the extracted points that a quadric must fit to. There are 3 examples with varying sigma values and the final surface on the bottom row.

there are large regions of missing data, this is particularly visible on the roof data, from which only the front is visible in each of the photographs.

9 Conclusions

We conclude by observing that state of the art methods for quadric fitting give reasonable results on noisy point clouds. Our algorithm provides a means to enforce a prior, allowing the algorithm to better fit the quadric that the points were drawn from, particularly when there is missing data or a large amount of noise. This has a practical use on real data sets, since it allows a user to specify the curvature of the surface where there are few points available.

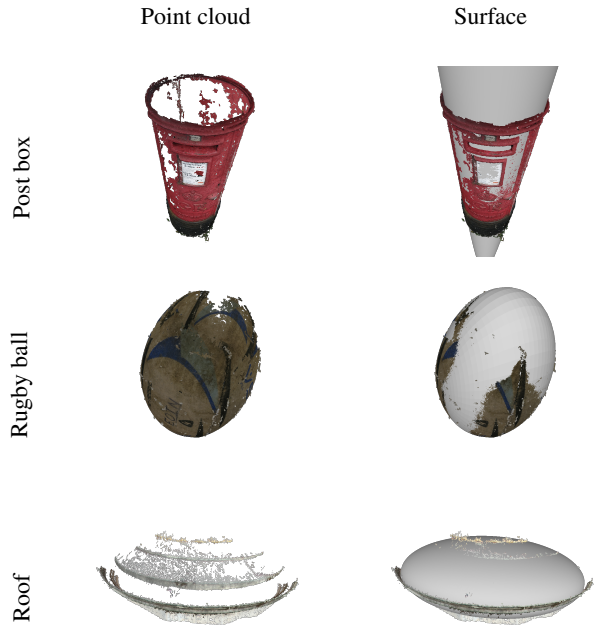


Fig. 7 Some examples of the quadric fitting algorithm on point cloud data.

A Symmetry of the matrix A

The matrix A is assumed symmetric without loss of generality. This is possible since the function,

$$f : \mathbb{R}^{M\times M} \to \mathbb{R}^{M} \tag{47}$$

$$A \mapsto \boldsymbol{x}^T A\boldsymbol{x} \tag{48}$$

is not injective, for any $\boldsymbol{x} \in \mathbb{R}^M$. For example, if we take,

$$B = \frac{1}{2}(A + A^T) \tag{49}$$

which is a symmetric matrix, then,

$$\boldsymbol{x}^T B\boldsymbol{x} = \boldsymbol{x}^T(\frac{1}{2}(A + A^T))\boldsymbol{x} \tag{50}$$

$$= \frac{1}{2}(\boldsymbol{x}^T A\boldsymbol{x} + \boldsymbol{x}^T A^T\boldsymbol{x}) \tag{51}$$

since $\boldsymbol{x}^T A^T\boldsymbol{x}$ is a scalar, it is also symmetric, and so,

$$\boldsymbol{x}^T A^T\boldsymbol{x} = (\boldsymbol{x}^T A^T\boldsymbol{x})^T \tag{52}$$

$$= \boldsymbol{x}^T(\boldsymbol{x}^T A^T)^T \tag{53}$$

$$= \boldsymbol{x}^T A\boldsymbol{x} \tag{54}$$

It then follows that,

$$\boldsymbol{x}^T B\boldsymbol{x} = \boldsymbol{x}^T A\boldsymbol{x} \tag{55}$$

The choice of symmetry then follows, because if A, $\boldsymbol{b}$ and c satisfy, $\boldsymbol{x}^T A\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x} + c = 0$, then so does B, and B is symmetric.

References

[1] S. J. Ahn, W. Rauh, H. S. Cho, and H.-J. Warnecke. Orthogonal distance fitting of implicit curves and surfaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):620–638, 2002.

[2] J. Andrews and C. H. Séquin. Type-constrained direct fitting of quadric surfaces. *Computer-Aided Design and Applications*, 11(1):107–119, 2014.

[3] A. Atieg and G. Watson. A class of methods for fitting a curve or surface to data by minimizing the sum of squares of orthogonal distances. *Journal of Computational and Applied Mathematics*, 158(2):277–296, 2003.

[4] D. Cohen-Steiner, P. Alliez, and M. Desbrun. Variational shape approximation. *ACM Trans. Graph.*, 23(3):905–914, 2004.

[5] M. Dai, T. S. Newman, and C. Cao. Least-squares-based fitting of paraboloids. *Pattern Recognition*, 40(2):504–515, 2007.

[6] D. Eberly. Distance from a point to an ellipse, an ellipsoid, or a hyperellipsoid. *Geometric Tools, LLC*, 2011.

[7] A. Fitzgibbon, M. Pilu, and R. B. Fisher. Direct least square fitting of ellipses. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):476–480, 1999.

[8] W. Gander, G. H. Golub, and R. Strebel. Least-squares fitting of circles and ellipses. *BIT Numerical Mathematics*, 34(4):558–578, 1994.

[9] R. Halır and J. Flusser. Numerically stable direct least squares fitting of ellipses. In *Proc. 6th International Conference in Central Europe on Computer Graphics and Visualization. WSCG*, volume 98, pages 125–132. Citeseer, 1998.

[10] B. Hamann. Curvature approximation for triangulated surfaces. In G. Farin, H. Noltemeier, H. Hagen, and W. Kndel, editors, *Geometric Modelling*, volume 8, pages 139–153. Springer Vienna, 1993.

[11] V. Kwatra and M. Han. Fast covariance computation and dimensionality reduction for sub-window features in images. In *Computer Vision–ECCV 2010*, pages 156–169. Springer, 2010.

[12] Q. Li and J. G. Griffiths. Least squares ellipsoid specific fitting. In *Geometric Modeling and Processing, 2004. Proceedings*, pages 335–340. IEEE, 2004.

[13] Y. Li, X. Wu, Y. Chrysathou, A. Sharf, D. Cohen-Or, and N. J. Mitra. Globfit: Consistently fitting primitives by discovering global relations. *ACM Trans. Graph.*, 30(4):52:1–52:12, 2011.

[14] G. K. Lott. Direct orthogonal distance to quadratic surfaces in 3d. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(9):1888–1892, 2014.

[15] G. Lukács, R. Martin, and D. Marshall. Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation. In *Computer VisionECCV'98*, pages 671–686. Springer, 1998.

[16] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. In *ACM SIGGRAPH 2005 Courses*, page 173. ACM, 2005.

[17] K. B. Petersen, M. S. Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7:15, 2008.

[18] S. Petitjean. A survey of methods for recovering quadrics in triangle meshes. *ACM Computing Surveys (CSUR)*, 34(2):211–262, 2002.

[19] M. W. Powell, K. W. Bowyer, X. Jiang, and H. Bunke. Comparing curved-surface range image segmenters. In *Computer Vision, 1998. Sixth International Conference on*, pages 286–291. IEEE, 1998.
[20] V. Pratt. Direct least-squares fitting of algebraic surfaces. *ACM SIGGRAPH computer graphics*, 21(4):145–152, 1987.
[21] M. Rouhani and A. D. Sappa. Implicit polynomial representation through a fast fitting error estimation. *Image Processing, IEEE Transactions on*, 21(4):2089–2098, 2012.
[22] O. Ruiz, S. Arroyave, and D. Acosta. Fitting of analytic surfaces to noisy point clouds. 2013.
[23] R. Schnabel, R. Wahl, and R. Klein. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226, 2007.
[24] J. Subrahmonia, D. B. Cooper, and D. Keren. Practical reliable bayesian recognition of 2d and 3d objects using implicit polynomials and algebraic invariants. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(5):505–519, 1996.
[25] G. Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (11):1115–1138, 1991.
[26] M. Vona and D. Kanoulas. Curved surface contact patches with quantified uncertainty. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1439–1446. IEEE, 2011.
[27] J. Wu and L. Kobbelt. Structure recovery via hybrid variational surface approximation. *Computer Graphics Forum*, 24(3):277–284, 2005.
[28] D.-M. Yan, Y. Liu, and W. Wang. Quadric surface extraction by variational shape approximation. In *Proceedings of the 4th International Conference on Geometric Modeling and Processing*, pages 73–86, 2006.



Daniel Beale is a post-doctoral research officer at the University of Bath. He obtained a PhD in Computer Science and a Master of Mathematics (MMath) from the University of Bath. He has worked in industrial positions as a software and systems engineer. His interests are in the application of mathematics, statistics and probability theory to problems in computing, particularly in the area of computer vision.



Yong-Liang Yang is a lecturer (assistant professor) in the Department of Computer Science at University of Bath. He received his Ph.D. degree and Masters in Computer Science from Tsinghua University. His research interests include geometric modelling, computational design and computer graphics in general.



Neill D. F. Campbell is a lecturer in Computer Vision, Graphics and Machine Learning in the Department of Computer Science at the University of Bath. Previously he was a Research Associate at University College London (where he is an Honorary Lecturer) working with Jan Kautz and Simon Prince. He completed his MEng and PhD in the Department of Engineering at the University of Cambridge under the supervision of Roberto Cipolla with George Vogiatzis and Carlos Hernàndez at Toshiba Research. His main area of interest is in learning models of shape and appearance in 2D and 3D.



Darren Cosker is a Royal Society Industrial Research Fellow at Double Negative Visual Effects, London, and a Reader/Associate Prof. at the University of Bath. He is the Director of the Centre for the Analysis of Motion, Entertainment Research and Applications (CAMERA), starting from September 2015. Previously, Darren held a Royal Academy of Engineering Research Fellowship, also at the University of Bath. His is interests are in the convergence of computer vision, graphics and psychology, with applications in movies and video games.



Peter Hall is professor of Visual Computing at the University of Bath. His research interests focus around the use of computer vision for computer graphics applications. His research interests are in Computer Graphics applications of Computer Vision. He is known for automatically processing real photographs and video into art, especially abstract styles such as Cubism, Futurism, etc. More recently he has published papers on classification and detection of objects regardless of how they are depicted: as photos, drawings, painting etc. He is also interested in using Computer Vision methods to acquire 3D dynamic models of complex natural phenomena such as trees, water, and fire for use in computer games, TV broadcast, and film special effects.