

# Generalized, Basis-Independent Kinematic Surface Fitting

James Andrews<sup>a</sup>, Carlo H. Séquin<sup>a</sup>

<sup>a</sup>UC Berkeley, Soda Hall, Berkeley, CA 94720

---

## Abstract

Kinematic surfaces form a general class of surfaces, including surfaces of revolution, helices, spirals, and more. Standard methods for fitting such surfaces are either specialized to a small subset of these surface types (either focusing exclusively on cylinders or exclusively on surfaces of revolution) or otherwise are basis-dependent (leading to scale-dependent results). Previous work has suggested re-scaling data to a fixed size bounding box to avoid the basis-dependence issues. We show that this method fails on some simple, common cases such as a box or a cone with small noise. We propose instead adapting a well-studied approximate maximum-likelihood method to the kinematic surface fitting problem, which solves the basis-dependence issue. Because this technique is not designed for a specific type of kinematic surface, it also opens the door to the possibility of new variants of kinematic surfaces, such as affinely-scaled surfaces of revolution.

**Keywords:** kinematic surface fitting, reverse engineering, slippable surfaces, velocity fields, approximate maximum likelihood, Taubin’s method

---

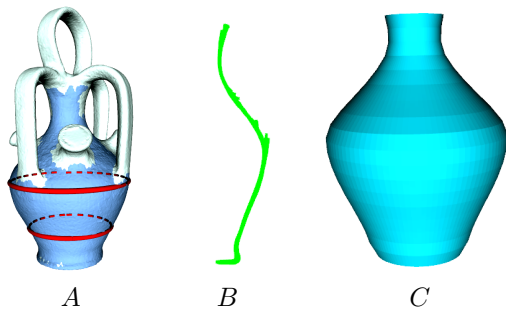


Figure 1: A simple example of kinematic surface fitting: (A) A slippable motion (illustrated by red streamlines) is found for some selection of the data (in blue). (B) The data (in green) is projected to some shared sweep plane, where it can be fit by a generator curve. (C) Advecting the generator curve along the slippable motion field generates the kinematic surface.

## 1. Introduction

A fundamental sub-problem of reverse engineering is to fit a primitive surface to a set of points [1]. Kinematic surfaces are a class of primitive surfaces notable for their general applicability. They can be used to classify a wide range of common surfaces: spheres, planes, cylinders, cones, surfaces of revolution, logarithmic spirals, helices, and more [2]. By chaining simple kinematic surfaces, even more interesting surfaces can be fit, including profile and developable surfaces [3]. A number of recent systems use these methods for reverse engineering tasks [4, 5, 6, 7].

A kinematic surface is a surface that is tangent everywhere to some easily parameterizeable, linear velocity field

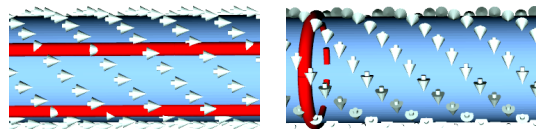


Figure 2: Visualization of two kinematic motions of a cylinder. The arrows show the direction of the fields at individual points, and the red lines are streamlines showing paths traced by following the velocity fields.

over space (such as those in Sec. 2.1). Given such a field and a curve in space (which we refer to as the “generator curve”), advecting the generator curve by the velocity field will generate a kinematic surface (Fig. 1). Such tangent-everywhere velocity fields are called the “slippable motions” of a surface [8]. Slippability of a point  $\mathbf{p}$  with normal  $\mathbf{n}$ , with respect to a field  $\mathbf{v}(\mathbf{p})$ , can be tested by checking that the normal is orthogonal to the velocity field:  $\mathbf{v}(\mathbf{p}) \cdot \mathbf{n} = 0$ . The full basis of slippable motions of a surface can be used to classify the surface type: for example, if a surface is slippable by both a pure rotation and a pure translation in the direction of the rotation axis (as in Fig. 2), then the surface is a cylinder.

The kinematic surface fitting problem consists of several sub-problems: segmentation of a surface into subsets that can be fit by separate kinematic surfaces [8, 2]; fitting a kinematic motion to a given set of data points; and finally, finding a generator curve [9]. We focus specifically on the problem of fitting the kinematic motion. We show that previous methods for fitting general kinematic motion can fail on some simple, common cases such as a box or a cone with small noise. We then show how to fix these problems.

---

Email address: [jima@eecs.berkeley.edu](mailto:jima@eecs.berkeley.edu) (James Andrews)

## 2. Background

Previous work on fitting the velocity fields of kinematic surfaces has used a common fitting method (Sec. 2.2) and problem formulation: Given a set of  $n$  points with their surface normals  $\{\mathbf{p}_i, \mathbf{n}_i\}$ , and a velocity field (Sec. 2.1) parameterized by some vector  $\mathbf{m}$ , find field parameters that are “most slippable” with respect to the data points.

### 2.1. Velocity Field Types

Three velocity fields have been proposed. In order of increasing generality, they are: First, a *constant field*, which accepts only translational motion [9]:

$$\mathbf{v}(\mathbf{p}) = \mathbf{c} \quad (1)$$

Second, a *helical field*, which adds optional rotational motion for helices and surfaces of revolution [10]:

$$\mathbf{v}(\mathbf{p}) = \mathbf{r} \times \mathbf{p} + \mathbf{c} \quad (2)$$

Finally, a *spiral field* which adds optional scaling motion for cones and logarithmic spirals [11]:

$$\mathbf{v}(\mathbf{p}) = \mathbf{r} \times \mathbf{p} + \mathbf{c} + \gamma \mathbf{p} \quad (3)$$

### 2.2. Common Fitting Method

Almost all kinematic surface fitting papers use a common direct fitting algorithm [10] to find the field parameters.

First, the slippability of the field with respect to the data is expressed in terms of some symmetric ‘covariance’ matrix  $\mathbf{M}$ , such that  $(\mathbf{v}(\mathbf{p}) \cdot \mathbf{n})^2 = \mathbf{m}^T \mathbf{M} \mathbf{m}$ , where  $\mathbf{m}$  is the vector of velocity field parameters. For example, for the spiral field with parameters  $\mathbf{m} = \langle \mathbf{r}_x, \mathbf{r}_y, \mathbf{r}_z, \mathbf{c}_x, \mathbf{c}_y, \mathbf{c}_z, \gamma \rangle$ , matrix  $\mathbf{M}$  will be:

$$\mathbf{M} := \sum_i \mathbf{f}(\mathbf{x}_i) \mathbf{f}(\mathbf{x}_i)^T \quad (4)$$

$$\text{where } \mathbf{x} := \langle p_x, p_y, p_z, n_x, n_y, n_z \rangle$$

$$\text{and } \mathbf{f}(\mathbf{x}) := \langle (\mathbf{p} \times \mathbf{n})_x, (\mathbf{p} \times \mathbf{n})_y, (\mathbf{p} \times \mathbf{n})_z, n_x, n_y, n_z, p \cdot \mathbf{n} \rangle$$

Second, a normalization is introduced to avoid degenerate solutions such as the field with  $\mathbf{v}(\mathbf{p}) = \mathbf{0}$  everywhere. The most straightforward solution would be to normalize by  $\|\mathbf{v}(\mathbf{p})\|$ :

$$\frac{\mathbf{v}(\mathbf{p}_i) \cdot \mathbf{n}_i}{\|\mathbf{v}(\mathbf{p}_i)\|} \quad (5)$$

However, this normalization has been largely avoided because it would make the field a *non-linear* function of its parameters, leading to a non-linear optimization problem [10]. Instead, previous work has normalized the cumulative squared error by some quadratic function  $q$  (Sec. 2.3) of the motion parameters  $\mathbf{m}$ :

$$\operatorname{argmin}_{\mathbf{m}} \frac{\frac{1}{n} \sum_{i=1}^n (\mathbf{v}(\mathbf{p}_i) \cdot \mathbf{n}_i)^2}{q(\mathbf{m})} \quad (6)$$

Note that this normalization can alternatively be viewed as a constraint on the solution vector: minimizing Eqn. 6 is equivalent to minimizing the non-normalized metric under the constraint  $q(\mathbf{m}) = 1$ .

Finally, this quadratic function  $q$  is expressed in terms of some (often singular) symmetric matrix  $\mathbf{N}$ :  $q(\mathbf{m}) = \mathbf{m}^T \mathbf{N} \mathbf{m}$ . Using the method of Lagrange multipliers to solve the constrained minimization problem,  $\mathbf{m}$  must be a solution to the generalized eigenvalue problem:

$$(\mathbf{M} - \lambda \mathbf{N}) \mathbf{m} = 0 \quad (7)$$

All eigenvectors corresponding to small eigenvalues of the matrix pencil  $(\mathbf{M} - \lambda \mathbf{N})$  are then slippable motions of the data points.

Some systems augment this core fitting method by iterative re-weighting (for example to downweight outliers) [10, 2] or subsequent application of a general, non-linear fitting technique (which requires a reasonable “initial estimate” from the core fitting method to perform well) [4].

### 2.3. Quadratic Normalization Functions

A few different quadratic normalizations have been proposed. The two commonly used normalizations are: First, the *unit constraint*, in which the full parameter vector is constrained to unit magnitude [8, 2]. For the spiral field, this becomes:

$$(\|\mathbf{r}\|^2 + \|\mathbf{c}\|^2 + \gamma^2) = 1 \quad (8)$$

Second, the *rotation constraint*, which just constrains the rotation axis  $\mathbf{r}$  [10]:

$$\|\mathbf{r}\|^2 = 1 \quad (9)$$

The constant field is simple enough that the constraint  $\|\mathbf{c}\|^2 = 1$  solves the problem perfectly; however this is only applicable to the constant field.

The choice of normalization fundamentally affects the resulting fit: The rotation constraint requires the solution to include a rotation component of constant magnitude, so it should only be used when it is known in advance that rotation is included in the desired solution [10].

The unit constraint is “basis-dependent:” the best fit will change if the data is scaled or translated. Previous work using the unit constraint suggested re-scaling the data to a fixed size to address this issue [8, 2]. This mitigates but does not eliminate the problem: We show in Sec. 3 that there is no fixed scale that can eliminate the artifacts of basis dependence.

## 3. Failure Cases of Current Methods

The kinematic surface fitting methods described in Sec. 2 work on many examples, shown in previous work [10, 8, 2, 7]. However, we found that they also fail on some common, simple cases. In this section, we demonstrate and explain the cases that cause these methods to fail.

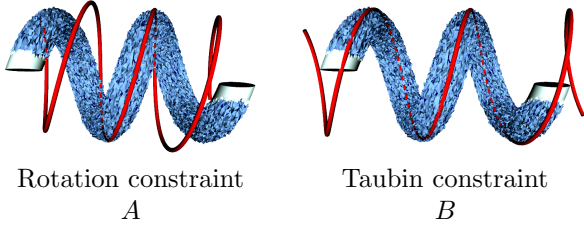


Figure 3: A spiral field (Eqn 3) (red streamlines) is fit to a blue selection of a helix with randomly perturbed vertices (Gaussian noise with  $\sigma = 0.4\%$  of bounding box size) using (A) the rotation constraint and (B) Taubin’s constraint.

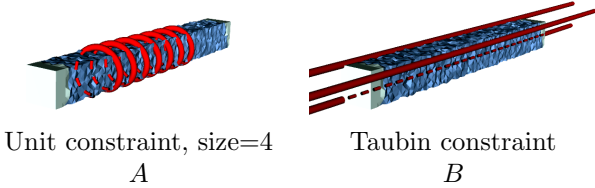


Figure 4: A spiral field (Eqn 3) (red streamlines) is fit to a blue selection of a box with randomly perturbed vertices (Gaussian noise with  $\sigma = 0.2\%$  of bounding box size) using (A) the unit constraint with bounding box size 4 and (B) Taubin’s constraint.

In each failure case, the key problem is that the slippability measured for each data point is scaled by  $\|\mathbf{v}(\mathbf{p})\|$  – a quantity that is unrelated to the actual tangency of the field to the surface at that point. While the quadratic normalizations of Sec. 2.3 avoid the degenerate case of  $\mathbf{v}(\mathbf{p}) = 0$  everywhere, they still permit “low velocity” fields for which  $\|\mathbf{v}(\mathbf{p})\|$  is reduced at every data point. The “most slippable” solutions under these constraints are therefore biased towards such “low velocity” fields. For noisy data, where the expected slippable motions have some error, these bias-favored solutions can be erroneously chosen as the most slippable fields.

We demonstrate the failure cases in practice on simple synthetic example meshes, for which the ideal solutions are readily apparent. Each example mesh is generated with approximately uniform vertex sampling. We introduce a small amount of Gaussian noise (with  $\sigma$  less than  $0.5\%$  of the bounding box size and smaller than half the average edge length), and we recompute the normal for each sample point by averaging face normals. These small-noise examples should not be challenging, but they cause the previous methods to perform poorly due to their systematic biases.

### 3.1. Rotation Constraint Failure Cases

The rotation constraint,  $\|\mathbf{r}\| = 1$ , requires a fixed magnitude rotation be part of the solution field, but does not specify the constraints on the translational or scaling motion of the field. Any translational or scaling motion in the field will therefore increase the magnitude of  $\|\mathbf{v}(\mathbf{p})\|$  everywhere, beyond what is mandated by the rotation constraint. This additional velocity scales the error at each

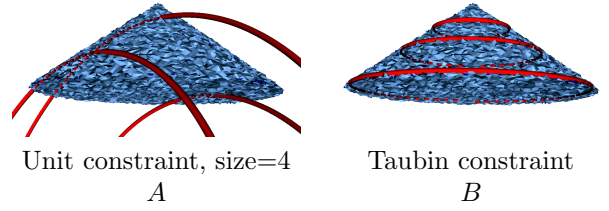


Figure 5: A spiral field (Eqn 3) (red streamlines) is fit to a cone with randomly perturbed vertices (Gaussian noise with  $\sigma = 0.4\%$  of bounding box size) using (A) the unit constraint with bounding box size 4 and (B) the Taubin constraint.

data point, causing the rotation constraint to be systematically biased against translational or scaling motion in the face of noise.

To demonstrate this, we fit a helix with small noise using the rotation constraint in Fig. 3. The rotation constraint underestimates the pitch (translational motion) of the helix. In contrast, a fitting method without this bias (introduced in Sec. 4.1) recovers the expected pitch.

### 3.2. Unit Constraint Failure Cases

The unit constraint is basis-dependent: fitting results depend on the scale and translation of the data points. Therefore, those who use it first center the data points around the origin, and scale the bounding box to a fixed size (e.g. so the longest edge of the box has unit length) [8, 2]. The biases of the unit constraint depend on the chosen size.

One source of bias in the unit constraint favors scaling and rotation at small scales: Velocities of linear scaling and rotational motions are proportional to the distance from the center or axis of the motion, so as data points come closer together, the velocities (and thus errors) from scaling and rotation become smaller. To demonstrate this bias, we fit four sides of a box with small noise using the unit constraint in Fig. 4. At scales with bounding box size 4 or smaller, the resulting fit has a significant, erroneous rotational component.

Another source of bias favors offsetting the rotation axis from the origin. As the constant parameter  $\mathbf{c}$  increases to achieve the offset, the rotation axis  $\|\mathbf{r}\|$  must scale down proportionally (to satisfy the unit constraint). Scaling down the rotation axis scales down the velocity (and thus error) of rotation. To demonstrate this bias, we fit a cone with small noise using the unit constraint in Fig. 5. At scales with bounding box size 4 or greater, the resulting fit erroneously offsets the rotation axis.

From these two examples, we see that the bias of the unit constraint can cause problems at small scales (sizes  $\leq 4$ ) and large scales (sizes  $\geq 4$ ) alike: no single scale works well for all cases.

#### 4. Improved Fitting Methods

The problems we have identified in kinematic surface fitting methods are similar to those faced by early methods for algebraic surface fitting [12]. “Approximate maximum likelihood” (AML) methods are a general class of methods for fitting algebraic surfaces and general parametric models, which have been applied to many other problems [13, 14, 15]. In this section, we show how to apply AML methods to the kinematic surface fitting problem to create an improved kinematic surface fitting method.

AML methods can apply to any parametric model that takes the form  $\mathbf{m} \cdot \mathbf{f}(\mathbf{x}) = 0$ . For our kinematic equations (Sec. 2.1) this holds – in the case of a spiral field, for example,  $\mathbf{m}$  would be the parameter vector  $\langle r_x, r_y, r_z, c_x, c_y, c_z, \gamma \rangle$ , and  $\mathbf{f}(\mathbf{x})$  would be the transformation defined in Eqn. 4.

The “maximum likelihood” (ML) method seeks to minimize the squared distance from each data point  $\mathbf{x}_i$  to the nearest corresponding point on the model surface  $\bar{\mathbf{x}}_i$ ; in other words, to minimize:

$$\sum_i \|\mathbf{x}_i - \bar{\mathbf{x}}_i\|^2, \text{ subject to } \mathbf{m} \cdot \mathbf{f}(\bar{\mathbf{x}}_i) = 0$$

Because  $\bar{\mathbf{x}}_i$  is the zero of  $\mathbf{m} \cdot \mathbf{f}(\mathbf{x})$  that is closest to  $\mathbf{x}_i$ , the distance to this root can be approximated to first order by the magnitude of one step of Newton’s method. This gives the AML distance:

$$\sum_i \frac{(\mathbf{m} \cdot \mathbf{f}(\mathbf{x}_i))^2}{\|\nabla_{\mathbf{x}}(\mathbf{m} \cdot \mathbf{f}(\mathbf{x}_i))\|^2} \quad (10)$$

Rewritten in terms of kinematic surface fitting, the AML method then becomes:

$$\sum_i \frac{(\mathbf{v}(\mathbf{p}_i) \cdot \mathbf{n}_i)^2}{\|\nabla_{\mathbf{p}}(\mathbf{v}(\mathbf{p}_i) \cdot \mathbf{n}_i)\|^2 + \|\mathbf{v}(\mathbf{p}_i)\|^2} \quad (11)$$

Note that the AML and ML methods are both scale dependent, because the closest element  $\bar{\mathbf{x}}_i$  can be different from  $\mathbf{x}_i$  in both position and normal. If the points are scaled up, differences in the normal are unchanged, but differences in the position increase. We can make this trade-off explicit: scale the data points to a fixed size bounding box (we scale it so the longest axis has length 1), then introduce a weight parameter  $w_{\mathbf{p}}$  that scales the contribution of the position-based term:

$$\sum_i \frac{(\mathbf{v}(\mathbf{p}_i) \cdot \mathbf{n}_i)^2}{w_{\mathbf{p}} \|\nabla_{\mathbf{p}}(\mathbf{v}(\mathbf{p}_i) \cdot \mathbf{n}_i)\|^2 + \|\mathbf{v}(\mathbf{p}_i)\|^2} \quad (12)$$

Note that as  $w_{\mathbf{p}}$  goes to zero, the AML fitting equation becomes the non-linear minimization (Eqn 5) previously proposed for kinematic surface fitting [10]. Therefore, in theory, all results on AML fitting apply directly to this fitting problem. However, this original non-linear minimization has numerical issues around singularities where  $\|\mathbf{v}(\mathbf{p})\|$  goes to zero: at these points, that slippability metric is undefined. A small, non-zero value for  $w_{\mathbf{p}}$  gives a more stable metric with no undefined points.

##### 4.1. Direct AML Method: Taubin’s Constraint

The AML distance metric is non-linear, requiring iterative methods or approximation to find a solution. One popular approximation is Taubin’s method [13], which approximates the non-linear AML metric (Eqn. 10) by summing all numerator and denominator elements separately:

$$\frac{\sum_i (\mathbf{m} \cdot \mathbf{f}(\mathbf{x}_i))^2}{\sum_i \|\nabla_{\mathbf{x}}(\mathbf{m} \cdot \mathbf{f}(\mathbf{x}_i))\|^2}$$

Like the previous kinematic surface fitting methods (Sec. 2), this is a direct method solveable by a small generalized eigenvalue problem. Like those methods, it rescales the cumulative squared error (as in Eqn. 6), so individual points are still scaled by local velocity. However, when  $w_{\mathbf{p}} := 0$ , this constraint ensures that the overall average squared velocities have a fixed magnitude – Taubin’s constraint becomes:

$$\frac{1}{n} \sum_{i=1}^n \|\mathbf{v}(\mathbf{p}_i)\|^2 = 1 \quad (13)$$

Because the average squared magnitude velocity is directly constrained, we can’t ‘cheat’ the Taubin-constrained error metric by choosing a field that globally reduces the velocity at all data points. This prevents failure cases of the variety described in Sec. 3.

Note that letting  $w_{\mathbf{p}} := 0$  ensures that Taubin’s method is basis independent, and does not cause stability issues: degenerate points where  $\mathbf{v}(\mathbf{p}_i)$  goes to zero simply do not contribute to either the numerator or denominator sums.

To implement Taubin’s method, we express the normalization in the form  $\mathbf{m}^T \mathbf{N} \mathbf{m}$  required by the standard fitting algorithm (Sec. 2.2). The matrix  $\mathbf{N} = (\nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x}_i))(\nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x}_i))^T$ ; for the spiral field (Eqn 3) this is:

$$\begin{aligned} \mathbf{N} = & \sum_{i=1}^n \begin{bmatrix} [\mathbf{p}_i]_{\times}^T [\mathbf{p}_i]_{\times} & -[\mathbf{p}_i]_{\times} & 0 \\ -[\mathbf{p}_i]_{\times}^T & \mathbf{I} & \mathbf{p}_i \\ 0 & \mathbf{p}_i^T & \mathbf{p}_i \cdot \mathbf{p}_i \end{bmatrix} \\ & + w_{\mathbf{p}} \sum_{i=1}^n \begin{bmatrix} [\mathbf{n}_i]_{\times}^T [\mathbf{n}_i]_{\times} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \mathbf{n}_i \cdot \mathbf{n}_i \end{bmatrix} \\ & \text{with } \mathbf{m} = \langle r_x, r_y, r_z, c_x, c_y, c_z, \gamma \rangle \end{aligned} \quad (14)$$

For generality, we have included the  $w_{\mathbf{p}}$  terms here; when applying the Taubin constraint this weight should be zero, but in a non-linear, iterative method (Sec. 4.2) it can be non-zero.

To demonstrate the Taubin constraint in practice, we show a number of practical test cases of the Taubin constraint in Figs. 3-6 and 8-10.

While the Taubin constraint works well in practice, it remains a biased approximation – it systematically places less weight than ideal on points where the velocity field is small, and more where the velocity field is large. To address this, we turn to iterative, non-linear AML methods.

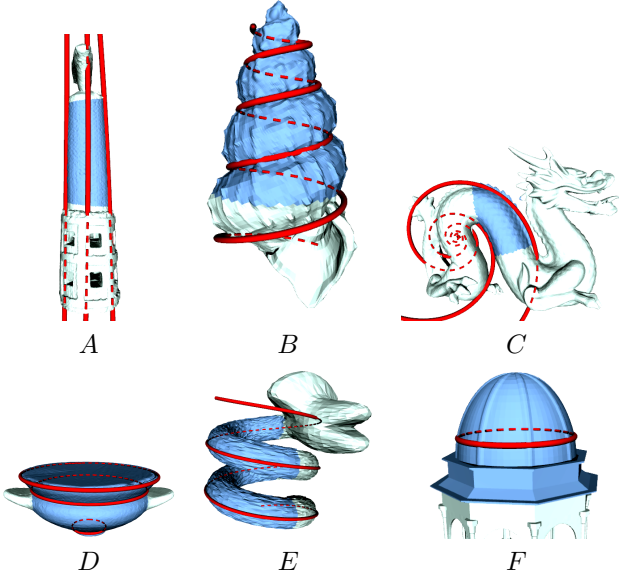


Figure 6: A spiral velocity field fit to a number of different selections (in blue) using Taubin’s constraint. Streamlines tracing the best fitting field are shown for each image.

#### 4.2. Iterative AML Methods: HEIV and Reduced

To minimize the true non-linear error term (either Eqn. 5 or Eqn. 12), we must use an iterative method. Fortunately, a number of iterative AML methods have been developed [15, 16, 17] – all of which have been shown to converge very quickly in theory for data with small noise [15]. All of these methods are based on solving a similar eigenvalue problem to the direct methods (Sec. 2.2) but iteratively adjusted to correct the weights of the data points. Because the weights can only be corrected with respect to one field, these iterative methods focus on finding a single best solution rather than the full basis of solutions provided by direct methods.

Previously the “reduced method” [16] has been suggested for use on the kinematic surface fitting problem [18], although without evaluation. This method simply iteratively re-weights the unit-constraint method. Specifically, it repeatedly solves the eigenvalue problem described in Sec. 2.2, with normalization matrix  $\mathbf{N} = \mathbf{I}$  and error matrix  $\mathbf{M}$  re-computed at the  $(j + 1)^{\text{th}}$  iteration as:

$$\mathbf{M}_{j+1} := \sum_i \frac{\mathbf{f}(\mathbf{x}_i)\mathbf{f}(\mathbf{x}_i)^T}{\|\nabla_{\mathbf{x}}(\mathbf{m}_j \cdot \mathbf{f}(\mathbf{x}_i))\|^2} \quad (15)$$

Where  $\mathbf{m}_j$  is the parameter vector at iteration  $j$ ;  $\mathbf{m}_0$  can be initialized by solving with any direct method. Unfortunately, this fails in the same way as the non-iterative unit constraint method: the inherent biases of that method are not addressed by re-weighting, and similar results to those of Figs. 4 and 5 occur. This result is consistent with the poor performance observed for the reduced method on algebraic curve fitting problems under mild noise [15].

In contrast, the “heteroscedastic errors-in-variables” (HEIV) method [17] performed much more successfully

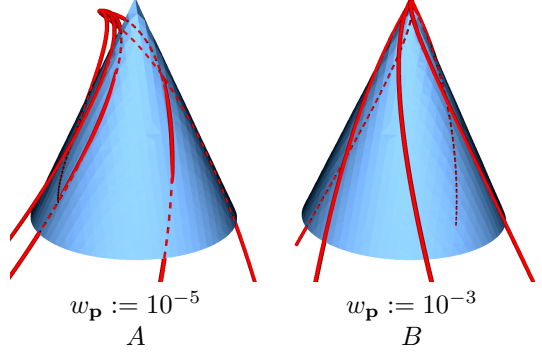


Figure 7: A spiral field (Eqn 3) is fit to a cone using the HEIV method with varying values for  $w_{\mathbf{p}}$ . The HEIV method never converges for  $w_{\mathbf{p}} := 10^{-5}$ ; we show the state after 101 iterations. The method converges in 2 iterations for  $w_{\mathbf{p}} := 10^{-3}$ .

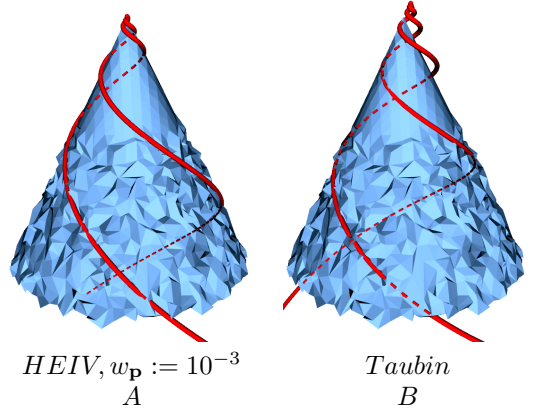


Figure 8: A spiral field (Eqn 3) is fit to a cone with Gaussian noise applied to the base ( $\sigma = 1\%$  bounding box size) using the HEIV method (converged in 3 iterations) and the Taubin method.

when evaluated in the context of algebraic curve fitting [15] – replicating the robustness under noise of the Taubin method, but with lower error. Intuitively this may be expected because this method can be seen as an iterative re-weighting of Taubin’s method. At each iteration, it solves a generalized eigenvalue problem with  $\mathbf{M}$  reweighted as in Eqn. 15 above, and  $\mathbf{N}$  reweighted as:

$$\mathbf{N}_{j+1} := \sum_i \left( \frac{(\mathbf{m}_j \cdot \mathbf{f}(\mathbf{x}_i))^2}{\|\nabla_{\mathbf{x}}(\mathbf{m}_j \cdot \mathbf{f}(\mathbf{x}_i))\|^4} \right) (\nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x}_i))(\nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x}_i))^T$$

This method performs similarly to the Taubin method on most examples we tested. Some care must be taken to choose the  $w_{\mathbf{p}}$  large enough for stability; we found  $w_{\mathbf{p}} \geq .001$  worked consistently, while smaller values could be unstable and thus fail to converge as shown in Fig. 7. We therefore set  $w_{\mathbf{p}} := .001$  for our tests. Advantages of HEIV over Taubin become evident when noise is distributed unevenly over the surface, in areas which Taubin will systematically over-weight, as shown in Fig. 8.



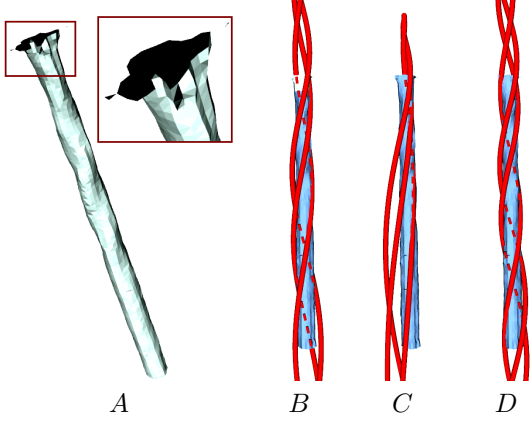


Figure 9: A spiral field (Eqn 3) is fit to a scanned model of a drill bit, on which outliers are concentrated at one end. In (B)-(D), the best-fitting vector field is visualized by red streamlines. (B) and (C) are fit without using RANSAC; in (B) the outliers at the end have been omitted manually by not selecting that portion of the mesh, while in (C) the outliers are included, significantly affecting the result. (D) is fit using RANSAC, and it gives a result similar to manually avoiding the outliers. The Taubin method is used for each fit.

## 5. Robustness to Outliers

Like any least-squares fitting method, these methods are all sensitive to outliers. Many methods can be, and have been, used to reduce the impact of such outlier points; in particular, M-estimators [19] and RANSAC [20] have both been suggested [10, 2]. The RANSAC approach is explained in detail, in the context of kinematic surface fitting, by [2]. We recommend this procedure, albeit using the new fitting techniques and normalized distances presented here instead of Eqn. 8. An example demonstrating the effectiveness of this approach is shown in Fig. 9. For this example we assumed that outliers have an error (computed by Eqn. 5) greater than .1, and that 90% of points are not outliers.

## 6. New Velocity Fields

Previous methods for kinematic surface fitting did not generalize well beyond the few field types listed in Sec. 2.1: the unit constraint's problems only become worse with more complex fields, and it is unclear how to apply the rotation constraint unless the field prominently features a rotation axis parameter. The Taubin and HEIV methods, in contrast, apply to any velocity field linear in its parameters  $\mathbf{m}$  – that is, any velocity field that can be expressed in the form  $\mathbf{v}(\mathbf{p}) := \sum_i m_i \mathbf{f}_i(\mathbf{p})$  where  $m_i$  are the elements of the parameter vector  $\mathbf{m}$ , and the functions  $\mathbf{f}_i(\mathbf{p})$  can be any functions from positions to vectors. Therefore these methods can be used to fit new, more general velocity fields. For sensible results, the class of fields chosen should also be closed under the Lie bracket operator – in other words, composing motions of multiple velocity fields in a class should result in motions which are also in that class.

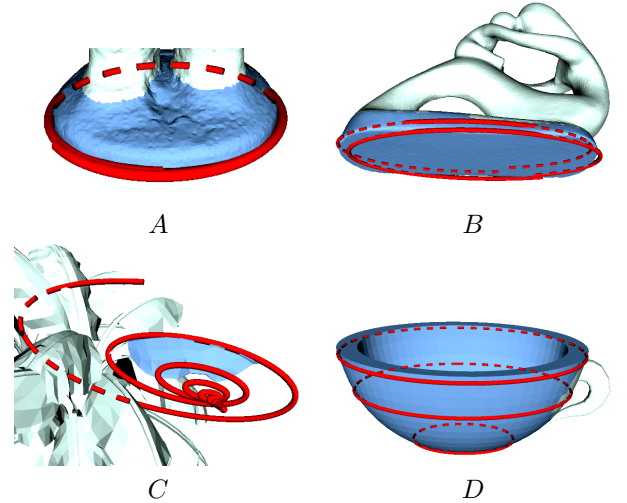


Figure 10: The general linear field (illustrated with red streamlines) is fit to a number of selections (in blue) on various objects.

The space of possible classes of velocity fields is enormous, and not easy to understand intuitively. But it opens doors to fitting some new, interesting primitives – and some simple primitives that were notably missing from the past repertoire of kinematic surfaces. For example: although spheres are handled by the more traditional kinematic surfaces, ellipsoids and general quadrics are not, because there is no support for rotation combined with some scaling. This would correspond to a non-linear field with some scaling matrix  $\mathbf{S}$  as a new parameter:

$$\mathbf{v}(\mathbf{p}) := \mathbf{S}^{-1}(\mathbf{r} \times (\mathbf{S}\mathbf{p})) + \mathbf{c} \quad (16)$$

Just by adding this scale factor, kinematic surfaces would now include all quadric surfaces as a subtype they could handle. This scaled equation is no longer linear in the parameters, but if we multiply out (letting  $\mathbf{A} = \mathbf{S}^{-1}[\mathbf{r}]_{\times}\mathbf{S}$ , where  $[\mathbf{r}]_{\times}$  is the matrix form of a cross product by  $\mathbf{r}$ ) we see that its fields are a subset of a class of general linear fields:

$$\mathbf{v}(\mathbf{p}) := \mathbf{A}\mathbf{p} + \mathbf{c} \quad (17)$$

where  $\mathbf{A}$  is an arbitrary  $3 \times 3$  matrix. This general linear field can be used to fit fields with rotation combined with some scaling, as we demonstrate in Fig. 10.

From this example it is clear that the more general fields do include at least one additional, useful shape primitive, and thus seem worthy of further investigation.

## 7. Discussion and Future Work

We identified a weakness in the standard algorithm for fitting kinematic surfaces, giving detailed examples of how this causes problems in practice as well as an explanation of the underlying source of these problems. We then presented a solution that is general, basis-independent, and

correctly handles the failure cases of the previous methods. This solution should improve the robustness of algorithms based on kinematic surface fitting. Its generality also makes it easy for us to introduce a new type of field.

The newly introduced field is a proof of concept that the generality of our method could apply to even broader applications. It also presents new challenges: with more complex velocity fields, a useful interpretation of the resulting parameters becomes more difficult. Developing a complete surface reconstruction pipeline that exploits the full range of possible kinematic surfaces will likely require exploration of additional new field types and new algorithms to fit and interpret those field types.

## 8. Acknowledgements

This work was supported in part by the National Science Foundation (NSF award #CMMI-1029662 (EDI)) and by Adobe Systems. Thanks to the Image-based 3D Models Archive, Tlcom Paris, and to the Stanford Computer Graphics Laboratory for some of the test meshes used.

## References

- [1] T. Varady, R. Martin, J. Cox, Reverse engineering of geometric models - an introduction, *Computer Aided Design* 29 (1997) 255–268.
- [2] M. Hofer, B. Odehnal, H. Pottmann, T. Steiner, J. Wallner, 3d shape recognition and reconstruction based on line element geometry, in: *Tenth IEEE International Conference on Computer Vision*, 2005.
- [3] H. Pottmann, H.-Y. Chen, I. K. Lee, Approximation by profile surfaces, *The Mathematics of Surfaces VIII* (1998) 17–36.
- [4] Y. Liu, H. Pottmann, W. Wang, Constrained 3d shape reconstruction using a combination of surface fitting and registration, *Computer-Aided Design* 38 (2006) 572–583.
- [5] G. Harary, A. Tal, The natural 3d spiral., *Computer Graphics Forum* 30 (2) (2011) 237–246.
- [6] A. Gferrer, J. Lang, A. Harrich, M. Hirz, J. Mayr, Car side window kinematics, *Computer-Aided Design* 43 (4) (2011) 410 – 416.
- [7] J. Andrews, H. Jin, C. H. Séquin, Interactive inverse 3d modeling, *Computer-Aided Design and Applications* 9 (6) (2012) 881–900.
- [8] N. Gelfand, L. Guibas, Shape segmentation using local slippage analysis, in: *Eurographics Symposium on Geometry Processing*, 2004.
- [9] T. Randrup, Approximation by cylinder surfaces, *Computer Aided Design* 30 (1998) 807–812.
- [10] H. Pottmann, T. Randrup, Rotational and helical surface approximation for reverse engineering, *Computing* 60 (1998) 307–322.
- [11] H. Pottmann, I.-K. Lee, T. Randrup, Reconstruction of kinematic surface from scattered data, *Proceedings of Symposium on Geodesy for Geotechnical and Structural Engineering* (1998) 483–488.
- [12] V. Pratt, Direct least-squares fitting of algebraic surfaces, *SIGGRAPH Comput. Graph.* 21 (4) (1987) 145–152.
- [13] 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 and Machine Intelligence* 13 (1991) 1115–1138.
- [14] K. Kanatani, N. Ohta, Comparing optimal three-dimensional reconstruction for finite motion and optical flow., *J. Electronic Imaging* 12 (3) (2003) 478–488.
- [15] N. Chernov, On the convergence of fitting algorithms in computer vision, *J. Math. Imaging Vis.* 27 (3) (2007) 231–239.
- [16] K. Kanatani, Further improving geometric fitting, in: *Proc. 5th Int. Conf. 3-D Digital Imaging and Modeling*, 2005, pp. 2–13.
- [17] Y. Leedan, P. Meer, Heteroscedastic regression in computer vision: Problems with bilinear constraint, *Int. J. Comput. Vision* 37 (2) (2000) 127–150.
- [18] H. Pottmann, J. Wallner, *Computational Line Geometry*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.
- [19] P. Huber, *Robust Statistics*, John Wiley and Sons, New York, 1981.
- [20] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Commun. ACM* 24 (6) (1981) 381–395.