Invariant fitting of arbitrary single-extremum surfaces.

Andrew W. Fitzgibbon
Robert B. Fisher
Department of Artificial Intelligence, Edinburgh University
5 Forrest Hill, Edinburgh EH1 2QL

Abstract

Besl and Jain's variable order surface fitting algorithm [1] is a useful method of constructing a noise-free reconstruction of $2\frac{1}{2}$D range images with a small number of primitive regions. The use of bivariate polynomials as the approximation basis functions is linear, fast and easy to render robust. Seeding fits from regions classified by differential geometry is an important step towards a viewpoint invariant segmentation.

However, in order to better approximate arbitrarily shaped surfaces, polynomials of high degree are needed. For a region-growing paradigm, the poor extrapolation power of high order polynomials slows convergence and generates "non-intuitive" segmentations when crossing curvature discontinuities. Such segmentations are difficult to match against traditional CAD-like models. Further, the instability of the segmentation makes invocation of the correct model from a large database extremely difficult.

We show that these algorithms must of necessity trade representational richness for repeatability. In this paper we describe a new method of satisfying the requirement for high representational richness while retaining the ease of manipulation and recognition of single-extremum surface patches.

By introducing a canonical reparameterised coordinate system, biquadratic patches can be made to approximate arbitrary single-extremum shapes in a viewpoint invariant manner. An iterative fitting algorithm is presented, which quickly converges to the appropriate description. Examples of the abilities of the new approach are supplied, and compared with alternative strategies.

1 Introduction

We are interested in recognising complex curved objects using range data, with industrial inspection being the intended application domain. Typical scenes contain one or many objects, possibly overlapping. Initially, we make the assumption that the objects in the scene are a subset of a database of known objects stored in the computer, and that novel objects will be indicated to the system by an external agent. Given this framework, a number of decisions remain to be made, choice of model representation and matching algorithms being most important. The data representation is predetermined, a $2\frac{1}{2}$D image of depth values, which must be converted into a form suitable for matching against the chosen model representation, hence the need for some form of data segmentation. The combination of model representation and segmentation output must be sufficient to allow the system to

1. **Invoke** a small number of plausible hypotheses of data-to-model pairings from the (possibly large) object database. While in industrial applications,

BMVC 1993 doi:10.5244/C.7.57

the number of objects can generally be controlled, this is not the case in less structured environments. It seems desirable that a system should be extendable to cope with such situations.

2. Establish **correspondences** between features on the model and points in the image. This is important in the industrial domain where inspection tasks are specified in terms of measurements to be made on or between labelled model features.

3. Compute the **pose** of the object in order to further verify correspondences, to direct the sensor to invisible parts of the object, and to display matching results in an visually intuitive form.

4. Identify novel objects and add **automatically acquired models** to its object database. This ability is particularly relevant in unstructured domains, but is useful even in restricted domains — where a-priori models of the objects are unavailable or expressed in terms from which it is difficult to derive a visual model.

1.1 The Correspondence Problem

Within this framework, we may consider the problem of establishing correspondences between model and data features to be the bottleneck process. If a scene containing a single rigid object is converted to a symbolic form which is exactly that chosen in the model, modulo a predefined class of 3D transformations, matching will be exact. As an example, consider Figure 3. The model built from the first viewpoint is being matched with a segmentation from the second viewpoint. The features used are planar and biquadratic patches. In this instance, the correspondence between patches which appear in both images is close to exact. If the matching system can rely on such output, it can be simpler and more reliable. If it must also be able to cope with segmentations such as that shown in Besl's paper [1] it will need to be considerably more complex.

1.2 Desirable Properties of an Object Representation

Many researchers, for example Marr [6] and Fan [3] have enumerated the desirable properties of an object modelling scheme. Here we concentrate on three in particular.

Repeatability: If we accept that finding the model-to-data correspondence is the key difficulty, this leads to the first requirement, that presenting the same object to the system produces the same segmentation. This is generally divided into two subgoals: **Stability** and **Viewpoint Invariance**. Stability is the property that images which differ by small amounts should produce segmentations which differ by small amounts. Figure 1 illustrates how a simple region growing algorithm's choice of seed point can lead to instability. Viewpoint invariance is the property that the segmentation, expressed in the object's reference frame, is invariant to the predefined class of 3D transformations mentioned above.

Generality of Representation: It seems reasonable that the final symbolic representation of the scene should be sufficient to reconstruct the original image, to ensure that no information has been lost in the conversion.

Concise Descriptions: Although matching a perfect segmentation will in general be exact, we cannot conclude that it will be quick. The speed of the

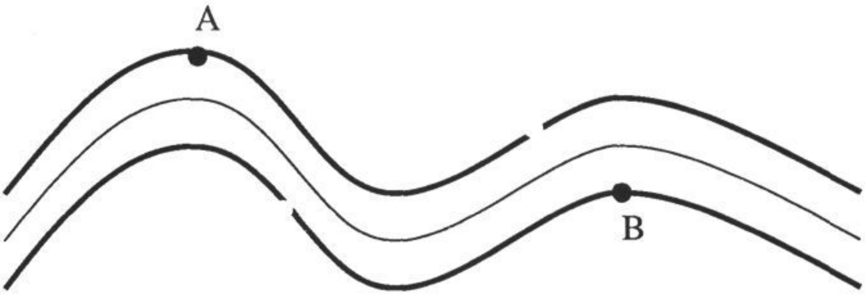

Figure 1: Demonstration of instability problem when growing high-order surfaces. The light curve is the data, the heavy curves are cubics fitted to the data. Depending on the choice of seed point (A or B), the breakpoint (shown as a gap) between the two regions changes considerably.

algorithm is dependent on the length of description — small rich descriptions will invoke and match quickly, whereas a valid, but large, description will in general take longer.

2 The Richness / Repeatability Tradeoff

Having chosen bivariate polynomials for region approximation, we must decide the highest order that will be used. The simplest constraint is that higher orders will model the data more accurately, while lower orders incur less computational expense. Besl chooses to limit to 4th order because it gives adequate results.

With the requirement for stability, however, additional constraints are imposed. In this case, the poor extrapolation power of high-order models is unacceptable. Consider Figure 1 where the order of fitting and choice of seed point drastically affects the resultant segmentation, even from a single viewpoint. We argue that this is because the segmentation is not sufficiently object-centered. In fact, stability is more easily guaranteed by restricting to single-extremum functions, which will not change curvature sign over their surface. In terms of polynomial order, this implies that we use only planes and biquadratics. Such a restriction in turn immediately begs consideration of the issue of representational richness, and does not address parameter invariance under change of viewpoint. We now discuss solutions to both these problems.

3 Invariant Fitting of Biquadratics

Perhaps the simplest parametric curved surface description is the explicit biquadratic patch, of the form:

$$F(\mathbf{x}_i) = F(x_i, y_i, z_i) = [a_1 ... a_6][1, x_i, y_i, x_i y_i, x_i^2, y_i^2]^T - z_i = 0$$

where the a_j are parameters, and $\mathbf{x}_i$ are the data points. As a representation, however, this clearly lacks viewpoint invariance and generality.

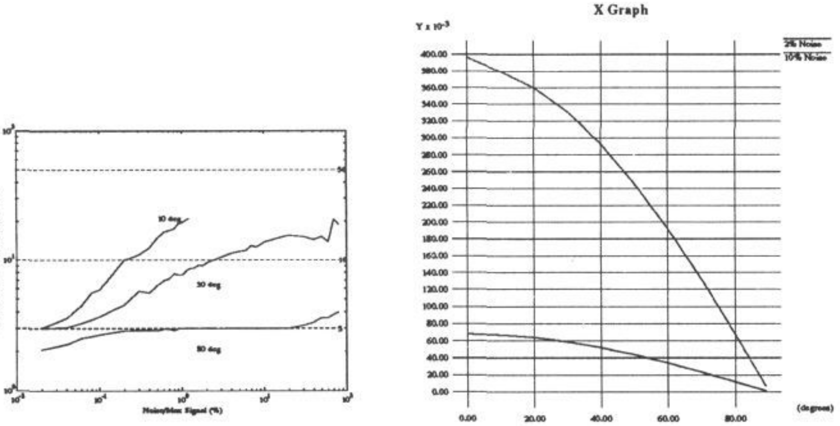

Figure 2: Performance of the canonical biquadratic fitting algorithm. The log-log graph on the left demonstrates the performance of the algorithm in fitting to the sloped cylinder $z = \sqrt{R^2 - y^2} + x \tan \theta$ for three values of θ. Increasing noise variance is on the X axis, number of iterations on the Y. The graph on the right shows the mean fit residuals as a function of surface slant. For more slanted surfaces, the rotated fit improves significantly, while the unrotated fit error is constant over all orientations.

3.1 A different representation

The essence of the lack of viewpoint invariance is that explicit bivariate polynomials have a 'canonical' reference frame, which is independent of the orientation of the surface patch away from the viewer. To eliminate this problem, we borrow an idea from the SMS modelling system [4, 5], which separates surface shape from extent and position. In this case, we consider the *canonical biquadratic*

$$B(\vec{\kappa}; \mathbf{u}) = B(\kappa_1, \kappa_2; u, v, w) = (\frac{\kappa_1}{2}u^2 + \frac{\kappa_2}{2}v^2 - w) = 0$$

which parameterises only the surface's principal curvatures at the origin. This representation also implicitly places the surface in a canonical reference frame. To fit an arbitrarily oriented surface, we include the required reference-frame transformation into the surface's equation:

$$F(\kappa_1, \kappa_2, R, \mathbf{t}; \mathbf{x}) = B(\kappa_1, \kappa_2; R\mathbf{x} + \mathbf{t})$$

where R is a 3×3 rotation matrix, and $\mathbf{t}$ is a translation vector. This representation is unique (up to a 180° rotation about the transformed Z axis) for a general biquadratic, where both curvature parameters are non-zero and unequal. When $\kappa_1 = \kappa_2$, the rotation about Z is unconstrained. If one curvature is zero, the translation has one degree of freedom. When both are zero, translation has two degrees of freedom.

3.2 Determining the transformation

Given a set of data points $\{\mathbf{x}_i\}_{i=1}^{n}$, we need a means of determining the surface parameters $(\kappa_1, \kappa_2, R, \mathbf{t})$, by minimising the least-square error measure

$$\epsilon^2 = \sum_{i=1}^{n} \left(\frac{B(\kappa_1, \kappa_2; R\mathbf{x}_i + \mathbf{t})}{\|\nabla B(\kappa_1, \kappa_2; R\mathbf{x}_i + \mathbf{t})\|} \right)^2$$

on the surface $\det(R) = 1$.

An initial estimate of the parameters is obtained by translating the points to the data centroid and fitting a 6-parameter biquadratic to the data. From this we calculate the normal at the origin $\mathbf{n} = norm(a_2, a_3, -1)$ and the elongation axis of the biquadratic in the X-Y plane $\mathbf{a} = (\cos\theta, \sin\theta, 0)$ with $\tan 2\theta = \frac{a_4}{a_5 - a_6}$. The rotation estimate R_k is the matrix which rotates $\mathbf{n}$ into the Z axis and $\mathbf{a}$ into the X-Z plane:

$$R_k = [(\mathbf{n} \times \mathbf{a}) \times \mathbf{n} \;\; \mathbf{n} \times \mathbf{a} \;\; \mathbf{n}]^T$$

An effective fitting algorithm can be constructed by rotating the data points by R_k and repeating the above process until

$$\mathrm{trace}(R_k R_k^T) > 3 - \epsilon$$

where ϵ is close to machine precision. The final rotation R is the inverse of $R^T = \prod_{k=1}^{n} R_k$. This algorithm has been implemented and converges in 3 to 5 iterations on real images taken in our lab. Figure 2 illustrates the number of iterations as a function of noise and initial angle to the viewer, by simulation. (Theoretical convergence calculations are difficult due to the re-fitting at each iteration.) Finally, it should be noted that Taubin [7] has recently developed a least-squares algorithm for implicit surface fitting which will fit this function using a generalized eigenvector method.

3.3 Matching two transformations

An immediate application demonstrating the usefulness of this representation is in the calculation of the registering transform between two images. With two corresponding patches $(\kappa_1^i, \kappa_2^i, R_i, \mathbf{t}_i)$, applying the rotation $R_2 R_1^{-1}$ to the first patch brings it into the same orientation as the second (again, modulo reflections). Figure 3 shows the registration between two images, where the rotation is calculated as described, and the translation is taken as the difference between the corresponded centroids of one of the planes.

4 Richness: modelling general surfaces

As noted in section 2, low-order polynomials do not closely approximate arbitrary curved surfaces. This implies that a representation based simply on biquadratics fails to satisfy the reconstructibility requirement. In this section we describe a modification of the standard biquadratic which maintains the curvature-sign preserving features of the single-extremum surface, while extending the representational power of the primitives. Experiments have been performed on the 1-D case, which we describe here. Extension to the 2-D case is discussed.

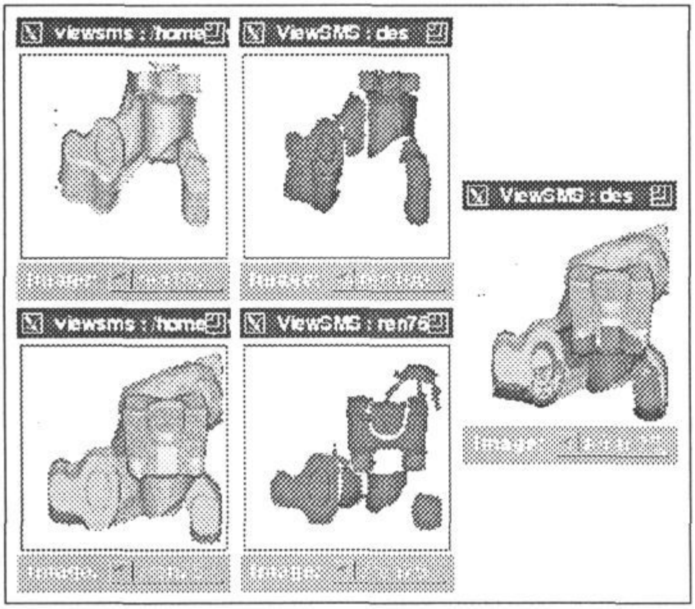

Figure 3: Calculating the rotation between 2 images. The two images on the left are raw data from two views, the middle column is the segmentation output and the left-hand image shows a model constructed from the first view (dark grey) superimposed on the data from the second viewpoint (light grey).

4.1 Monotonic reparameterisation

Figure 4 illustrates a parabola fitted to non-quadratic data. In the absence of noise the parabola cannot model the underlying circle. However, we can alter the shape of the parabola by reparameterising its X axis to "broaden the shoulders" of the curve. The single-extremum property may be retained by ensuring that the reparameterisation is monotonic. Without loss of generality, we may consider a parabola at the origin $y = f(x) = ax^2$. The reparameterised equation is $y = f(u)$, $u = g(x)$. Then $y(x)$ has only a single extremum as $y'(x) = f'(g(x))g'(x) = 2ag(x)g'(x)$. As $g(x)$ is monotonic, its derivative has constant sign, so that the only zero of $y'(x)$ is the single zero of $g(x)$, implying that $y(x)$ has a single extremum.

The advantage of this new parameterisation is that the monotonic function can be smoothed very heavily and still retain the underlying shape (see Figure 5 and Chapter 8 of Blake [2], for example).

4.2 Calculation of g

We now describe how to calculate the discrete approximation $\{g_i\}_{i=1}^{n}$ to the monotonic function g. First, we generate a disparity map at each data point which defines how the parabola needs to be stretched to meet that point. In the current implementation, this map is the signed distance between the point and the approximate nearest point on the curve (obtained by assuming that $f'(x_0) = f'(x)$, where x_0 is the data point, and x is the true nearest point on the curve. The projection

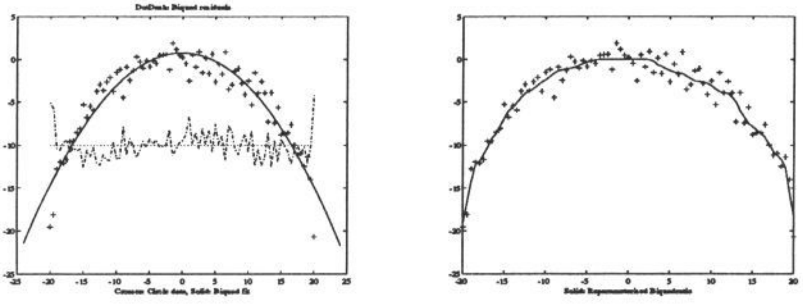

Figure 4: Monotonic smoothing. The left image shows a parabola fitted to noisy data sampled from a sphere. The right image shows the fit after the parabola's X axis has been smoothly monotonically reparameterised.

of the disparity vectors onto the X axis then defines a reparameterisation which exactly warps the parabola to fit the data. In general, this reparameterisation is neither monotonic nor smooth, and needs to be altered before use. The method used is to minimise $\sum_{i=1}^{n}(d_i - g_i)^2$ where d_i is the disparity measure at point i and g_i is constrained to be monotonic. The implementation solves the non-negative least-squares (NNLS) problem

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} \Delta_1 \\ \Delta_2 \\ \Delta_3 \\ \vdots \\ \Delta_n \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{pmatrix} - \min_{i=1}^{n} d_n$$

By ensuring that the Δ_i are non-negative, and because of the construction of the coefficient matrix, the technique finds the closest monotonic function to the disparity map if we set $g_i = \sum_{i=1}^{n} \Delta_i + \min_{i=1}^{n} d_n$. Results of the algorithm are shown in Figure 4.

4.3 Extension to Two Dimensions

Extending the 1D technique to two dimensions involves (a) defining what comprises monotonicity in 2D; and (b) devising a fitting technique that can quickly approach the required representation. We have cast the first problem as follows: Consider the original parameterisation as inkblots on a rubber sheet of constant elasticity. If the sheet is heated, say, its elasticity will locally change, moving the inkblots in a single-extremum preserving way. Figure 6 shows this technique applied to a paraboloid, with a hand-chosen "heating matrix". Our current work is now centred around finding an efficient fitting algorithm to decide the initial values for the heating matrix, based on raw data points.

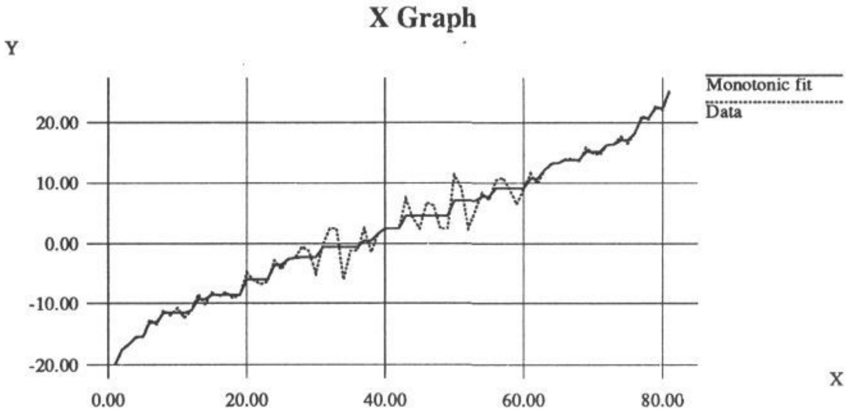

Figure 5: The results of the monotonic optimization algorithm on noisy disparity data.

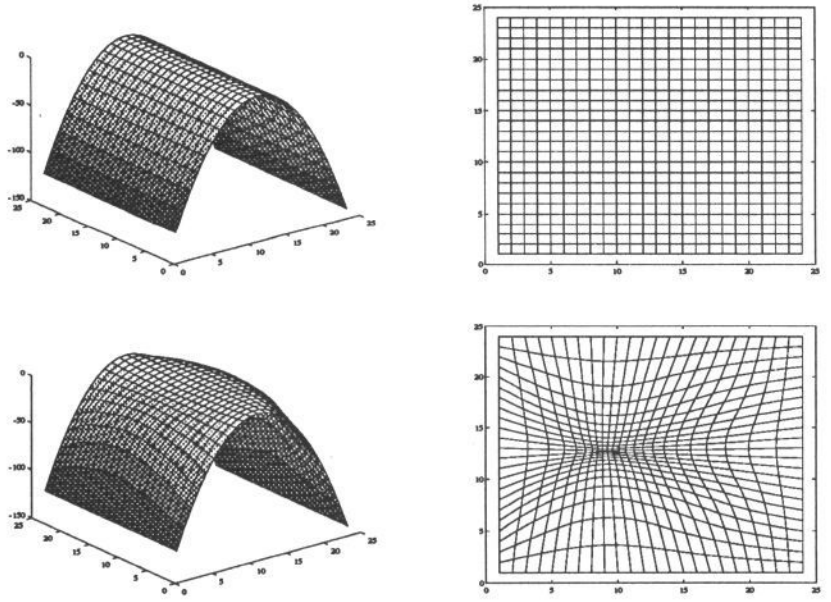

Figure 6: Using the rubber sheet to deform the biquadratic. The top two images show the original paraboloid and its parameter lines. The bottom images show the deformed parameter lines with the corresponding biquadratic.

5 Conclusions

We have introduced two new representations to the biquadratic. The first directly tackles the problems of viewpoint invariance of the surface patch, while the second extends the generality of the basic biquadratic. By restricting to the simplest type of curved surface, quick and robust algorithms can be used. The utility of the rotated biquadratic has been demonstrated on real images, and results from the monotonic reparameterisation look promising. The system shares similarities with other deformable modelling schemes, but is designed more for $2\frac{1}{2}$D images.

Future work will concentrate on improving the speed of the NNLS optimization, which currently uses a standard pseudo-inverse algorithm. In our case, however, the form of the matrices are *a-priori* limited, which may allow the development of an improved specialised method. Work is also progressing on the use of piecewise monotonicity as a general adaptive smoothing paradigm.

6 Acknowledgements

Funding for this work was provided by ACME grant GR/H/86905. The authors would like to thank the members of the Machine Vision Unit at Edinburgh University for their help and advice.

References

[1] P. J. Besl and R. C. Jain. Segmentation through variable-order surface fitting. *IEEE T-PAMI*, 10(2):167–192, March 1988.

[2] A. Blake and A. Zisserman. *Visual Reconstruction.* MIT Press, London, 1987.

[3] T.-J. Fan. *Describing and recognizing 3-D objects using surface properties.* Springer-Verlag, 1990.

[4] R. B. Fisher. SMS: A suggestive modelling system for computer vision. *Image and Vision Computing*, 5(2):98–104, May 1986.

[5] A. W. Fitzgibbon and R. B. Fisher. Suggestive modeling for machine vision. In *Proceedings, SPIE OE/Technology '92*, November 1992.

[6] D. Marr. *Vision.* W. H. Freeman Company, New York, 1982.

[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 T-PAMI*, 13(11):1115–1138, November 1991.