A Buyer's Guide to Euclidean Elliptical Cylindrical and Conical Surface Fitting *

Petko Faber and Bob Fisher
Division of Informatics, University of Edinburgh,
Edinburgh, EH1 2QL, UK
npf|rbf@dai.ed.ac.uk

Abstract

The ability to construct CAD or other object models from edge and range data has a fundamental meaning in building a recognition and positioning system. While the problem of model fitting has been successfully addressed, the problem of efficient high accuracy and stability of the fitting is still an open problem. In the past researchers have used approximate distance functions rather than the real Euclidean distance because of computational efficiency. We now feel that machine speeds are sufficient to ask whether it is worth considering Euclidean fitting again. This paper address the problem of estimation of elliptical cylinder and cone surfaces to 3D data by a constrained Euclidean fitting. We study and compare the performance of various distance functions in terms of correctness, robustness and pose invariance, and present our results improving known fitting methods by closed form expressions of the real Euclidean distance.

1 Motivation

Shape analysis of objects is a key problem in computer vision with several important applications in manufacturing, such as quality control and reverse engineering. However, the application of shape in computer vision has been limited to date by the difficulties in its computation. To build a recognition and positioning system based on implicit curves and surfaces it is imperative to solve the problem of how curves and surfaces can be fitted to the data extracted from single or multiple 3D images. The fitting process is necessary for automatically constructing object models and for building intermediate representations from observations during the recognition. Implicit polynomial surfaces are potentially among the most useful object or data representations for use in computer vision and image analysis. Their power appears by their ability to smooth noisy data, to interpolate through sparse or missing data, their compactness and their form being commonly used in numerous constructions. An implicit surface is the zero set of a smooth function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ of the n variables: $\mathcal{Z}(f) = \{\vec{x} : f(\vec{x}) = 0\}$. Let $f(\vec{x})$ be an *implicit polynomial* of degree d given by

$$f(\vec{x}) = \sum_{\substack{(i+j+k)\leq d \\ \{i,j,k\}\geq 0}} a_{ijk} \cdot x^i \cdot y^j \cdot z^k = 0 \; . \tag{1}$$

*The work was funded by the CAMERA (*CA*d *M*odelling of Built *E*nvironments from *R*ange *A*nalysis) project, an EC TMR network (ERB FMRX-CT97-0127).

BMVC 2001 doi:10.5244/C.15.54

Then, we only have to determine the parameter set $\{a_{ijk}\}$ that describes the given data best. Parameter estimation is usually cast as an optimization problem, which can be solved in many ways because of different optimization criteria and several possible parameterizations. Generally, the literature on fitting can be divided into three general techniques: least-squares fitting (e.g. [1, 3, 9, 11, 13]), Kalman filtering (e.g. [4, 5, 6]), and robust clustering techniques (e.g. [2, 7]). While the clustering methods are based on mapping data points to the parameter space, such as the Hough transform and the accumulation methods, the least-squares methods are centered on finding the sets of parameters that minimize some distance measures between the data points and the curves and surfaces. Given a finite set of data points $\mathcal{D} = \{\vec{x}_p\}$, $p \in [1, P]$, the problem of fitting a general curve and surface $\mathcal{Z}(f)$ to $\mathcal{D}$ is usually cast as minimizing a distance measure

$$\frac{1}{P}\sum_{p=1}^{P} \mathrm{dist}\,(\vec{x}_p, \mathcal{Z}(f)) \rightarrow \mathrm{Minimum} \tag{2}$$

from the data points to the curve or surface $\mathcal{Z}(f)$, a function of the set of parameters $\{a_{ijk}\}$ of the polynomial. The distance from the point $\vec{x}_p$ to the zero set $\mathcal{Z}(f)$ is defined as the minimum of the distances from $\vec{x}_p$ to points $\vec{x}_t$ in the zero set $\mathcal{Z}(f)$:

$$\mathrm{dist}(\vec{x}_p, \mathcal{Z}(f)) = \min\{\| \vec{x}_p - \vec{x}_t \| : f(\vec{x}_t) = 0\} \ . \tag{3}$$

In the past researchers have often replaced the real Euclidean distance by an approximation because of computational efficiency. For approximation, often the result of evaluating the characteristic polynomial $f(\vec{x})$ is taken, or the first order approximation, suggested by Taubin [15], is used. However, experiments with the Euclidean distance show the limitations of approximations regarding quality and accuracy of the fitting results. Additionally, when using an approximation, the invariance of the fitting to Euclidean transformations is not guaranteed. But on the other hand it is obvious that accuracy and stability of the fitting has a substantial impact on the recognition performance especially in reverse engineering where we desire an accurate reconstruction of 3D geometric models of objects from range data. Thus it is very important to get good shape estimates from the data.
Combining Eq. (2) (to minimizes the distance error of all points) and Eq. (3) (to get the distance error of one point) to get the fitting algorithm over all data points will lead to algorithms of different computational complexity (see Table 1).

Algorithm	distance computation	minimization
Algebraic	closed form	closed form
Taubin	closed form	iterative
Euclidean	sometimes closed, sometimes iterative	iterative

Table 1: Computational properties of the three distance measures: algebraic distance, Taubin's approximation, and Euclidean distance

Given the doubly iterative nature of the Euclidean form, many researchers have avoided it. However, we show below that the time difference is not actually so bad as maybe expected. In this paper we

1. summarize how to compute the Euclidean distance function for elliptical cylinder and cone surfaces, These are probably the most commonly encountered shapes in reverse engineering (pipes and blending) after planar surfaces.

2. give an efficient algorithm for least-squares fitting using the Euclidean distance, and
3. compare this fitting to well-known fitting distance measures (algebraic and Taubin) and concludes that the Euclidean distance is much more accurate and stable without extraordinary computational expense.

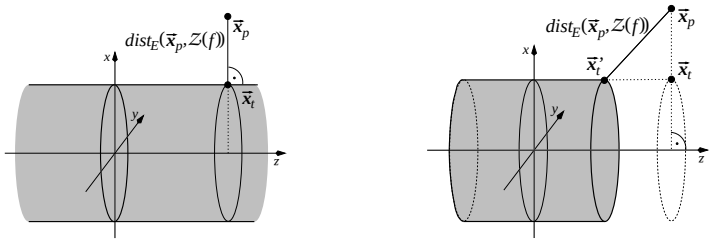
2 Closed-form expression for the Euclidean distance

For primitive surfaces like planes, cylinders, and cones, there is a closed form expression for the Euclidean distance from a point to the zero set and we use these. While the closed form expression for the distance between a point and a plane is trivial, we will show how the Euclidean distance can be estimated in a closed form for an elliptical cylinder and an elliptical cone.
It is a well-known that the most fitting methods are not pose invariant (see Sec. 4). To improve the pose invariance of the fittings we *normalize* the data set in terms of general moments to affine transformations at first. For a detailed description on how to normalize the data set we refer to the appropriate, numerous literature (e.g. [10, 14]). Applying the normalization we get the data set in a so-called *standard position* with respect to a given transformation, here the Affine transformation. The standard position can be understood as a special representative of the equivalence class corresponding to the transformation. If the data set is in standard position, it is ensured that we get a) more stable fitting results for all methods, and b) the framework to estimate the Euclidean distance can be simplified for the elliptical cylinder and elliptical cone substantially. In particular, we normalize by moving the data into standard position, translation by the centroid to the origin, rotation of the principal axes of the data aligned with the coordinate axes, and anisotropic scaling to fit in the unit cube.

2.1 Elliptical cylinder

If the generatix g of the elliptical cylinder is parallel to one axis, e.g. here the z axis, any arbitrary normalized elliptical cylinder can be expressed by the implicit polynomial $\mathcal{Z}(f) = a_{200}x^2 + a_{020}y^2 + a_{110}xy + a_{100}x + a_{010}y + a_{001}z + a_{000}$ with $a_{200}, a_{020} > 0$. The estimation of the Euclidean distance can be reduced to the estimation of the Euclidean distance to the directrix d, here an ellipse $\mathcal{Z}(d) = a_{200}x^2 + a_{020}y^2 + a_{110}xy + a_{100}x + a_{010}y + a_{000}$, $a_{200}, a_{020} > 0$. Thus, to estimate the Euclidean distance for an elliptical cylinder we only have to estimate the Euclidean distance to an ellipse. For a detailed description on how to estimate the Euclidean distance between a point $\vec{x}_p$ and the ellipse $\mathcal{Z}(f)$ in closed form we refer to [8].
If the Euclidean distance is given, we have to consider if the estimated $\vec{x}_t$ belongs to the cylinder element or not, which means if the generatrix g of the elliptical cylinder is infinite in length or not. In case of a finite generatrix we can simply estimate the point $\vec{x}_r \in \mathcal{Z}(f)$ nearest to $\vec{x}_t$ (see Fig. 1b.).

$$\mathrm{dist}_E\left(\vec{x}_p, \mathcal{Z}(f)\right) = \begin{cases} \|\vec{x}_p - \vec{x}_t\| & \text{if } g \to \infty \\ \|\vec{x}_p - \vec{x}_r\| & \text{else} \end{cases} \tag{4}$$



a. Infinite generatrix g b. Finite generatrix g

Figure 1: Estimation of $\mathrm{dist}_E(\vec{x}_p, \mathcal{Z}(f))$ for an elliptical cylinder. $\vec{x}_t$ is the nearest point to $\vec{x}_p$ on the (infinite) cylinder. If $\vec{x}_t$ does not belong to the cylinder element, $\vec{x}_r$ can be used.

2.2 Elliptical Cone

If the generatix g of the elliptical cone is parallel to one axis, e.g. here the z axis, any arbitrary normalized elliptical cone can be expressed by the implicit polynomial $\mathcal{Z}(f) = a_{200}x^2 + a_{020}y^2 + a_{002}z^2 + a_{110}xy + a_{100}x + a_{010}y + a_{001}z$ with $a_{200}, a_{020} > 0$ and $a_{002} < 0$. From the $\{a_{ijk}\}$ the fixed point $\vec{x}_0$ (vertex of the cone) can be directly deduced.

$$\vec{x}_0 = -\begin{pmatrix} a_{200} & \frac{1}{2}a_{110} & 0 \\ \frac{1}{2}a_{110} & a_{020} & 0 \\ 0 & 0 & a_{002} \end{pmatrix}^{-1} \cdot \begin{pmatrix} a_{100} \\ a_{010} \\ a_{001} \end{pmatrix} \tag{5}$$

The estimation of the Euclidean distance can be done in two steps. First the estimation of the Euclidean distance can be reduced to the estimation of the Euclidean distance to the directrix d, here an ellipse $\mathcal{Z}(d) = a'_{200}x^2 + a'_{020}y^2 + a'_{110}xy + a'_{100}x + a'_{010}y + a'_{000}$, $a'_{200}, a'_{020} > 0$. Now, the parameters A and B (axis of the directrix d) can be deduced from the $\{a'_{ijk}\}$

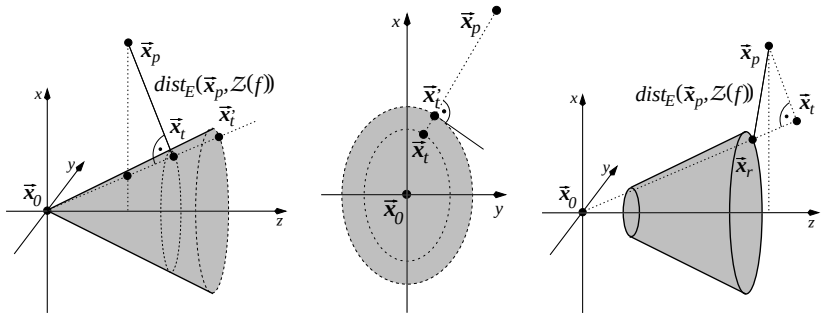
$$A = \sqrt{h/\lambda_1}, \quad B = \sqrt{h/\lambda_2}$$

$$\begin{aligned} \lambda_{1,2} &= \frac{1}{2}\left(a'_{200} + a'_{020} \pm \sqrt{(a'_{200} - a'_{020})^2 + (a'_{110})^2}\right) \\ h &= -\left(a'_{200}x_c^2 + a'_{020}y_c^2 + a'_{110}x_c y_c + a'_{100}x_c + a'_{010}y_c + a'_{000}\right) \end{aligned} \tag{6}$$

Note, this step is similar to that in the previous section, but $\{a'_{ijk}\}$ are different to $\{a_{ijk}\}$. In the second step the $\vec{x}_t$ with $(\vec{x}_o - \vec{x}_t) \perp (\vec{x}_p - \vec{x}_t)$ is determined, which yields the Euclidean distance. The only problem left is how to estimate $\vec{x}_t$. Therefore, we estimate $\vec{x}_t'$ located on the directrix d of the cone (see Fig. 2b.). Then $\vec{x}_t$ can be estimated using simple geometric relations (see Fig. 2a.). Then, the estimation of the Euclidean distance is trivial using Eq. 4. In the case that the estimated $\vec{x}_t$ does not belong to the cone element (see Fig. 2c.), the reasoning is similar to that in the previous section.

3 Euclidean Surface Fitting

To overcome the problems with the approximate distance metrics, it is natural to use instead the Euclidean distance, which is invariant to transformations in Euclidean space



a. (Infinite) generatrix g b. Directrix d c. Finite generatrix g

Figure 2: Estimation of $\mathrm{dist}_E(\vec{x}_p, \mathcal{Z}(f))$ for an elliptical cone. $\vec{x}_t$ is the nearest point to $\vec{x}_p$ on the (infinite) cone. If $\vec{x}_t$ does not belong to the cone element, $\vec{x}_r$ can be used.

and is not biased. In Sec. 2 we showed how the Euclidean distance can be estimated between a point and an elliptical cylinder or an elliptical cone. Given the Euclidean distance $\mathrm{dist}_E(\vec{x}_p, \mathcal{Z}(f))$ for each point the following fitting method can be used:

1. The Euclidean fitting requires an initial estimate for the parameters $\{a_{ijk}\}$ and we have found (cf. Sec. 4) that the results of Taubin's fitting method are better than the others. We get the initial parameter set $\{a_{ijk}\}^{[0]}$.

2. In the second step the estimation is updated using the Levenberg-Marquardt (*LM*) algorithm $\{a_{ijk}\}^{[s+1]} = F_{LM}(\{a_{ijk}\}^{[s]})$. The *LM* algorithm has become the standard of nonlinear optimization routines, because it combines the inherent stability of the Steepest Gradient Descent with the quadratic convergence rate of the Gauss-Newton method.

3. Finally the new estimation $\{a_{ijk}\}^{[s+1]}$ is evaluated by a so-called *M-estimator* $\mathcal{L}$ on the basis of $\mathrm{dist}_E(\vec{x}_p, \mathcal{Z}(f))$. $\{a_{ijk}\}^{[s+1]}$ will be accepted if $\mathcal{L}(\{a_{ijk}\}^{[s+1]}) < \mathcal{L}(\{a_{ijk}\}^{[s]})$ and the fitting will be continued with step 2. Otherwise the fitting is terminated and $\{a_{ijk}\}^{[s]}$ is the desired solution. The problem of selecting a suitable error function $\mathcal{L}$ was discussed in e.g. [12] and [17]. A good choice may be the so-called $\mathcal{L}_\nu$ (*least power*) function which represents a family of functions including the two commonly used functions $\mathcal{L}_1$ (*absolute power*) with $\nu = 1$ and $\mathcal{L}_2$ (*least squares*) with $\nu = 2$. In practise, we use $\mathcal{L}_{1.2}$ for 3 iterations, remove the 10% worst data points using a histogram analysis and continue the fitting with the commonly used $\mathcal{L}_2$ estimator.

4 Experimental results

In the previous section we described how to estimate the distance between a point and an elliptical cylinder or an elliptical cone and how to approximate a surface by an Euclidean fitting based on the estimated Euclidean distance. In this section we summarize empirical testing of the proposed fitting method in terms of efficiency, correctness and robustness for both simulated and real data. In case of simulated data we have generated data sets which describe (elliptical) cylinders and cones. The 3D data were generated by adding

isotropic Gaussian noise $\sigma = \{1\%, 5\%, 10\%\}$. Additionally the surfaces were partially occluded. The visible surfaces were varied between $1/2$ (maximal case) and $1/6$ of the full 3D cylinder. The performance of Euclidean fitting (*EF*) is compared with Algebraic fitting (*AF*), and Taubin's fitting (*TF*). Finally, we look to the pose invariance of the fitting methods. For all experiments we include in all three fitting methods the same constraints which describe the expected surface type to enforce the fitting of a special surface type.

4.1 Efficiency

A good fitting algorithm has to be as efficient as possible in terms of run time and formal complexity. While the problem of computational cost is no longer a really hard problem because of the rapidly increasing machine speed, we should guarantee the fitting with acceptable computational cost as well as the algorithm with relatively low complexity. All algorithms have been implemented in C and the computation was performed on a Pentium III 466 MHz. The average computational costs for the *AF*, *TF*, *EF* are in Tab. 2.

Table 2: Average computational costs in milliseconds per 1000 points.

	AF	*TF*	*EF*
Circular cylinder	3.583	3.625	12.375
Elliptical cylinder	13.292	13.958	241.667
Circular cone	15.667	15.833	288.375
Elliptical cone	15.042	15.375	291.958

As expected the *AF* and *TF* supply the best performance. The *EF* algorithm requires a repeated search for the point x_t closest to x_p and the calculation of the Euclidean distance. A quick review of the values in Tab.2 shows that the computational costs increase if we fit an elliptical cylinder, a circular or an elliptical cone, because the distance estimation is more complicated. In summary the efficiency is a *con* of *EF*, but is bounded by a factor of about 20 times the performance of the other algorithms and is still computationally reasonable for up to 10^6 data points if real-time performance is not needed.

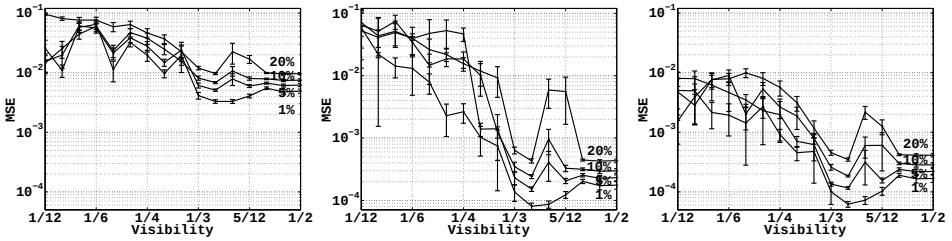
4.2 Correctness

It is obviously that the fitting result should describe the data set by the correct surface type. That means that it should not fit a false type to the data. To verify the correctness we tested if the fitting result of the (constrained) eigenvalue analysis corresponds to the general surface invariants. If one solution satisfies the conditions for the surface type, it is assumed that the fitting is correct in sense of an interpretable real surface. Otherwise, the fitting will be defined as failure. In our experiments *AF* failed up to 23%, especially in case of a higher noise level and a sparse data set (see Sec. 4.3). For *TF* and *EF* we had no failures in our experiments. In summary the correctness is a *pro* of *EF*.

4.3 Robustness

A fitting method must degrade gracefully with increasing noise in the data, with a decrease in the available relevant data, and with an increase in the irrelevant data. To evaluate the robustness of the proposed *EF*, we use synthetic generated data describing an elliptical cylinder by adding isotropic Gaussian noise $\sigma = \{1\%, 5\%, 10\%, 20\%\}$ and partially occlusion exposing between $1/2$ (maximal case) and $1/6$ of the full 3D cylinder. In the

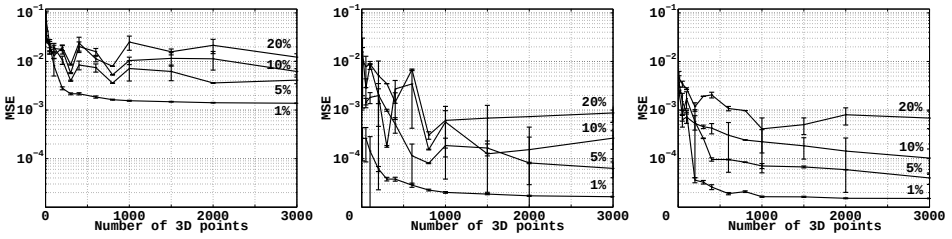
first experiment the number of 3D points for the simulated cylinder was $n = 100$ and to measure the average fitting error each experiment runs 100 times. The reported error is the Euclidean geometric distance between the 3D data points and the estimated surfaces. The mean squares errors (*MSE*'s) and standard deviations of the different fittings are in Fig. 3.



a. Algebraic fitting. b. Taubin's fitting. c. Euclidean fitting.

Figure 3: Average least squares error fitting a synthetic generated cylinder with added Gaussian noise $\sigma = \{1\%, 5\%, 10\%, 20\%\}$. The visible surfaces were observed between $1/2$ (maximal case) and $1/6$ of the full 3D cylinder. The number of trials was 100.

As expected, *TF* and *EF* yield the best results respect with to the mean and standard deviation, and the mean for *EF* is always lower than for the other two algorithms. The results of *AF* are only partially acceptable, because of the mean and the standard deviation. In the direct comparison of *TF* with *EF* the results of *EF* are much better. As mentioned in Sec.4.2, *AF* can sometimes give wrong results which means that the fitted curve or surface types does not come up with our expectations. We removed all failed fittings out of the considerations.
In the second experiment, the number of 3D points was stepwise decreased from $n = 3000$ down to $n = 30$ 3D data points to evaluate the behaviour of the several fitting methods. Each experiment runs 100 times. The mean squares errors (*MSE*'s) and standard deviations of the different fittings are in Fig. 4.



a. Algebraic fitting. b. Taubin's fitting. c. Euclidean fitting.

Figure 4: Average least squares error fitting a synthetic generated cylinder with added Gaussian noise $\sigma = \{1\%, 5\%, 10\%, 20\%\}$. The number of 3D points was stepwise decreased from 3000 up to 50. The visible surface was $5/12$ of the full 3D cylinder. The number of trials was 100.

As expected, *TF* and *EF* yield also the best results in this experiment. With decreased point density especially the *AF* becomes more and more unstable which is reflected in the

mean and standard deviation. Unexpectedly, the *EF* is very stable even with only $n = 10$ 3D data points. Reviewing some visualized fitting results for a real data set (see Fig. 5) we can assume that the results of *EF* are much better in the sense of robustness and the expected surface than the results of *AF* and *TF*. While the robustness (see Fig. 3 and 4) of *TF* is better than *AF*, the fitted surfaces of both fitting methods are mostly larger than the optimal surface. The results of *EF* are both robust and they correspond much more to our expectations. In summary the robustness is clearly a *pro* of *EF*.

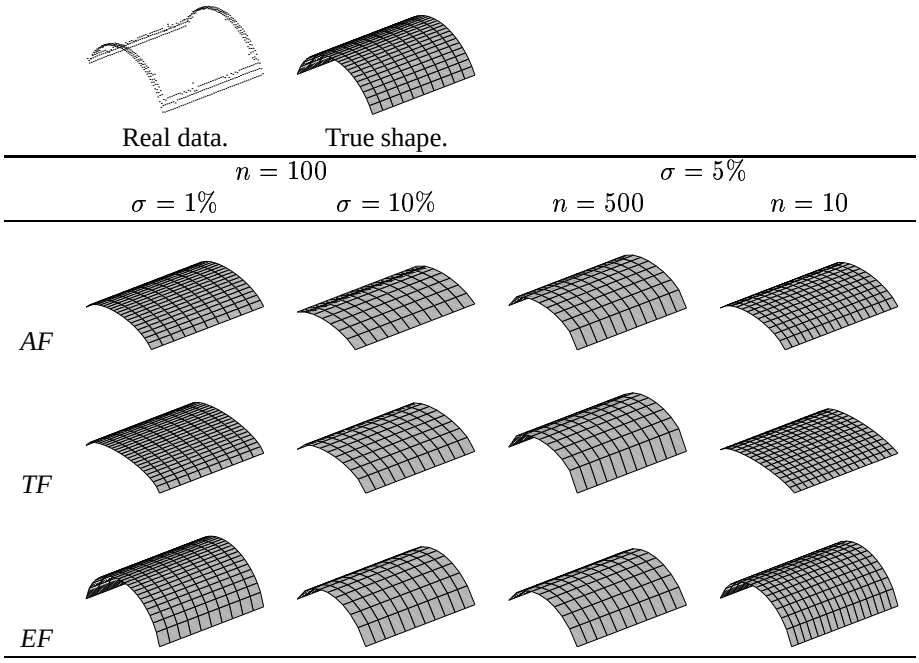


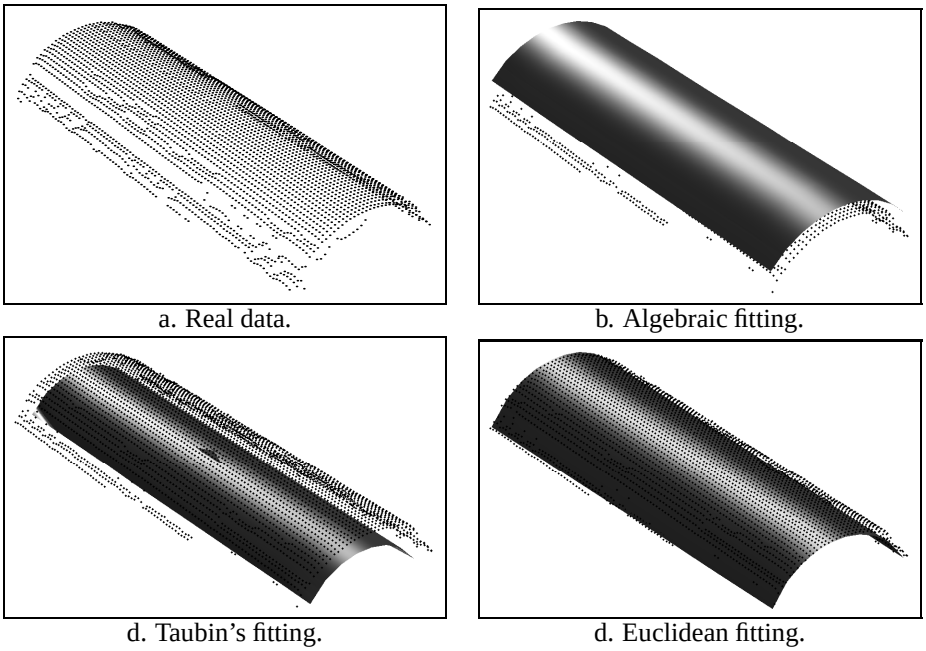
Figure 5: Fitting results for a real data set. The visible surface was $1/6$ of the full 3D cylinder. Note, the maximal visible surface is $1/2$ of the full 3D cylinder.

4.4 Pose invariance

It is obviously that the fitting results should be pose invariant. But, it is well known that this reasonable and necessary requirement cannot be always guaranteed by all three viewed fitting methods. To evaluate the pose invariance we use a real data set (see Fig. 6a.) describing an elliptical cylinder. The normalized data set was a) shifted, b) rotated, and c) both rotated and shifted. A quick review of the residuals (*MSE*) in Tab. 3 shows *AF* and *TF* are not pose invariant while the *EF* is pose invariant. To illustrate the pose dependency, the fitting results for position 3 are visualized in Fig. 6.
As we can see *AF* yields an elliptical cylinder with correct direction but too large radii, while the results of *TF* is an elliptical cylinder with a correct direction but too small radii. The result of *EF* describes the data set best (cmp. Tab. 3). In summary pose invariance is clearly a *pro* of *EF*.

Table 3: Residuals fitting an elliptical cylinder (see Fig. 6). The normalized cylinder was shifted by $t = [0.3, 0.2, 0.1]$ (pos. 1), rotated by $\vartheta = \pi/12$ and $n = [0.5, 1.0, 0.5]$ (pos. 2), shifted and rotated (pos. 3).

		normal pos	position 1	position 2	position 3
AF	$[10^{-3}]$	0.5242	2.0181	2.6950	1.8271
TF	$[10^{-3}]$	0.5024	1.5143	2.0277	1.3817
EF	$[10^{-3}]$	0.4021	0.4152	0.8634	0.6088



a. Real data. b. Algebraic fitting.

d. Taubin's fitting. d. Euclidean fitting.

Figure 6: Fitting results for real range data ($\approx$ 3300 points). The normalized data set was shifted by $t = [0.3, 0.2, 0.1]$ and rotated by $\vartheta = \pi/12$ and $n = [0.5, 1.0, 0.5]$.

5 Conclusion

We revisited the Euclidean fitting of elliptical cylindrical and conical surfaces to 3D data to investigate if it is worth considering Euclidean fitting again. The focus was on the quality and robustness of Euclidean fitting compared with the commonly used Algebraic and Taubin's fitting. Now, we can conclude that robustness and accuracy increases sufficiently compared to the other methods and Euclidean fitting is more stable with increased noise, as well as invariant to Euclidean transformations.

The main disadvantage of the Euclidean fitting, computational cost, has become less important due to rising computing speed. In our experiments the computational costs of Euclidean fitting were only about 2-19 times worse than Taubin's fitting. This relation probably cannot be improved substantially in favor of Euclidean fitting, but the absolute computational costs are becoming an insignificant deterrent to usage, especially if high accuracy is required.

References

[1] Allan, F. E. The general form of the orthogonal polynomial for simple series with proofs of their simple properties. In *Proc. Royal Soc. Edinburgh*, pp. 310–320, 1935.

[2] Besl, P. J. and R. C. Jain. Three-dimensional object recognition. *Computing Survey*, 17(1):75–145, March 1985.

[3] Bookstein, F. L. Fitting conic sections to scattered data. *CGIP*, 9:56–71, 1979.

[4] Chui, C. K. and G. Chen. *Kalman filtering with real time applications*. Springer, Berlin-Heidelberg-New York, 1987.

[5] Degeeter, J. and H. van Brussel and J. Deschutter and M. Decreton. *A smoothly constrained Kalman filter*. *IEEE Trans. on PAMI*, 19(10):1171–1177, 1997.

[6] Dickmanns, E. D. and V. Graefe. *Dynamic monocular machine vision*. *MVA*, 1:223–240, 1988.

[7] Duda, R. O. and P. E. Hart. The use of Hough transform to detect lines and curves in pictures. *Comm. Assoc. Comp. Machine*, 15:11–15, 1972.

[8] Faber, P. and R. B. Fisher. Estimation of General Curves and Surfaces to Edge and Range Data by Euclidean Fitting. *submitted to the MVA*.

[9] Fitzgibbon, A. W. and R. B. Fisher. A buyer's guide to conic fitting. In *6th BMVC*, pp. 513–522, 1995.

[10] Galvez, J. M. and M. Canton Normalization and shape recognition of three-dimensional objects by 3D moments. *PR*, 26:667–681, 1993.

[11] Kanatani, K. Renormalization for biased estimation. In *4th ICCV*, pp. 599–606, 1993.

[12] Rey, W. J. J. *Introduction to Robust and Quasi-Robust Statistical Methods*. Springer, Berlin-Heidelberg-New York, 1983.

[13] Rosin, P. L. A note on the least square fitting of ellipses. *PRL*, 14:799–808, 1993.

[14] Rothe, I. and H. Süße and K.Voss. The method of normalization to determine invariants. *IEEE Trans. on PAMI*, 18(4):366–376, 1996.

[15] Taubin, G. Estimation of planar curves, surfaces and non-planar space curves defined by implicit equations, with applications to edge and range image segmentation. *IEEE Trans. on PAMI*, 13(11):1115–1138, 1991.

[16] Taubin, G. An improved algorithm for algebraic curve and surface fitting. In *4th ICCV*, pp. 658–665, 1993.

[17] Zhang, Z. Parameter estimation techniques: a tutorial with application to conic fitting. *IVC*, 15:59–76, 1997.