Algebraic Curves That Work Better

Tolga Tasdizen
Division of Engineering
Brown University
Providence, RI 02912-9104
tt@lems.brown.edu

Jean-Philippe Tarel
Division of Engineering
Brown University
Providence, RI 02912-9104
jpt@lems.brown.edu

David B. Cooper
Division of Engineering
Brown University
Providence, RI 02912-9104
cooper@lems.brown.edu

Abstract

An algebraic curve is defined as the zero set of a polynomial in two variables. Algebraic curves are practical for modeling shapes much more complicated than conics or superquadrics. The main drawback in representing shapes by algebraic curves has been the lack of repeatability in fitting algebraic curves to data. A regularized fast linear fitting method based on ridge regression and restricting the representation to well behaved subsets of polynomials is proposed, and its properties are investigated. The fitting algorithm is of sufficient stability for very fast position-invariant shape recognition, position estimation, and shape tracking, based on new invariants and representations, and is appropriate to open as well as closed curves of unorganized data. Among appropriate applications are shape-based indexing into image databases.

1 Introduction

Algebraic $2D$ curves (and $3D$ surfaces) are extremely powerful for shape recognition and single-computation pose estimation because of their fast fitting, invariants, and interpretable coefficients, [1, 2, 5, 7]. Significant advantages over Fourier Descriptors are their simple applicability to non-star shapes, to open and/or non-ordered curve data that may contain gaps. A weakness has been the stability of estimated coefficients. This paper, studies the problem and provides a solution. The classical least squares fitting of algebraic curves, Sec. 3, especially the more interesting cases of higher degree polynomials, suffers three major problems:

- local inconsistency with the continuity of the dataset,
- local over-sensitivity of the polynomial zero set around the data to small data perturbations,
- global instability of coefficients due to excessive degrees of freedom in the polynomial.

Substituting an approximate Euclidean distance for algebraic distance [7], is much more stable than the classical least squares algorithm, in many cases gives useful fits, but in other cases the improvement is not sufficient to solve these major problems. Similarly, the use of the exact Euclidean distance provides better results than the algebraic distance [4], nevertheless the fitting is sometimes not stable enough and the minimization is solved iteratively, a time consuming process. Another attempt to improve the stability of the fit was the development of fitting algorithms which ensure that the obtained zero set is bounded [8, 2], but the latter is for $2nd$ degree curves only and increased stability for both and fitting speed for the former are still desired. The problem of an excessive number of parameters in *implicit polynomial* (IP) representations was first studied in [3] in the framework of Bayesian estimation. The linear 3L fitting algorithm [1] exhibits significantly improved curve representation accuracy and stability but there is significant value to further improvement in coefficient stability in order that algebraic curves be generally applicable for object-recognition purposes.

Following a short summary on algebraic curves in Sec. 2 and the classical least squares fitting in Sec. 3, the solution of the first and second problems by the 3L method [1] is analyzed from a new point of view in Sec. 4. In Sec. 5, we present a new linear algorithm which produces accurate and stable curve-data representations and stable coefficients. Results of object recognition experiments based on this algorithm and a new set of invariants [5] are presented in Sec. 6.

2 Representations of Algebraic Curves

Formally, an algebraic curve is specified by a 2D Implicit Polynomial (IP) of degree n given by $f_n(x, y) = \sum_{0 \le j+k \le n} a_{jk} x^j y^k = 0$. The *homogeneous binary poly-*

nomial of degree r in x and y is called a *form*, i.e., $a_{20}x^2 + a_{11}xy + a_{02}y^2$ is the 2nd degree form. The homogeneous polynomial of degree n is the so-called leading form. An algebraic curve of degree 2 is a conic, degree 3 a cubic, degree 4 a quartic, and so on.

Polynomial $f_n(x,y)$ is represented by the coefficient vector $(a_{jk})_{0\leq j,k;\ 0\leq j+k\leq n}$ which has dimension $p = \frac{1}{2}(n+1)(n+2)$:

$$f_n(x,y) = Y^t A \tag{1}$$

where [1] $A = [a_{00}\ a_{10} \ldots a_{n0}\ a_{n-1\ 1}\ \ldots\ a_{0n}]^t$ and $Y = [1\ \ x\ \ \ldots\ \ x^n\ \ x^{n-1}y\ \ \ldots\ \ y^n]^t$. In general, the vector notation is convenient for IP fitting since fitting can be set within a linear framework as detailed in Sec. 3. A shape is represented by the *zero set* of $f_n(x)$, i.e., the set of points $\{x,y\}$ satisfying the IP equation $f_n(x,y) = 0$.

3 Classical Least Squares Fitting

The classical and simplest way to fit an algebraic curve to data is to minimize the algebraic distance over the set of given data points $(x_j, y_j)_{1\leq j\leq m}$ with the least squares criterion, that is

$$e_a = \sum_{1\leq j\leq m} (f_n(x_j,y_j))^2 = A^t \left(\sum_{1\leq j\leq m} Y_j Y_j^t \right) A \tag{2}$$

by using vector representation of f_n as in (1). $S = \sum_{1\leq j\leq m} Y_j Y_j^t$ is the *scatter matrix* of the monomials.

To avoid the trivial zero solution in the minimization of (2), a constraint such as $\|A\|^2 = 1$ is imposed. The solution is given by the unit eigenvector A associated with the smallest eigenvalue of S, [7]. Although this algorithm is affine invariant [7], most of the time it is not of any practical use due to the following major problems: The fitted zero set does not respect the continuity of the original data set as illustrated in Fig. 4(a) and Fig. 1; thus classical LS fitting is not useful for shape representation. Results are highly sensitive to small errors in the data. Even seemingly negligible errors in the data can lead to zero sets that have no resemblance to the results that would be obtained if there were no errors in the data, Fig. 1. Even with low degrees, depending on the structure of the given data set, S may not provide a stable unique eigenvector A under small perturbations. For example, several eigenvalues can have similar values to the smallest one, so the solution will span a subspace in the coefficient space when small perturbations are added to the data set. Consequently, classical LS fitting is also practically useless for recognition purposes.

[1] Superscript ^{t} denotes vector and matrix transpose.

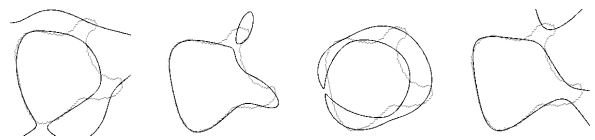


Figure 1: **Classical least squares algorithm gives unstable 4th degree IP fits under even the smallest perturbations to the data.**

4 Gradient-one Fitting

It is well known that some polynomials, in particular polynomials of high degree, are ill-conditioned in the sense that a tiny change applied to certain coefficients result in extreme variations in the roots of the polynomial, [6]. Loosely define the set of well-conditioned polynomials to be polynomials for which small or large changes in the coefficients produce small or large changes, respectively, in the roots, and vice versa. It is shown in [6] that a polynomial in one variable is stable in the above sense if its root locations and first derivative values at the root locations are all "close" to ± 1.0. In $2D$, a set of polynomials satisfying these constraints exactly are the powers of the unit circle: $\frac{1}{2n}((x^2+y^2)^n - 1)$. Members of the set of polynomials "close" to these polynomials in the coefficient space are well-conditioned.

The first requirement for stable fitting is to apply a data set standardization to force the data points to be close to the unit circle, and thus indirectly to force the zero set of the polynomial to be as close as possible to the unit circle. The data set standardization consists of centering the data-set centroid at the origin of the coordinate system and then scaling the set by dividing each point by the average of the eigenvalues of the 2×2 matrix of second order moments.

The second requirement is to control the value of the first derivatives along the zero set, i.e, the gradient of the 2D polynomial:

$$\nabla f_n(x_j,y_j) = \begin{bmatrix} \frac{\partial f_n}{\partial x} \\ \frac{\partial f_n}{\partial y} \end{bmatrix} (x_j,y_j) \tag{3}$$

The necessity to introduce information about the first derivatives was first pointed out in [1] and handled in a linear way with the so-called 3-levels (3L) fitting algorithm. Here we present another approach based directly on the gradients. The gradient vector along the zero set of the polynomial is always perpendicular to the curve defined by the zero set. Since the local tangents to the data curve can easily be computed by a fast distance transform, we propose to constrain the polynomial gradient at each data point to be perpendicular to these local tangents and to have unit norm. This will force the zero set of the polynomial to respect the local continuity of the data set. The proposed fitting

technique is set as a linear least squares problem with the additional constraints that the directional derivatives of the IP in the direction of the local tangents and normals must be as close as possible to 0 and 1, respectively. These constraints add two terms to (2) to yield

$$e_g = \sum_{1 \le j \le m} (f_n(x_j, y_j))^2 + \mu(N_j^t \nabla f_n - 1)^2 + \mu(T_j^t \nabla f_n)^2 \tag{4}$$

where T_j and N_j are the local tangent and normal at (x_j, y_j) and μ is the relative weight on the gradient with respect to the f_n^2 term and is fixed at $\frac{1}{8}$ for all experiments. With the use of the vector representation of f_n, this minimization is a linear least squares problem. Indeed, by using the vector notation (1) in (3), we deduce the vector form of the gradient: $\nabla f_n = \nabla Y^t A$. After substitution in (4), we expand e_g as:

$$\begin{aligned} e_g = A^t \underbrace{\sum Y_j Y_j^t}_{S} A + \mu A^t \underbrace{\sum \nabla Y_j N_j N_j^t \nabla Y_j^t}_{S_N} A + \\ \mu A^t \underbrace{\sum \nabla Y_j T_j T_j^t \nabla Y_j^t}_{S_T} A - 2\mu A^t \underbrace{\sum \nabla Y_j N_j}_{G_N} + \mu m \end{aligned} \tag{5}$$

S is the scatter matrix of the monomials as introduced before, S_N and S_T are the scatter matrices of the directional derivatives of monomials in directions perpendicular and tangent to the data set, respectively, and G_N is the average gradient of the monomials in the normal direction. The solution that minimizes e_g is then formally derived as:

$$A = \mu(\underbrace{S + \mu(S_N + S_T)}_{\mathcal{S}})^{-1} G_N \tag{6}$$

We named this algorithm gradient-one fitting. *Gradient-one fitting is Euclidean invariant, but not affine invariant*,[6]. Gradient-one fitting is also *scale invariant* due to the data standardization step. If the data standardization step has to be omitted, (5) and (6) can easily be modified to preserve scale invariance, [6]. However, data standardization should be used whenever possible because it improves the condition number of $\mathcal{S}$ and hence the numerical stability of $\mathcal{S}^{-1}$.

In comparison to the classical least squares fits, Fig. 4(a), results obtained with the gradient-one algorithm provide better shape representations that are locally consistent with the continuity of the data set, Fig. 4(b). It can be seen in Fig. 2 that the zero sets of the resulting fits are stable under data perturbations. Even though, the perturbations in Fig. 2(a) are much larger than those in Fig. 1, the changes in the fitted zero sets are much smaller in Fig. 2(b).

(a) (b)

Figure 2: **(a) An example perturbation superimposed on the original shape, (b) 10 superimposed 4th degree polynomial fits with the gradient-one algorithm to such perturbed data sets.**

5 Ridge Regression Fitting

5.1 Unstable Subspaces

Although, local stability of the zero set around the data is excellent with gradient-one fitting, there is still significant room for improvement in the stability of the coefficients of the polynomial and the global behaviour of the polynomial. Due to the problem of multicollinearity, coefficient vectors in certain subspaces of the coefficient space may produce very similar zero sets around the data set. As an example, assume that the data is a set of aligned points along $x - y = 0$, and that we are trying to fit a full conic. If we do the fit many times under small random noise, we can observe that the resulting coefficient vectors span a 3 dimensional subspace containing the solutions $x(x-y)$ and $y(x-y)$ as well as $x - y$. This is a consequence of the fact that each of these three solutions and all of their linear combinations fit the original data set equally well. The global instability of polynomials is also evident in the extra pieces of the zero set that lie away from the data. Indeed, these pieces are extremely sensitive to small perturbations in the data even though the zero set around the data is stable.

We now examine the multicollinearity and global instability problems. $\mathcal{S}$, defined in (6), is symmetric positive definite since it is a sum of scatter matrices, and thus can be written as $\mathcal{S} = U^t \Lambda U$ where U is a rotation in the coefficient space. The elements of Λ and the columns of U are the eigenvalues and eigenvectors of $\mathcal{S}$, respectively. If there is exact multicollinearity in the data, $\mathcal{S}$ will be singular and one or more eigenvalues will be 0. However, this rarely is the case; a much more common occurrence is near multicollinearity where some eigenvalues are very small compared to others and $\mathcal{S}$ is nearly singular with a very large condition number. Least Squares Estimation (LSE) is dedicated to finding the coefficient vector A that globally minimizes the error function in (4). Eigenvectors of $\mathcal{S}$ associated with the very small eigenvalues do not contribute to the polynomial significantly around the dataset; thus such vectors multiplied with large scalars

will be added into the solution in pursuit of slightly better solutions. This will result in very large variances for coefficients in the subspaces spanned by these eigenvectors. In Fig. 3 the graph of a goodness of fit function in two variables is shown. Notice that the function drops off steeply with the stable variable V, but changes only slightly with unstable W. Thus, the solution of LSE which seeks the highest point on the graph, marked LS in the figure, will move along the unstable *ridge* (shown in the figure with a heavier line) with the addition of small amounts of noise to the data. Consequently, the variance of the variable W under noise will be much larger than that of V. What we desire is that scalars multiplying such eigenvectors be pushed to zero rather than up to unstabily-cancelling infinities. This requires modifying LSE as we explain next.

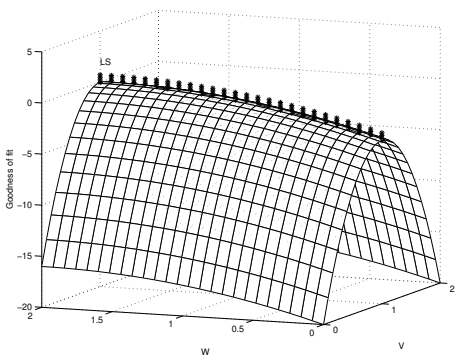


Figure 3: **Graph of an error function of two variables; here V is the stable variable while W is relatively unstable. The unstable ridge is marked by a heavier line.**

5.2 Ridge Regression (RR)

Since the solution has to move along the ridge, the stabilization of the LSE is known as *Ridge Regression* [9], referred to as RR in the rest of the paper. RR modifies $\mathcal{S}$ so that it is closer to what it would be for data in which there is no collinearity, that is, data in which all the explanatory variables are uncorrelated with one another. The modified coefficient vector, A_{rr}, is obtained by

$$A_{rr} = \mu(\mathcal{S} + kD)^{-1} G_N \tag{7}$$

where D is a diagonal matrix and k is the RR parameter. The specific form of D will be explained at the end of this section. When there is collinearity, (7) biases the solution closer to G_N. For the example given in Sec. 5.1, $G_N = [0 \quad n \quad -n \quad 2\bar{x}_j \quad \bar{y}_j - \bar{x}_j \quad -2\bar{y}_j]^t$. Thus, if the data set is centered at the origin, the solution obtained by RR is biased toward $[0 \quad 1 \quad -1 \quad 0 \quad 0 \quad 0]^t$, i.e., the equation of the line $x - y = 0$ we are searching for. If D in (7) were chosen to be the identity matrix, it can be shown [9] that

$$A_{rr} = U \Delta U^t A \tag{8}$$

Δ is a diagonal matrix of shrinkage factors and U is as defined in Sec. 5.1. In other words, RR modifies the LSE by first rotating it to obtain uncorrelated components, shrinking each component by some amount and finally restoring the original coordinate system by another rotation. The crucial point is the amount of shrinkage applied to each component. It is shown in [9] that

$$\Delta = diag(\delta_i) \ , \ \delta_i = \frac{\lambda_i}{\lambda_i + k} \tag{9}$$

where k is the RR parameter and λ_i are the eigenvalues of $\mathcal{S}$, i.e., the diagonal components of Λ. The shrinkage factor δ_i multiplies the *i'th* eigenvalue of $\mathcal{S}^{-1}$ which is λ_i^{-1}, thus the coefficient of the *i'th* eigenvector is shrunk by a factor of δ_i in the solution. Since the eigenvectors related to the very small eigenvalues of $\mathcal{S}$ are unstable, we would like to shrink them while leaving other eigenvectors largely unaffected. With (7), we accomplish this as shown by (9).

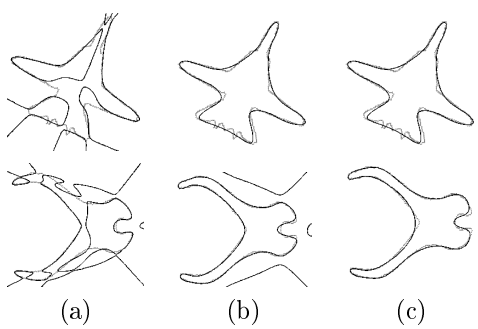


Figure 4: **(a) Classical Fitting Algorithm. (b) Gradient-one Fitting Algorithm. (c) RR Fitting Algorithm. Degree 6 and 8 are used for the airplane and pliers shapes, respectively.**

Fig. 4(c) shows fits of degrees 6 and 8 obtained by RR. Comparing these results with the results from standard gradient-one fitting shown in Fig. 4(b), we observe two important properties of RR: (i) the extra pieces of the zero set in the fit to the pliers shape is gone and both fits are bounded, and (ii) the smoothing introduced around the data set is negligible. These properties follow from the fact that stable dimensions are left largely unaffected by RR while unstable ones are shrunk to insignificant values. The effect of increasing k from 0 to higher values is shown in Fig. 5. Notice that the unbounded pieces that are close to the data in fitting with no RR ($k = 0$), start to move away

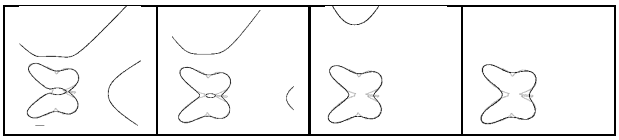


Figure 5: 6th **degree polynomial fits with the gradient-one algorithm and RR with increasing values of parameter k from left to right. Observe that the extra components are moving away from the data set.**

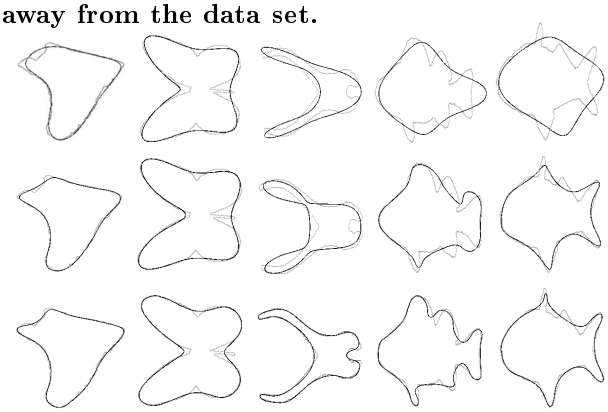


Figure 6: **Fits with 4th, 6th, and 8th degrees. No extra components are close to data sets. The RR parameter was chosen manually for each shape in this example.**

with increasing k. Actually, these pieces totally disappear and the polynomial zero set becomes bounded. As k is increased, $\mathcal{S} + kD$ approaches D, and A_{rr} in (7) approaches the limit $A_{lim} = \frac{\mu}{k} D^{-1} G_N$. Indeed for closed data curves and even degree polynomials, it can be shown using the divergence theorem for closed 2D curves that the limiting IP curve given by A_{lim} is always bounded, [6]. Hence, for closed data shapes and even degree IPs it is always possible to choose a k that will give a bounded IP curve. Fig. 6 shows more examples of 4th, 6th, and 8th degrees fits to illustrate this fact.

Rotational Invariance and the ridge matrix D

Invariance of the fitting algorithm to Euclidean transformations of the data is important to insure the repeatability of the results. A_{rr} in (8) is not rotationally invariant. Choosing D to be a diagonal matrix with elements

$$D_{vv} = \frac{i!j!}{(i+j)!} \sum_{k,l \geq 0; k+l=i+j} \frac{(k+l)!}{k!l!} \underbrace{\sum_{q=1}^{m} x_q^{2k} y_q^{2l}}_{diag.\ elements\ of\ S} \tag{10}$$

for $i, j \geq 0; i+j \leq n$ where $v = j + \frac{(i+j+1)(i+j)}{2}$ (ordering of the monomial matrices) preserves the rotational invariance of gradient-one fitting while achieving the desired regularization, [6]. Elements of D are invariantly weighted sums of the diagonal of the monomial scattering matrix S. With this choice of D, RR is very closely related to *weight decay* regularization used to overcome problems of *overfitting* in iterative optimization.

5.3 Choosing the RR Parameter

The *bias* of an estimator is the distance between the true value of the parameter being estimated, A_{true}, and the expected value of the estimator, $\overline{A_{rr}}$. The *variance* of an estimator is its expected square deviation from its expected value, $\overline{|| A_{rr} - \overline{A_{rr}} ||^2}$. k controls the *bias-variance* tradeoff. Usually, the variance is significantly reduced by deliberately introducing a small amount of bias so that the net effect is a reduction in *total mean squared error* which is defined as $bias^2 + variance$. Introducing bias is equivalent to restricting the range of functions for which a model can account. Typically this is achieved by removing degrees of freedom. Contrary to other approaches such as *Principal Component Methods*, RR does not explicitly remove degrees of freedom but instead smoothly reduces the variability of parameters. This makes the model less sensitive to small perturbations. Selection of the parameter k in practice can be done in one of two ways depending on what the resulting fit will be used for:

Choosing k for Shape Modeling. Here the main goal of fitting is to obtain a good representation of the shape without too much smoothing or extraneous pieces of the zero set. The smallest value of k that achieves this goal can be chosen by the user as in Fig. 6. Or k can be chosen automatically in an iterative trial-and-error approach since fitting for modeling can usually be done off-line. k can be increased from 0 to larger values until significant amounts of error start to be introduced into the fit.

Choosing k for Recognition. Here the main goal is to minimize the total mean squared error of estimator A_{rr}. Such an optimal value of k is empirically shown to exist and is found in Sec. 6. Choosing the optimal value of k analytically remains to be done in our future work. Optimal values of k could differ for different data sets. In [6], it is shown that k can be computed from a data independent threshold τ, on the condition number of $\mathcal{S} + kD$. The optimal value of τ will be data set independent.

6 Experiments

We comment on how the perturbed data sets used in the experiments were generated. The most commonly used shape perturbation model in Computer Vision is white noise, which adds independent amounts of noise to each data point. White noise when used with very small standard deviations is good for simulating quan-

tization errors; however, it is not a good model for generating deformations of a shape as might be sketched by a human or as might appear after segmentation from an image of an object taken under slightly different viewing conditions. The perturbation model we propose is colored noise (averaged white noise). A white noise sequence is generated and convolved with an averaging window. The standard deviation of the white noise sequence is chosen so that the resulting colored noise sequence will have the desired standard deviation as a percentage of the shape size. The obtained colored noise sequence is then added in the direction perpendicular to the data curve at each point, Fig. 7. Another type of perturbation used in our experiments is missing data where a random point on the given shape is picked and a number (a percentage of the total number of points) of consecutive points are removed, Fig. 7.

Figure 7: **A few shapes perturbed with 10% colored noise and 10% missing data.**

A set of 27 objects, Fig. 8, including real world objects and artificial free-form shapes ranging from simple to complex, were used for the experiments. Note that some objects have very similar shapes such as the fighter aircrafts, eels, and fishes. This makes object recognition for this set of objects a non-trivial task.

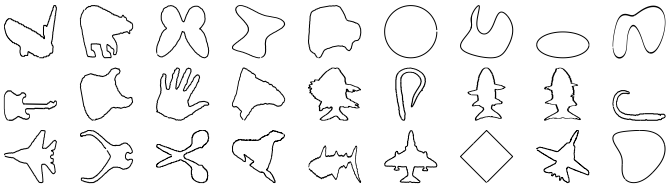
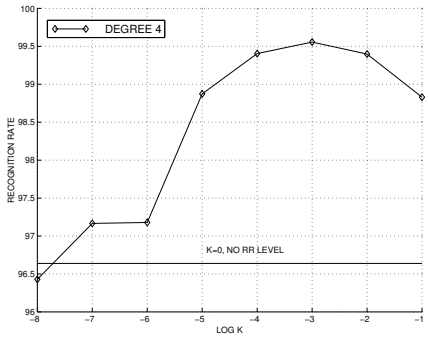
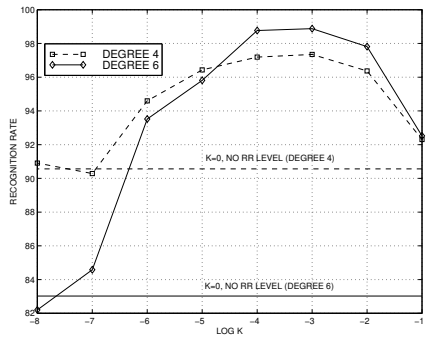


Figure 8: **Objects used in the experiments.**

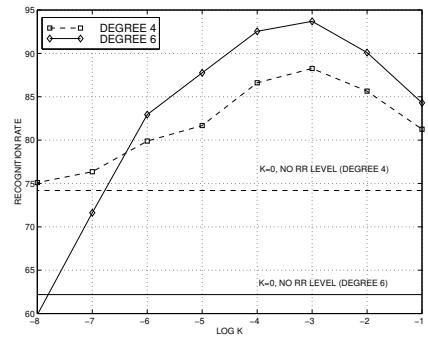
Recognition performance was tested under 3 perturbation models which are combinations of colored noise, missing data and rotation. Given a perturbation model, 1000 samples, i.e., perturbed shapes are generated from each base shape and fit with an IP. Then, a recently developed *complete set of invariants* [5] are computed for each coefficient sample. One of the important advantages of this set of invariants for recognition is that each invariant function is either a linear or quadratic function of the coefficients or an angle between 2 coefficients. This leads us to believe that they should be more robust than highly non-linear algebraic invariants. Finally, a mean vector and full covariance



(a)



(b)



(c)

Figure 9: **Pertubation models are (a) 10% colored noise + rotation, (b) 10% colored noise + 10% missing data + rotation and (c) 10% colored noise + 20% missing data + rotation.**

matrix in the invariant space is learned for each object. Test sets (100 samples of each object) are generated in the same manner independently from the training set.

Average recognition rates are plotted against the logarithm of the RR parameter k in Fig. 9. Recognition rates obtained without using RR are shown with the horizontal lines. In Fig. 9(a) $4th$ degree polynomials were used with a perturbation model of 10% colored noise and random rotations. Optimal choice of the ridge regression parameter provides approximately 3% increase over the already high rate of 96.5%. Note that there is an optimal value of k; this is expected since k controls the *bias-variance* tradeoff in invariant space and some value of k will minimize $bias^2 + variance$. The other experiments verify this fact with the fur-

ther important implication that for this set of objects, best recognition performance is obtained using approximately $k = 10^{-3}$ regardless of the degree of the polynomial or the perturbation model being used. One question to be investigated in future work is whether this optimal value of k will generalize to larger sets of objects.

The experiments presented in Fig. 9(b) use a stronger perturbation model combining 10% colored noise, 10% missing data and random rotations. Both $4th$ and $6th$ degree polynomials were tested. For degree 4, optimal choice of k provides 7% improvement in recognition achieving approximately 97%. For degree 6, a much more substantial 16% improvement is obtained raising the best recognition performance to approximately 99%. Using $6th$ degree IPs provides only a 2% advantage over using using $4th$ degree. For some non-optimal values of k and with no RR it actually does worse. There are two important deductions here: 1. Since $6th$ degree IPs have more coefficients (degrees of freedom) they are more prone to problems of unstable subspaces then $4th$ degree IPs, especially for simpler shapes that may not require a $6th$ degree polynomial. Since this is exactly the problem RR sets out to solve, the observation made above is expected. 2. It might seem tempting to restrict object recognition to the use of $4th$ degree IPs; however, as will be made clear in the next example there are much more substantial gains to be made with the use of higher degrees in some cases. We now use even a stronger model of perturbation, by keeping the 10% colored noise and rotation and doubling the amount of missing data to 20%. Robustness to missing data crucially depends on a good representation. Fig. 9(c) confirms this statement; $4th$ degree IPs yield a top recognition rate of approxiamtely 88%, $6th$ degree IPs are able to improve this rate to approximately 94%. Having established that using high degree IPs are necessary in certain problems, it is also very important to once more realize the crucial role played by RR in the success of high degree IPs; using the optimal value of k provided a gain of over 35% compared to no RR for $6th$ degree IPs in this example.

7 Conclusions

In the continuing quest for achieving maximum stability in the representation of curve data by algebraic curves and in the stability of the polynomial coefficients, this paper makes two important contributions. The first is an understanding of the role of data normalization and polynomial gradient-constraint in improving representation and coefficient stability. This also sheds light on why the 3L fitting algorithm [1] is so much more stable than previous fitting algorithms. The second contribution is the use of rotation-invariant RR, in the fitting, for improving the stability of both the representation and the coefficients even further. The RR drives those portions of the polynomial zero-set, that are not appropriate to the curve data, far from the data. It also shrinks to near-zero projections of polynomial coefficients in those subspaces that are not important for representing the curve data. The remaining coefficients are stable and result in increased stability when used for pose-invariant object recognition or object pose estimation. The exact same methodology of gradient-one fitting and RR can be used for 3D surface fitting to data in x,y,z. Also, the gradient-one least squares fitting can be extended to include higher order directional derivatives, e.g., to impose curvature constraints on the fitted shape representation.

Acknowledgments

This work was partially supported by NSF Grant #IIS-9802392.

References

[1] Z. Lei, M. M. Blane, and D. B. Cooper. 3L fitting of higher degree implicit polynomials. In *Proc. of Third IEEE Workshop on Applications of Computer Vision*, pages 148–153, Sarasota, Florida, Dec. 1996.

[2] M. Pilu, A. Fitzgibbon, and R.Fisher. Ellipse-specific direct least-square fitting. In *Proc. IEEE ICIP*, Lausanne, Switzerland, Sept. 1996.

[3] J. Subrahmonia, D. B. Cooper, and D. Keren. Practical reliable Bayesian recognition of 2D and 3D objects using implicit polynomials and algebraic invariants. *IEEE Trans. on PAMI*, 18(5):505–519, May 1996.

[4] S. Sullivan, L. Sandford, and J. Ponce. Using geometric distance fits for 3-D object modeling and recognition. *IEEE Trans. on PAMI*, 16(12):1183–1196, Dec. 1994.

[5] J.-P. Tarel and D. Cooper. A new complex basis for implicit polynomial curves and its simple exploitation for pose estimation and invariant recognition. In *Proc. IEEE CVPR*, pages 111–117, Santa Barbara, California, June 1998.

[6] T. Tasdizen, J.-P. Tarel, and D. Cooper. Improving the stability of algebraic curves for applications. Technical Report 176, LEMS, Division of Engineering, Brown University, 1998.

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

[8] G. Taubin, F. Cukierman, S. Sullivan, J. Ponce, and D. Kriegman. Parameterized families of polynomials for bounded algebraic curve and surface fitting. *IEEE Trans. on PAMI*, 16(3):287–303, March 1994.

[9] H. D. Vinod and A. Ullah. *Recent Advances In Regression Methods*. Statistics: Textbooks and Monographs. Marcel Dekker, Inc., 1981.